



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ

ΗΛΕΚΤΡΟΛΟΓΩΝ

ΜΗΧΑΝΙΚΩΝ

ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

Σχεδιασμός και Ανάπτυξη Διαδραστικής Εφαρμογής για την Διακοπή Καπνίσματός με Χρήση Τεχνολογικών Παιχνιδοποίησης και Παραγωγικής Τεχνητής Νοημοσύνης.

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Νικόλαος Δ. Κάσσαρης

Επιβλέπων : Παναγιώτης Δ. Τσανάκας

Κοσμήτορας ΣΗΜΜΥ, ΕΜΠ
Καθηγητής ΕΜΠ

Αθήνα, Σεπτέμβριος 2025



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ

ΗΛΕΚΤΡΟΛΟΓΩΝ

ΜΗΧΑΝΙΚΩΝ

ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

Σχεδιασμός και Ανάπτυξη Διαδραστικής Εφαρμογής για την Διακοπή Καπνίσματός με Χρήση Τεχνολογικών Παιχνιδοποίησης και Παραγωγικής Τεχνητής Νοημοσύνης.

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Νικόλαος Δ. Κάσσαρης

Επιβλέπων : Παναγιώτης Δ. Τσανάκας

Κοσμήτορας ΣΗΜΜΥ, ΕΜΠ
Καθηγητής ΕΜΠ

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 29^η Σεπτεμβρίου 2025.

.....
Π. Τσανάκας
Κοσμήτορας ΣΗΜΜΥ, ΕΜΠ
Καθηγητής ΕΜΠ

.....
Γ. Ματσόπουλος
Καθηγητής ΕΜΠ

.....
Α. Μενύχτας
Επίκουρος Καθηγητής ΠΑ.ΠΕΙ.

Αθήνα, Σεπτέμβριος 2025

.....
Νικόλαος Δ. Κάσσαρης

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

Copyright © Νικόλαος Δ. Κάσσαρης, 2025

Με επιφύλαξη παντός δικαιώματος. All rights reserved.

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τον συγγραφέα.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον συγγραφέα και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις επίσημες θέσεις του Εθνικού Μετσόβιου Πολυτεχνείου.

Περίληψη

Το κάπνισμα παραμένει μία από τις κύριες αιτίες πρόωρου θανάτου παγκοσμίως. Οδηγεί σε πολλές σοβαρές παθήσεις, όπως καρδιοπάθειες, καρκίνο και αναπνευστικά προβλήματα. Παρόλο που οι κίνδυνοι για την υγεία είναι ευρέως γνωστοί, ο εθισμός στη νικοτίνη καθιστά εξαιρετικά δύσκολη τη διακοπή για πολλούς ανθρώπους. Ο συνδυασμός ψυχολογικών και σωματικών παραγόντων απαιτεί ποικίλες στρατηγικές για την αποτελεσματική υποστήριξη της προσπάθειας διακοπής.

Η παρούσα εργασία περιγράφει τον σχεδιασμό και την ανάπτυξη ενός πληροφοριακού συστήματος που βοηθά τους χρήστες στη διαδικασία διακοπής του καπνίσματος. Χρησιμοποιεί τεχνολογίες Flutter και Firebase. Η εφαρμογή παρέχει εξατομικευμένες συμβουλές, παρακολουθεί την πρόοδο του χρήστη (όπως χρήματα που εξοικονομήθηκαν και την βελτίωση της υγείας του) και του βάζει στόχους. Κατά την πρώτη είσοδο του, ο χρήστης συμπληρώνει ένα ερωτηματολόγιο, το οποίο οδηγεί σε μια εξατομικευμένη εμπειρία.

Η εφαρμογή διαθέτει επίσης έναν διαδραστικό ψηφιακό βοηθό (chatbot) που βασίζεται σε μεγάλο γλωσσικό μοντέλο (LLM) με υποστήριξη τεχνολογίας RAG (Retrieval-Augmented Generation), παρέχοντας βοήθεια σε πραγματικό χρόνο μέσω διαλόγου. Επιπλέον, περιλαμβάνει πίνακα κατάταξης, φίλους, σύστημα επιτευγμάτων με πόντους, καθημερινά ενθαρρυντικά μηνύματα, ειδοποιήσεις, τεχνικές αντιμετώπισης στερητικών συμπτωμάτων με δυνατότητα καταγραφής τους, διάλογο με άλλους χρήστες και απλά δισδιάστατα παιχνίδια που αποσπούν την προσοχή του χρήστη σε στιγμές επιθυμίας για κάπνισμα.

Η εργασία συνοδεύεται από αναλυτική τεκμηρίωση, UML διαγράμματα, λειτουργική ανάλυση και προτάσεις για μελλοντικές βελτιώσεις. Στόχος είναι η αξιοποίηση της ψηφιακής τεχνολογίας για την ενίσχυση της ψυχολογικής υποστήριξης, προσφέροντας με αυτόν τον τρόπο μια καινοτόμα και εξατομικευμένη εμπειρία στους χρήστες.

Λέξεις-Κλειδιά: διακοπή καπνίσματος, εθισμός, Flutter, Firebase, Μεγάλα Γλωσσικά Μοντέλα, Langchain, RAG, παιχνιδοποίηση, υγεία, φορητή συσκευή, εφαρμογή, Dart.

Abstract

Smoking remains one of the top causes of early death worldwide. It leads to many serious health issues, such as heart disease, cancer, and respiratory problems. Although the health risks are well known, nicotine addiction makes it very hard for many people to quit. The mix of mental and physical factors requires different strategies to effectively help people stop smoking.

This thesis outlines the design and development of an information system that helps users during the process of smoking cessation. It uses Flutter and Firebase technologies. The app offers personalized advice, tracks the user's progress (like money saved and cigarettes avoided and the improvement of his health) and sets goals. When users first open the app, they fill out a questionnaire that creates a customized experience.

The app also features an interactive chatbot powered by a Large Language Model (LLM) with Retrieval-Augmented Generation (RAG) support. This chatbot provides real time conversation assistance. In addition, the app has a leaderboard for friends, achievements through experience points, daily motivational messages, notifications, techniques to cope with cravings and the ability to document them, chat with other users and simple 2D games to help distract users during cravings.

This work includes thorough documentation, UML diagrams, functional analysis, suggestions for future improvements, and app evaluation. The goal is to use digital technology to improve psychological support and motivation, providing this way a unique and personalized experience for users.

Keywords: smoking cessation, addiction, Flutter, Firebase, Large Language Models (LLM), Langchain, RAG, gamification, health, mobile, app, dart.

Ευχαριστίες

Η ολοκλήρωση αυτής της διπλωματικής εργασίας σηματοδοτεί το τέλος μιας μακρόχρονης και απαιτητικής, καθώς και βαθιά ουσιαστικής πορείας. Σε αυτό το σημείο, νιώθω την ανάγκη να ευχαριστήσω εγκάρδια όλους εκείνους που στάθηκαν δίπλα μου στα φοιτητικά αλλά και συνολικά στα μαθητικά μου χρόνια και με βοήθησαν να φτάσω ως εδώ.

Πρώτα απ' όλα, θα ήθελα να εκφράσω την ευγνωμοσύνη μου στον επιβλέποντα καθηγητή μου, κύριο Π. Τσανάκα και τον κύριο Α. Μενύχτα. Η διαθεσιμότητα και η επικοινωνία που είχαμε καθ' όλη τη διάρκεια της εργασίας, καθώς και η εμπιστοσύνη που μου έδειξαν, ήταν κρίσιμη στην περάτωση της εργασίας. Επίσης, η καθοδήγησή τους και ο χρόνος που αφιέρωσαν στηρίζαν την προσπάθειά μου και ήταν ουσιαστικές. Θα ήθελα ακόμη να εκφράσω την εκτίμησή μου στους καθηγητές που συμμετείχαν στην εξεταστική επιτροπή, για τον χρόνο και την προσοχή που αφιέρωσαν στην αξιολόγηση της παρούσας εργασίας.

Πέρα όμως από τον ακαδημαϊκό χώρο, ένα μεγάλο ευχαριστώ ανήκει στους φίλους μου, που στάθηκαν δίπλα μου σε αυτή την απαιτητική διαδρομή. Με χιούμορ, κατανόηση και ειλικρινή στήριξη, όντας οι περισσότεροι και συνάδελφοι, με βοήθησαν να παραμένω δημιουργικός και αισιόδοξος, ακόμη και στις πιο πιεστικές περιόδους.

Πάνω απ' όλα όμως, θέλω να ευχαριστήσω την οικογένειά μου, τους γονείς μου και τις αδερφές μου για την αγάπη, την υπομονή και την πίστη τους σε μένα. Ήταν πάντα εκεί, πρόθυμοι να ακούσουν και να με ενθαρρύνουν. Σταθήκαν δίπλα μου, ακόμα και όταν οι συνθήκες ήταν δύσκολες. Η παρουσία τους αποτέλεσε τη γερή βάση πάνω στην οποία χτίστηκε όλη αυτή η διαδρομή.

Η διπλωματική αυτή εργασία είναι αφιερωμένη σε όλους εσάς. Σας ευχαριστώ θερμά.

Αθήνα, Σεπτέμβριος 2025

Νικόλαος Δ. Κάσσαρης

Περιεχόμενα

Περίληψη	5
Abstract.....	7
Ευχαριστίες.....	9
Περιεχόμενα.....	11
Ευρετήριο Εικόνων.....	14
Ευρετήριο Πινάκων	16
Ευρετήριο Κώδικα.....	16
1. Εισαγωγή	18
1.1 Το Φαινόμενο	18
1.1.1 Το Κάπνισμα στη Σύγχρονη Κοινωνία	18
1.1.2 Ποσοστά και Σχετικά Στατιστικά.....	19
1.1.3 Αντίκτυπος	20
1.1.4 Αιτίες.....	21
1.2 Η Σημασία της Απεξάρτησης.....	22
1.3 Ψηφιακές Προκλήσεις και Ανάγκες	23
1.4 Ορισμός του Προβλήματος και Ερευνητικές Υποθέσεις	24
1.5 Η Δομή της Εργασίας.....	25
2. Σχετικό Υπόβαθρο και Βιβλιογραφική Ανασκόπηση	27
2.1 Ο Ρόλος των Ψηφιακών Εφαρμογών.....	27
2.2 Υπάρχουσες Εφαρμογές	27
2.3 Ηλεκτρονική Υγεία και Ψυχολογική Υποστήριξη	32
2.4 Η Τεχνητή Νοημοσύνη στην Υγεία.....	33
2.5 Κριτική και Καινοτομίες της Παρούσας Εργασίας	34

3.	Ανάλυση Απαιτήσεων και Περιγραφή Χρηστών	37
3.1	Στόχοι και Αντικείμενο της Εργασίας	37
3.1.1	Κύριος Σκοπός και Όραμα	37
3.1.2	Συγκεκριμένοι Στόχοι και Υποσυστήματα.....	38
3.1.3	Οριοθέτηση Λειτουργιών και Περιορισμοί.....	39
3.2	Γενική Περιγραφή του Συστήματος	39
3.3	Χρήστες και Χαρακτηριστικά.....	40
3.4	Απαιτήσεις Συστήματος.....	41
3.5	Περιπτώσεις Χρήσης.....	45
4.	Τεχνολογίες και Εργαλεία Υλοποίησης	49
4.1	Επιλογή Flutter και η Δομή του Συστήματος.....	49
4.2	Αρχιτεκτονική Συστήματος, UI και State Management.....	50
4.3	Firebase: Authentication, Firestore, Messaging	52
4.4	Μεγάλα Γλωσσικά Μοντέλα, Langchain και RAG.....	53
4.5	Python Backend FastAPI.....	55
4.6	Android Studio	57
4.7	Μοντελοποίηση Συστήματος.....	59
4.7.1	Διαγράμματα UML	59
4.7.2	Σχεδίαση με PlantUML	60
5.	Σχεδίαση Λύσης και Υποσυστημάτων	62
5.1	Η Αρχιτεκτονική Client-Server.....	62
5.2	Δεδομένα του Συστήματος.....	68
5.3	Ανάλυση Υποσυστημάτων	71
5.3.1	Σύστημα Αυθεντικοποίησης.....	72
5.3.2	Προφίλ και Εξατομίκευση.....	74
5.3.3	Επιτεύγματα και Υγειονομικοί Στόχοι	75
5.3.4	Υπενθυμίσεις και Ημερήσιες Συμβουλές	77
5.3.5	Chatbot Υποστήριξης	79
5.3.6	Πίνακας Κατάταξης και Κοινωνικές Λειτουργίες.....	81
5.3.7	Αναπνευστικές Ασκήσεις και Παιχνίδια	83
5.3.8	Σύστημα Καταγραφής Στερητικών Συμπτωμάτων	84
5.4	Φυσική Αρχιτεκτονική και Διαμόρφωση του Συστήματος.....	85
6.	Υλοποίηση της Εφαρμογής.....	89
6.1	Ενσωμάτωση Firebase και Firestore	89

6.2	Υλοποίηση Chatbot σε Python	90
6.3	Ανάκτηση Γνώσης με RAG	94
6.4	Διαχείριση Ειδοποιήσεων	98
6.5	Υλοποίηση Επιτευγμάτων και Πίνακα Κατάταξης.....	99
6.6	Παιχνίδια, Αναπνοές, Συμβουλές	100
6.7	Φίλοι, Συνομιλίες, Καταγραφές και Λοιπά.....	101
7.	Παρουσίαση και Λειτουργία του Συστήματος	105
8.	Αξιολόγηση και Συμπεράσματα	120
8.1	Απόδοση Chatbot και Συστήματος.....	120
8.1.1	Αξιολόγηση LLM και RAG	120
8.1.2	Μέθοδοι Αξιολόγησης.....	122
8.1.3	Αποτελέσματα της Αξιολόγησης.....	124
8.2	Ανάλυση της Εμπειρίας Χρήστη	126
8.3	Περιορισμοί του Συστήματος.....	127
8.4	Συμπεράσματα και Προτάσεις για Μελλοντική Επέκταση	129
	Αρχεία Συστήματος	131
	Παραπομπές.....	132

Ευρετήριο Εικόνων

Εικόνα 1: Ποσοστά Καπνιστών στην Ε.Ε.....	19
Εικόνα 2: Συχνότητα Καπνιστών στην Ε.Ε.....	20
Εικόνα 3: Οθόνες Εφαρμογής Quit Tracker	28
Εικόνα 4: Οθόνες Εφαρμογής Smoke Free.....	29
Εικόνα 5: Οθόνες Εφαρμογής Kwit.....	30
Εικόνα 6: Οθόνες Εφαρμογής QuitNow	30
Εικόνα 7: Use Case Diagram	48
Εικόνα 8: LLM Flow	54
Εικόνα 9: Παράδειγμα Ροής RAG.....	55
Εικόνα 10: Class Diagram.....	64
Εικόνα 11: Component Diagram.....	67
Εικόνα 12: Chatbot Component Diagram	67
Εικόνα 13: Entity-Relationship Diagram	70
Εικόνα 14: Activity Diagram-User Sign Up and Questionnaire	73
Εικόνα 15: Sequence Diagram-User Sign Up and Questionnaire.....	73
Εικόνα 16: Activity Diagram-App Home Screen	74
Εικόνα 17: Sequence Diagram-Load Home Screen	75
Εικόνα 19: Activity Diagram- Achievements Unlocking	76
Εικόνα 18: Activity-Diagram Daily Goals	76
Εικόνα 20: Sequence Diagram-Daily Goals.....	76
Εικόνα 21: Sequence Diagram-Unlock Achievement	76
Εικόνα 22: Activity Diagram-View Achievements.....	77
Εικόνα 23: Sequence Diagram-View Achievement.....	77
Εικόνα 24: Activity Diagram-Health Goals.....	77
Εικόνα 25: Sequence Diagram-Health Goals	77
Εικόνα 26: Activity Diagram-Daily Tips	78
Εικόνα 27: Sequence Diagram-Daily Tips	78
Εικόνα 28: Activity Diagram- Notification.....	79
Εικόνα 29: Sequence Diagram-Notifications	79
Εικόνα 30: Activity Diagram-Chatbot.....	80
Εικόνα 31: Sequence Diagram-Chatbot.....	80
Εικόνα 32: Activity Diagram-Leaderboard.....	81
Εικόνα 33: Activity Diagram- Community Chat	81
Εικόνα 34: Activity Diagram-Add Friend	82
Εικόνα 35: Sequence Diagram-Add Friend.....	82
Εικόνα 36: Sequence Diagram-Leaderboard	82

Εικόνα 37: Sequence Diagram-Community Chat.....	82
Εικόνα 39: Sequence Diagram-Mini Games.....	83
Εικόνα 38: Activity Diagram-Mini Games	83
Εικόνα 41: Sequence Diagram-Breathing Techniques	84
Εικόνα 40: Activity Diagram-Breathing Technique	84
Εικόνα 42: Activity Diagram-Craving Logs.....	85
Εικόνα 43: Sequence Diagram-Craving Logs	85
Εικόνα 44: Deployment Diagram	86
Εικόνα 45: Chatbot Deployment Diagram	87
Εικόνα 46: Ροή RAG με Classifier	92
Εικόνα 47: Αρχιτεκτονική του Chatbot με RAG και Contextual Prompt Injection.....	93
Εικόνα 48: Vector DB Retriever.....	94
Εικόνα 49: Δημιουργία Vector DB	96
Εικόνα 51: Σύνδεση/Εγγραφή Χρήστη	105
Εικόνα 50: Οθόνη Εισαγωγής.....	105
Εικόνα 52: Ανάκτηση Κωδικού Πρόσβασης.....	105
Εικόνα 53: Εισαγωγικά Μηνύματα Έπειτα από την Πρώτη Σύνδεση του Χρήστη	106
Εικόνα 54: Ερωτηματολόγιο Πρώτης Σύνδεσης	106
Εικόνα 55: Αρχική Οθόνη	107
Εικόνα 56: Καθημερινοί Στόχοι (Daily Goals)	108
Εικόνα 57: Οθόνη Στατιστικών.....	108
Εικόνα 58: Drawer Αρχικής Οθόνης.....	108
Εικόνα 59: Οθόνη Ρυθμίσεων	108
Εικόνα 60: Οθόνη Προόδου Στόχων Υγείας.....	109
Εικόνα 62: Οθόνη Ημερήσιων Συμβουλών (Daily Tips)	110
Εικόνα 61: Beat Cravings Οθόνη	110
Εικόνα 63: Αναδυόμενο Παράθυρο με Λεπτομέρειες Ημερήσιας Συμβουλής.....	110
Εικόνα 65: Οθόνη Καθοδήγησης Αναπνευστικών Ασκήσεων	111
Εικόνα 64: Αναπνευστικές Ασκήσεις	111
Εικόνα 67: Παιχνίδια (Pac Man, Φιδάκι, Flappy Bird)	112
Εικόνα 66: Οθόνη Παιχνιδιών	112
Εικόνα 68: Επιτεύγματα Ολοκληρωμένα (αριστερά) και Προς Ολοκλήρωση (Δεξιά).....	112
Εικόνα 69: Οθόνη Επικοινωνίας με Ειδικό και Οθόνες Ανακατεύθυνσης με Αυτόματη Συμπλήρωση.	113
Εικόνα 70: Οθόνη Επικοινωνίας	113
Εικόνα 71: Οθόνη Ομαδικής Συνομιλίας.....	113
Εικόνα 72: Λίστα με τις καταγραφές	114
Εικόνα 73: Προσθήκη Νέας Καταγραφής	114
Εικόνα 74: Πλοήγηση στις Καταγραφές Άλλων Χρηστών	114
Εικόνα 76: Δημιουργία Social Tag	115
Εικόνα 75: Προσθήκη Φίλου	115

Εικόνα 77: Χρήστης με Κατάταξη εκτός δεκάδας.....	115
Εικόνα 79: Όροι Χρήσης, Καρτέλα Επικοινωνίας, Αποσύνδεση Λογαριασμού.....	116
Εικόνα 78: Πληροφορίες Λογαριασμού	116
Εικόνα 80: Επανεκκίνηση Προόδου και Διαγραφή Λογαριασμού	116
Εικόνα 81: Αξιολόγηση Εφαρμογής.....	116
Εικόνα 82: Ειδοποιήσεις	116
Εικόνα 84: Chatbot στην πρόοδο της υγείας και η απαντήσεις του	117
Εικόνα 83: Chatbot στους ημερήσιους στόχους (Daily Goals).....	117
Εικόνα 85: Chatbot στις ημερήσιες συμβουλές.....	118
Εικόνα 86: Εξατομικευμένη Απάντηση του Chatbot.....	118
Εικόνα 87: Διαχείριση ερωτήσεων και λαθών Chatbot	119
Εικόνα 88: Αποτελέσματα Αξιολόγησης στο TruLens Dashboard	125
Εικόνα 89: TruLens Dashboard	125

Ευρετήριο Πινάκων

Πίνακας 1: Λειτουργίες Δημοφιλών Εφαρμογών Διακοπής Καπνίσματος.....	31
Πίνακας 2: Σύγκριση Κύριων Χαρακτηριστικών Υφιστάμενων Εφαρμογών με την παρούσα	35
Πίνακας 3: Οριοθέτηση Λειτουργιών και Περιορισμοί	39
Πίνακας 4: Λειτουργικές Απαιτήσεις Συστήματος.....	42
Πίνακας 5: Μη Λειτουργικές Απαιτήσεις.....	44
Πίνακας 6: Περιπτώσεις Χρήσης Συστήματος	45
Πίνακας 7: Απαιτήσεις Συστήματος που Υλοποιήθηκαν	127

Ευρετήριο Κώδικα

Κώδικας 1: FastAPI Endpoint	90
Κώδικας 2: Semantic Classifier.....	91
Κώδικας 3: Contextual Prompt Injection	92
Κώδικας 4: Retriever	93
Κώδικας 5: Fallback Search.....	95
Κώδικας 6: Memory Prunning.....	96
Κώδικας 7: Document Splitting	97

1. Εισαγωγή

Το κάπνισμα αποτελεί μία από τις σημαντικότερες αιτίες πρόωρης θνησιμότητας και ασθενειών παγκοσμίως. Παρότι οι επιπτώσεις του είναι ευρέως γνωστές και έχει σημειωθεί πρόοδος στην ευαισθητοποίηση του κοινού, εκατομμύρια άνθρωποι εξακολουθούν να ξεκινούν ή να διατηρούν αυτήν τη βλαβερή συνήθεια. Η καταπολέμηση του καπνίσματος δεν περιορίζεται πλέον μόνο σε πολιτικές απαγόρευσης και αυστηρούς νόμους, αλλά περιλαμβάνει και την αξιοποίηση τεχνολογικών εργαλείων, όπως εφαρμογές για κινητά τηλέφωνα, διαδραστικά βοηθήματα και υποστήριξη μέσω τεχνητής νοημοσύνης.

Η παρούσα εργασία αποσκοπεί στη μελέτη του φαινομένου του καπνίσματος, των επιπτώσεών του, της κοινωνικής και ψυχολογικής του διάστασης. Επίσης, εστιάζει στην ανάλυση και δημιουργία ενός συστήματος που στοχεύει στην υποστήριξη των ατόμων που επιθυμούν να διακόψουν το κάπνισμα.

1.1 Το Φαινόμενο

Το κάπνισμα είναι η εισπνοή καπνού που προέρχεται από την καύση ουσιών όπως φύλλα καπνού, με τη χρήση προϊόντων όπως τα τσιγάρα, τα πούρα, οι πίπες ή ακόμα και πιο σύγχρονα καπνικά προϊόντα, όπως οι ηλεκτρονικοί ατμιστές, συσκευές θέρμανσης καπνού κτλ. Η βασική εθιστική ουσία στον καπνό είναι η νικοτίνη, η οποία προκαλεί φυσιολογική και ψυχολογική εξάρτηση. Το κάπνισμα δεν αποτελεί μόνο μία προσωπική συνήθεια ή επιλογή, αλλά μια συμπεριφορά που συχνά συνδέεται με κοινωνικά, πολιτισμικά και ψυχολογικά πλαίσια.

Η ιστορία του καπνίσματος χρονολογείται από αρχαίους πολιτισμούς, όπου χρησιμοποιείτο σε τελετουργικά. Ωστόσο, η εκβιομηχάνιση του τσιγάρου και η μαζική του κατανάλωση ξεκίνησε κυρίως τον 20ό αιώνα. Από τότε, τα ποσοστά καπνιστών αυξήθηκαν ραγδαία, μέχρι την ευρεία διάδοση των επιστημονικών στοιχείων που ανέδειξαν τις αρνητικές επιπτώσεις του καπνού στην υγεία [1].

1.1.1 Το Κάπνισμα στη Σύγχρονη Κοινωνία

Αν και έχουν γίνει σημαντικές εκστρατείες για την πρόληψη και την ενημέρωση σχετικά με τις συνέπειες του καπνίσματος, αυτό εξακολουθεί να αποτελεί μια από τις κυριότερες αιτίες πρόωρης θνησιμότητας παγκοσμίως. Ο Παγκόσμιος Οργανισμός Υγείας (ΠΟΥ) εκτιμά ότι πάνω από 8 εκατομμύρια άτομα χάνουν τη ζωή τους κάθε χρόνο λόγω του καπνίσματος, ενώ εκατοντάδες εκατομμύρια ακόμα υποφέρουν από σοβαρές σχετιζόμενες ασθένειες [2].

Το κάπνισμα πλέον σχετίζεται έντονα όχι μόνο με τη νικοτίνη, αλλά και με κοινωνικούς, πολιτισμικούς και ψυχολογικούς παράγοντες. Η αύξηση του καπνίσματος σε νεαρά άτομα, κυρίως σε αναπτυσσόμενες χώρες, καθώς και η χρήση εναλλακτικών μεθόδων, όπως το ηλεκτρονικό τσιγάρο (vaping), προκαλεί ιδιαίτερη ανησυχία λόγω της ελκυστικότητάς του στους νέους.

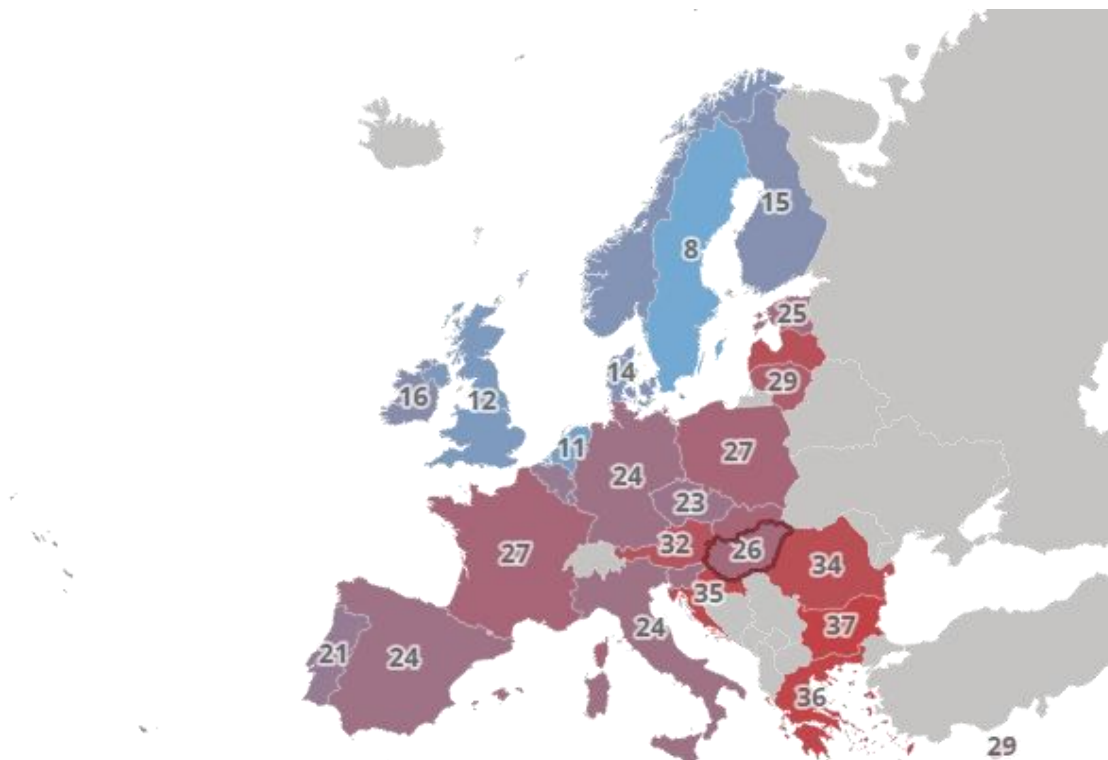
Στη σύγχρονη κοινωνία, το κάπνισμα δεν είναι μόνο μια προσωπική επιλογή ή συνήθεια, αλλά αποτυπώνει επίσης τις επιρροές του κοντινού περιβάλλοντος, της διαφήμισης, της πρόσβασης σε καπνικά προϊόντα και της ψυχικής κατάστασης των ατόμων. Η ενισχυμένη έκθεση σε στρεσογόνες καταστάσεις, η κοινωνική αποδοχή σε συγκεκριμένα περιβάλλοντα και η ψυχολογική πίεση είναι παράγοντες που προάγουν τη χρήση.

1.1.2 Ποσοστά και Σχετικά Στατιστικά

Σύμφωνα με τον Π.Ο.Υ., περισσότεροι από 1,25 δισεκατομμύρια άνθρωποι παγκοσμίως είναι καπνιστές, εκ των οποίων το 80% ζει σε χώρες με χαμηλό ή μεσαίο εισόδημα [3]. Στην Ελλάδα, σύμφωνα με τα στοιχεία της Ελληνικής Στατιστικής Αρχής (ΕΛΣΤΑΤ), περίπου το 27% του πληθυσμού άνω των 15 ετών δηλώνει ότι καπνίζει, με τους άνδρες να υπερτερούν σημαντικά των γυναικών [4].

Αξιοσημείωτο είναι επίσης ότι, παρά την ενημέρωση και τις προειδοποιήσεις στις συσκευασίες των καπνικών προϊόντων, ένας σημαντικός αριθμός εφήβων και νεαρών ενηλίκων εξακολουθεί να ξεκινά το κάπνισμα. Στην Ευρωπαϊκή Ένωση, το 29% των νέων ηλικίας 15-24 ετών έχει δοκιμάσει να καπνίσει, ενώ ένα σημαντικό ποσοστό εξακολουθεί να καπνίζει καθημερινά [5].

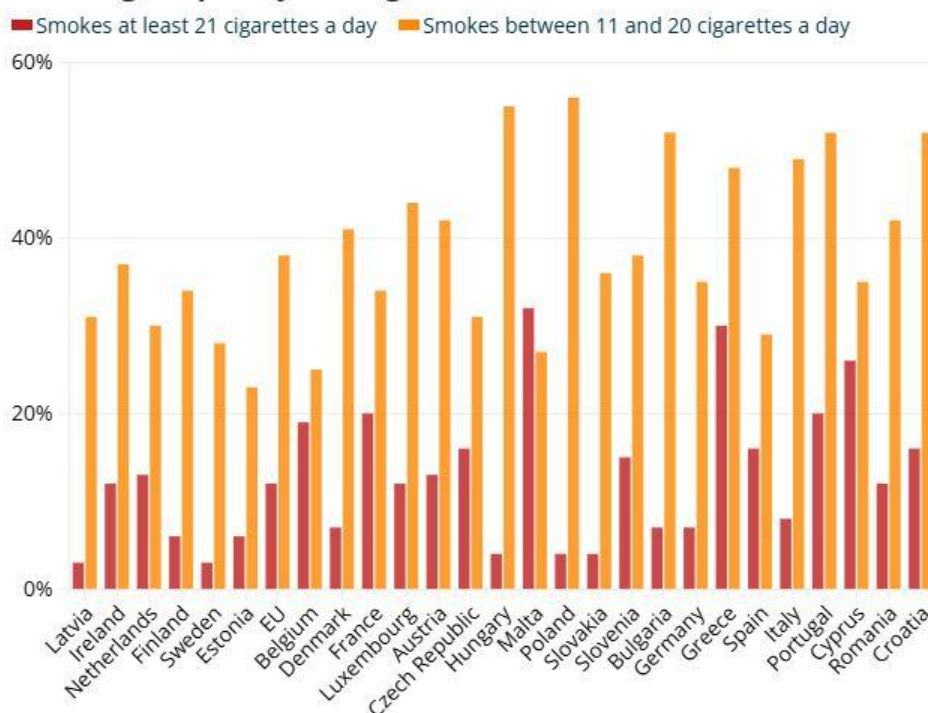
Σχεδόν το ένα τέταρτο των πολιτών της ΕΕ (24%) είναι καπνιστές. Η τελευταία έρευνα του Ευρωβαρόμετρου για το κάπνισμα υπογραμμίζει ότι το ποσοστό των καπνιστών έμεινε σχεδόν αμετάβλητο από το 2020 έως το 2023, με απλώς 1% μείωση. Οι πιο βαρύνες καπνιστές είναι στην Ανατολική και Νοτιοανατολική Ευρώπη, με 37% στη Βουλγαρία, 36% στην Ελλάδα, 35% στην Κροατία και 34% στη Ρουμανία. Η Σουηδία διαθέτει το μικρότερο ποσοστό καπνιστών (8%), με την Ολλανδία να ακολουθεί (11%) και τη Δανία (14%). Οι άνδρες είναι πιο πιθανό να καταναλώνουν προϊόντα καπνού σε σύγκριση με τις γυναίκες (28% έναντι 21%). Ταυτόχρονα, το 56% αναφέρει ότι δεν έχει καπνίσει ποτέ στη ζωή του [6].



Εικόνα 1: Ποσοστά Καπνιστών στην Ε.Ε.

Σχετικά με τα καπνικά προϊόντα, τα πακέτα τσιγάρων συνεχίζουν να είναι τα πιο διαδεδομένα, καθώς το 77% των καπνιστών τα επιλέγει, ενώ το 23% επιλέγει τα στριφτά τσιγάρα. Η ημερήσια κατανάλωση διαφέρει σε μεγάλο βαθμό από χώρα σε χώρα, με τον μέσο όρο να ανέρχεται σε 14 τσιγάρα ανά ημέρα, ποσοστό που παραμένει αμετάβλητο από το 2020. Στην Ελλάδα, το 30% καπνίζει τουλάχιστον 21 τσιγάρα καθημερινά, δηλαδή πάνω από ένα πακέτο, ενώ το 48% καπνίζει 11 με 20 τσιγάρα, που αντιστοιχούν σε μισό με ένα πακέτο. Τα δεδομένα του Ευρωβαρόμετρου αποκαλύπτουν επίσης ότι το ποσοστό των χρηστών ηλεκτρονικών τσιγάρων ανέρχεται μόλις στο 3% [6].

Smoking frequency among EU smokers



Source: Eurobarometer (2023)



Εικόνα 2: Συχνότητα Καπνιστών στην Ε.Ε.

Το κάπνισμα παραμένει ο μεγαλύτερος κίνδυνος για την υγεία που μπορεί να αποφευχθεί και ευθύνεται για 700.000 θανάτους ετησίως μόνο στην ΕΕ. Επιπλέον, περίπου το 50% των καπνιστών πεθαίνουν πρόωρα, κατά μέσο όρο 14 χρόνια νωρίτερα από τους μη καπνιστές [6].

1.1.3 Αντίκτυπος

Οι επιπτώσεις του καπνίσματος είναι πολύ σοβαρές για την υγεία. Το κάπνισμα συνδέεται άμεσα με την ανάπτυξη σοβαρών νοσημάτων όπως ο καρκίνος πνεύμονα, νεφρών και στόματος, καρδιοαγγειακές παθήσεις, εγκεφαλικά επεισόδια και χρόνια αποφρακτική πνευμονοπάθεια (ΧΑΠ)[7]. Ο ΠΟΥ εκτιμά ότι το κάπνισμα προκαλεί πάνω από 8 εκατομμύρια θανάτους ετησίως, εκ των οποίων περίπου 1,2 εκατομμύρια οφείλονται σε παθητικό κάπνισμα [8].

Πέρα από τις βιολογικές συνέπειες, το κάπνισμα προκαλεί σημαντικές οικονομικές και κοινωνικές συνέπειες. Οι δαπάνες για την ιατρική φροντίδα των καπνιστών είναι υψηλότερες, ενώ η μείωση της παραγωγικότητας εξαιτίας ασθενειών και πρόωρης θνησιμότητας συνιστά σοβαρή οικονομική επιβάρυνση για τα συστήματα υγείας. Ανάμεσα σε άλλα, η συνήθεια του καπνίσματος συχνά σχετίζεται με άγχος, κατάθλιψη και διάφορες ψυχολογικές διαταραχές, σχηματίζοντας έναν φαύλο κύκλο που καθιστά την αποχή ιδιαίτερα δύσκολη.

1.1.4 Αιτίες

Η έναρξη και διατήρηση της συνήθειας του καπνίσματος αποτελεί ένα πολυπαραγοντικό φαινόμενο, το οποίο επηρεάζεται από βιολογικούς, ψυχολογικούς, κοινωνικούς και περιβαλλοντικούς παράγοντες. Η κατανόηση αυτών των αιτιών είναι καθοριστικής σημασίας για τον σχεδιασμό αποτελεσματικών στρατηγικών πρόληψης και διακοπής.

Η επιστημονική βιβλιογραφία κατατάσσει τους καπνιστές σε διάφορες κατηγορίες, ανάλογα με τη συχνότητα, τα κίνητρα και το επίπεδο εξάρτησης:

- **Κοινωνικοί Καπνιστές:** Καπνίζουν περιστασιακά, συνήθως σε κοινωνικές περιστάσεις ή για να ενταχθούν σε ένα κοινωνικό σύνολο. Δεν εμφανίζουν συνήθως υψηλό επίπεδο εξάρτησης, αλλά διατρέχουν κίνδυνο να εξελιχθούν σε τακτικούς καπνιστές.
- **Τακτικοί Καπνιστές:** Καπνίζουν καθημερινά και παρουσιάζουν σωματική και ψυχολογική εξάρτηση από τη νικοτίνη. Αυτή η ομάδα εμφανίζει τις μεγαλύτερες δυσκολίες στη διακοπή και συνήθως χρειάζεται υποστήριξη για να απεξαρτηθεί.

Η αναγνώριση του τύπου καπνιστή είναι κρίσιμη για την επιλογή της κατάλληλης μεθόδου διακοπής, καθώς κάθε κατηγορία έχει διαφορετικές ανάγκες και ευαισθησίες

Κοινωνική Επιρροή και Περιβάλλον

Μία από τις πιο συχνές αιτίες που ξεκινούν το κάπνισμα, κυρίως στους εφήβους και νεαρούς ενήλικες, είναι η επιρροή του κοινωνικού τους κύκλου. Οι φίλοι, η οικογένεια και τα πρότυπα (όπως καπνιστές σε ταινίες ή στα social media) δρουν ως ισχυρά πρότυπα προς αντιγραφή. Οι νέοι που βρίσκονται σε περιβάλλον με καπνιστές έχουν μεγαλύτερη πιθανότητα να ξεκινήσουν το κάπνισμα [9]. Επιπλέον, η πολιτισμική αποδοχή του καπνίσματος και η διαφήμιση των καπνικών προϊόντων στο παρελθόν ενίσχυσαν την αντίληψη ότι το κάπνισμα είναι φυσιολογική ή ελκυστική συμπεριφορά, ιδιαίτερα σε άνδρες.

Βιολογική Προδιάθεση

Η νικοτίνη επιδρά στον εγκέφαλο ενεργοποιώντας την εκτόξευση νευροδιαβιβαστών όπως η ντοπαμίνη, που προκαλεί αίσθηση ικανοποίησης και επιβράβευσης. Σε άτομα που έχουν βιολογική τάση για αυξημένη ευαισθησία σε αυτά τα ερεθίσματα, η νικοτίνη μπορεί να δημιουργήσει γρηγορότερη και πιο έντονη εξάρτηση. Επίσης, η παρουσία γενετικών ποικιλιών σε κάποιους ανθρώπους (π.χ. στον υποδοχέα νικοτίνης) έχει συνδεθεί με μεγαλύτερη δυσκολία στην απεξάρτηση και αυξημένο κίνδυνο εθισμού [10].

Ψυχολογικοί Παράγοντες

Το κάπνισμα συχνά λειτουργεί ως μηχανισμός αντιμετώπισης του άγχους, της κατάθλιψης, του στρες ή της πλήξης. Άτομα με διαταραχές, είτε διάθεσης είτε αγχώδεις, παρουσιάζουν υψηλότερα ποσοστά καπνίσματος και μεγαλύτερη εξάρτηση. Σε περιόδους έντονης συναισθηματικής πίεσης ή προσωπικών δυσκολιών, το κάπνισμα χρησιμοποιείται ως μέσο

ανακούφισης και εκτόνωσης, παρότι η μακροχρόνια του χρήση ενδέχεται να επιδεινώνει τις ψυχικές καταστάσεις.

Διάθεση για Πειραματισμό και Περιέργεια

Πολλοί έφηβοι ξεκινούν το κάπνισμα από περιέργεια ή την επιθυμία να δοκιμάσουν κάτι απαγορευμένο ή ενήλικο. Η ηλικία έναρξης του καπνίσματος είναι κρίσιμος παράγοντας, καθώς η πρώιμη έκθεση αυξάνει σημαντικά την πιθανότητα ανάπτυξης εξάρτησης και μακροχρόνιας χρήσης.

Οικονομικοί και Επαγγελματικοί Παράγοντες

Τα άτομα που εργάζονται σε περιβάλλοντα με υψηλό στρες, βάρδιες ή χαμηλή κοινωνική υποστήριξη εμφανίζουν αυξημένη πιθανότητα να ξεκινήσουν το κάπνισμα. Το ίδιο ισχύει και για άτομα με χαμηλότερο μορφωτικό ή οικονομικό επίπεδο, γεγονός που συνδέεται με περιορισμένη πρόσβαση σε ενημέρωση και υπηρεσίες απεξάρτησης.

Ελλιπής Ενημέρωση και Παραπληροφόρηση

Σε ορισμένες περιπτώσεις, το κάπνισμα ξεκινά λόγω έλλειψης επαρκούς πληροφόρησης για τις συνέπειες του. Αν και οι περισσότεροι γνωρίζουν ότι το κάπνισμα είναι επιβλαβές, πολλοί υποτιμούν τον βαθμό του κινδύνου ή θεωρούν ότι θα το κόψουν εύκολα όταν χρειαστεί. Η παραπλανητική διαφήμιση στο παρελθόν, καθώς και η ύπαρξη προϊόντων που εμφανίζονται ως «λιγότερο βλαβερά» (π.χ. light ή ηλεκτρονικά τσιγάρα), συμβάλλουν επίσης στην υποβάθμιση των κινδύνων.

1.2 Η Σημασία της Απεξάρτησης

Η διακοπή του καπνίσματος αποτελεί μία από τις πιο ωφέλιμες παρεμβάσεις για τη βελτίωση της υγείας σε ατομικό και πληθυσμιακό επίπεδο. Ανεξαρτήτως ηλικίας ή διάρκειας καπνιστικής συνήθειας, τα οφέλη από τη διακοπή είναι σημαντικά και εμφανίζονται τόσο άμεσα όσο και μακροπρόθεσμα. Η σημασία της διακοπής επεκτείνεται σε πολλαπλούς τομείς: σωματική και ψυχική υγεία, κοινωνική και οικονομική ευημερία, καθώς και προστασία του περιβάλλοντος.

Βελτίωση της Σωματικής Υγείας

Η διακοπή του καπνίσματος μειώνει δραστικά τον κίνδυνο πρόωρου θανάτου και νοσηρότητας από πλήθος ασθενειών. Ο Παγκόσμιος Οργανισμός Υγείας αναφέρει ότι το κάπνισμα συνδέεται με πάνω από 25 διαφορετικά νοσήματα, τα περισσότερα εκ των οποίων μπορούν να προληφθούν με τη διακοπή [11].

- **Καρκίνος:** Η διακοπή μειώνει σημαντικά τον κίνδυνο για καρκίνο του πνεύμονα, του λάρυγγα, της ουροδόχου κύστης και άλλων οργάνων [12].
- **Καρδιαγγειακά νοσήματα:** Μέσα σε μόλις 1 χρόνο από τη διακοπή, ο κίνδυνος εμφάνισης στεφανιαίας νόσου μειώνεται στο μισό σε σχέση με έναν ενεργό καπνιστή [13].
- **Αναπνευστικά προβλήματα:** Η λειτουργία των πνευμόνων βελτιώνεται σταδιακά και τα συμπτώματα όπως χρόνιος βήχας ή δύσπνοια μειώνονται αισθητά [14].
- **Σακχαρώδης διαβήτης:** Η διακοπή σχετίζεται με καλύτερο γλυκαιμικό έλεγχο και μειωμένο κίνδυνο για μικρό και μακρό αγγειοπάθειες [15].

Ψυχολογική Ευεξία και Ποιότητα Ζωής

Ενώ πολλοί καπνιστές ισχυρίζονται ότι το τσιγάρο τους βοηθά να ανακουφιστούν από το άγχος, πολλές έρευνες αποδεικνύουν ότι η διακοπή του καπνίσματος σχετίζεται με καλύτερη ψυχική ευεξία. Μέσω της διακοπής, μειώνεται το άγχος και τα συμπτώματα κατάθλιψης σε μεσοπρόθεσμο διάστημα, ενώ ενισχύεται η αίσθηση ελέγχου και η αυτοεκτίμηση [16]. Επίσης, βελτιώνεται η ποιότητα ύπνου και η συγκέντρωση. Τέλος, η καθημερινότητα γίνεται πιο ευχάριστη, καθώς οι πρώην καπνιστές ανακτούν την αίσθηση της όσφρησης και της γεύσης.

Οικονομικά Οφέλη

Το κάπνισμα αποτελεί μία ιδιαίτερα δαπανηρή συνήθεια. Ένας καπνιστής που καταναλώνει ένα πακέτο ημερησίως μπορεί να εξοικονομήσει χιλιάδες ευρώ ετησίως διακόπτοντας το. Το κόστος δεν περιορίζεται μόνο στην αγορά τσιγάρων, αλλά περιλαμβάνει και την ιατρική περίθαλψη λόγω νοσημάτων που σχετίζονται με το αυτό. Οι ασφαλιστικές εισφορές και οι παροχές ζωής είναι συχνά φθηνότερες για μη καπνιστές. Επιπλέον, βελτιώνεται και η παραγωγικότητα στον εργασιακό χώρο, καθώς περιορίζονται τα διαλείμματα και οι ημέρες απουσίας λόγω ασθενειών.

Προστασία Οικείων και Μείωση Παθητικού Καπνίσματος

Η διακοπή του καπνίσματος δεν προστατεύει μόνο τον ίδιο τον καπνιστή αλλά και το περιβάλλον του, ιδιαίτερα τα παιδιά και τις εγκύους. Το παθητικό κάπνισμα έχει αποδειχθεί ότι προκαλεί καρκίνο, άσθμα, βρογχίτιδα και συνδέεται με ξαφνικό βρεφικό θάνατο [17]. Η απομάκρυνση του καπνίσματος από το σπίτι, το αυτοκίνητο και τους χώρους εργασίας βελτιώνει σημαντικά την υγεία των μη καπνιστών και συμβάλλει στην ευρύτερη κουλτούρα της πρόληψης.

Περιβαλλοντικά Οφέλη

Τα προϊόντα καπνού και τα απορρίμματά τους έχουν σοβαρές συνέπειες στο περιβάλλον. Κάθε χρόνο απορρίπτονται πάνω από 4,5 τρισεκατομμύρια αποτσίγαρα, τα οποία περιέχουν τοξικά υλικά που ρυπαίνουν το νερό και το έδαφος. Επίσης, η καλλιέργεια καπνού συμβάλλει στην αποψίλωση δασών, χρήση χημικών και απώλεια βιοποικιλότητας. Έτσι, η διακοπή του καπνίσματος μειώνοντας τη ζήτηση αυτών των προϊόντων, ενισχύει την περιβαλλοντική βιωσιμότητα.

1.3 Ψηφιακές Προκλήσεις και Ανάγκες

Καθώς η τεχνολογία ενσωματώνεται ολοένα και περισσότερο στην καθημερινότητα των ανθρώπων, προκύπτει μια νέα προοπτική στην προσέγγιση προβλημάτων δημόσιας υγείας, όπως η διακοπή του καπνίσματος. Οι ψηφιακές εφαρμογές και τα ψηφιακά εργαλεία υγείας (digital health tools) προσφέρουν καινοτόμες λύσεις, αλλά συνοδεύονται από συγκεκριμένες προκλήσεις και ανάγκες, τόσο από την πλευρά των χρηστών όσο και των επαγγελματιών υγείας.

Οι χρήστες εφαρμογών διακοπής καπνίσματος δεν αποτελούν μια ομοιογενή ομάδα. Διαφέρουν ως προς την ηλικία, το φύλο, την ψυχολογική κατάστασή τους, τη διάρκεια καπνίσματος, το κίνητρο διακοπής και τις προηγούμενες προσπάθειες τους για απεξάρτηση. Επομένως, οι ψηφιακές λύσεις πρέπει να είναι εξατομικευμένες, προσαρμοσμένες στις ανάγκες κάθε χρήστη, για να έχουν ουσιαστικό αντίκτυπο. Πολλές υπάρχουσες εφαρμογές

αποτυγχάνουν να προσφέρουν δυναμικό, προσωποποιημένο περιεχόμενο, βασιζόμενες σε στατικό υλικό χωρίς αναγνώριση των ιδιαιτεροτήτων του κάθε χρήστη.

Παρόλο που η πρόσβαση στην τεχνολογία είναι ευρεία και διαδεδομένη, υπάρχουν ακόμη ομάδες ατόμων με περιορισμένη πρόσβαση σε υπηρεσίες δικτύου. Το ψηφιακό χάσμα είναι εμφανές κυρίως σε ηλικιωμένα άτομα, τα οποία συνήθως δεν είναι εξοικειωμένα με τις ψηφιακές εφαρμογές και σε άτομα με χαμηλότερο κοινωνικοοικονομικό υπόβαθρο, που ίσως δεν έχουν σταθερή πρόσβαση στο διαδίκτυο ή σε έξυπνες συσκευές. Επιπλέον, ακόμα και όταν υπάρχει πρόσβαση, η έλλειψη ψηφιακού αλφαριθμητισμού (digital literacy) δυσχεραίνει τη σωστή χρήση των εφαρμογών. Αυτό εντείνει την ανάγκη για απλά, φιλικά προς τον χρήστη περιβάλλοντα και διεπαφές.

Η αδυναμία διατήρησης της αλληλεπίδρασης του χρήστη είναι μια από τις μεγαλύτερες προκλήσεις των ψηφιακών εφαρμογών. Πολλοί χρήστες εγκαθιστούν την εφαρμογή, αλλά τη χρησιμοποιούν για μικρό χρονικό διάστημα, με αποτέλεσμα να μην επιτυγχάνεται η επιθυμητή αλλαγή συμπεριφοράς. Για την αύξηση της συμμόρφωσης (adherence), είναι απαραίτητο να χρησιμοποιούνται μηχανισμοί παιχνιδοποίησης (Gamification) (π.χ. πόντοι, επιτεύγματα, πίνακες κατάταξης) [18]. Επιπλέον, απαραίτητη είναι και η χρήση ειδοποιήσεων (push notifications) με προσεκτικά επιλεγμένα και όχι ενοχλητικά μηνύματα και συναισθηματική υποστήριξη (π.χ. chatbot, κοινότητες υποστήριξης, αλληλεπίδραση με άλλους πρώην καπνιστές).

Επίσης, με την αυξανόμενη χρήση εφαρμογών υγείας, προκύπτουν σοβαρές ανησυχίες ως προς την προστασία προσωπικών δεδομένων των χρηστών. Οι εφαρμογές συλλέγουν ευαίσθητες πληροφορίες, όπως ιατρικό ιστορικό, συνήθειες καπνίσματος, τοποθεσία, ηλικία και φύλο. Η μη συμμόρφωση με κανονισμούς όπως ο GDPR (General Data Protection Regulation) στην Ευρωπαϊκή Ένωση μπορεί να οδηγήσει σε σημαντικές παραβιάσεις ιδιωτικότητας. Οι χρήστες απαιτούν διαφάνεια σχετικά με τη χρήση των δεδομένων τους και τη δυνατότητα διαγραφής τους.

Επιπλέον, αρκετές εφαρμογές που κυκλοφορούν στα ηλεκτρονικά καταστήματα (Google Play, AppStore) δεν βασίζονται σε επιστημονικά τεκμηριωμένα δεδομένα. Αυτό είναι ιδιαίτερα προβληματικό όταν προτείνονται μέθοδοι διακοπής ή όταν δίνονται οδηγίες που δεν είναι σύμφωνες με τις κατευθυντήριες οδηγίες οργάνων δημόσιας υγείας [19]. Η ανάγκη για συνεργασία με ιατρικούς φορείς και επαγγελματίες υγείας είναι κρίσιμη ώστε το περιεχόμενο να είναι έγκυρο και ασφαλές για τον χρήστη.

Τέλος, ένα από τα πιο σοβαρά κενά που παρατηρούνται στις περισσότερες ψηφιακές παρεμβάσεις είναι η απουσία συστηματικής αξιολόγησης της αποτελεσματικότητάς τους. Ελάχιστες εφαρμογές έχουν υποβληθεί σε κλινικές δοκιμές ή σε μακροχρόνια αξιολόγηση της επίδρασής τους στη διακοπή καπνίσματος [20]. Η ανάγκη για τεκμηριωμένη τεχνολογία (evidence-based technology) είναι επιτακτική, προκειμένου να ενισχυθεί η εμπιστοσύνη των χρηστών και να καθιερωθεί η εφαρμογή ως αξιόπιστο εργαλείο υποστήριξης.

1.4 Ορισμός του Προβλήματος και Ερευνητικές Υποθέσεις

Στον σύγχρονο ψηφιακό κόσμο, εφαρμογές κινητών τηλεφώνων (mobile health apps) έχουν αναδειχθεί ως νέα μέσα παρέμβασης, αξιοποιώντας τεχνολογικά εργαλεία όπως ειδοποιήσεις, η παρακολούθηση προόδου, τα κίνητρα μέσω παιχνιδοποίησης (gamification),

και η προσωποποιημένη υποστήριξη. Ωστόσο, η πλειονότητα αυτών των εφαρμογών δεν είναι βασισμένη σε επιστημονικές κατευθυντήριες γραμμές, παρουσιάζει χαμηλά ποσοστά διατήρησης χρήσης και στερείται ουσιαστικής προσαρμογής στις ανάγκες του εκάστοτε χρήστη.

Επιπλέον, παραμένει περιορισμένη η κατανόηση του τρόπου με τον οποίο οι χρήστες αλληλοεπιδρούν με τις ψηφιακές εφαρμογές, καθώς και του αντίκτυπου που έχει η χρήση τέτοιων εφαρμογών στην ψυχολογική ενίσχυση, την αποτελεσματικότητα και την τελική επιτυχία διακοπής. Στο πλαίσιο αυτό, το πρόβλημα που επιχειρείται να διερευνηθεί στην παρούσα εργασία είναι:

Πώς μπορεί μια ψηφιακή εφαρμογή διακοπής καπνίσματος, βασισμένη σε τεκμηριωμένες πρακτικές, προσωποποιημένα δεδομένα και μηχανισμούς παρακίνησης, να επηρεάσει θετικά την πορεία διακοπής του καπνίσματος και την εμπλοκή του χρήστη στη διαδικασία;

Σε σχετικές μελέτες, προτάθηκαν συστηματικά πλαίσια για την αναθεώρηση και την ανάλυση εφαρμογών κινητών τηλεφώνων, που έχουν σχεδιαστεί για τη διακοπή του καπνίσματος. Οι έρευνες ανέδειξαν περιορισμούς στην προσβασιμότητα και τις δυνατότητες μεταξύ των υφιστάμενων εφαρμογών, επισημαίνοντας την ανάγκη για πιο ολοκληρωμένες και τεκμηριωμένες εφαρμογές [21].

Συνεπώς, η εργασία αυτή αποσκοπεί στην επίλυση αυτών των προβλημάτων, εισάγοντας τις εξής ερευνητικές υποθέσεις:

- Η χρήση μιας ψηφιακής εφαρμογής που προσφέρει προσωποποιημένα δεδομένα, ψυχολογική υποστήριξη και στοιχεία παιχνιδοποίησης αυξάνει την πιθανότητα επιτυχούς διακοπής του καπνίσματος.
- Οι χρήστες που αλληλοεπιδρούν τακτικά με τις λειτουργίες της εφαρμογής (π.χ. προβολή επιτευγμάτων, αναπνευστικές ασκήσεις, ειδοποιήσεις) παρουσιάζουν υψηλότερα επίπεδα αποτελεσματικότητας σε σχέση με όσους δεν τις χρησιμοποιούν συχνά.
- Η ύπαρξη πίνακα κατάταξης και συστήματος επιβράβευσης ενισχύει το κίνητρο του χρήστη και παρατείνει τη χρήση της εφαρμογής σε βάθος χρόνου.
- Οι χρήστες που καταγράφουν οικονομικά και υγειονομικά οφέλη (π.χ. χρήματα που εξοικονομήθηκαν, βελτίωση της κυκλοφορίας του αίματος) αισθάνονται μεγαλύτερη πρόοδο και ενισχύεται η προσπάθειά τους.
- Η κοινωνική διάσταση της εφαρμογής (π.χ. προσθήκη φίλων, ομαδική συνομιλία) σχετίζεται θετικά με την αίσθηση δέσμευσης και τη μείωση υποτροπών.

Αυτές οι υποθέσεις θα αξιολογηθούν τόσο ποιοτικά όσο και ποσοτικά, μέσα από δεδομένα χρήσης, ερωτηματολόγια, και ανάλυση αλληλεπίδρασης με την εφαρμογή.

1.5 Η Δομή της Εργασίας

Η παρούσα εργασία είναι οργανωμένη σε εννέα κύρια κεφάλαια, καθένα από τα οποία συμβάλλει στη σφαιρική προσέγγιση του προβλήματος της διακοπής καπνίσματος μέσω ψηφιακής εφαρμογής, από τη θεωρητική τεκμηρίωση έως την τελική υλοποίηση και αξιολόγηση.

Στο **Κεφάλαιο 1** παρουσιάζεται η εισαγωγή στο πρόβλημα του καπνίσματος, η σημασία της διακοπής του, οι ψηφιακές προκλήσεις και ανάγκες, καθώς και οι ερευνητικές υποθέσεις που τίθενται.

Το **Κεφάλαιο 2** εστιάζει στη βιβλιογραφική ανασκόπηση, παρουσιάζοντας το σχετικό υπόβαθρο για το κάπνισμα, τις υφιστάμενες εφαρμογές, τις ψηφιακές τεχνολογίες υγείας (mHealth) και τη χρήση προηγμένων τεχνολογιών όπως τα chatbots και τα συστήματα RAG.

Στο **Κεφάλαιο 3** περιγράφονται αναλυτικά οι στόχοι της εφαρμογής, το όραμα, τα υποσυστήματα, οι λειτουργικοί περιορισμοί και τα αναμενόμενα αποτελέσματα. Επίσης, αναλύονται οι απαιτήσεις του συστήματος, ο ρόλος των διαφορετικών τύπων χρηστών και οι λειτουργικές και μη λειτουργικές απαιτήσεις,

Το **Κεφάλαιο 4** παρουσιάζει τις τεχνολογίες και τα εργαλεία που χρησιμοποιήθηκαν για την ανάπτυξη της εφαρμογής, όπως Flutter, Firebase, Python, Langchain, RAG κτλ.

Στο **Κεφάλαιο 5** αναπτύσσεται η σχεδίαση της λύσης με βάση την αρχιτεκτονική client server και την ανάλυση των επιμέρους υποσυστημάτων. Η σχεδίαση της λύσης και της αρχιτεκτονικής ενισχύεται με μοντελοποίηση με UML διαγράμματα (Activity, Sequence, Class, Component, ER και Deployment).

Στο **Κεφάλαιο 6** περιγράφεται λεπτομερώς η υλοποίηση της εφαρμογής, με αναφορές στον κώδικα, τις οθόνες, τη διασύνδεση με Firebase, την υλοποίηση του chatbot, των επιτευγμάτων και όλων των λειτουργιών της εφαρμογής.

Το **Κεφάλαιο 7** αφορά την παρουσίαση της λειτουργικότητας του συστήματος με σενάρια χρήσης, screenshots, ροές δεδομένων, διαχείριση σφαλμάτων.

Τέλος, το **Κεφάλαιο 8** περιλαμβάνει την αξιολόγηση του συστήματος μέσω δοκιμών, ανάλυσης UX, απόδοσης και ποιοτικών/ποσοτικών στοιχείων, ενώ καταλήγει σε συμπεράσματα και προτάσεις για μελλοντική εξέλιξη της εφαρμογής.

Στο τέλος της εργασίας παρατίθενται ο σύνδεσμος για τον πηγαίο κώδικα της εφαρμογής και οι σχετικές βιβλιογραφικές παραπομπές.

2. Σχετικό Υπόβαθρο και Βιβλιογραφική Ανασκόπηση

Η ραγδαία τεχνολογική πρόοδος και η αυξανόμενη ανάγκη για αποδοτικές παρεμβάσεις στον τομέα της δημόσιας υγείας, έχουν οδηγήσει στην ανάδειξη νέων ψηφιακών εργαλείων υποστηρίζοντας την διακοπή καπνίσματος. Το κεφάλαιο αυτό εξετάζει το σχετικό υπόβαθρο, παρουσιάζοντας αρχικά την κοινωνική διάσταση του φαινομένου του καπνίσματος. Στη συνέχεια, παρουσιάζεται ο ρόλος των ψηφιακών εφαρμογών στην αλλαγή συμπεριφοράς του χρήστη, καθώς και η αξιολόγηση της αποτελεσματικότητάς τους βάσει της υπάρχουσας βιβλιογραφίας και σχετικών ερευνών και μελετών.

2.1 Ο Ρόλος των Ψηφιακών Εφαρμογών

Με την εξάπλωση των smartphones και της πρόσβασης στο διαδίκτυο, οι ψηφιακές εφαρμογές έχουν αναδειχθεί σε σύγχρονα εργαλεία προώθησης της υγείας. Στον τομέα της διακοπής καπνίσματος, έχουν δημιουργηθεί πολυάριθμες εφαρμογές. Αυτές στοχεύουν στην ενίσχυση της κινητοποίησης του χρήστη, την παροχή πληροφορίας και τη διαχείριση υποτροπών. Τα ψηφιακά εργαλεία πλεονεκτούν έναντι των παραδοσιακών παρεμβάσεων, καθώς προσφέρουν προσωποποιημένες εμπειρίες και λειτουργούν σε πραγματικό χρόνο. Επιπλέον, είναι διαθέσιμα οποιαδήποτε στιγμή και ενισχύουν την αίσθηση αυτονομίας του χρήστη. Επίσης, επιτρέπουν την καταγραφή δεδομένων, οπτικοποιούν την πρόοδο με επιτεύγματα, ενισχύουν το κίνητρο μέσω υπενθυμίσεων και στοιχείων παιχνιδοποίησης.

Επιστημονικές μελέτες δείχνουν ότι εφαρμογές οι οποίες ενσωματώνουν θεωρητικά μοντέλα αλλαγής συμπεριφοράς (όπως το Transtheoretical Model ή το COM-B Model) παρουσιάζουν μεγαλύτερη αποτελεσματικότητα [22]. Επιπλέον, η χρήση ψηφιακών βοηθών (chatbots), που προσφέρουν ψυχολογική υποστήριξη μέσω τεχνητής νοημοσύνης και προσωποποιημένων ειδοποιήσεων φαίνεται να αυξάνει τα ποσοστά επιτυχούς διακοπής [23].

Ωστόσο, υπάρχουν σημαντικές προκλήσεις όπως η εμπιστοσύνη του χρήστη, η διασφάλιση της ιδιωτικότητας και η αποφυγή της πληθώρας ημιτελών ή χαμηλής ποιότητας εφαρμογών χωρίς επιστημονική τεκμηρίωση. Η επιλογή εργαλείων που βασίζονται σε αποδεδειγμένα πρότυπα και προσφέρουν ποιοτικό περιεχόμενο κρίνεται κρίσιμη για την ενίσχυση της αποτελεσματικότητας τέτοιων παρεμβάσεων.

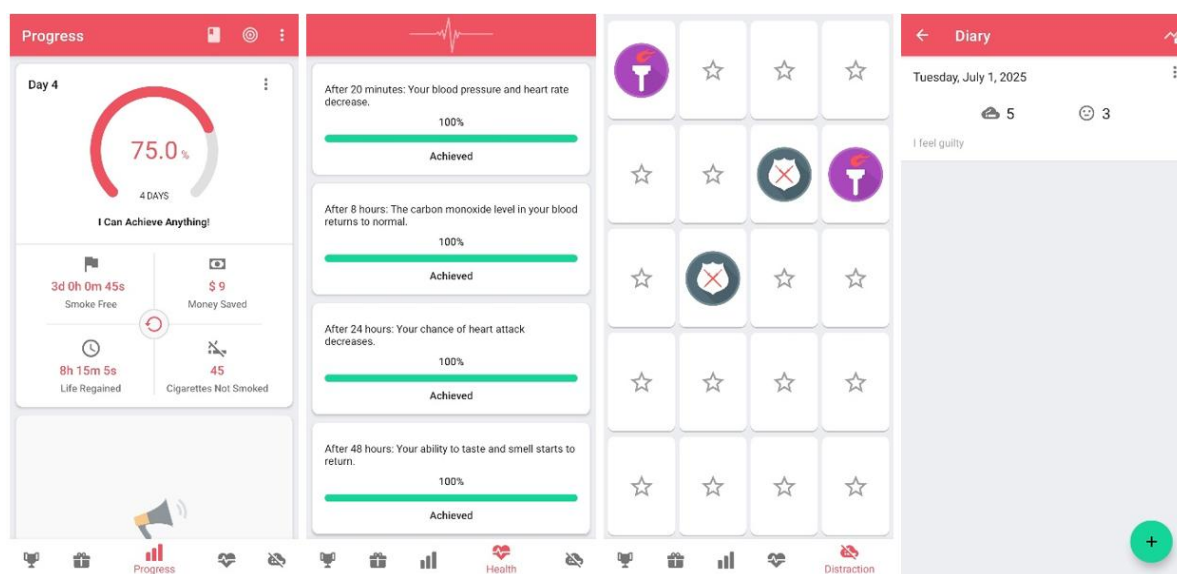
2.2 Υπάρχουσες Εφαρμογές

Τα τελευταία χρόνια ο χώρος των ψηφιακών εφαρμογών έχει κατακλυστεί από πλήθος εφαρμογών διακοπής καπνίσματος, με διαφορές στην ποιότητα, την επιστημονική τεκμηρίωση και τη χρηστικότητα. Σύμφωνα με πρόσφατη έρευνα, περισσότερες από 500 εφαρμογές σχετικές με το κάπνισμα είναι διαθέσιμες τόσο σε Android όσο και σε iOS καταστήματα, αλλά λίγες εξ αυτών βασίζονται σε επιστημονικές προσεγγίσεις. Ορισμένες από τις πιο διαδεδομένες εφαρμογές παγκοσμίως είναι:

Quit Tracker

Το Quit Tracker είναι μια καλά αναγνωρισμένη εφαρμογή που βοηθά τους χρήστες να παρακολουθούν την πρόοδό τους στη διακοπή του καπνίσματος. Προσφέρει λειτουργίες όπως υπολογισμούς εξοικονόμησης χρημάτων, χρόνο ζωής που δεν χάνεται, ενώ περιλαμβάνει και ενθαρρυντικές ειδοποιήσεις.

Η εφαρμογή παρακολουθεί διάφορες μετρήσεις, όπως εξοικονόμηση χρημάτων, κερδισμένο χρόνο ζωής και πρόοδο προς την απεξάρτηση, παρέχοντας στους χρήστες πολύτιμες πληροφορίες σχετικά με τις συνήθειες τους. Λειτουργίες όπως οι υπενθυμίσεις και τα στατιστικά στοιχεία της υγείας τους, μπορούν να βοηθήσουν τους χρήστες να παραμείνουν σε καλό δρόμο και να αισθάνονται ικανοποίηση για την πρόοδό τους. Επιπλέον, πολλοί χρήστες βρίσκουν την εφαρμογή εύκολη στην πλοήγηση και τη χρήση. Η εφαρμογή ενσωματώνει στοιχεία της Cognitive Behavioral Therapy (CBT) με σύγχρονο και προσιτό τρόπο.

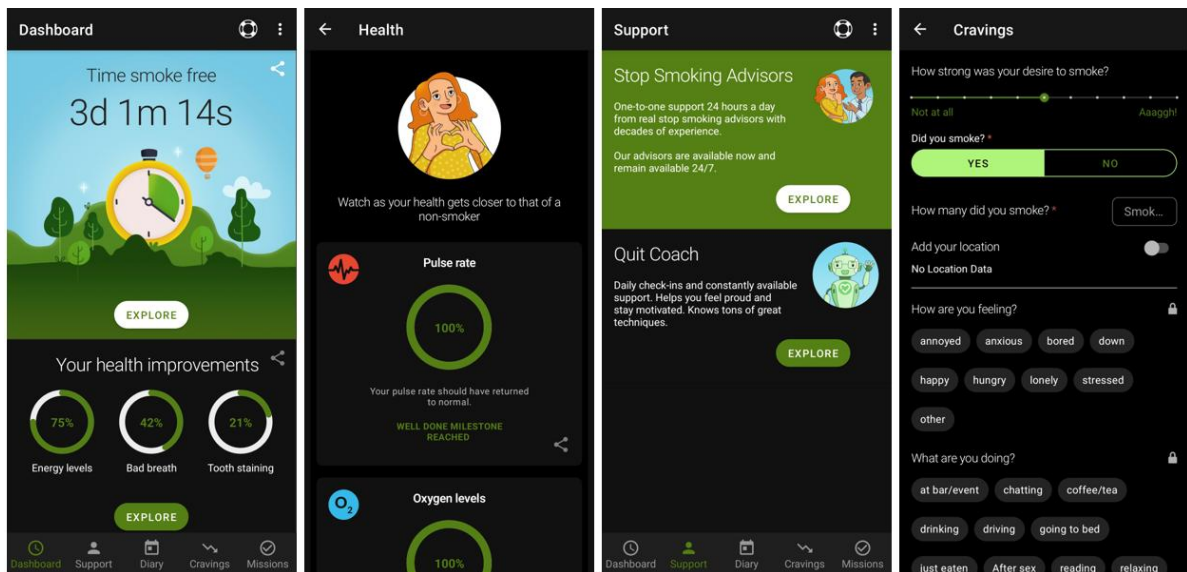


Εικόνα 3: Οθόνες Εφαρμογής Quit Tracker

Ωστόσο, η δωρεάν έκδοση της εφαρμογής περιέχει διαφημίσεις που είναι ενοχλητικές για την πλειοψηφία των χρηστών. Επιπλέον, οι επιλογές και οι δυνατότητες της εφαρμογής είναι αρκετά περιορισμένες, αφού οι λειτουργίες περιλαμβάνουν μόνο τις βασικές και τις συνηθισμένες σε τέτοιου είδους εφαρμογές, χωρίς να έχει επιπλέον επιλογές για τον χρήστη (ή και περισσότερες λεπτομέρειες όσον αφορά τους στόχους, την υγεία κτλ.), όπου θα την διαφοροποιήσει από τις άλλες.

Smoke Free

Οι χρήστες εκτιμούν την διεπαφή παρακολούθησης προόδου, συμπεριλαμβανομένων των ημερών από την διακοπή, των χρημάτων που εξοικονομήθηκαν και των βελτιώσεων στην υγεία. Η εφαρμογή περιλαμβάνει λειτουργίες όπως καθημερινές αποστολές, ημερολόγιο και chatbot για να βοηθήσει τους χρήστες να παραμείνουν επικεντρωμένοι στους στόχους τους. Η δυνατότητα παρακολούθησης της προόδου της υγείας, όπως ο μειωμένος κίνδυνος καρδιακών παθήσεων, αποτελεί θετικό κίνητρο για πολλούς χρήστες. Ορισμένοι χρήστες έχουν βρει το κοινωνικό κομμάτι της εφαρμογής και την ικανότητά της να συνδέεται με άλλους χρήσιμη στην προσπάθεια απεξάρτησης τους.



Εικόνα 4: Οθόνες Εφαρμογής Smoke Free

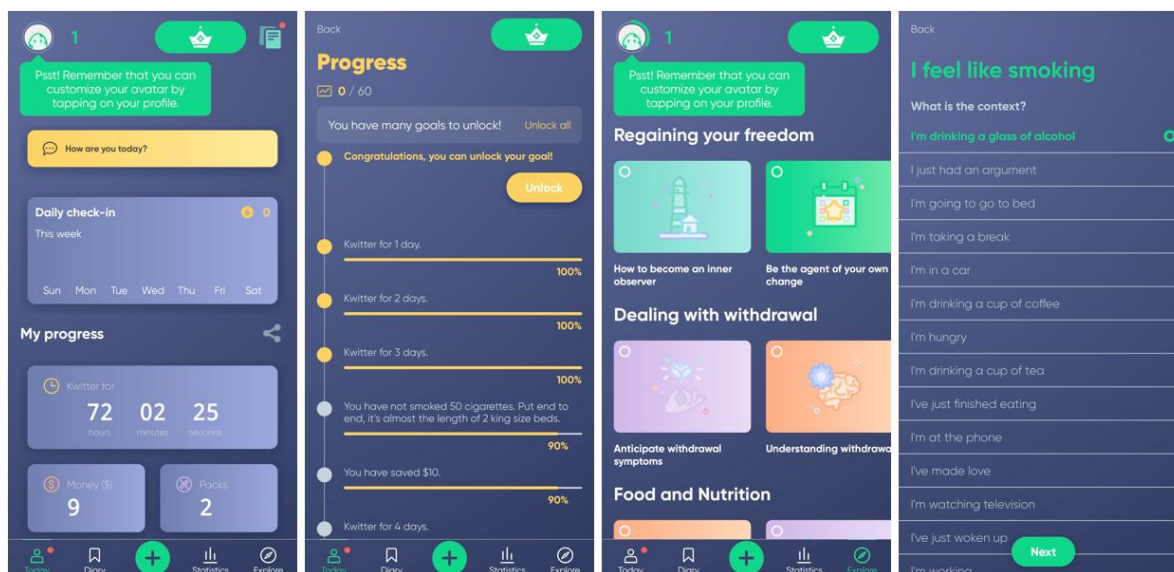
Παρ' όλα αυτά, ορισμένοι χρήστες βρίσκουν περιορισμένες τις επιλογές εξατομίκευσης της εφαρμογής, ιδιαίτερα όσον αφορά την προσαρμογή του κόστους του τσιγάρου μετά την αρχική ρύθμιση. Μερικοί χρήστες έχουν αναφέρει προβλήματα με την επαναφορά της προόδου όταν σκόπευαν να συνεχίσουν να παρακολουθούν την πρόοδο τους μετά από υποτροπή στο κάπνισμα. Επίσης, η εφαρμογή, περιλαμβάνει ελάχιστες επιλογές για την παρακολούθηση των στόχων, συμπεριλαμβανομένης και της δυνατότητας προσθήκης προσαρμοσμένων συναισθημάτων και triggers. Τέλος, οι περισσότερες λειτουργίες της εφαρμογής, μαζί με τις πιο σημαντικές (στόχοι, συμβουλές, βραβεία, chatbot), είναι επιπληρωμή στο premium plan. Στο free plan οι δυνατότητες και η βοήθεια της εφαρμογής είναι αρκετά περιορισμένη.

Kwit

Το Kwit ενσωματώνει παιχνιδοποιημένα στοιχεία όπως ορόσημα, επίπεδο και ξεκλείδωμα προκλήσεων, τα οποία μπορεί να δίνουν κίνητρο στους χρήστες. Η εφαρμογή παρέχει κάρτες με κίνητρα και στόχους, καθώς και τεχνικές αναπνοής για να βοηθήσει τους χρήστες να αντιμετωπίσουν την επιθυμία τους για κάπνισμα και να συνεχίσουν την καλή τους πορεία. Οι χρήστες μπορούν να παρακολουθούν το χρόνο από τη διακοπή, τα χρήματα που εξοικονομούνται και τις βελτιώσεις στην υγεία. Οι χρήστες βρίσκουν την εφαρμογή χρήσιμη στην επιβράβευση στόχων που επιτεύχθηκαν και στην κοινή παρακολούθηση της προόδου τους με φίλους τους. Το Kwit βασίζεται σε Cognitive Behavioral Therapy και άλλες επιστημονικά αναγνωρισμένες μεθόδους για τη θεραπεία του εθισμού.

Ορισμένοι χρήστες θεωρούν ότι το κόστος της έκδοσης premium είναι απαγορευτικό. Η λειτουργία καθημερινού check-in της εφαρμογής επικεντρώνεται σε μεγάλο βαθμό στα αρνητικά συναισθήματα, τα οποία μπορεί να είναι αποθαρρυντικά. Η δωρεάν έκδοση της εφαρμογής είναι πολύ βασική και δεν διαθέτει τις προηγμένες δυνατότητες της έκδοσης premium. Τέλος, η εφαρμογή περιλαμβάνει την επιλογή να καπνίζετε με επίγνωση κατά τη διάρκεια των στερητικών, η οποία είναι αντιπαραγωγική για το στόχο της απεξάρτησης.

Το Kwit είναι μια καλά αναγνωρισμένη εφαρμογή με έντονη εστίαση στη ενθάρρυνση και την παρακολούθηση της προόδου. Ωστόσο, τα premium χαρακτηριστικά του είναι ακριβά και ορισμένοι χρήστες βρίσκουν την προσέγγισή του, στα αρνητικά συναισθήματα και στα στερητικά, προβληματική.

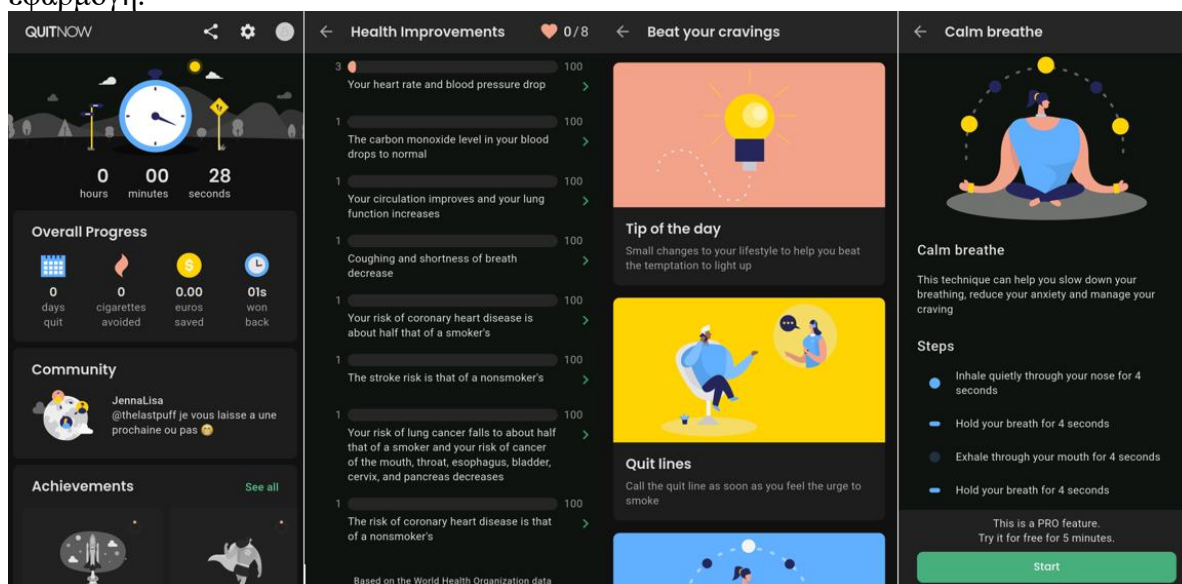


Εικόνα 5: Οθόνες Εφαρμογής Kwit

QuitNow

Το ενεργό φόρουμ κοινότητας της εφαρμογής παρέχει στους χρήστες έναν χώρο για να συνδεθούν, να μοιραστούν εμπειρίες και να προσφέρουν ενθάρρυνση. Η εφαρμογή παρακολουθεί βασικές μετρήσεις, όπως ημέρες χωρίς καπνό, εξοικονόμηση χρημάτων και βελτιώσεις στην υγεία, οι οποίες δίνουν κίνητρο στους χρήστες. Οι χρήστες μπορούν να προσαρμόσουν τις ρυθμίσεις της εφαρμογής, τις ειδοποιήσεις, ενώ περιλαμβάνει και home screen widgets. Η εφαρμογή προσφέρει μια ολοκληρωμένη ενότητα FAQ με βάση τα ευρήματα του Παγκόσμιου Οργανισμού Υγείας και ένα πολύ βασικό και απλό chatbot για να απαντήσει σε συγκεκριμένες ερωτήσεις. Η δωρεάν έκδοση προσφέρει μια ποικιλία χαρακτηριστικών.

Ορισμένοι χρήστες θεωρούν ότι το φόρουμ της κοινότητας αποσπά την προσοχή και είναι γεμάτο με άσχετο περιεχόμενο. Επιπλέον, η εφαρμογή θα μπορούσε να επωφεληθεί από πιο εμπειρισταωμένες συμβουλές για τη διαχείριση των στερητικών και συγκεκριμένων τεχνικών διακοπής, όπως η θεραπεία αντικατάστασης νικοτίνης. Ορισμένοι χρήστες αναφέρουν προβλήματα με το UI της εφαρμογής που δεν συγχρονίζεται σωστά με την κύρια εφαρμογή.



Εικόνα 6: Οθόνες Εφαρμογής QuitNow

Πίνακας 1: Λειτουργίες Δημοφιλών Εφαρμογών Διακοπής Καπνίσματος

Εφαρμογή	Quit Tracker	QuitNow	Smoke Free	Kwit
Βαθμολογία PlayStore	4,8	4,5	4,8	4,8
Google Adds	✓	✓	✓	✓
Premium Plan	✓	✓	✓	✓
ChatBot	-	✓	✓	-
Leaderboard	-	-	✓	-
Achievements	✓	✓	✓	✓
Distraction Games	✓	-	-	-
Contact with Experts	-	✓	✓	-
Gamification Features	-	-	-	✓
Cravings Diary	✓	✓	✓	✓
Breathing Techniques	-	✓	-	-
Community Chat	-	✓	-	-
Friends	-	✓	-	✓
Reset Button	✓	-	-	✓

Οι περισσότερες εφαρμογές ενσωματώνουν λειτουργίες όπως η παρακολούθηση της προόδου, ειδοποιήσεις, αναφορές για τη βελτίωση της υγείας, χρήσιμα άρθρα και τεχνικές διαχείρισης στερητικών (cravings). Ωστόσο, πολλές από αυτές αποτυγχάνουν να ενσωματώσουν δομικά μοντέλα αλλαγής συμπεριφοράς ή να αξιολογηθούν με τυχαιοποιημένες ελεγχόμενες μελέτες. Η έλλειψη δεδομένων για την αποτελεσματικότητά τους, η απουσία αλληλεπίδρασης με επαγγελματίες υγείας και η κακή σχεδίαση διεπαφής χρήστη υποβαθμίζουν την εμπειρία και τη διατήρηση χρήσης.

Επιπλέον, η χρήση κοινών στρατηγικών όπως το gamification ή η συλλογή επιτευγμάτων χωρίς σύνδεση με εσωτερικά κίνητρα και πραγματική ψυχολογική ενίσχυση μπορεί να οδηγήσει σε προσωρινή συμμετοχή και όχι σε διαρκή και ριζική αλλαγή. Συνεπώς, αναδεικνύεται η ανάγκη για εφαρμογές που συνδυάζουν τεχνικά χαρακτηριστικά με τεκμηριωμένες ψυχολογικές παρεμβάσεις, προσαρμοσμένες στις ανάγκες του κάθε χρήστη.

Η παρούσα εργασία αποσκοπεί ακριβώς στη δημιουργία ενός τέτοιου συστήματος, που να ενσωματώνει ποιοτικά στοιχεία από τα καλύτερα παραδείγματα της αγοράς, ενισχύοντας ταυτόχρονα την επιστημονική εγκυρότητα, τη διαδραστικότητα και την εξατομίκευση.

2.3 Ηλεκτρονική Υγεία και Ψυχολογική Υποστήριξη

Ο όρος mobile health (mHealth) αναφέρεται στη χρήση τεχνολογιών όπως smartphones, tablets, wearables για την υποστήριξη της υγείας και της ιατρικής φροντίδας. Ο τομέας αυτός έχει γνωρίσει ταχύτατη ανάπτυξη, ειδικά στις παρεμβάσεις πρόληψης και τροποποίησης συμπεριφοράς, όπου η καθημερινή πρόσβαση και η αμεσότητα επικοινωνίας αποτελούν καθοριστικά πλεονεκτήματα [24].

Στη διακοπή καπνίσματος, η mHealth προσφέρει μοναδικά πλεονεκτήματα σε σχέση με τις παραδοσιακές μεθόδους:

- **Συνεχής παρουσία:** Ο χρήστης έχει διαρκή πρόσβαση σε υποστήριξη, ακόμη και σε κρίσιμες στιγμές.
- **Προσωποποίηση:** Τα δεδομένα του χρήστη (ημερομηνία διακοπής, φύλο, αριθμός τσιγάρων, ψυχολογική κατάσταση) αξιοποιούνται για την παροχή στοχευμένου περιεχομένου.
- **Αλληλεπίδραση:** Οι ειδοποιήσεις, τα επιτεύγματα και οι συμβουλές σε πραγματικό χρόνο ενισχύουν το κίνητρο.
- **Ανάλυση δεδομένων:** Οι εφαρμογές συλλέγουν πληροφορίες για την πρόοδο του χρήστη, διευκολύνοντας την παρέμβαση όταν υπάρχει κίνδυνος υποτροπής [20].

Ένα κρίσιμο στοιχείο της επιτυχίας της mHealth, είναι η ενσωμάτωση ψυχολογικής υποστήριξης. Τεχνικές όπως η ενθάρρυνση, mindfulness, οι ασκήσεις αναπνοής και η Cognitive Behavioral Therapy, αποδεικνύονται αποτελεσματικές στην καταπολέμηση του εθισμού και την ενίσχυση της αυτοσυγκράτησης [25]. Παράλληλα, η χρήση ψηφιακών βοηθών (π.χ. chatbots), έχει αποδειχθεί ιδιαίτερα χρήσιμη στη δημιουργία μιας διαρκούς και μη επικριτικής παρουσίας. Έτσι, ενισχύεται η ψυχολογική ασφάλεια του χρήστη. Οι χρήστες είναι πιο πρόθυμοι να ανοιχτούν σε ανώνυμους, απρόσωπους, πάντα διαθέσιμους ψηφιακούς βοηθούς, ειδικά όταν αντιμετωπίζουν ενοχές ή απογοήτευση λόγω πιθανής υποτροπής [26].

Το mHealth δεν αποτελεί απλώς ένα τεχνικό εργαλείο, αλλά μια πλατφόρμα ψυχολογικής ενδυνάμωσης. Οι εφαρμογές που συνδυάζουν λειτουργικότητα, προσβασιμότητα και επιστημονική τεκμηρίωση, έχουν τη δυνατότητα να δώσουν ουσιαστική βοήθεια στη διαδικασία της διακοπής του καπνίσματος, προάγοντας ταυτόχρονα την ευρύτερη ψυχική υγεία του χρήστη.

Σχετικές Έρευνες

Συμφώνα με μελέτες που διερεύνησαν τις αντιλήψεις των καπνιστών για τις κοινές λειτουργίες και τα χαρακτηριστικά των εφαρμογών διακοπής του καπνίσματος, εντοπίστηκαν βασικές λειτουργίες όπως η παρακολούθηση, η κοινωνική υποστήριξη, η απόσπαση της προσοχής και οι υπενθυμίσεις. Επιπλέον, χαρακτηριστικά εφαρμογών όπως, η εξατομίκευση, οι διαφορετικές μορφές περιεχομένου, η διαδραστικότητα και η προστασία προσωπικών δεδομένων επισημάνθηκαν ως ζωτικής σημασίας για την αφοσίωση των χρηστών [27]. Επίσης, αξιολογήθηκε και η ποιότητα του περιεχομένου και η αποτελεσματικότητα της παρέμβασης των εφαρμογών που έχουν σχεδιαστεί για τη διακοπή του καπνίσματος. Τα αποτελέσματα έδειξαν ότι ενώ πολλές εφαρμογές είναι διαθέσιμες, υπάρχουν περιορισμένα δεδομένα σχετικά με την τήρησή τους σε τεκμηριωμένες κατευθυντήριες γραμμές θεραπείας, υπογραμμίζοντας την ανάγκη για περαιτέρω έρευνα σε αυτόν τον τομέα [28].

Σε μια άλλη ολοκληρωμένη ανασκόπηση συντέθηκαν στοιχεία σχετικά με την αποτελεσματικότητα διάφορων παρεμβάσεων ηλεκτρονικής υγείας (eHealth), που

αποσκοπούν στην υποβοήθηση της διακοπής του καπνίσματος. Τα ευρήματα έδειξαν ότι οι διαδικτυακές πλατφόρμες υγείας (mHealth) αύξησαν σημαντικά τα ποσοστά αποχής από το κάπνισμα σε σύγκριση με τις συμβατικές μεθόδους. Τα προσωποποιημένα μηνύματα και οι πληροφορίες μέσω διαδικτύου, σε συνδυασμό με θεραπεία αντικατάστασης νικοτίνης ήταν μεταξύ των αποτελεσματικών στρατηγικών που εντοπίστηκαν [29].

2.4 Η Τεχνητή Νοημοσύνη στην Υγεία

Η ενσωμάτωση τεχνητής νοημοσύνης (AI) και συστημάτων φυσικής γλώσσας στην υγειονομική φροντίδα έχει οδηγήσει στην ταχεία εξάπλωση ψηφιακών βοηθών (chatbots). Τα ψηφιακά αυτά εργαλεία έχουν εξελιχθεί, από απλές αυτοματοποιημένες απαντήσεις, σε έξυπνα συστήματα που μπορούν να κατανοήσουν, να ανταποκριθούν και να προσαρμοστούν στις ανάγκες του χρήστη με προηγμένο επίπεδο εξατομίκευσης και ακρίβειας.

Τα chatbots στην υγειονομική περίθαλψη χρησιμοποιούνται σε αρκετές εφαρμογές για παρακολούθηση των συμπτωμάτων, υποστήριξη της ψυχικής υγείας, διαχείριση χρόνιων ασθενειών και ενίσχυση της συμμόρφωσης στη θεραπεία. Συγκεκριμένα, στον τομέα της διακοπής καπνίσματος, τα chatbots παρέχουν 24ωρη υποστήριξη, ενίσχυση κινήτρου, απαντήσεις σε όλες τις κρίσιμες στιγμές και ενίσχυση της αυτοσυγκράτησης του χρήστη [30].

Ωστόσο, τα συμβατικά chatbots περιορίζονται στην κατανόηση σύνθετων ερωτήσεων και στην παροχή ποιοτικής, εξατομικευμένης πληροφορίας σε πραγματικό χρόνο. Εδώ είναι που τα μεγάλα γλωσσικά μοντέλα (LLMs) και τα εργαλεία όπως το Langchain και το Retrieval-Augmented Generation (RAG) έρχονται να ενισχύσουν ριζικά τις δυνατότητες των συστημάτων συνομιλίας.

Έρευνες δείχνουν ότι, η ενσωμάτωση chatbots σε εφαρμογές διακοπής καπνίσματος αυξάνει σημαντικά τη χρήση τους. Σε σχετική μελέτη, το ποσοστό χρήσης διπλασιάστηκε, ενώ η αναφερόμενη διακοπή καπνίσματος αυξήθηκε σημαντικά. Επιπλέον, σε τυποποιημένες διεπαφές chatbot, εμπνευσμένες από Motivational Interviewing (MI), δεν παρατηρήθηκε διαφορά στην εμπλοκή χρήστη σε σύντομες αλληλεπιδράσεις. Πρ'όλα αυτά, καταγράφηκε αύξηση στη θεραπευτική σύνδεση και τη διάθεση διακοπής [23]. Ανασκοπήσεις δείχνουν ότι τα NLP-chatbots μπορούν βελτιώσουν τη πρόθεση απεξάρτησης, τη φυσική δραστηριότητα ή τις διατροφικές συνήθειες. Χρησιμοποιώντας τεχνικές όπως TTM, Behavior Change Wheel, και τροφοδοτώντας δεδομένα συμπεριφοράς για εξατομικευμένη υποστήριξη [27]. Επίσης, τα chatbots που παρέχουν προσωποποιημένα μηνύματα, σχολιασμό για τα στατιστικά (π.χ. χρήματα που εξοικονομήθηκαν), πίνακες κατάταξης και ειδοποιήσεις, αυξάνουν την αίσθηση της δέσμευσης και της ευθύνης.

Νέες Τεχνολογίες στην Υπηρεσία της Υγείας

Το Langchain αποτελεί ένα framework που επιτρέπει την κατασκευή εφαρμογών βασισμένων σε LLMs, όπως το GPT-4, δίνοντας στους developers τη δυνατότητα να διαχειρίζονται σύνθετες αλληλεπιδράσεις, να ενσωματώνουν εργαλεία όπως εξωτερικές βάσεις δεδομένων, APIs και να σχεδιάζουν πολύπλοκες ροές διαλόγου. Στην υγειονομική περίθαλψη, το Langchain μπορεί να χρησιμοποιηθεί για να κατασκευάσει chatbots που ενσωματώνουν ιατρικές βάσεις δεδομένων και δίνουν επαληθεύσιμες απαντήσεις με παραπομπές σε αξιόπιστες πηγές. Επιπλέον, μπορούν να προσαρμόζουν τη γλώσσα και το ύφος ανάλογα με τον χρήστη (π.χ. ηλικία, φύλο) και να υποστηρίζουν συνεχή διάλογο, όπου η εφαρμογή θυμάται το ιστορικό και τις προτιμήσεις του χρήστη.

Από την άλλη, η τεχνική Retrieval-Augmented Generation (RAG) λειτουργεί ως μηχανισμός αναζήτησης και παραγωγής απάντησης. Το μοντέλο αντλεί πληροφορίες από μια δομημένη βάση γνώσης (π.χ. ιατρικά αρχεία, επιστημονικές δημοσιεύσεις) και παράγει απάντηση βασισμένη σε αυτό το υλικό. Έτσι, μειώνεται η πιθανότητα λανθασμένων απαντήσεων από τα LLMs και ενισχύεται η αξιοπιστία της πληροφορίας.

Το παρόν σύστημα ενσωματώνει ένα chatbot βασισμένο σε Langchain και RAG για την υποστήριξη των χρηστών που προσπαθούν να διακόψουν το κάπνισμα. Το σύστημα επιτρέπει την αλληλεπίδραση σε φυσική γλώσσα με εξατομικευμένες συμβουλές και την άμεση ανάκτηση έγκυρων πληροφοριών από βάση δεδομένων με επιστημονικά άρθρα, στατιστικά υγείας, και τεχνικές διαχείρισης επιθυμιών. Επίσης, προσφέρει εκπαίδευση σε τεχνικές αναπνοής και mindfulness μέσω συζήτησης, καθώς και συνεχή υποστήριξη και ενθάρρυνση, ειδικά σε κρίσιμες στιγμές υψηλού κινδύνου υποτροπής.

Η ενσωμάτωση αυτών των τεχνολογιών δεν αντικαθιστά τον ρόλο των επαγγελματιών υγείας. Προσφέρει όμως προσιτή, άμεση και προσαρμόσιμη υποστήριξη, η οποία μπορεί να ενισχύσει την αποτελεσματικότητα της παρέμβασης. Αυτό είναι ιδιαίτερα χρήσιμο για άτομα που έχουν χαμηλή πρόσβαση σε συμβουλευτικές υπηρεσίες.

Λοιπές Έρευνες και Μελέτες

Μελέτες που σύγκριναν chatbot με Motivational Interviewing (MI) και chatbot με confrontational προσέγγιση, έδειξαν υψηλότερη εμπειρία χρηστών (εμπιστοσύνη, συμπάθεια) στην MI ομάδα, ενώ και οι δύο αύξησαν τη διάθεση για διακοπή [31]. Επιπλέον. Με την ευελιξία των Retrieval-Augmented Generation (RAG), επιτρέπονται εξατομικευμένες απαντήσεις, χωρίς επανεκπαίδευση LLM. Σε σχετική έρευνα, βρέθηκε ότι τα chatbot μειώνουν το Self-Report Habit Index (SRHI) [32].

2.5 Κριτική και Καινοτομίες της Παρούσας Εργασίας

Η παρούσα εργασία διαφοροποιείται ουσιαστικά από τις υφιστάμενες προσεγγίσεις τόσο ως προς τη συνολική φιλοσοφία υποστήριξης των καπνιστών όσο και ως προς την τεχνολογική καινοτομία του τρόπου παρέμβασης. Παρόλο που έχουν αναπτυχθεί αρκετές παρόμοιες εφαρμογές, η παρούσα προσέγγιση ενσωματώνει πολλαπλές διαστάσεις υποστήριξης, συνδυάζοντας την επιβράβευση, την κοινωνική αλληλεπίδραση και την προσωποποιημένη παρέμβαση.

Η συγκριτική ανάλυση ανταγωνιστικών εφαρμογών αποτελεί έναν κρίσιμο πυλώνα στη διαδικασία σχεδιασμού και ανάπτυξης μιας εφαρμογής, καθώς παρέχει τη δυνατότητα εντοπισμού των αδυναμιών και πλεονεκτημάτων των ήδη υπαρχουσών λύσεων στην αγορά. Η εφαρμογή που αναπτύσσεται στα πλαίσια της παρούσας διπλωματικής εργασίας, επιχειρεί να προσφέρει έναν καινοτόμο και ολιστικό τρόπο υποστήριξης χρηστών που επιθυμούν να διακόψουν το κάπνισμα. Για τον σκοπό αυτό, έγινε αναλυτική σύγκριση με τις πλέον δημοφιλείς εφαρμογές διακοπής καπνίσματος που είναι διαθέσιμες παγκοσμίως. Παρακάτω παρουσιάζεται συγκριτικός πίνακας με τα κύρια χαρακτηριστικά των εφαρμογών:

Πίνακας 2: Σύγκριση Κύριων Χαρακτηριστικών Υφιστάμενων Εφαρμογών με την παρούσα

Εφαρμογή	Εξατομίκευση	Chatbot	Friends	Ψυχολογική υποστήριξη	Κοινότητα	Gamification	Group Chat	Επιστημονική Τεκμηρίωση
Quit Tracker	-	-	-	-	-	✓ Μέτρια	-	N/A
Kwit	✓ Μέτρια	-	✓	✓	-	✓	-	✓
Smoke Free	-	-	-	-	-	-	-	✓
QuitNow	✓ Μέτρια	-	✓	✓	✓	✓	✓	N/A
Η Παρούσα Εργασία	✓ Προηγμένη	✓	✓	✓	✓	✓	✓	✓

Από την σύγκριση προκύπτει, ότι αρκετές εφαρμογές ενσωματώνουν βασικές λειτουργίες όπως η παρακολούθηση προόδου και τα επιτεύγματα, ελάχιστες όμως συνδυάζουν ένα ευρύ φάσμα χαρακτηριστικών με σύγχρονες τεχνολογίες. Η σύγκριση αναδεικνύει ότι καμία υπάρχουσα εφαρμογή δεν αξιοποιεί πλήρως ταυτόχρονα τεχνικές AI, στοιχεία gamification, κοινωνικές αλληλεπιδράσεις και επιστημονικά τεκμηριωμένες παρεμβάσεις. Επομένως, η εφαρμογή που παρουσιάζεται προσφέρει έναν καινοτόμο συνδυασμό τεχνολογιών, ο οποίος αναμένεται να έχει θετικό αντίκτυπο στην αποτελεσματικότητα της απεξάρτησης.

Αρχικά, το προτεινόμενο πληροφοριακό σύστημα που υλοποιήθηκε, προσφέρει πλήρη εξατομίκευση. Με βάση τα δεδομένα του χρήστη (ηλικία, φύλο, διάρκεια καπνίσματος, αριθμός τσιγάρων, κόστος), δημιουργείται προσωπικό πλάνο απεξάρτησης, καθώς και εξατομικευμένες συμβουλές. Χρησιμοποιείται chatbot, για καθημερινή συνομιλία με το χρήστη, ενσωματώνοντας τεχνικές Motivational Interviewing (MI) και cognitive behavioral therapy. Η ικανότητά του να απαντά φυσικά και με ενσυναίσθηση, ξεπερνά τις κλασικές αυτοματοποιημένες απαντήσεις που προσφέρουν οι υφιστάμενες εφαρμογές.

Η εφαρμογή ενσωματώνει σύστημα επιβράβευσης (πόντοι, επίπεδα), προσωπικά επιτεύγματα και παιχνίδια απόσπασης προσοχής (π.χ. Pac-Man), που βοηθούν τον χρήστη κατά τις περιόδους εμφάνισης στερητικών συμπτωμάτων. Η ενσωμάτωση τεκμηριωμένων χαρακτηριστικών στις εφαρμογές διακοπής του καπνίσματος μπορεί να ενισχύσει σημαντικά την αποτελεσματικότητά τους.

Παρακάτω είναι εξετάζονται συγκεκριμένες λειτουργίες, όπως mini-games, για απόσπαση της προσοχής, οικονομική παρακολούθηση των χρημάτων που εξοικονομούνται, τεχνικές αναπνοής και παρακολούθηση της προόδου αποκατάστασης της υγείας, που υποστηρίζεται από σχετικά ευρήματα ερευνών.

Η ενσωμάτωση mini-games σε εφαρμογές διακοπής καπνίσματος χρησιμεύει ως μηχανισμός απόσπασης της προσοχής για τη διαχείριση της επιθυμίας για τσιγάρο. Για παράδειγμα, τα παιχνίδια που αναπτύχθηκαν στο πλαίσιο σχετικών ερευνών, αποτελούν παράδειγμα αυτής της προσέγγισης. Σχεδιάστηκαν για να αποσπούν τους χρήστες σε στιγμές στερητικών συμπτωμάτων, παρέχοντας μια διαδραστική εμπειρία. Τα αποτελέσματα έδειξαν ότι η ενσωμάτωση τέτοιων παιχνιδιών είναι αποδεκτή και δυνητικά αποτελεσματική στην υποβοήθηση των προσπαθειών διακοπής του καπνίσματος [34]. Μεταξύ των πορισμάτων των ερευνών, το παιχνίδι οδήγησε σε μείωση των επιπέδων άγχους κατά την περίοδο στέρησης,

ιδιαίτερα μεταξύ εκείνων που απείχαν για πάνω από 48 ώρες. Αυτό δηλώνει ότι τα παιχνίδια που βασίζονται στην απόσπαση της προσοχής μπορούν να είναι αποτελεσματικά εργαλεία για την υποστήριξη των προσπαθειών διακοπής του καπνίσματος [35]. Επιπλέον, διαπιστώθηκε ότι η ενσωμάτωση ενός ψηφιακού παιχνιδιού οδήγησε σε αύξηση 20% στη συνολική αλληλεπίδραση με την εφαρμογή και 30% αύξηση στη χρήση των εργαλείων διακοπής του καπνίσματος της εφαρμογής. Αυτό υποδεικνύει ότι η παιχνιδοποίηση μπορεί να βελτιώσει την αλληλεπίδραση του χρήστη με πόρους διακοπής [35].

Στο σύστημα που σχεδιάστηκε για την παρούσα εργασία, χρήστης επιπλέον θα μπορεί να βλέπει την πρόοδο των φίλων του και να βρίσκει κίνητρο μέσω peer motivation. Έτσι, θα αποφεύγει να ενδώσει στο κάπνισμα αφού η πρόοδος του παρακολουθείται και από τους φίλους του. Η ύπαρξη πίνακα κατάταξης (leaderboard) βασισμένου σε ημέρες απεξάρτησης και πόντων (XP) συνδέεται άμεσα με αυξημένη παραμονή στην εφαρμογή [18].

Επιπλέον, οι ειδοποιήσεις είναι στηριγμένες σε συναισθηματική προσέγγιση και ενίσχυση κινήτρου (π.χ. Σήμερα έχεις αποφύγει 8 τσιγάρα μπράβο σου!), ενώ η υλοποίηση του συστήματος με Retrieval-Augmented Generation επιτρέπει την προσθήκη νέου υλικού (π.χ. επιστημονικά άρθρα) χωρίς ανάγκη επανεκπαίδευσης του μοντέλου.

Η εφαρμογή προσφέρει επίσης και τεχνικές αναπνοής για αντιμετώπιση συμπτωμάτων στέρησης νικοτίνης. Η εφαρμογή τεχνικών αναπνοής σε παρόμοιες εφαρμογές, μπορεί να βοηθήσει στη διαχείριση του στρες και στη μείωση των στερητικών. Εφαρμογές που επικεντρώνονται σε τεχνικές διαφραγματικής αναπνοής για τη διαχείριση του στρες, δείχνουν από σχόλια των χρηστών, ότι είναι χρήσιμες κατά τη διάρκεια των cravings. Η πλειοψηφία των συμμετεχόντων τις βρίσκουν ευεργετικές και εκφράζουν προθυμία να τις χρησιμοποιήσουν κατά τη διάρκεια προσπαθειών διακοπής καπνίσματος.

Επίσης, οι χρήστες θα μπορούν να συνομιλήσουν με άλλους χρήστες της εφαρμογής που προσπαθούν να ξεπεράσουν το εθισμό τους στο κάπνισμα. Σύμφωνα με έρευνες, οι πιθανότητες απεξάρτησης είναι πολύ μεγαλύτερες αν ο χρήστης είναι ενταγμένος σε ένα σύνολο και συναναστρέφεται με άτομα με τους ίδιους στόχους.

Η παρούσα εφαρμογή συνδυάζει την τεχνολογική καινοτομία με την ψυχολογική θεωρία, προσφέροντας ένα πολυδιάστατο εργαλείο απεξάρτησης που μπορεί να λειτουργήσει τόσο ως υποστηρικτής όσο και ως σύμμαχος στην καθημερινότητα του καπνιστή. Το chatbot σε τέτοιου είδους εφαρμογή, μαζί με μηχανισμούς παιχνιδοποίησης και κοινωνικής εμπλοκής, είναι σπάνιο έως και ανύπαρκτο στην τρέχουσα βιβλιογραφία, καθιστώντας την παρούσα πρόταση πρωτοποριακή στον τομέα.

3. Ανάλυση Απαιτήσεων και Περιγραφή Χρηστών

Η ανάλυση απαιτήσεων αποτελεί κρίσιμο βήμα για τον σαφή καθορισμό των αναγκών που καλείται να καλύψει η εφαρμογή διακοπής καπνίσματος. Αρχικά, το σύστημα απευθύνεται στον τελικό χρήστη, δηλαδή άτομα που θέλουν να διακόψουν ή να μειώσουν το κάπνισμα. Οι χρήστες αυτοί διαθέτουν διαφορετικά επίπεδα τεχνογνωσίας και κινητοποιούνται από ποικίλα κίνητρα, όπως τα οικονομικά, η υγεία ή η κοινωνική αποδοχή.

Η εφαρμογή πρέπει να υποστηρίζει λειτουργίες όπως η εγγραφή και σύνδεση χρήστη, τη συμπλήρωση ερωτηματολογίου, για την εξατομίκευση του προφίλ (π.χ. λόγοι διακοπής, προηγούμενες προσπάθειες) και τη συνεχή παρακολούθηση προόδου (π.χ. ημερήσιοι στόχοι, χρήματα που εξοικονομήθηκαν). Αυτές οι λειτουργίες πρέπει να είναι διαθέσιμες με μενού φιλικό προς τον χρήστη και αισθητικά σωστά διαμορφωμένο, ώστε να μην προκαλεί σύγχυση ούτε υπερφόρτωση δεδομένων. Οι βασικές κατηγορίες απαιτήσεων και χαρακτηριστικών καλύπτουν την εξατομικευμένη αλληλεπίδραση χρήστη, τεχνολογικές δεξιότητες και αξιοπιστία.

Στις επόμενες παραγράφους, θα εμβαθύνουμε στα διαγράμματα περιπτώσεων χρήσης, στις λειτουργικές και μη λειτουργικές απαιτήσεις αλλά και στην ανάλυση της εμπειρίας χρήσης, με έμφαση στην ψυχολογική υποστήριξη και την τέχνο-κοινωνική διάσταση της εφαρμογής.

3.1 Στόχοι και Αντικείμενο της Εργασίας

Στο κεφάλαιο αυτό παρουσιάζεται η κεντρική επιδίωξη της διπλωματικής εργασίας το αντικείμενο που καλύπτει, καθώς και οι ειδικοί στόχοι, οι περιορισμοί και τα αναμενόμενα αποτελέσματα του συστήματος που σχεδιάστηκε.

3.1.1 Κύριος Σκοπός και Όραμα

Ο δικός μου κύριος σκοπός είναι η δημιουργία μιας εφαρμογής ολοκληρωμένης, προσωποποιημένης και έξυπνης και επιστημονικά τεκμηριωμένης, που υποστηρίζει ανθρώπους που θέλουν να διακόψουν το κάπνισμα. Η εφαρμογή θα ανταποκρίνεται εξατομικευμένα στο προφίλ και τις ανάγκες κάθε χρήστη, μέσα από απαντημένα ερωτηματολόγια, ενώ θα ενσωματώνει ένα έξυπνο chatbot, με στόχο τη συνεχή ψυχολογική υποστήριξη, απαντώντας σε απορίες, επιβραβεύοντας επιτυχίες, και βοηθώντας σε δύσκολες στιγμές. Επιπλέον, θα ενδυναμώνει την αίσθηση κοινωνικής σύνδεσης με φίλους και άλλα άτομα μέσα από λειτουργίες leaderboard, ομαδικές συνομιλίες. Κύρια λειτουργία της θα αποτελεί η εποπτεία της προόδου, είτε οικονομικά (χρήματα που αποταμιεύτηκαν) είτε ως προς την υγεία (ημέρες αποχής, αριθμός τσιγάρων που απέφυγαν). Τέλος, θα προάγει τη συνέπεια και ψυχική ανθεκτικότητα, αξιοποιώντας στοιχεία παιχνιδοποίησης, όπως πόντους (XP), badges, επιτεύγματα, καθημερινές προκλήσεις και απλές αναπνευστικές τεχνικές ως εναλλακτικές στάσεις.

Το όραμα είναι η εφαρμογή να μεταμορφωθεί σε έναν αξιόπιστο ψηφιακό βοηθό για την απεξάρτηση, όχι απλώς ως εργαλείο, αλλά ως συμπληρωματική μέθοδος συμμετοχής,

ενθάρρυνσης και διαρκούς ενδυνάμωσης. Η φιλοδοξία είναι να επέλθει αλλαγή τρόπου ζωής μέσω της τεχνολογίας. Η εφαρμογή βασίζεται στο πλαίσιο της παροχής προσωποποιημένης, συνεχούς και ενεργητικής υποστήριξης στον χρήστη, όχι ως απλός οδηγός, αλλά σαν βοηθός σε ένα απαιτητικό ταξίδι απεξάρτησης.

3.1.2 Συγκεκριμένοι Στόχοι και Υποσυστήματα

Το σύστημα έχει δομηθεί γύρω από πέντε κεντρικούς στόχους, καθένας από τους οποίους υποστηρίζεται από συγκεκριμένα υποσυστήματα. Ο πρώτος στόχος είναι η εξατομίκευση και ο προγραμματισμός, όπου ο χρήστης συμπληρώνει ένα εκτενές ερωτηματολόγιο και με βάση τα δεδομένα που προκύπτουν (όπως διάρκεια καπνίσματος, προσωπικοί λόγοι και συνήθειες), δημιουργείται ένα εξατομικευμένο πλάνο απεξάρτησης από το chatbot.

Ο δεύτερος στόχος αφορά την ψηφιακή και συναισθηματική υποστήριξη, η οποία προσφέρεται μέσω του chatbot. Ο χρήστης μπορεί να συνομιλεί με έναν ψηφιακό βοηθό που απαντά επιστημονικά και συναισθηματικά, αξιοποιώντας ενημερωμένη τεκμηρίωση μέσω RAG, ενώ το bot διαχειρίζεται feedback, διαθέτει μνήμη και αντλεί προσωπικά δεδομένα από το ερωτηματολόγιο.

Ο επόμενος στόχος είναι η παρακολούθηση της προόδου και η επιβράβευση. Το σύστημα καταγράφει τις ημέρες αποχής, τα τσιγάρα που δεν καταναλώθηκαν και τα χρήματα που εξοικονομήθηκαν, ενώ παράλληλα προσφέρει πόντους, επίπεδα και εμφανίζει πίνακα κατάταξης για την ενίσχυση του κινήτρου.

Άλλος ένας στόχος σχετίζεται με την κοινωνική δυναμική, καθώς οι χρήστες έχουν τη δυνατότητα να προσθέτουν φίλους, να παρακολουθούν την πρόοδο άλλων και να ανταλλάσσουν ενθαρρυντικά και παρακινητικά μηνύματα, καλλιεργώντας έτσι ένα αίσθημα κοινότητας.

Ο τελευταίος στόχος είναι η καθημερινή παρέμβαση, η οποία επιτυγχάνεται με τις ημερήσιες συμβουλές και στόχους, καθώς και την αποστολή ειδοποιήσεων. Κάθε πρωί ή απόγευμα ο χρήστης λαμβάνει σύντομα μηνύματα με συμβουλές υγείας, υπενθύμιση προόδου, γεγονός που ενισχύει τη συνέπεια και την αφοσίωσή του στο πλάνο διακοπής. Αυτή η αρχιτεκτονική εξασφαλίζει ότι η εφαρμογή ανταποκρίνεται δυναμικά σε κρίσιμες ανάγκες, όπως η διαχείριση της επιθυμίας, η ψυχολογική υποστήριξη και η κοινωνική αλληλεπίδραση.

Επίσης, η εργασία φιλοδοξεί να αυξήσει την αλληλεπίδραση των χρηστών, μέσω προσωποποιημένων μηνυμάτων και επιτευγμάτων που στηρίζονται σε δεδομένα του χρήστη (π.χ. χρηματική εξοικονόμηση, χρόνοι χωρίς κάπνισμα). Αναμένεται ότι χρήστες που χρησιμοποιούν ταυτόχρονα την εφαρμογή και ενδεχομένως φαρμακευτική αγωγή ή επίβλεψη από επαγγελματία, θα έχουν μεγαλύτερη πιθανότητα απεξάρτησης, σύμφωνα με μελέτες που δείχνουν υψηλότερες επιτυχίες όταν συνδυάζονται ψηφιακά εργαλεία και θεραπεία. Επιπλέον, η ενσωμάτωση chatbot μπορεί να ενισχύσει την εγκυρότητα και σχετικότητα των συμβουλών, όπως υποδεικνύεται σε πειραματικές εφαρμογές υγείας. Συνολικά, αναμένονται βελτιωμένα επίπεδα αποτελεσματικότητας, ψυχολογικής υποστήριξης και διαρκούς δέσμευσης σε σχέση με παραδοσιακές εφαρμογές που δεν παρέχουν εξατομίκευση ή ενεργή υποστήριξη.

3.1.3 Οριοθέτηση Λειτουργιών και Περιορισμοί

Η εφαρμογή περιορίζεται σε υποστηριζόμενη πλατφόρμα Flutter (Android). Δεν περιλαμβάνονται θεραπείες με φαρμακευτική υποστήριξη. Επιπλέον, το chatbot εξυπηρετεί μόνο χρήστες που ερωτούν για το κάπνισμα, ενώ άσχετες ερωτήσεις απαντώνται με μήνυμα για αλλαγή θέματος. Ορισμένοι περιορισμοί σχετίζονται με το κόστος χρήσης APIs όπως OpenRouter και Firebase, τις απαιτήσεις σύνδεσης για RAG, και την υπολογιστική ισχύ. Αναμένεται ότι η παρούσα έκδοση θα στοχεύει σε πρώτη υλοποίηση για τις απαιτήσεις της διπλωματικής εργασίας.

Πίνακας 3: Οριοθέτηση Λειτουργιών και Περιορισμοί

Τομέας	Περιγραφή
Ιατρική Θεραπεία	Δεν αντικαθιστά επαγγελματική διάγνωση ή θεραπεία.
Chatbot Αντιδράσεις	Μπορεί να αποτύχει σε ακραίες ψυχολογικές περιπτώσεις ή συναισθηματικά τραύματα.
Ακρίβεια Χρήστη	Βασίζεται σε δεδομένα που δηλώνει μόνος ο χρήστης με πιθανές ανακρίβειες.
Hardware	Απαιτεί σύνδεση στο διαδίκτυο για αρκετές λειτουργίες.
Δεδομένα και RAG	Απαιτεί συνεχή συντήρηση της βάσης γνώσεων σύμφωνα με τις νεότερες επιστημονικές έρευνες για ακρίβεια και επικαιρότητα.

3.2 Γενική Περιγραφή του Συστήματος

Η αρχιτεκτονική του βασίζεται σε μια δομή client server, όπου το frontend είναι υλοποιημένο με τη χρήση του Flutter framework, προσφέροντας ταυτόχρονα υψηλή αισθητική, λειτουργική διεπαφή και πλήρη συμβατότητα με πολλαπλά μεγέθη οθονών. Το backend σύστημα βασίζεται στην πλατφόρμα Firebase της Google, η οποία προσφέρει αυθεντικοποίηση χρηστών (Firebase Authentication), αποθήκευση και διαχείριση δεδομένων (Cloud Firestore), καθώς και δυνατότητα αποστολής ειδοποιήσεων (Firebase Cloud Messaging).

Η λειτουργία της εφαρμογής ξεκινά με μια οθόνη εισαγωγής όπου ο χρήστης έχει τη δυνατότητα να δημιουργήσει λογαριασμό είτε μέσω email/κωδικού πρόσβασης είτε μέσω λογαριασμού Google. Μετά την επιτυχημένη σύνδεση, εμφανίζεται ένα προσωποποιημένο ερωτηματολόγιο. Σε αυτό ο χρήστης καλείται να δηλώσει πληροφορίες όπως το φύλο του, τη μέση ημερήσια κατανάλωση τσιγάρων, το κόστος του πακέτου, τη χρονική διάρκεια του καπνίσματος, καθώς και τους βασικούς του στόχους ή λόγους για διακοπή. Τα δεδομένα αυτά αποθηκεύονται με ασφάλεια στο Firestore και χρησιμοποιούνται για την προσωποποίηση της εμπειρίας χρήσης, τον υπολογισμό στατιστικών, καθώς και για την προσαρμογή ειδοποιήσεων και μηνυμάτων υποστήριξης.

Η κεντρική λειτουργικότητα της εφαρμογής περιλαμβάνει μια κεντρική οθόνη που παρουσιάζει σε πραγματικό χρόνο στατιστικά, όπως οι ημέρες χωρίς κάπνισμα, τα χρήματα που έχουν εξοικονομηθεί, και τα τσιγάρα που έχουν αποφευχθεί. Τα δεδομένα αυτά ανανεώνονται αυτόματα ανά 1 λεπτό, με βάση την ημερομηνία έναρξης διακοπής που έχει ορίσει ο χρήστης. Η ενότητα αυτή περιλαμβάνει επίσης μια σειρά από δυναμικές μπάρες προόδου που αναπαριστούν επιμέρους επιτεύγματα (π.χ. αποτοξίνωση από νικοτίνη, αποκατάσταση κυκλοφορίας αίματος, κ.λπ.) βάσει επιστημονικών δεδομένων από τον Παγκόσμιο Οργανισμό Υγείας και το NHS (World Health Organization).

Ένα βασικό στοιχείο καινοτομίας είναι η ενσωμάτωση ενός chatbot, για την παροχή προσωποποιημένης, υποστηρικτικής αλληλεπίδρασης με τον χρήστη. Το chatbot αντλεί δεδομένα από ένα δυναμικό corpus πληροφοριών, περιλαμβάνοντας επιστημονικές οδηγίες για το πώς να αντιμετωπιστεί η επιθυμία για κάπνισμα, απαντήσεις σε σχετικά ερωτήματα και τεχνικές mindfulness και αναπνοής. Οι απαντήσεις βασίζονται είτε στο ιστορικό συνομιλίας είτε στα δεδομένα που έχει καταχωρήσει ο χρήστης, παρέχοντας αίσθηση προσωπικού ψηφιακού καθοδηγητή.

Υποστηρίζονται επίσης κοινωνικές δυνατότητες, επιτρέποντας στους χρήστες να προσθέτουν φίλους, να παρακολουθούν την πρόοδό τους και να εμφανίζονται σε πίνακες κατάταξης. Οι χρήστες κατατάσσονται είτε σε νέους (ημ/νια διακοπής κάτω των 40 ημερών) είτε σε πιο έμπειρους (άνω των 40 ημερών), προκειμένου να διατηρηθεί η δίκαιη συγκριτική αξιολόγηση. Αυτή η μορφή κοινωνικής ενίσχυσης έχει αποδειχθεί ιδιαίτερα αποτελεσματική για την αύξηση της δέσμευσης σε παρεμβάσεις διακοπής καπνίσματος.

Η τεχνική υλοποίηση της υποδομής του συστήματος επιτρέπει την ταυτόχρονη εξυπηρέτηση πολλών χρηστών, με τη χρήση Firestore indexes και triggers, ενώ ο χρονισμός των ειδοποιήσεων και των updates πραγματοποιείται με cron jobs που τρέχουν στην cloud πλατφόρμα. Όλες οι ενέργειες που σχετίζονται με τα προσωπικά δεδομένα (π.χ. δημιουργία, ενημέρωση ή διαγραφή λογαριασμού).

Συνολικά, το σύστημα συνδυάζει λειτουργίες αυτοπαρακολούθησης, κοινωνικής σύγκρισης, υποστηρικτικής καθοδήγησης και θετικής ενίσχυσης μέσω gamification. Ο χρήστης όχι μόνο πληροφορείται για την πρόοδό του αλλά αποκτά την αίσθηση ότι έχει έναν ψηφιακό σύμμαχο που τον καταλαβαίνει και τον καθοδηγεί. Ο συνδυασμός αυτών των στοιχείων δημιουργεί ένα ολοκληρωμένο οικοσύστημα ψηφιακής υποστήριξης για τη διακοπή καπνίσματος, εναρμονισμένο με τις σύγχρονες τάσεις της eHealth τεχνολογίας και της ψυχολογίας της συμπεριφοράς.

3.3 Χρήστες και Χαρακτηριστικά

Η επιτυχής σχεδίαση και ανάπτυξη μιας ψηφιακής εφαρμογής προϋποθέτει την εις βάθος κατανόηση του προφίλ των τελικών χρηστών της. Οι χρήστες που απευθύνονται σε τέτοιου τύπου εφαρμογές δεν αποτελούν μια ομοιογενή ομάδα, αλλά διαφοροποιούνται ως προς τα κίνητρα, το επίπεδο εξοικείωσης με την τεχνολογία, την ψυχολογική τους κατάσταση, το ιστορικό τους ως καπνιστές, καθώς και τις κοινωνικές παραμέτρους που τους χαρακτηρίζουν. Η συγκεκριμένη εφαρμογή έχει σχεδιαστεί ώστε να υποστηρίξει προσαρμόσιμες λειτουργίες, οι οποίες ανταποκρίνονται σε αυτή την διαφορετικότητα.

Μία βασική κατηγοριοποίηση των χρηστών είναι η διάκριση ανάμεσα σε νέους χρήστες που επιχειρούν να διακόψουν το κάπνισμα για πρώτη φορά και σε χρήστες με προηγούμενες αποτυχημένες απόπειρες. Οι πρώτοι συχνά εμφανίζουν υψηλό αρχικό κίνητρο, αλλά ενδέχεται να στερούνται της εμπειρίας και της γνώσης για τους τρόπους με τους οποίους μπορούν να διαχειριστούν την εξάρτηση από τη νικοτίνη και τις ψυχολογικές προκλήσεις της διαδικασίας. Αντίθετα, οι δεύτεροι μπορεί να έχουν βαθύτερη επίγνωση των δυσκολιών, αλλά συνοδεύονται συχνά από συναισθήματα απογοήτευσης ή παραίτησης, που απαιτούν διαφορετική προσέγγιση ενίσχυσης. Η εφαρμογή αναγνωρίζει αυτούς τους δύο τύπους

χρηστών ήδη από το πρώτο ερωτηματολόγιο σύνδεσης, προσαρμόζοντας τις τεχνικές καθοδήγησης μέσω του chatbot.

Μια επιπλέον διάσταση διαφοροποίησης των χρηστών αφορά το φύλο και την ηλικία. Έρευνες έχουν δείξει ότι άνδρες και γυναίκες συχνά αντιμετωπίζουν διαφορετικά ψυχολογικά και σωματικά συμπτώματα κατά τη διάρκεια της διακοπής. Οι γυναίκες, για παράδειγμα, εμφανίζουν μεγαλύτερη ευαισθησία στο στρες και στις μεταβολές διάθεσης, και συχνά χρειάζονται υποστήριξη που εστιάζει περισσότερο στις συναισθηματικές και κοινωνικές πτυχές. Από την άλλη πλευρά, οι άνδρες τείνουν να ανταποκρίνονται καλύτερα σε στρατηγικές που βασίζονται σε ανταμοιβές και επιτεύγματα. Η εφαρμογή λαμβάνει υπόψη αυτές τις διαφορές και προσαρμόζει τα μηνύματα καθοδήγησης με βάση τις δηλωμένες παραμέτρους φύλου.

Οι χρήστες διακρίνονται επίσης ανάλογα με τον βαθμό ψηφιακής εξοικείωσης. Ένα σημαντικό ποσοστό χρηστών άνω των 45 ετών έχει περιορισμένη εμπειρία με προηγμένες τεχνολογικές λειτουργίες και, ως εκ τούτου, η εφαρμογή έχει σχεδιαστεί με γνώμονα την ευχρηστία και την προσβασιμότητα. Το UI είναι λιτό, με μεγάλα κουμπιά, καθαρή τυπογραφία και σαφές οπτικό υλικό. Αντίθετα, οι νεότεροι χρήστες είναι εξοικειωμένοι με έννοιες όπως gamification, κοινωνική αλληλεπίδραση και δυναμικά στατιστικά, και αξιοποιούν πλήρως τις επεκτάσεις όπως οι πίνακες κατάταξης και τα επιτεύγματα με πόντους και εικονίδια προόδου.

Συμπερασματικά, το σύστημα δεν αντιμετωπίζει τον χρήστη ως απλό δέκτη περιεχομένου, αλλά ως ένα μοναδικό άτομο με διακριτές ανάγκες, ιστορικό και προσδοκίες. Μέσω της αξιοποίησης των δεδομένων που δηλώνει ο ίδιος, αλλά και της διαρκούς ανάλυσης της συμπεριφοράς χρήσης, η εφαρμογή επιδιώκει να παρέχει στοχευμένη, εξατομικευμένη και διαχρονικά σχετική υποστήριξη, στοιχείο κρίσιμο για την επιτυχία οποιασδήποτε παρέμβασης που αφορά τη αλλαγή συμπεριφοράς.

3.4 Απαιτήσεις Συστήματος

Η ανάπτυξη ενός συστήματος, όπως αυτό που παρουσιάζεται στην παρούσα εργασία, απαιτεί την πλήρη κατανόηση τόσο των λειτουργικών όσο και των μη λειτουργικών απαιτήσεων. Οι λειτουργικές απαιτήσεις ορίζουν τη βασική συμπεριφορά και τα χαρακτηριστικά του συστήματος. Οι μη λειτουργικές απαιτήσεις καθορίζουν τα ποιοτικά χαρακτηριστικά της εφαρμογής, όπως η απόδοση, η ασφάλεια και η ευχρηστία. Η σαφής αποτύπωσή τους αποτελεί βασικό βήμα στον κύκλο ζωής ανάπτυξης λογισμικού, εξασφαλίζοντας ότι η τελική υλοποίηση ανταποκρίνεται στις ανάγκες των τελικών χρηστών.

Λειτουργικές Απαιτήσεις

Οι λειτουργικές απαιτήσεις αφορούν τα βασικά υποσυστήματα και τη συμπεριφορά της εφαρμογής. Η εφαρμογή παρέχει πλήθος λειτουργιών που ενισχύουν τη δέσμευση του χρήστη και υποστηρίζουν την αλλαγή της συμπεριφοράς του. Μία από τις κύριες λειτουργίες είναι η δημιουργία και παρακολούθηση προφίλ χρηστών. Οι χρήστες εισάγουν δεδομένα όπως φύλο, ηλικία, αριθμό ημερήσιων τσιγάρων και οικονομικό κόστος, τα οποία χρησιμοποιούνται για την παροχή προσωποποιημένης εμπειρίας. Η εφαρμογή καταγράφει καθημερινή πρόοδο, εμφανίζει στατιστικά (όπως χρήματα που εξοικονομήθηκαν ή τσιγάρα που δεν καταναλώθηκαν), και παρουσιάζει σχετικές ειδοποιήσεις και επιτεύγματα.

Επιπλέον, μια σημαντική λειτουργική απαίτηση είναι η υποστήριξη συστήματος ειδοποιήσεων (push notifications), το οποίο υπενθυμίζει στους χρήστες καθημερινά τις επιτυχίες τους, παρέχει κίνητρα και χρήσιμες συμβουλές. Η ενσωμάτωση τεχνικών υποβοηθούμενης αναπνοής μέσω κατάλληλων animation ή ήχων είναι κρίσιμη για στιγμές κρίσης και επιθυμίας για κάπνισμα. Άλλη βασική λειτουργία είναι η δυνατότητα επικοινωνίας με άλλους χρήστες μέσω ενός συστήματος ομαδικής συνομιλίας (group chatroom), με στόχο την ανταλλαγή εμπειριών και την ενίσχυση του αισθήματος κοινότητας. Τέλος, το σύστημα επιτευγμάτων, πινάκων κατάταξης και mini-games στοχεύει στην ενίσχυση του engagement και της θετικής ψυχολογίας, μέσω gamification.

Μη Λειτουργικές Απαιτήσεις

Οι μη λειτουργικές απαιτήσεις καθορίζουν τα ποιοτικά χαρακτηριστικά της εφαρμογής και παίζουν καθοριστικό ρόλο στην αποδοχή και μακροχρόνια χρήση της. Πρώτα από όλα, η εφαρμογή πρέπει να χαρακτηρίζεται από υψηλή διαθεσιμότητα και αξιοπιστία. Η απρόσκοπτη λειτουργία της, χωρίς σφάλματα ή διακοπές, είναι ζωτικής σημασίας για τη διατήρηση της εμπιστοσύνης του χρήστη.

Η ευχρηστία (usability) αποτελεί μία από τις πιο κρίσιμες μη λειτουργικές απαιτήσεις. Η εφαρμογή απευθύνεται σε χρήστες διαφόρων ηλικιών και τεχνολογικής εξοικείωσης. Το περιβάλλον πρέπει να είναι απλό, αισθητικά ευχάριστο και προσβάσιμο. Ακόμη, η ασφάλεια και προστασία προσωπικών δεδομένων είναι κεντρικές απαιτήσεις, ειδικά λόγω της αποθήκευσης ευαίσθητων προσωπικών πληροφοριών.

Επιπλέον, η απόδοση είναι ιδιαίτερα σημαντική για την εμπειρία χρήστη. Η εφαρμογή πρέπει να φορτώνει γρήγορα και να διαχειρίζεται αποτελεσματικά τους πόρους της συσκευής. Πρέπει να μπορεί να χρησιμοποιηθεί χωρίς καθυστερήσεις ακόμα και σε παλαιότερες ή χαμηλότερης ισχύος συσκευές. Επίσης, πρέπει να διασφαλίζεται και η επεκτασιμότητα (scalability), ώστε το σύστημα να υποστηρίζει μελλοντικά περισσότερους χρήστες ή νέες λειτουργίες χωρίς να υποβαθμίζεται η απόδοση. Τέλος, οι απαιτήσεις συντήρησης (maintainability), καθώς και η διαλειτουργικότητα (interoperability) με άλλα APIs και υπηρεσίες όπως Firebase, Langchain, ή εξωτερικά προγράμματα αναλύσεων, είναι αναγκαίες για τη μελλοντική υποστήριξη και συνεχή εξέλιξη της εφαρμογής.

Πίνακας 4: Λειτουργικές Απαιτήσεις Συστήματος

ID	Κατηγορία	Τίτλος	Περιγραφή Απαίτησης	Προτεραιότητα
FR1	Αuth/ση	Δημιουργία Λογαριασμού	Το σύστημα πρέπει να επιτρέπει στον χρήστη να δημιουργήσει λογαριασμό μέσω email ή Google Sign-In. Η διαδικασία αυτή περιλαμβάνει καταχώρηση βασικών στοιχείων και αποθήκευσή τους στη βάση Firestore με χρήση του UID από το Firebase Authentication.	Υψηλή
FR2	Αuth/ση	Σύνδεση / Αποσύνδεση	Ο χρήστης πρέπει να μπορεί να συνδεθεί στον λογαριασμό του, με αυτόματη διατήρηση της συνεδρίας του (session persistence). Αν η σύνδεση χαθεί, η εφαρμογή πρέπει να επιτρέπει γρήγορη επανασύνδεση χωρίς απώλεια δεδομένων.	Υψηλή
FR3	Αuth/ση	Διαγραφή Λογ/μου	Ο χρήστης πρέπει να μπορεί να διαγράψει τον λογαριασμό του και όλα τα δεδομένα του.	Υψηλή

FR4	Πρόοδος	Πρόοδος Χρήστη	Στην αρχική οθόνη της εφαρμογής πρέπει να προβάλλονται οι καθημερινοί στόχοι (Daily Goals), η συνολική πρόοδος της υγείας (Health Progress) και στατιστικά όπως μέρες χωρίς κάπνισμα, τσιγάρα που δεν καπνίστηκαν και χρήματα που εξοικονομήθηκαν.	Υψηλή
FR5	Επιτεύγματα και XP	Εξέλιξη Υγείας	Το υποσύστημα Health Progress πρέπει να παρουσιάζει δυναμικές μπάρες προόδου που απεικονίζουν την αποκατάσταση συγκεκριμένων λειτουργιών του σώματος (π.χ. καρδιακός ρυθμός, πνευμονική ικανότητα) με βάση τον χρόνο που έχει περάσει από τη διακοπή.	Υψηλή
FR6	Ευεξία	Daily Tips	Η εφαρμογή πρέπει να προσφέρει καθημερινές συμβουλές (Daily Tips), οι οποίες ξεκλειδώνονται μία ανά ημέρα. Οι ήδη ξεκλειδωμένες συμβουλές παραμένουν διαθέσιμες για μελλοντική χρήση.	Υψηλή
FR7	Ευεξία	Mini Games	Ο χρήστης πρέπει να έχει τη δυνατότητα να παίζει μικρά παιχνίδια (Mini Games), τα οποία λειτουργούν ως μέσο απόσπασης κατά τη διάρκεια έντονων επιθυμιών για κάπνισμα. Παραδείγματα παιχνιδιών είναι Snake, Pac-Man και Flappy Bird.	Μεσαία
FR8	Επιτεύγματα και XP	Achievements	Το σύστημα επιτευγμάτων (Achievements) πρέπει να καταγράφει την πρόοδο του χρήστη, να δίνει XP και να εμφανίζει ολοκληρωμένα επιτεύγματα με οπτική ένδειξη (π.χ. Completed overlay).	Υψηλή
FR9	Κοινωνικές Λειτουργίες	Προσθήκη Φίλων	Οι χρήστες πρέπει να μπορούν να προσθέτουν φίλους με αναζήτηση username ή email. Το προφίλ φίλων εμφανίζει βασικές πληροφορίες προόδου, όπως μέρες χωρίς κάπνισμα και επίπεδο.	Μεσαία
FR10	Κοινωνικές Λειτουργίες	Leaderboard	Ο πίνακας κατάταξης (Leaderboard) πρέπει να ταξινομεί τους χρήστες σε δύο κατηγορίες (<40 ημέρες ή ≥40 ημέρες αποχής), να εμφανίζει τους 10 πρώτους της πρώτης κατηγορίας και να διασφαλίζει ότι ο τρέχων χρήστης εμφανίζεται πάντα, ακόμη κι αν δεν είναι στην κορυφή.	Υψηλή
FR11	Παρ/θηση Προόδου	Craving Logs	Ο χρήστης πρέπει να μπορεί να δημιουργήσει Craving Logs, καταγράφοντας τον χρόνο που εμφανίστηκε η επιθυμία, την έντασή της και τον τρόπο αντιμετώπισής της (π.χ. αναπνοές, παιχνίδι, βόλτα).	Υψηλή
FR12	Κοινωνικές Λειτουργίες	Craving Logs Browsing	Η εφαρμογή πρέπει να παρέχει πρόσβαση στα παλιά Craving Logs του χρήστη, αλλά και ανώνυμη περιήγηση στα logs άλλων χρηστών, ώστε να μοιράζεται η εμπειρία αντιμετώπισης.	Μεσαία
FR13	Ευεξία	Breatthing Techniques	Πρέπει να υπάρχει ξεχωριστή οθόνη για τεχνικές αναπνοής (Breathing Techniques), η οποία καθοδηγεί τον χρήστη με animation και χρονόμετρα για να ακολουθήσει συγκεκριμένα μοτίβα αναπνοής.	Μεσαία
FR14	Αναζήτηση Βοήθειας	Επιλογές Βοήθειας	Μέσω της οθόνης Talk to Someone ο χρήστης μπορεί να επιλέξει να μιλήσει είτε με το Chatbot, είτε με γιατρό (μέσω	Υψηλή

			τηλεφώνου ή email), είτε να συμμετάσχει σε Community Chat.	
FR15	Κοινωνικές Λειτουργίες	Community Chat	Το Community Chat πρέπει να έχει όριο χρηστών ανά δωμάτιο και να δημιουργεί νέα δωμάτια όταν γεμίζουν.	Υψηλή
FR16	Chatbot	RAG και Μνήμη	Το Chatbot πρέπει να λειτουργεί με LangChain, RAG, memory, ώστε να απαντά σχετικές ερωτήσεις.	Υψηλή
FR17	Chatbot	Backend και Server	Το chatbot πρέπει να τρέχει μέσω FastAPI και Ngrok για ασφαλή επικοινωνία.	Υψηλή
FR18	Παρακινήσεις	Ειδοποιήσεις	Ο χρήστης πρέπει να λαμβάνει push notifications (motivational, tips, reminders).	Υψηλή
FR20	Εξατομίκευση	Restart Progress	Ο χρήστης πρέπει να μπορεί να επαναφέρει την πρόοδό του (restart progress) αν το επιθυμεί.	Μεσαία
FR21	Εξατομίκευση	Προσωπικά Στατιστικά	Η εφαρμογή πρέπει να υπολογίζει χρήματα που εξοικονομούνται και τσιγάρα που αποφεύγονται.	Υψηλή
FR22	Εξατομίκευση	Ερωτ/γιο	Ο χρήστης πρέπει να απαντήσει στο First-time Login Questionnaire (φύλο, τσιγάρα/ημέρα, κόστος, διάρκεια συνήθειας).	Υψηλή

Πίνακας 5: Μη Λειτουργικές Απαιτήσεις

ID	Κατηγορία	Τίτλος	Περιγραφή Απαίτησης	Προτεραιότητα
NRF1	Τεχνολογία	Χρήση Flutter	Η εφαρμογή αναπτύσσεται σε Flutter ώστε να υποστηρίζεται με ενιαίο κώδικα τόσο σε Android όσο και σε iOS, μειώνοντας κόστος και χρόνο συντήρησης.	Υψηλή
NFR2	Backend	Firebase Authentication και Firestore	Χρησιμοποιείται Firebase για αυθεντικοποίηση χρηστών και Firestore για real-time βάση δεδομένων, με δυνατότητα κλιμάκωσης και αξιόπιστη απόδοση.	Υψηλή
NFR3	Ασφάλεια Δεδομένων	Ασφάλεια δεδομένων και GDPR Συμμόρφωση	Όλα τα δεδομένα είναι προσβάσιμα μόνο από τον εκάστοτε χρήστη και δίνεται η δυνατότητα οριστικής διαγραφής τους, σύμφωνα με το GDPR.	Υψηλή
NFR4	Απόδοση	Χρόνος Απόκρισης Εφαρμογής	Οι βασικές λειτουργίες (π.χ. φόρτωση αρχικής οθόνης, αποθήκευση δεδομένων) πρέπει να ολοκληρώνονται σύντομα.	Υψηλή
NFR5	Απόδοση Chatbot	Χρόνος Απόκρισης Συνομιλίας	Το Chatbot πρέπει να απαντά σύντομα, ώστε η εμπειρία συνομιλίας να είναι φυσική.	Μεσαία
NFR6	UI	Responsive Σχεδίαση	Το περιβάλλον χρήσης πρέπει να προσαρμόζεται σε διαφορετικά μεγέθη οθονών (smartphone, tablet) χωρίς να αλλοιώνεται η εμπειρία.	Υψηλή
NFR7	Scalability	Αυτόματη Δημιουργία Chat Rooms	Όταν ένα community chat room γεμίσει, δημιουργείται αυτόματα νέο δωμάτιο.	Υψηλή

NFR8	Επεκτ/τα Κώδικα	Χρήση Provider	Η διαχείριση κατάστασης γίνεται με Provider ώστε η αρχιτεκτονική να παραμένει καθαρή, επεκτάσιμη και εύκολα συντηρήσιμη.	Υψηλή
NFR9	Διαθ/τα	Fallback Μηχανισμός Chatbot	Αν παρουσιαστεί πρόβλημα στο Chatbot, εμφανίζονται προκαθορισμένα μηνύματα υποστήριξης ώστε να διασφαλιστεί η συνέχεια της εμπειρίας.	Υψηλή
NFR10	Χρηστικότητα (UX/UI)	Μινιμαλιστικός Σχεδιασμός	Το UI ακολουθεί αρχές καθαρού και μινιμαλιστικού design, μειώνοντας το γνωστικό φορτίο και ενισχύοντας την καθημερινή χρήση.	Υψηλή
NFR11	Ειδοποιήσεις	Background Notifications	Οι ειδοποιήσεις πρέπει να αποστέλλονται έγκαιρα και αξιόπιστα ακόμη κι όταν η εφαρμογή δεν εκτελείται στο παρασκήνιο.	Υψηλή
NFR12	Συγχρονισμός	Cloud Sync	Όλα τα δεδομένα χρήστη συγχρονίζονται αυτόματα στο cloud, ώστε η πρόοδος να είναι διαθέσιμη από οποιαδήποτε συσκευή.	Υψηλή

Η καταγραφή και ανάλυση των λειτουργικών και μη λειτουργικών απαιτήσεων συνιστά ένα από τα πλέον κρίσιμα στάδια στον σχεδιασμό ενός πληροφοριακού συστήματος.

3.5 Περιπτώσεις Χρήσης

Κατά την ανάλυση απαιτήσεων μιας εφαρμογής, είναι ζωτικής σημασίας η καταγραφή των περιπτώσεων χρήσης (use cases), οι οποίες περιγράφουν με σαφήνεια τις αλληλεπιδράσεις των χρηστών με το σύστημα.

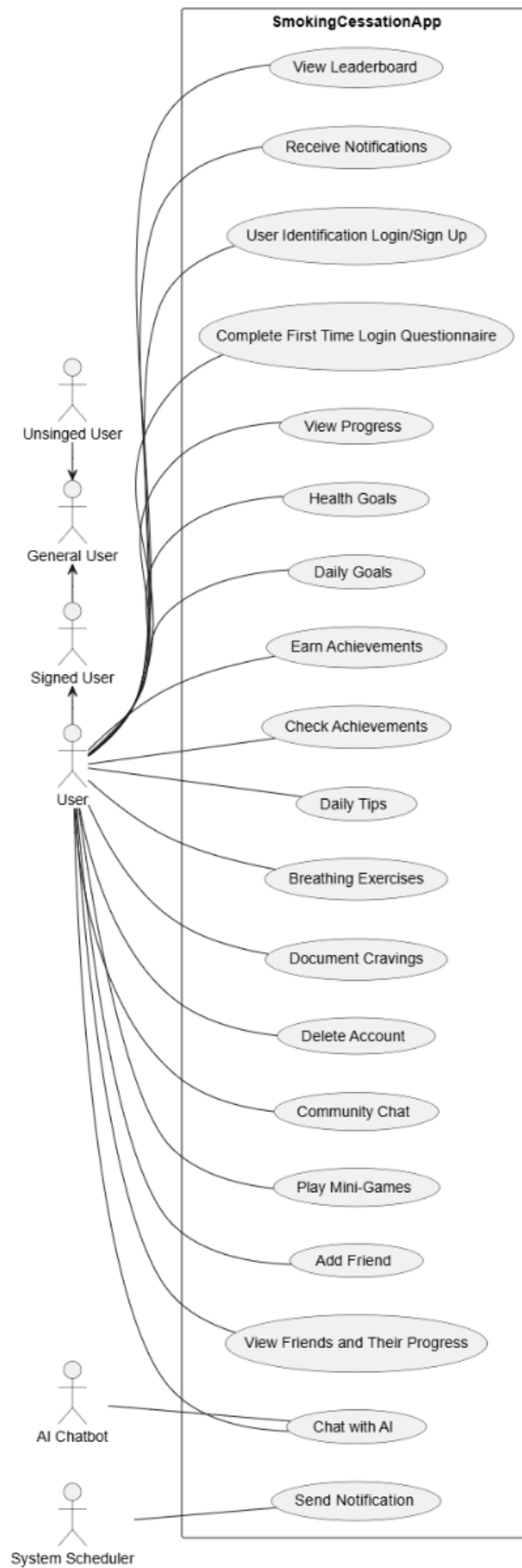
Πίνακας 6: Περιπτώσεις Χρήσης Συστήματος

ID	Τίτλος	Actor	Προϋποθέσεις	Περιγραφή	Αποτέλεσμα
UC1	Δημ/γία Λογ/μού	Χρήστης	Ο χρήστης διαθέτει email ή Google account.	Ο χρήστης ανοίγει την εφαρμογή, επιλέγει Sign Up, συμπληρώνει τα στοιχεία του ή συνδέεται μέσω Google.	Ο λογαριασμός δημιουργείται και αποθηκεύεται στο Firebase, ο χρήστης οδηγείται στο ερωτηματολόγιο πρώτης χρήσης.
UC2	Σύνδεση Χρήστη	Χρήστης	Ο λογαριασμός είναι ήδη καταχωρημένος.	Ο χρήστης εισάγει email και κωδικό ή επιλέγει σύνδεση μέσω Google.	Επιτυχής σύνδεση και μετάβαση στην αρχική οθόνη με τα daily goals.
UC3	Ερωτ/γιο	Χρήστης	Επιτυχής δημιουργία λογαριασμού.	Ο χρήστης απαντά σε ερωτήσεις σχετικά με το κάπνισμα, π.χ. τιγάρα/ημέρα, κόστος, διάρκεια καπνίσματος.	Δημιουργείται προσωποποιημένο προφίλ και προσαρμοσμένα daily tips.

UC4	Daily Goals	Χρήστης	Ολοκληρωμένη αρχικοποίηση προφίλ.	Ο χρήστης βλέπει καθημερινούς στόχους στην αρχική οθόνη, με δυνατότητα Learn More.	Ο χρήστης παρακολουθεί την πρόοδο και μπορεί να δει περισσότερες πληροφορίες μέσω του chatbot.
UC5	Daily Tips	Χρήστης	Ο λογαριασμός είναι ενεργός.	Ο χρήστης λαμβάνει μία ημερήσια συμβουλή που ξεκλειδώνονται σταδιακά και παραμένουν διαθέσιμες.	Ο χρήστης έχει πρόσβαση σε καθημερινή καθοδήγηση για το κόψιμο του καπνίσματος.
UC6	Επιτ/τα και XP	Χρήστης	Ο χρήστης προχωρά σε στόχους.	Η εφαρμογή εμφανίζει λίστα επιτευγμάτων, κάποια ολοκληρωμένα και κάποια σε εξέλιξη, με XP.	Ο χρήστης παρακολουθεί την πρόοδο του και ενισχύεται το κίνητρό του.
UC7	Leaderboard	Χρήστης	Σύνδεση στο internet.	Ο χρήστης βλέπει την κατάταξη στην κατηγορία του (<40 ή ≥40 ημέρες), με εμφάνιση top-10 και της δικής του θέσης.	Ενίσχυση ανταγωνισμού και κίνητρο συνέχισης.
UC8	Community Chat	Χρήστης	Ο χρήστης είναι συνδεδεμένος.	Ο χρήστης εισέρχεται σε chat room. Αν το δωμάτιο είναι γεμάτο, δημιουργείται αυτόματα νέο.	Ο χρήστης επικοινωνεί με άλλα μέλη, μοιράζεται εμπειρίες και λαμβάνει υποστήριξη.
UC9	Chatbot	AI Chatbot	Ενεργή σύνδεση στο Internet και πρόσβαση στο chatbot.	Ο χρήστης πληκτρολογεί μια ερώτηση σχετικά με το κάπνισμα ή την υγεία. Το chatbot επεξεργάζεται την ερώτηση, αντλεί δεδομένα με RAG και λαμβάνει υπόψη τη μνήμη συνομιλίας.	Ο χρήστης λαμβάνει απάντηση σχετική, προσωποποιημένη και τεκμηριωμένη με βάση επιστημονικές πηγές.
UC10	Καταγραφή Craving Logs	Χρήστης	Ο χρήστης είναι συνδεδεμένος στην εφαρμογή.	Ο χρήστης καταγράφει ένα craving, εισάγοντας ένταση, αιτία (π.χ. μετά το φαγητό) και τρόπο διαχείρισης.	Δημιουργείται νέα καταχώρηση στη προσωπική συλλογή cravings στο Firestore.
UC11	Προβολή Craving Logs	Χρήστης	Υπάρχουν διαθέσιμες καταγραφές.	Ο χρήστης βλέπει το ιστορικό των δικών του cravings αλλά και ανώνυμες καταγραφές άλλων χρηστών για έμπνευση.	Ο χρήστης αντλεί γνώση και υποστήριξη από τις εμπειρίες άλλων, παρακολουθώντας ταυτόχρονα τη δική του πρόοδο.
UC12	Προσθήκη Φίλου	Χρήστης	Και οι δύο χρήστες υπάρχουν στο Firestore.	Ο χρήστης αναζητά άλλους μέσω username/social tag και στέλνει αίτημα φιλίας.	Δημιουργία δικτύου φίλων.

UC13	Παρακολούθηση Προόδου	Χρήστης	Ο χρήστης είναι συνδεδεμένος στην εφαρμογή.	Καταγραφή ημερών αποχής, τσιγάρων που δεν αγοράστηκαν και χρημάτων που εξοικονομήθηκαν.	Εμφάνιση προόδου, πόντων, επιπέδων και leaderboard.
UC14	Αποστολή Ειδοποίησης	System Scheduler	Έχει δώσει άδεια στις ειδοποιήσεις και τις έχει ενεργοποιήσει.	Ο χρήστης λαμβάνει ενθαρρυντικά μηνύματα.	Ενθάρρυνση και απόδοση κινήτρου.
UC15	Health Goals	Χρήστης	Ο χρήστης είναι συνδεδεμένος στην εφαρμογή.	Ο χρήστης βλέπει την πρόοδο της υγείας του σε σχετική οθόνη, μέσω μπάρας προόδου και ποσοστό επίτευξης.	Ενθάρρυνση και απόδοση κινήτρου.
UC16	Mini Games	Χρήστης	Ο χρήστης είναι συνδεδεμένος στην εφαρμογή.	Ο χρήστης παίζει απλά παιχνίδια που βρίσκονται σε σχετικό μενού και είναι ενσωματωμένα στο σύστημα.	Απόσπαση προσοχής από επιθυμία καπνίσματος.
UC17	Παρ/ση Προόδου Φίλων	Χρήστης	Ο χρήστης είναι συνδεδεμένος στην εφαρμογή και έχει προσθέσει φίλους.	Ο χρήστης έχει την δυνατότητα στην λίστα με τους φίλους του να δει μερικά στατιστικά και την πρόοδό τους.	Ενίσχυση ανταγωνισμού και κίνητρο συνέχισης.
UC18	Διαγραφή Λογ/μου	Χρήστης	Ο λογαριασμός είναι ήδη καταχωρημένος, έχει πρόσφατα συνδεθεί ο χρήστης.	Ο χρήστης έχει την δυνατότητα από την οθόνη ρυθμίσεων να διαγράψει τον λογαριασμό του.	Διαγράφονται όλα τα δεδομένα του χρήστη και το προφίλ του οριστικά.

Τα Use Case διαγράμματα (διαγράμματα περιπτώσεων χρήσης) αποτελούν βασικό εργαλείο στη φάση της ανάλυσης απαιτήσεων ενός πληροφοριακού συστήματος, καθώς περιγράφουν τις λειτουργικές απαιτήσεις του συστήματος με τρόπο κατανοητό τόσο από προγραμματιστές όσο και από μη τεχνικούς ενδιαφερόμενους. Ο κύριος στόχος τους είναι η απεικόνιση των αλληλεπιδράσεων μεταξύ των χρηστών (actors) και των λειτουργιών (use cases) της εφαρμογής. Τα διαγράμματα αυτά επιτρέπουν την καθαρή κατανόηση του τι μπορεί να κάνει ο χρήστης, πώς το σύστημα ανταποκρίνεται και που υπάρχουν πιθανές επεκτάσεις ή περιορισμοί. Παράλληλα, εντοπίζονται εύκολα τα σημεία διασύνδεσης μεταξύ των υποσυστημάτων, γεγονός που διευκολύνει τη σχεδίαση και υλοποίηση της εφαρμογής με συνέπεια και αποδοτικότητα.



Εικόνα 7: Use Case Diagram

4. Τεχνολογίες και Εργαλεία Υλοποίησης

4.1 Επιλογή Flutter και η Δομή του Συστήματος

Το Flutter αποτελεί ένα σύγχρονο εργαλείο ανάπτυξης εφαρμογών, που γνωρίζει διαρκώς αυξανόμενη απήχηση στην κοινότητα των επαγγελματιών προγραμματιστών και στην ερευνητική κοινότητα. Αναπτύχθηκε από τη Google και παρουσιάστηκε για πρώτη φορά δημόσια στο συνέδριο Dart Developer Summit τον Μάιο του 2017, ενώ η πρώτη σταθερή έκδοση κυκλοφόρησε τον Δεκέμβριο του 2018. Ο στόχος του Flutter είναι να διευκολύνει τη δημιουργία native εφαρμογών για κινητά, web και desktop, μέσω ενός ενιαίου κώδικα (single codebase), χωρίς συμβιβασμούς στην απόδοση ή την εμπειρία χρήστη [39].

Το Flutter χρησιμοποιεί ως βασική του γλώσσα προγραμματισμού τη Dart, επίσης σχεδιασμένη από τη Google. Η Dart είναι αντικειμενοστραφής, διαθέτει ισχυρή στατική τυποποίηση και υποστηρίζει τόσο Just in Time (JIT) όσο και Ahead of Time (AOT) compilation. Αυτό σημαίνει ότι μπορεί να προσφέρει ταχύτατη ανανέωση του UI κατά την ανάπτυξη μέσω της λειτουργίας Hot Reload, αλλά και υψηλή απόδοση κατά την εκτέλεση της εφαρμογής σε παραγωγικά περιβάλλοντα [40].

Αυτό που διαφοροποιεί το Flutter από άλλα frameworks όπως React Native ή Xamarin είναι το γεγονός ότι δεν χρησιμοποιεί τα native UI components του λειτουργικού συστήματος, αλλά αντ' αυτού βασίζεται σε μια δική του μηχανή γραφικών, τη Skia Graphics Engine. Έτσι, το Flutter έχει πλήρη έλεγχο των εικονοστοιχείων (pixels) στην οθόνη και μπορεί να δημιουργήσει ομοιογενή εμπειρία χρήστη σε όλες τις πλατφόρμες, ανεξαρτήτως συσκευής ή λειτουργικού συστήματος. Ταυτόχρονα, αυτή η προσέγγιση εξαλείφει ασυμβατότητες και διαφορές μεταξύ Android και iOS, γεγονός που απλοποιεί τον σχεδιασμό του UI και μειώνει σημαντικά το debugging time [39].

Το Widget-based αρχιτεκτονικό μοντέλο του Flutter είναι ένα ακόμη χαρακτηριστικό που το καθιστά ιδιαίτερα ελκυστικό. Όλα, από ένα απλό κουμπί μέχρι σύνθετες σελίδες και animations, είναι widgets που μπορούν να συνδυαστούν, να κληρονομηθούν και να προσαρμοστούν. Υπάρχει ξεκάθαρος διαχωρισμός μεταξύ StatelessWidget (μη δυναμικού περιεχομένου) και StatefulWidget (δυναμικού περιεχομένου που αλλάζει κατά την αλληλεπίδραση του χρήστη).

Η Dart Virtual Machine μαζί με το rendering engine επιτρέπουν στο Flutter να μεταφράζει απευθείας τις οδηγίες σε χαμηλού επιπέδου εντολές χωρίς να περνά από native bridges, όπως γίνεται σε άλλα frameworks. Αυτή η σχεδίαση προσφέρει πολύ καλή απόδοση και μειωμένη καθυστέρηση στις διεπαφές χρήστη.

Στο πλαίσιο της παρούσας εφαρμογής, το Flutter επιλέχθηκε για την ταχύτατη ανάπτυξη και ανανέωση του UI μέσω του hot reload και τη γενικότερη ευελιξία του στον χειρισμό σύνθετων διεπαφών. Η λειτουργική αρχιτεκτονική της εφαρμογής βασίστηκε σε μια modular προσέγγιση με διαχωρισμό μεταξύ οθονών, παροχών δεδομένων (providers), υπηρεσιών (services), και μοντέλων (models). Το routing μεταξύ των οθονών πραγματοποιείται μέσω Navigator, ενώ ο χειρισμός κατάστασης επιτυγχάνεται μέσω του πακέτου provider, που ενσωματώνεται άφογα με το widget tree του Flutter.

Το αρχείο `main.dart` χρησιμεύει ως το σημείο εισόδου, όπου ρυθμίζεται η αρχική εκκίνηση της εφαρμογής, η δρομολόγηση (routing) μεταξύ των οθονών, καθώς και η σύνδεση με την κατάσταση της εφαρμογής (state).

Ο σχεδιασμός των οθονών βασίστηκε σε προκαθορισμένο σχεδιαστικό σύστημα που περιλαμβάνει grid-based διατάξεις, consistent typography και custom animated components μέσω του `flutter_animate`. Η αισθητική της εφαρμογής είναι εναρμονισμένη με τη θεματολογία της, δίνοντας έμφαση στην καθαρότητα, στην ενθαρρυντική παρουσίαση της προόδου του χρήστη και στην εργονομική παρουσίαση δεδομένων (όπως στόχοι υγείας, leaderboard, επιτεύγματα κ.λπ.).

Η εφαρμογή ακολουθεί επίσης τη φιλοσοφία του modular design. Κάθε λειτουργικό χαρακτηριστικό, όπως η φόρμα πρώτης εισόδου, το ημερήσιο tip, ή το widget για αναπνευστικές ασκήσεις, έχει υλοποιηθεί ως ξεχωριστό Widget ή Screen, καθιστώντας εύκολο debugging και την επεκτασιμότητα. Η διαχείριση κατάστασης έγινε με το πακέτο provider, το οποίο επιτρέπει την αμφίδρομη επικοινωνία μεταξύ UI και δεδομένων, διευκολύνοντας την ενημέρωση του interface σε πραγματικό χρόνο χωρίς περιττή πολυπλοκότητα.

Τέλος, το Flutter συνοδεύεται από πλούσια τεκμηρίωση και υποστήριξη. Η ενσωμάτωσή του με Firebase, REST APIs, animation libraries και AI SDKs (όπως αυτό του DeepSeekAI για το chatbot) επιτρέπει γρήγορη και σταθερή ανάπτυξη προηγμένων λειτουργιών όπως ειδοποιήσεις, authentication, real-time database και φυσικά LLM integrations για personalized υπηρεσίες στον χρήστη.

Τέλος, η δυνατότητα hot reload του Flutter επέτρεψε την ταχεία ανάπτυξη και δοκιμή της εφαρμογής, μειώνοντας σημαντικά τον αναπτυξιακό κύκλο και προσφέροντας ένα agile μοντέλο υλοποίησης, ιδανικό για ερευνητικά και πειραματικά project όπως το παρόν.

4.2 Αρχιτεκτονική Συστήματος, UI και State Management

Η αρχιτεκτονική μιας εφαρμογής επηρεάζει άμεσα την ευχρηστία, επεκτασιμότητα και συντηρησιμότητά της. Η ανάγκη για καθαρό διαχωρισμό μεταξύ της επιχειρησιακής λογικής (business logic) και του γραφικού περιβάλλοντος (UI) στις σύγχρονες εφαρμογές είναι ζωτικής σημασίας. Αυτό είναι ιδιαίτερα σημαντικό ειδικά όταν η εφαρμογή περιλαμβάνει ποικίλα δεδομένα, δυναμική ενημέρωση, αλληλεπίδραση με εξωτερικά APIs, authentication και πολλαπλές οθόνες. Στην παρούσα εφαρμογή, η διαχείριση κατάστασης επιτυγχάνεται με το Provider, ένα από τα πιο δημοφιλή και συνιστώμενα patterns για Flutter. Το Provider προσφέρει καθαρή αρχιτεκτονική και αποδοτική διαχείριση κατάστασης (state management), εξασφαλίζοντας ευελιξία, modular σχεδιασμό, και σαφή διαχωρισμό μεταξύ UI και επιχειρησιακής λογικής.

Κατά τα πρώτα χρόνια του Flutter, οι developers χρησιμοποιούσαν κυρίως το `setState()` για την αλλαγή του UI, το οποίο όμως ήταν κατάλληλο μόνο για μικρές εφαρμογές. Καθώς το Flutter κέρδιζε έδαφος στην παραγωγή πολύπλοκων εφαρμογών, αναδύθηκε η ανάγκη για πιο κλιμακούμενα μοντέλα διαχείρισης κατάστασης. Αυτή η προσέγγιση ήταν απλή αλλά γρήγορα αποδείχθηκε ανεπαρκής για μεγαλύτερες εφαρμογές, καθώς προκαλούσε σύγχυση, ελαττωμένη απόδοση και κακή αρχιτεκτονική δομή. Η κοινότητα αναζήτησε καλύτερες λύσεις, με την εμφάνιση αρχιτεκτονικών όπως BLoC (Business Logic Component), Redux,

MobX, και τελικά, την ανάπτυξη του Provider από τον Remi Rousselet. Το 2019, ο Remi Rousselet εισήγαγε τη βιβλιοθήκη Provider, η οποία έγινε το πρότυπο στην κοινότητα του Flutter και αργότερα υιοθετήθηκε επίσημα από την Google, καθώς συνδυάζει απλότητα, απόδοση και modularity. Βασίζεται στην υποδομή των InheritedWidgets του Flutter, αλλά με πιο φιλική σύνταξη και περισσότερες δυνατότητες.

Το Provider είναι ένα inherited widget wrapper και ένα dependency injection framework και state management solution που επιτρέπει στο Flutter να διαχειρίζεται και να μοιράζει την κατάσταση (state) και services σε ολόκληρο το widget tree, χωρίς περίπλοκη κωδικοποίηση ή επανυπολογισμούς. Οι providers δημιουργούνται στο root της εφαρμογής (ή σε κάποιο subtree) και παρέχουν δεδομένα ή λογική σε οποιοδήποτε child widget το ζητήσει, χωρίς να χρειάζεται να περάσει χειροκίνητα μέσω constructors ή callbacks.

Ο κύκλος ζωής του Provider ξεκινά με τη δημιουργία του, η οποία ορίζεται στο main.dart ή σε κάποιο MultiProvider block και περιλαμβάνει instances μοντέλων, υπηρεσιών ή ρυθμίσεων. Στη συνέχεια, ακολουθεί η κατανάλωση από το UI, όπου τα widgets μπορούν να χρησιμοποιούν Consumer, Provider.of, context.watch, context.read και άλλα για να αποκτήσουν πρόσβαση στα δεδομένα και να αντιδράσουν σε αλλαγές. Η ενημέρωση του state γίνεται μέσω notifyListeners() από τις αντίστοιχες ChangeNotifier κλάσεις, αναγκάζοντας το UI να επαναδομηθεί μόνο εκεί που χρειάζεται.

Κύρια στοιχεία του Provider είναι το ChangeNotifier, μια κλάση που ειδοποιεί τους listeners όταν αλλάζει κάποιο state. Το ChangeNotifierProvider, το τυλίγει ένα widget και του παρέχει πρόσβαση στο μοντέλο. Άλλα σημαντικά στοιχεία του αποτελούν το Consumer οι μέθοδοι context.watch() και context.read(), που επιτρέπουν στο widget να αντιδρά σε αλλαγές ή να αποκτή πρόσβαση στο μοντέλο.

Τα πλεονεκτήματα του περιλαμβάνουν χαμηλό learning curve, καλή υποστήριξη για reactive UI, καθαρό διαχωρισμό μεταξύ UI και business logic και καταλληλότητα για μικρές και μεγάλες εφαρμογές. Η χρήση του Provider ως αρχιτεκτονική λύση προσφέρει επιπλέον πλεονεκτήματα στην εφαρμογή, όπως απόδοση, αφού το UI ανανεώνεται μόνο εκεί όπου αλλάζει το σχετικό state. Προσφέρει επίσης καθαρή αρχιτεκτονική, καθώς ο διαχωρισμός UI και λογικής καθιστά τον κώδικα πιο επεκτάσιμο και συντηρήσιμο. Επιπλέον, η σύνδεση με Firebase είναι αρκετά εύκολη, αφού η διαχείριση της αυθεντικοποίησης και των Firestore reads/writes μπορεί να οργανωθεί σε services που εκτίθενται μέσω provider. Τέλος, η ενσωμάτωση με widgets είναι απλή, καθώς οποιοδήποτε widget μπορεί να αποκτήσει εύκολα πρόσβαση στην κατάσταση χωρίς πολύπλοκο passing από γονικά components.

Ο Provider αποτελεί την ραχοκοκαλιά της διαχείρισης δεδομένων και κατάστασης της παρούσας εφαρμογής. Επιτρέπει την εύελικτη, επεκτάσιμη και καθαρή οργάνωση των modules της εφαρμογής, συνδέοντας το UI με το state χωρίς περίπλοκη λογική. Μαζί με το backend που έχει υλοποιηθεί για το chatbot, συνθέτει ένα σύγχρονο, modular σύστημα όπου το frontend έχει σημαντική αυτονομία και λογική μέσω Flutter/Provider, και το Python backend εξυπηρετεί εξειδικευμένες λειτουργίες τεχνητής νοημοσύνης. Μερικοί από τους providers που υλοποιήθηκαν είναι:

UserProvider: Διαχειρίζεται τις πληροφορίες του χρήστη (όνομα, φύλο, ημερομηνία διακοπής, XP, rank, UID) και ενημερώνεται κατά την αυθεντικοποίηση ή μετά από αλλαγές του Firestore. Εξυπηρετεί επίσης το UI του Home Screen, του Profile Drawer και του Settings.

LeaderboardProvider: Διατηρεί τη λίστα των κορυφαίων χρηστών, υπολογίζει rankings, διαχωρίζει τους χρήστες ανάλογα με τη διάρκεια διακοπής καπνίσματος και ενημερώνεται με βάση τα δεδομένα του Firestore.

AchievementsProvider: Διαχειρίζεται τους στόχους/επιτεύγματα και την πρόοδό τους. Παρέχει το κατάλληλο progress σε XP, περιγραφές, καθώς και αν κάποιος στόχος έχει ολοκληρωθεί ή όχι.

NotificationProvider: Υπεύθυνος για τις ειδοποιήσεις, τόσο άμεσες (triggered από in-app events) όσο και προγραμματισμένες (scheduled), που λειτουργούν ακόμη κι αν η εφαρμογή είναι στο background.

4.3 Firebase: Authentication, Firestore, Messaging

Το Firebase αποτελεί μια πλήρη Backend as a Service (BaaS) πλατφόρμα, η οποία ανήκει στη Google και προσφέρει ένα ευρύ φάσμα υπηρεσιών για την ανάπτυξη εφαρμογών κινητών, ιστού και desktop. Αρχικά δημιουργήθηκε το 2011 από τον Andrew Lee και James Tamplin με στόχο την παροχή real-time δεδομένων μέσω ενός απλού API. Το 2014 εξαγοράστηκε από τη Google και σταδιακά μετατράπηκε σε μία εκτενή πλατφόρμα ανάπτυξης εφαρμογών με πλήθος εργαλείων, όπως authentication, real-time και cloud βάσεις δεδομένων, αποστολή ειδοποιήσεων, παρακολούθηση σφαλμάτων (Crashlytics), A/B testing, remote config, analytics και πολλά άλλα [41].

Το Firebase προσφέρει cloud υποδομές που απαλλάσσουν τον προγραμματιστή από την ανάγκη δημιουργίας και συντήρησης ενός δικού του backend. Αντίθετα, προσφέρει έτοιμες λύσεις, πλήρως παραμετροποιήσιμες, με δυνατότητα ενσωμάτωσης σε εφαρμογές που βασίζονται σε Flutter, React Native, Android, iOS, ή ακόμη και σε εφαρμογές web. Η ενσωμάτωση του Firebase με το Flutter γίνεται μέσω των επίσημων firebase πακέτων, τα οποία καλύπτουν όλο το εύρος των λειτουργιών του οικοσυστήματος.

Firebase Authentication

Η υπηρεσία Firebase Authentication αποτελεί μία από τις πιο δημοφιλείς λύσεις ταυτοποίησης χρηστών. Παρέχει έτοιμα APIs και SDKs για τη διαχείριση εγγραφών, συνδέσεων, αποστολής επιβεβαιωτικών email, και επαναφοράς κωδικών. Υποστηρίζει πολλαπλές μεθόδους ταυτοποίησης όπως email/κωδικός, Google, Facebook, Apple ID, phone number (OTP), και anonymous login.

Η αρχιτεκτονική της βασίζεται σε secure tokens (JWT), που αποδίδονται σε κάθε συνεδρία χρήστη και χρησιμοποιούνται για την πιστοποίηση των requests σε άλλες υπηρεσίες του Firebase, όπως το Firestore και το Cloud Functions. Όλα τα δεδομένα επικοινωνίας είναι πλήρως κρυπτογραφημένα και η υπηρεσία υποστηρίζει OAuth 2.0 για εξωτερικές ταυτοποιήσεις.

Στην παρούσα εφαρμογή, το Firebase Auth χρησιμοποιείται για δημιουργία και ταυτοποίηση λογαριασμών χρηστών μέσω email και Google login, για διαχείριση ενεργών συνεδριών (session management), καθώς και για ανάκτηση στοιχείων του χρήστη και χρήση UID ως μοναδικό αναγνωριστικό στις βάσεις δεδομένων.

Cloud Firestore

Το Cloud Firestore είναι μία NoSQL βάση δεδομένων εγγράφων (document database) που αντικατέστησε σταδιακά την παλαιότερη Firebase Realtime Database, προσφέροντας πιο ευέλικτη και επεκτάσιμη υποδομή. Αντί για παραδοσιακές γραμμικές βάσεις (π.χ. SQL), το Firestore αποθηκεύει τα δεδομένα σε μορφή JSON-like εγγράφων, τα οποία είναι οργανωμένα σε collections.

Κάθε έγγραφο (document) μπορεί να περιέχει key-value pairs, arrays, embedded objects και μπορεί να ερωτηθεί με indexing και φίλτρα (queries). Η λειτουργία του υποστηρίζεται από Google Cloud Infrastructure και επιτρέπει πραγματικό συγχρονισμό σε πολλαπλές πλατφόρμες με ελάχιστο latency. Επίσης, υποστηρίζει offline access με τοπικό caching, κάτι ιδιαίτερα σημαντικό για εφαρμογές όπως η παρούσα.

Στο πλαίσιο της παρούσας εφαρμογής, το Firestore αξιοποιείται για, αποθήκευση και ανάκτηση προφίλ χρήστη με χρήση UID, καταγραφή επιτευγμάτων, XP και ημερομηνίας διακοπής καπνίσματος, για αποθήκευση εξατομικευμένων δεδομένων (π.χ. tips, πρόοδοι, φίλοι), καθώς και για την υλοποίηση chatrooms (μέσω subcollections ανά συνομιλία).

Firebase Cloud Messaging (FCM)

Η υπηρεσία Firebase Cloud Messaging επιτρέπει την αποστολή ειδοποιήσεων και δεδομένων (data messages) σε εφαρμογές Android, iOS και Web. Υποστηρίζει και στοχευμένη αποστολή (targeted messaging) ανά χρήστη, topic ή device token, καθώς και προγραμματισμό ειδοποιήσεων μέσω cloud functions ή τρίτων εργαλείων. Τα μηνύματα μπορούν να είναι είτε απλά (notification payload, για εμφάνιση στο UI), είτε πιο σύνθετα (data payload, για επεξεργασία εντός του app). Επιπλέον, παρέχεται analytics για την παρακολούθηση της αποδοτικότητας κάθε καμπάνιας ειδοποιήσεων.

Στην εφαρμογή αυτή, το FCM χρησιμοποιείται για αποστολή motivational ειδοποιήσεων με βάση την πρόοδο του χρήστη, υπενθυμίσεις για breathing techniques και daily tips και για ενημερώσεις για νέα επιτεύγματα.

Η επιλογή του Firebase έγινε με βάση συγκεκριμένων κριτηρίων. Αρχικά, αποτελεί ένα ενιαίο οικοσύστημα που καλύπτει authentication, αποθήκευση δεδομένων, ειδοποιήσεις και analytics. Ταυτόχρονα, υποστηρίζει άμεση ενσωμάτωση με Flutter μέσω των επίσημων πακέτων. Προσφέρει επεκτασιμότητα και σταθερότητα, βασισμένη στις υποδομές της Google, ενώ προφανής είναι η απλότητα στην ανάπτυξη, χωρίς ανάγκη για custom backend. Τέλος, προσφέρει υψηλή ασφάλεια, μέσω κανόνων πρόσβασης (security rules) που βασίζονται σε ρόλους χρηστών και auth state.

4.4 Μεγάλα Γλωσσικά Μοντέλα, Langchain και RAG

Η ενσωμάτωση μεγάλων γλωσσικών μοντέλων (Large Language Models - LLMs) σε εφαρμογές, όπως αυτή που αναπτύσσεται στο πλαίσιο της παρούσας εργασίας, έχει αλλάξει ολοκληρωτικά τον τρόπο με τον οποίο οι χρήστες αλληλοεπιδρούν με την πληροφορία. Για την υλοποίηση ενός εξατομικευμένου, ευφυούς chatbot, το σύστημα αξιοποιεί ένα backend υλοποιημένο σε Python, με βιβλιοθήκες όπως το Langchain και με σύνδεση σε LLM APIs για την εκτέλεση ερωτήσεων/απαντήσεων σε πραγματικό χρόνο, χρησιμοποιώντας και την τεχνική RAG (Retrieval-Augmented Generation).

Η έννοια των μεγάλων γλωσσικών μοντέλων καθιερώθηκε με την εμφάνιση του GPT-2 από την OpenAI το 2019, κυρίως με την επανάσταση που έφερε το GPT-3 το 2020. Τα μοντέλα αυτά εκπαιδεύονται πάνω σε τεράστιες ποσότητες φυσικού κειμένου και μπορούν να παράγουν απαντήσεις με συνεκτικότητα και ακρίβεια. Με την αύξηση ενδιαφέροντος για conversational agents, πολλές εταιρείες και πλατφόρμες άρχισαν να προσφέρουν LLM APIs, επιτρέποντας την εύκολη ενσωμάτωση της υπολογιστικής τους ισχύος σε εξωτερικές εφαρμογές [42].



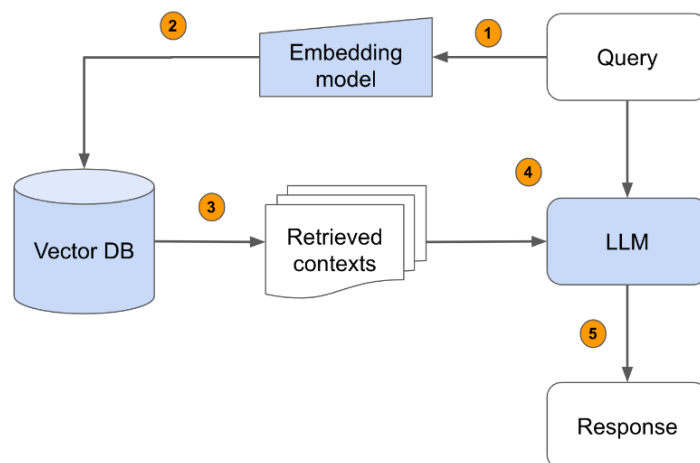
Εικόνα 8: LLM Flow

Παρόλα αυτά, τα LLMs έχουν σημαντικούς περιορισμούς, κυρίως στην ανάκτηση σύγχρονης ή domain-specific πληροφορίας. Τα μοντέλα είναι στατικά και δεν γνωρίζουν πληροφορίες που εμφανίστηκαν μετά την ημερομηνία εκπαίδευσής τους. Για να αντιμετωπιστεί αυτό το ζήτημα, αναπτύχθηκε η τεχνική Retrieval Augmented Generation, η οποία συνδυάζει τη ισχύ του LLM με την ανάκτηση σχετικής πληροφορίας από μία εξωτερική βάση δεδομένων ή μια βάση γνώσης.

Το Langchain είναι μια open-source βιβλιοθήκη της Python που δημιουργήθηκε το 2022 και αναπτύσσεται ενεργά μέχρι σήμερα. Ο σκοπός του Langchain είναι να παρέχει ένα ενιαίο πλαίσιο για τη δημιουργία σύνθετων και modular εφαρμογών που χρησιμοποιούν LLMs. Αυτό το πετυχαίνει διευκολύνοντας την ενσωμάτωση λειτουργιών όπως την αλληλεπίδραση με APIs LLMs (OpenAI, Cohere, OpenRouter, κ.ά.). Επιπλέον, βοηθάει στην ανάκτηση πληροφορίας μέσω Document Loaders και Vector Stores, καθώς και στην χρήση της RAG αρχιτεκτονικής με pipelines (Retrievers και Prompt Templates). Τέλος, διευκολύνει την δημιουργία agents με memory και context και προσφέρει Chain of Thought reasoning.

Το Langchain βασίζεται στη φιλοσοφία του modular abstraction, προσφέροντας building blocks για την κατασκευή chains, δηλαδή αλυσίδων λειτουργιών όπου η μία οδηγεί στην άλλη. Στην παρούσα εργασία, το Langchain χρησιμοποιείται για την κατασκευή ενός chatbot, ώστε οι απαντήσεις του να μην είναι απλώς γενικές, αλλά να αντλούν πληροφορία από ένα ειδικά διαμορφωμένο knowledge base με επιστημονικά δεδομένα για τη διακοπή του καπνίσματος, άρθρα, και παραπομπές [43].

Η λογική του RAG ακολουθεί την εξής ροή. Αρχικά, ο χρήστης εισάγει ένα ερώτημα στο chatbot, π.χ. Τι συμβαίνει στον οργανισμό μου την πρώτη εβδομάδα χωρίς τσιγάρο;. Έπειτα, το Langchain χρησιμοποιεί έναν Retriever για να αναζητήσει σχετικά αποσπάσματα από τη βάση δεδομένων κειμένων (π.χ. .txt αρχεία, markdown ή PDFs που έχουν γίνει embed). Στην συνέχεια, τα σχετικά κείμενα ενσωματώνονται (pass as context) μαζί με το ερώτημα του χρήστη στο prompt που στέλνεται στο LLM. Τέλος, το LLM (μέσω του OpenRouter API) απαντά λαμβάνοντας υπόψη όχι μόνο το ερώτημα, αλλά και την πληροφορία που του δόθηκε από τον Retriever. Αυτός ο μηχανισμός ενισχύει δραματικά την ακρίβεια, την επιστημονική εγκυρότητα και την contextual συνάφεια των απαντήσεων. Ειδικά σε ευαίσθητα θέματα υγείας, όπως η διακοπή του καπνίσματος, η παροχή απαντήσεων βασισμένων σε έγκυρα επιστημονικά δεδομένα είναι καθοριστικής σημασίας.



Εικόνα 9: Παράδειγμα Ροής RAG

Χρήση OpenRouter API ως γέφυρα με LLMs

Το OpenRouter.ai αποτελεί ένα από τα πιο ευέλικτα και ισχυρά APIs για την πρόσβαση σε διαφορετικά LLMs. Σε αντίθεση με APIs όπως το openai.com, το OpenRouter επιτρέπει τη σύνδεση με πολλαπλούς παρόχους μοντέλων, όπως OpenAI (GPT-4, GPT-3.5), Anthropic (Claude), Mistral, Google (Gemini), Meta (LLaMA), Cohere, Perplexity κτλ. Επιπλέον, υποστηρίζει routing με βάση την ταχύτητα ή την οικονομικότητα. Επιπλέον, παρέχει επιλογές fine-tuning και streaming output, απαραίτητα για responsive chat εφαρμογές. Μέσω ενός και μόνο API key, ο developer έχει πρόσβαση σε δεκάδες LLMs με ενοποιημένο τρόπο.

Στην παρούσα εργασία, το OpenRouter χρησιμοποιείται ως πύλη πρόσβασης σε μοντέλα μεγάλης ακρίβειας και απαντητικής ποιότητας, τα οποία συνεργάζονται με τον μηχανισμό RAG για να παράγουν συμβουλές, επιστημονικές εξηγήσεις και προσωπικά μηνύματα υποστήριξης στον χρήστη που προσπαθεί να διακόψει το κάπνισμα.

4.5 Python Backend FastAPI

Στην αρχιτεκτονική του συστήματος μας, η βασική δομή του backend χωρίζεται σε δύο μέρη. Αρχικά, η πλειοψηφία της επιχειρησιακής λογικής, διαχείρισης κατάστασης (state management) και της αλληλεπίδρασης με τον χρήστη υλοποιείται απευθείας στο περιβάλλον Flutter μέσω του Provider. Έπειτα, μια σημαντική αλλά απομονωμένη λειτουργία της εφαρμογής, το chatbot, υποστηρίζεται από ένα ξεχωριστό backend, το οποίο φιλοξενείται με τη βοήθεια του FastAPI και διασυνδέεται μέσω Ngrok κατά την ανάπτυξη και τις δοκιμές.

Το FastAPI είναι ένα μοντέρνο και υψηλών επιδόσεων web framework για τη γλώσσα Python, το οποίο παρουσιάστηκε για πρώτη φορά το 2018. Η δημιουργία του προέκυψε ως απάντηση σε ανάγκες που δεν κάλυπταν πλήρως υπάρχοντα frameworks, όπως το Flask και το Django. Η βασική ιδέα πίσω από το FastAPI ήταν η δημιουργία ενός εργαλείου που θα επιτρέπει τη συγγραφή γρήγορων, ασφαλών και πλήρως τυποποιημένων RESTful APIs, με όσο το δυνατόν λιγότερη προσπάθεια και χωρίς επανάληψη κώδικα [44].

Αυτό που καθιστά το FastAPI μοναδικό είναι η χρήση του τύπου δεδομένων (type annotations) της Python3.6+, σε συνδυασμό με τη βιβλιοθήκη Pydantic για την επαλήθευση

και την αυτόματη μετατροπή των εισερχόμενων δεδομένων. Αυτό επιτρέπει στον προγραμματιστή να δημιουργεί endpoints που είναι ακριβώς τεκμηριωμένα, αυτόματα διασφαλισμένα ως προς την εγκυρότητα των δεδομένων και πλήρως περιγράψιμα μέσω OpenAPI/Swagger χωρίς επιπλέον κώδικα.

Το FastAPI βασίζεται στο ASGI (Asynchronous Server Gateway Interface) και είναι χτισμένο πάνω στο Starlette framework, κάτι που του προσδίδει εξαιρετική υποστήριξη για ασύγχρονο προγραμματισμό. Η επιλογή του ASGI επιτρέπει στο FastAPI να υποστηρίζει πραγματικά ασύγχρονες λειτουργίες με `async/await`, γεγονός που το καθιστά ιδιαίτερα κατάλληλο για μοντέρνες εφαρμογές με υψηλή απόδοση, όπως APIs που καλούν εξωτερικά LLMs (π.χ. HuggingFace).

Μέσα σε σύντομο χρονικό διάστημα από την κυκλοφορία του, το FastAPI γνώρισε τεράστια αποδοχή από την κοινότητα της τεχνητής νοημοσύνης και της ανάπτυξης web APIs. Αρκετοί οργανισμοί το υιοθέτησαν και έγινε πρότυπο για Pythonic APIs που βασίζονται σε microservices, machine learning και AI.

Στην εφαρμογή αυτή, το FastAPI χρησιμοποιείται για τη δημιουργία ενός API endpoint, το οποίο δέχεται ερωτήσεις των χρηστών (σε μορφή κειμένου) και τις διαβιβάζει στο LLM μέσω της βιβλιοθήκης LangChain. Η απάντηση που δημιουργείται από το μοντέλο επιστρέφεται στο frontend ως JSON response. Το συγκεκριμένο API λειτουργεί αποκλειστικά για την επεξεργασία φυσικής γλώσσας και δεν εμπλέκεται σε καμία άλλη επιχειρησιακή διαδικασία της εφαρμογής.

Είναι κρίσιμο να υπογραμμιστεί ότι το Python backend δεν χειρίζεται καμία άλλη επιχειρησιακή λειτουργία της εφαρμογής πέρα από την υποστήριξη του chatbot. Όλα τα υπόλοιπα στοιχεία του backend, όπως η αποθήκευση και επεξεργασία δεδομένων προόδου του χρήστη, η εμφάνιση επιτευγμάτων, οι ειδοποιήσεις, ο υπολογισμός των στατιστικών και η προσωποποιημένη εμπειρία χρήσης, υλοποιούνται πλήρως στο Flutter μέσω Provider, που εξασφαλίζει αποδοτική και οργανωμένη διαχείριση της κατάστασης της εφαρμογής. Η επιλογή αυτής της υβριδικής προσέγγισης επιτρέπει τη σαφή διάκριση αρμοδιοτήτων μεταξύ των τεχνολογιών. Έτσι, διατηρείται η συνολική αρχιτεκτονική απλή, ευέλικτη και επεκτάσιμη.

Ngrok και Ανάπτυξη τούνελ για προσωρινή απομακρυσμένη πρόσβαση

Το Ngrok είναι ένα ευρέως διαδεδομένο εργαλείο για την προσωρινή έκθεση τοπικών υπηρεσιών στον παγκόσμιο ιστό μέσω ασφαλών HTTPS τούνελ. Δημιουργήθηκε το 2014 ως ένα open-source εργαλείο με στόχο την απλοποίηση της διαδικασίας του remote testing και της συνεργασίας κατά την ανάπτυξη web εφαρμογών. Καθώς η χρήση του αυξήθηκε, εξελίχθηκε σε μια επαγγελματική SaaS υπηρεσία με προηγμένες δυνατότητες και υποστήριξη για authentication, rate limiting, webhooks, custom domains και παρακολούθηση δικτύου .

Η ανάγκη που καλύπτει το Ngrok δημιουργείται κατά την ανάπτυξη ενός backend, ο server αυτός εκτελείται τοπικά, σε localhost, και είναι προσβάσιμος μόνο από τον ίδιο υπολογιστή. Σε περιπτώσεις όπου χρειάζεται η δοκιμή της εφαρμογής από εξωτερικές συσκευές, για παράδειγμα από ένα smartphone στο οποίο τρέχει η εφαρμογή, αυτό δημιουργεί εμπόδιο. Αυτό συμβαίνει διότι δεν είναι δυνατή η άμεση πρόσβαση στο localhost από άλλη συσκευή, εκτός αν ο server φιλοξενηθεί στο διαδίκτυο ή αν υπάρξει χειροκίνητη ρύθμιση του router και firewall, πράγμα πολύπλοκο και μη πρακτικό.

Ακριβώς εδώ επεμβαίνει το Ngrok, δημιουργώντας έναν ασφαλή, προσωρινό και άμεσα διαθέσιμο δημόσιο URL (HTTPS) που συνδέεται με τη θύρα στην οποία εκτελείται η τοπική εφαρμογή. Το εργαλείο αυτό δημιουργεί έναν end-to-end encrypted tunnel μεταξύ της τοπικής μηχανής και του cloud server του Ngrok. Έτσι, επιτρέπεται η πρόσβαση στην εφαρμογή μέσω του διαδικτύου χωρίς κανένα επιπλέον configuration.

Ο τρόπος λειτουργίας του είναι απλός, γεγονός που έχει συμβάλει σημαντικά στη δημοφιλία του. Με την εντολή `ngrok http 8000`, για παράδειγμα, δημιουργείται ένα προσωρινό domain, όπως `https://abc123.ngrok-free.app`, το οποίο δρομολογεί όλα τα αιτήματα προς την τοπική θύρα 8000. Αυτό το URL μπορεί στη συνέχεια να χρησιμοποιηθεί από οποιονδήποτε client, όπως μια εφαρμογή στο κινητό, για να επικοινωνήσει απευθείας με το backend.

Εκτός από τη διευκόλυνση του development, το Ngrok προσφέρει web-based dashboard για την παρακολούθηση των εισερχόμενων αιτημάτων, τη δυνατότητα επανάληψης requests για debugging, καθώς και στατιστικά χρήσης. Στην επί πληρωμή του εκδοχή, δίνει επιπλέον δυνατότητα μόνιμων custom subdomains και HTTP basic authentication, χαρακτηριστικά χρήσιμα για πιο ασφαλή και επαναλαμβανόμενη χρήση.

Στην παρούσα εφαρμογή, το Ngrok χρησιμοποιείται αποκλειστικά για την ανάπτυξη και δοκιμή της μονάδας chatbot, η οποία εκτελείται τοπικά μέσω FastAPI. Κατά την περίοδο ανάπτυξης, ενεργοποιείται το Ngrok για να εκθέσει το endpoint `/chat` που δέχεται τα μηνύματα του χρήστη, ώστε η εφαρμογή Flutter να μπορεί να στείλει αιτήματα HTTP στο backend, ακόμη και όταν αυτό δεν έχει ανέβει σε cloud ή σε δημόσιο server.

Η επιλογή του Ngrok καθιστά τη διαδικασία δοκιμής του συστήματος άμεση, ευέλικτη και πλήρως ανεξάρτητη από σύνθετες διαδικασίες deployment. Αποτελεί με αυτόν τον τρόπο ένα εργαλείο γεφύρωσης μεταξύ της τοπικής ανάπτυξης και του πραγματικού περιβάλλοντος χρήσης. Επιπλέον, ενδείκνυται και για την παρουσίαση λειτουργιών σε τρίτους (π.χ. επιβλέποντες, testers, χρήστες), χωρίς την ανάγκη μεταφόρτωσης της εφαρμογής σε εξωτερικό server ή πλατφόρμα cloud.

4.6 Android Studio

Το Android Studio είναι το επίσημο Integrated Development Environment (IDE) για την ανάπτυξη εφαρμογών Android. Δημιουργήθηκε και υποστηρίζεται από τη Google, σε συνεργασία με την εταιρεία JetBrains και βασίζεται στο IntelliJ IDEA. Η πρώτη του έκδοση παρουσιάστηκε το 2013, κατά τη διάρκεια του συνεδρίου Google I/O και αντικατέστησε το παλαιότερο Eclipse Android Development Tools (ADT), το οποίο είχε χρησιμοποιηθεί ευρέως κατά την προηγούμενη δεκαετία [45].

Η βασική ανάγκη για το Android Studio προέκυψε από τον αυξανόμενο όγκο και τη πολυπλοκότητα της ανάπτυξης του Android, καθώς οι εφαρμογές γίνονταν πιο σύνθετες και απαιτητικές. Η Google αποφάσισε να προσφέρει ένα πλήρως ενοποιημένο περιβάλλον το οποίο θα εξυπηρετούσε τις ανάγκες των αρχαρίων και των επαγγελματιών προγραμματιστών Android.

Το Android Studio παρέχει ένα ολοκληρωμένο σύνολο εργαλείων για κάθε στάδιο του κύκλου ζωής μιας εφαρμογής Android, από τον σχεδιασμό του UI μέχρι την τελική διάθεση

της εφαρμογής στο Google Play Store. Μερικές από τις βασικές του δυνατότητες περιλαμβάνουν:

- **Ενσωματωμένο Emulator (Android Virtual Device AVD):** Επιτρέπει την άμεση προσομοίωση συσκευών Android με διάφορες διαστάσεις οθόνης, εκδόσεις λειτουργικού και χαρακτηριστικά hardware, χωρίς την ανάγκη για φυσική συσκευή.
- **Code Editor με Υποστήριξη για Kotlin, Java και Dart (με Flutter Plugin):** Ο επεξεργαστής κώδικα προσφέρει έξυπνη συμπλήρωση (IntelliSense), refactoring, linting και debugging δυνατότητες.
- **UI Designer (Layout Editor):** Επιτρέπει τον οπτικό σχεδιασμό διεπαφών χρήστη μέσω drag-and-drop εργαλείων, ενώ παράλληλα διατηρεί συγχρονισμένη την XML περιγραφή του UI.
- **Gradle Build System:** Ένα πανίσχυρο εργαλείο αυτοματοποίησης του build, που υποστηρίζει modular αρχιτεκτονικές, πολλαπλά flavors και δυναμική παραμετροποίηση της εφαρμογής.
- **Firebase Integration:** Υπάρχει άμεση υποστήριξη για υπηρεσίες Firebase (authentication, cloud messaging, real-time database, analytics, κ.ά.) με auto-setup wizard.
- **Profiling και Performance Tools:** Ολοκληρωμένα εργαλεία για τη μέτρηση απόδοσης (CPU, RAM, GPU, δικτύου), την ανίχνευση leaks και την αξιολόγηση του frame rate της εφαρμογής.
- **Debugging σε Πραγματικό Χρόνο:** Υποστηρίζει debugging με breakpoints, watch expressions, live memory inspection και debugging σε πραγματικές ή εικονικές συσκευές.
- **Flutter και Dart Integration:** Μέσω των αντίστοιχων plugins, το Android Studio μετατρέπεται σε πλήρες περιβάλλον ανάπτυξης Flutter εφαρμογών. Περιλαμβάνει hot reload, debugger, device manager και υποστήριξη για pubspec.yaml.

Στην παρούσα εργασία, το Android Studio χρησιμοποιήθηκε κυρίως σε συνδυασμό με το Flutter SDK για την εκτέλεση και δοκιμή σε Android Emulator. Ο ενσωματωμένος emulator του Android Studio (Android Virtual Device Manager) χρησιμοποιείται συχνά για τον έλεγχο της λειτουργίας της εφαρμογής σε διαφορετικές οθόνες, αναλύσεις και εκδόσεις Android. Με αυτόν τον τρόπο εξασφαλίζεται η συμβατότητα και η ομαλή εμπειρία χρήσης σε ευρύ φάσμα συσκευών. Επιπλέον, χρησιμοποιήθηκε για την καταγραφή και ανάλυση των σφαλμάτων, καθώς παρέχει εξειλιγμένα εργαλεία debugging και logcat για την καταγραφή λαθών, exceptions και επιδόσεων. Αυτά χρησιμοποιούνται για την επίλυση προβλημάτων σε πραγματικό χρόνο κατά την εκτέλεση της εφαρμογής σε emulator ή φυσική συσκευή. Επίσης, το Android Studio επιτρέπει την διαχείριση Android SDKs, αφού η εγκατάσταση, ενημέρωση και συντήρηση των διαφόρων εκδόσεων του Android SDK, build-tools και system images γίνεται μέσα από αυτό. Επιπλέον, επιτρέπει και στην δημιουργία Signed APK ή App Bundle (AAB). Για τη δοκιμή της εφαρμογής σε φυσική συσκευή, γίνεται build και export του APK αρχείου μέσω του Android Studio, το οποίο στη συνέχεια φορτώνεται τοπικά για χειροκίνητο testing. Τέλος, επιλέχθηκε να χρησιμοποιηθεί και για το Flutter Integration. Το Android Studio διαθέτει πλήρη υποστήριξη για το Flutter μέσω του επίσημου plugin της Google. Αυτό επιτρέπει την παρακολούθηση της εφαρμογής σε πραγματικό χρόνο και την εκτέλεση hot reload. Συμβατική είναι και η ανάλυση widget tree και η χρήση Flutter inspector, κάτι που είναι κρίσιμο κατά τον οπτικό και λειτουργικό έλεγχο της εφαρμογής.

Παρότι το Visual Studio Code αποτέλεσε το κύριο περιβάλλον ανάπτυξης του κώδικα λόγω της ευελιξίας του, το Android Studio χρησιμοποιήθηκε εκτενώς για προχωρημένες εργασίες διαγνωστικής, testing και προετοιμασίας builds.

Η χρήση του Android Studio προσφέρει σημαντικά πλεονεκτήματα όπως, επίσημη υποστήριξη από την Google και πλήρως ολοκληρωμένο περιβάλλον για native και cross-platform Android ανάπτυξη. Επίσης, διαθέτει πλούσια τεκμηρίωση, συνεχή ενημέρωση και μεγάλη κοινότητα υποστήριξης. Προσφέρει πλήρη συμβατότητα με Firebase, και Material Design, καθώς και ευκολία στη διαχείριση φυσικών και εικονικών συσκευών.

Το Android Studio αποτελεί αναπόσπαστο εργαλείο στην ανάπτυξη Android εφαρμογών και παραμένει το πρότυπο IDE για επαγγελματίες στον χώρο. Στην παρούσα εργασία, η χρήση του συνδυάστηκε ιδανικά με τα εργαλεία του Flutter και τα debugging χαρακτηριστικά του Android, συμβάλλοντας καταλυτικά στην σταθερότητα και την επαγγελματική δοκιμή της εφαρμογής. Ενσωματώνοντας προηγμένα εργαλεία testing, debugging και performance profiling, το Android Studio ανέλαβε τον ρόλο της αξιολόγησης και δοκιμής του λογισμικού, πριν από τη διάθεσή του σε εξωτερικούς χρήστες.

4.7 Μοντελοποίηση Συστήματος

4.7.1 Διαγράμματα UML

Η σωστή σχεδίαση και τεκμηρίωση ενός λογισμικού συστήματος αποτελεί βασικό πυλώνα για την επιτυχημένη υλοποίησή του. Σε έργα που εμπλέκουν πολλαπλά υποσυστήματα, διαφορετικές τεχνολογίες και ομάδες ανάπτυξης, η ανάγκη για σαφή, τυποποιημένη επικοινωνία γίνεται επιτακτική. Σε αυτό το πλαίσιο, η γλώσσα UML (Unified Modeling Language) αποτελεί ένα καθολικά αποδεκτό εργαλείο για την οπτική αναπαράσταση και μοντελοποίηση λογισμικών συστημάτων.

Η χρήση UML διαγραμμάτων επιτρέπει στους αναλυτές, σχεδιαστές, προγραμματιστές και μετόχους ενός έργου να έχουν κοινή αντίληψη για τη λειτουργικότητα, τη ροή, τις σχέσεις, και τις αλληλεπιδράσεις των επιμέρους στοιχείων ενός πληροφοριακού συστήματος. Μέσα από διάφορους τύπους διαγραμμάτων όπως use case, activity, sequence, component κ.ά. μπορούμε να απεικονίσουμε την πολυπλοκότητα ενός συστήματος σε πολλαπλά επίπεδα. Αυτό, συμβάλλει στον καθαρό σχεδιασμό, τη συντήρηση και τη μελλοντική επέκταση του συστήματος.

Η UML προέκυψε από την ανάγκη ενοποίησης των διαφορετικών μεθοδολογιών ανάλυσης και σχεδιασμού που κυριαρχούσαν στις δεκαετίες του 1980 και 1990. Εκείνη την εποχή κυκλοφορούσαν πολλές εναλλακτικές για την αναπαράσταση αντικειμενοστραφών συστημάτων, όπως οι μέθοδοι Booch, OMT (Object Modeling Technique) του Rumbaugh και OOSE (Object-Oriented Software Engineering) του Jacobson. Το 1994, οι τρεις κορυφαίοι εκπρόσωποι αυτών των μεθόδων Grady Booch, James Rumbaugh και Ivar Jacobson συνεργάστηκαν στην εταιρεία Rational Software, με στόχο τη δημιουργία μιας ενοποιημένης γλώσσας μοντελοποίησης. Το 1997, η Object Management Group (OMG) αναγνώρισε επισήμως την UML έκδοση 1.1 ως πρότυπο για την οπτική μοντελοποίηση αντικειμενοστραφών συστημάτων [37].

Έκτοτε, η UML εξελίχθηκε και καθιερώθηκε ως η πιο διαδεδομένη και καθολικά αποδεκτή γλώσσα μοντελοποίησης, ιδίως στον χώρο της ανάπτυξης λογισμικού, της ανάλυσης απαιτήσεων, και της σχεδίασης αρχιτεκτονικής.

Η UML είναι μια τυποποιημένη γραφική γλώσσα μοντελοποίησης που χρησιμοποιείται για την αναπαράσταση της δομής, συμπεριφοράς, και αλληλεπίδρασης των στοιχείων ενός πληροφοριακού συστήματος. Δεν είναι γλώσσα προγραμματισμού, αλλά ένα εργαλείο σχεδιασμού και τεκμηρίωσης, ανεξάρτητο από γλώσσες υλοποίησης (όπως Java, Python ή Dart). Χρησιμοποιείται εκτενώς σε όλα τα στάδια του κύκλου ζωής του λογισμικού όπως στην κατανόηση και ανάλυση απαιτήσεων, την σχεδίαση αρχιτεκτονικής και ροής δεδομένων και την επικοινωνία μεταξύ μελών της ομάδας ανάπτυξης. Επιπλέον, χρησιμοποιείται και για την τεκμηρίωση για μελλοντική συντήρηση ή επέκταση, καθώς και για την υποστήριξη της υλοποίησης μέσω αυτόματης δημιουργίας κώδικα ή reverse engineering.

Μερικά από τα πλεονεκτήματα της UML είναι η τυποποιημένη αναπαράσταση, αφού προσφέρει έναν κοινό κώδικα επικοινωνίας μεταξύ διαφορετικών ρόλων. Η πολυεπίπεδη ανάλυση παρέχει εργαλεία τόσο για υψηλού επιπέδου σχεδιασμό (use case, component) όσο και για λεπτομερή ροή ελέγχου και δεδομένων (sequence, activity, class). Τέλος, η υποστήριξη αντικειμενοστραφούς προσέγγισης αντιπροσωπεύει πλήρως την εννοιολογική και λογική δομή των αντικειμένων του συστήματος.

Στην παρούσα εργασία, η UML χρησιμοποιήθηκε για τη μοντελοποίηση της αρχιτεκτονικής, της συμπεριφοράς των χρηστών, των σχέσεων μεταξύ υποσυστημάτων, και της ροής δεδομένων, αξιοποιώντας την PlantUML για την παραγωγή καθαρών και διαλειτουργικών σχημάτων.

4.7.2 Σχεδίαση με PlantUML

Για την παραγωγή των UML διαγραμμάτων που συνοδεύουν τη μελέτη και την τεκμηρίωση του συστήματος, αξιοποιήθηκε το εργαλείο PlantUML, ένα σύγχρονο και εξαιρετικά ευέλικτο εργαλείο δημιουργίας διαγραμμάτων μέσω γραπτής περιγραφής σε απλή γλώσσα. Η χρήση του PlantUML επιλέχθηκε για λόγους αυτοματισμού, ευκολίας συντήρησης, συνεργασίας σε περιβάλλοντα version control (όπως GitHub) και συμβατότητας με σύγχρονα IDEs και τεκμηριωτικά συστήματα.

Το PlantUML είναι ένα εργαλείο ανοιχτού κώδικα (open-source), που επιτρέπει τη δημιουργία UML διαγραμμάτων με τη χρήση απλής συντακτικής γλώσσας βασισμένης σε κείμενο. Αντί να σχεδιάζονται τα διαγράμματα χειροκίνητα μέσω GUI, ο χρήστης γράφει ένα σενάριο σε μορφή plain text (με την κατάληξη .puml), και το εργαλείο το μετατρέπει αυτόματα σε οπτικό διάγραμμα (σε μορφές όπως PNG, SVG ή PDF) [38]. Η φιλοσοφία του βασίζεται στην αρχή ότι η πληροφορία ως κώδικας είναι πιο εύκολα επαναχρησιμοποιήσιμη, μεταφέρσιμη και παρακολουθήσιμη από ένα παραδοσιακό αρχείο εικόνας.

Η επιλογή του PlantUML έγινε βάσει συγκεκριμένων πλεονεκτημάτων που το καθιστούν ιδιαίτερα χρήσιμο για την ανάπτυξη και τεκμηρίωση συστημάτων. Αρχικά, η σύνταξή του είναι απλή και εύκολη στην εκμάθηση, ενώ ταυτόχρονα προσφέρει σαφή αναπαράσταση σχέσεων και ροών. Ένα ακόμη σημαντικό πλεονέκτημα είναι η ευκολία τροποποίησης. Η αλλαγή σε ένα διάγραμμα δεν απαιτεί χειροκίνητη παρέμβαση στο γραφικό περιβάλλον αλλά γίνεται απλώς με την αλλαγή του αντίστοιχου κειμένου. Παράλληλα, τα διαγράμματα αποθηκεύονται ως αρχεία κειμένου, γεγονός που επιτρέπει την εύκολη ενσωμάτωσή τους σε pipelines ανάπτυξης και την παρακολούθησή τους σε git repositories, εξασφαλίζοντας έτσι πλήρη υποστήριξη σε διαδικασίες CI/CD. Επιπλέον, το PlantUML είναι συμβατό με πολλά εργαλεία, όπως τα IDEs VS Code και IntelliJ, αλλά και με πλατφόρμες όπως Docusaurus,

MkDocs και LaTeX. Τέλος, προσφέρει εκτενή υποστήριξη σε όλους τους βασικούς τύπους UML διαγραμμάτων, όπως use case, class, component, activity, sequence, deployment και ER diagrams, ενώ υποστηρίζει και διαγράμματα πέρα από την UML, όπως Gantt, wireframes και JSON viewers.

Κατά τη διάρκεια της ανάπτυξης του πληροφοριακού συστήματος, χρησιμοποιήθηκαν αρχεία .puml για την παραγωγή των διαγραμμάτων UML. Αυτά τα αρχεία ενσωματώθηκαν στην τεκμηρίωση και βρίσκονται παράλληλα με τον πηγαίο κώδικα. Όλες οι αλλαγές στην αρχιτεκτονική ή στις ροές της εφαρμογής φαίνονται άμεσα στα διαγράμματα, επιτυγχάνοντας συνέπεια μεταξύ κώδικα και τεκμηρίωσης.

5. Σχεδίαση Λύσης και Υποσυστημάτων

Η επιτυχία μιας εφαρμογής εξαρτάται σε μεγάλο βαθμό από τη σωστή αρχιτεκτονική και τη διαχείριση των επιμέρους υποσυστημάτων της. Στο παρόν κεφάλαιο περιγράφεται αναλυτικά η αρχιτεκτονική του συστήματος. Επιπλέον, περιγράφεται και η διάσπαση σε επιμέρους λειτουργικά υποσυστήματα, καθένα από τα οποία αναλαμβάνει έναν κρίσιμο ρόλο στη συνολική εμπειρία του χρήστη και στη σωστή λειτουργία του συστήματος.

5.1 Η Αρχιτεκτονική Client-Server

Η αρχιτεκτονική Client-Server αποτελεί μία από τις πλέον διαδεδομένες αρχιτεκτονικές στον χώρο της ανάπτυξης λογισμικού, ειδικά σε εφαρμογές κινητών τηλεφώνων και διαδικτυακές εφαρμογές. Στην περίπτωση μας, η συγκεκριμένη προσέγγιση επιλέχθηκε λόγω της ευελιξίας, της επεκτασιμότητας και της διαχώρισης των ευθυνών μεταξύ του client και του server. Η εφαρμογή είναι χτισμένη με το Flutter στο client side και αξιοποιεί μία σειρά από backend μηχανισμούς όπως το Firebase για την αυθεντικοποίηση και την αποθήκευση δεδομένων, ενώ το Python backend (μέσω FastAPI) χρησιμοποιείται αποκλειστικά για την υποστήριξη του chatbot.

Ο client, δηλαδή η εφαρμογή Flutter που τρέχει στη συσκευή του χρήστη, είναι υπεύθυνος για την απόδοση του γραφικού περιβάλλοντος, την αλληλεπίδραση με τον χρήστη. Επίσης, ευθύνεται και για την αποστολή και λήψη δεδομένων, καθώς και για την τοπική διαχείριση της κατάστασης μέσω Provider. Οι λειτουργίες αυτές περιλαμβάνουν προβολή επιτευγμάτων, υπολογισμό ημερών χωρίς κάπνισμα, εμφάνιση ειδοποιήσεων, προσαρμογή περιεχομένου με βάση το ερωτηματολόγιο και πολλά άλλα.

Στην πλευρά του server, η εφαρμογή βασίζεται σε δύο στοιχεία, το Firebase και το Python backend. Το Firebase χρησιμοποιείται για την αυθεντικοποίηση χρηστών (μέσω Google ή email/password), την αποθήκευση και λήψη δεδομένων χρηστών σε πραγματικό χρόνο (Cloud Firestore), καθώς και για τον συγχρονισμό μεταξύ πολλών χρηστών (όπως στην περίπτωση του leaderboard). Από την άλλη πλευρά, το backend σε Python, παρέχει τις απαραίτητες διεπαφές API για την αλληλεπίδραση με το Langchain που εξυπηρετεί το chatbot. Η επικοινωνία μεταξύ της εφαρμογής Flutter και του Python backend γίνεται μέσω HTTP requests, χρησιμοποιώντας Ngrok για την έκθεση του τοπικού server κατά τη διάρκεια της ανάπτυξης.

Ένα βασικό πλεονέκτημα αυτής της αρχιτεκτονικής είναι η διαχωρισμένη ευθύνη. Η λογική του interface (UI) και η διαχείριση κατάστασης (State Management) βρίσκονται αποκλειστικά στον client, ενώ ο server αναλαμβάνει υπολογιστικά πιο απαιτητικές ή ευαίσθητες διεργασίες όπως η εκτέλεση LLM μοντέλων και η αποθήκευση δεδομένων. Αυτή η προσέγγιση διευκολύνει την επεκτασιμότητα, καθώς μπορούν να γίνουν αλλαγές στη μία πλευρά χωρίς να επηρεαστεί απαραίτητα η άλλη, π.χ. αντικατάσταση του backend του chatbot ή εισαγωγή νέου μοντέλου LLM στο server.

Ιδιαίτερη σημασία δόθηκε επίσης και στην ασφάλεια των δεδομένων μέσω των μηχανισμών του Firebase (π.χ. Firestore Rules), στην απόδοση μέσω caching και τοπικής αποθήκευσης στο client. Επιπλέον, η χρήση διαχείριση κατάστασης στο Flutter επιτρέπει την αποδοτική

ενημέρωση του UI, διασφαλίζοντας ομαλή εμπειρία χρήσης ακόμη και με συχνές αλλαγές κατάστασης (π.χ. νέα επιτεύγματα, ενημερώσεις από chatbot, αλλαγές στο leaderboard).

Η επιλογή της συγκεκριμένης αρχιτεκτονικής client server στο σύστημα μας, εξασφαλίζει τη διατήρηση μιας καθαρής δομής, την ευκολία στη συντήρηση και εξέλιξη. Επίσης, υποστηρίζει και την ικανότητα ενσωμάτωσης σύγχρονων τεχνολογιών όπως τα LLM APIs και την cloud-based προσωποποίηση των δεδομένων.

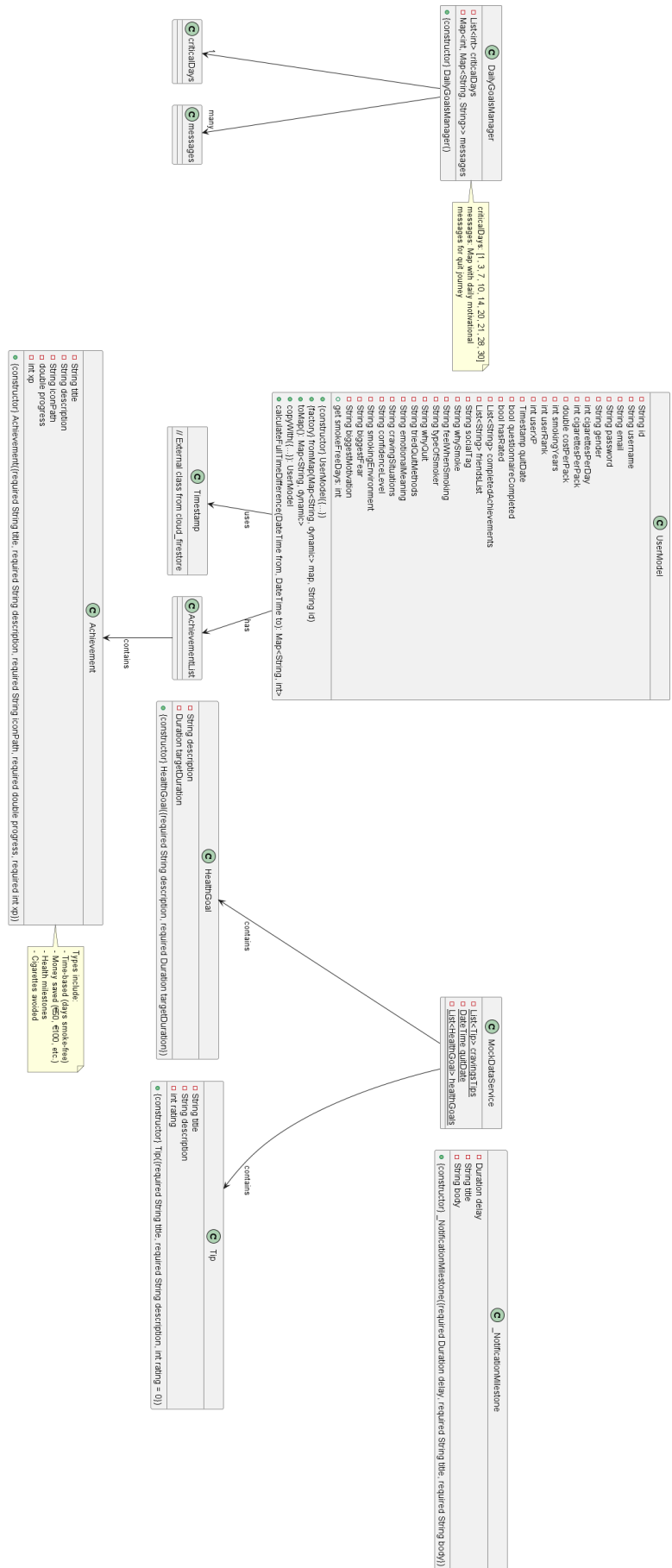
Σε επίπεδο εφαρμογής, ο κώδικας αποτελείται από διάφορες κλάσεις που αναλαμβάνουν ξεχωριστούς ρόλους. Η κλάση MockDataService για παράδειγμα περιλαμβάνει δεδομένα που είναι σταθερά και χρησιμεύουν για σκοπούς δοκιμής και για να εμφανίζονται άμεσα στον χρήστη χωρίς σύνδεση στο διαδίκτυο. Σε αυτή ορίζονται στατικές λίστες με συμβουλές, μία λίστα με στόχους υγείας με συγκεκριμένη χρονική διάρκεια κτλ. Οι στόχοι υγείας αναπαρίστανται από την κλάση HealthGoal, η οποία αποθηκεύει την περιγραφή του στόχου και τη χρονική διάρκεια που απαιτείται για να επιτευχθεί. Αντίστοιχα, η κλάση Tip αντιπροσωπεύει μια απλή συμβουλή με τίτλο, περιγραφή και μία πιθανή αξιολόγηση.

Η σημαντικότερη κλάση της εφαρμογής είναι η UserModel, καθώς σε αυτήν αποτυπώνεται όλο το προφίλ του χρήστη. Η κλάση αυτή περιλαμβάνει προσωπικά στοιχεία όπως όνομα, email και φύλο, στοιχεία που σχετίζονται με το κάπνισμα, όπως ο αριθμός τσιγάρων που καπνίζει ημερησίως, το κόστος του πακέτου και τα χρόνια καπνίσματος, καθώς και πληροφορίες προόδου όπως ο αριθμός πόντων (XP) και το επίπεδο κατάταξης (rank). Επίσης, αποθηκεύει απαντήσεις σε ερωτήσεις που σχετίζονται με την ψυχολογία του χρήστη και τους λόγους που καπνίζει ή θέλει να διακόψει. Περιλαμβάνει και δυναμικές λίστες με φίλους και με επιτεύγματα που έχουν ολοκληρωθεί. Η κλάση παρέχει μεθόδους για εύκολη μετατροπή των δεδομένων σε μορφή χάρτη (toMap) και για κατασκευή ενός αντικειμένου από δεδομένα που έρχονται από το Firestore (fromMap).

Μία ακόμη βασική κλάση είναι η Achievement, που αναπαριστά τα επιτεύγματα της εφαρμογής. Τα επιτεύγματα μπορεί να είναι χρονικά, δηλαδή να βασίζονται στις ημέρες που ο χρήστης παραμένει χωρίς κάπνισμα, οικονομικά, που σχετίζονται με τα χρήματα που εξοικονομεί, επιτεύγματα υγείας ή επιτεύγματα που αφορούν τον αριθμό τσιγάρων που έχει αποφύγει. Κάθε επίτευγμα περιέχει περιγραφή, εικονίδιο, πόντους XP και την πρόοδο προς την ολοκλήρωσή του. Επιπλέον, υπάρχει και η κλάση DailyGoalsManager, που κρατά συγκεκριμένες κρίσιμες ημέρες και τα μηνύματα ενθάρρυνσης που εμφανίζονται στον χρήστη. Παράλληλα, η κλάση _NotificationMilestone μοντελοποιεί τα επιτεύγματα για τις ειδοποιήσεις που ενεργοποιούνται μετά από συγκεκριμένο χρονικό διάστημα.

Τα class diagrams (διαγράμματα κλάσεων) είναι από τα πιο θεμελιώδη και χρήσιμα διαγράμματα στη γλώσσα UML. Περιγράφουν τη δομή των αντικειμένων (objects) που αποτελείται το σύστημα, τις σχέσεις μεταξύ τους, τα χαρακτηριστικά (attributes) και τις μεθόδους (methods) που διαθέτουν. Η λειτουργία τους είναι σημαντική στη φάση της σχεδίασης, διότι δίνουν μια απεικόνιση του μοντέλου δεδομένων και της αρχιτεκτονικής του συστήματος.

Στην περίπτωση μας, το διάγραμμα αυτό αναπαριστά τα βασικά μοντέλα που χρησιμοποιούνται από το frontend (Flutter/Provider), το backend (Python Langchain API), αλλά και για τη διαχείριση δεδομένων στον Firestore.



Εικόνα 10: Class Diagram

Οι σχέσεις μεταξύ αυτών των κλάσεων περιλαμβάνουν:

- **Συσχέτιση (Association):** Π.χ. κάθε User έχει μια λίστα από Achievement.
- **Κληρονομικότητα (Inheritance)** αν υπάρχουν εξειδικευμένες παραλλαγές κάποιων κλάσεων.
- **Συνάθροιση (Aggregation) ή Σύνθεση (Composition),** ανάλογα με το αν τα αντικείμενα υπάρχουν ανεξάρτητα από τον κάτοχό τους ή όχι.

Τα διαγράμματα κλάσεων λειτουργούν ως γέφυρα μεταξύ σχεδίασης και προγραμματισμού. Με αυτά επιτρέπεται η οπτικοποίηση της δομής των δεδομένων πριν από την υλοποίηση του κώδικα. Παράλληλα, διευκολύνουν τον καλύτερο διαχωρισμό ευθυνών ανά κλάση. Επιπλέον, κάνουν πιο εύκολη την επαναχρησιμοποίηση κώδικα μέσω καλά σχεδιασμένων μοντέλων και την τεκμηρίωση του συστήματος για μελλοντική συντήρηση και επέκταση.

Στην παρούσα εφαρμογή, οι κλάσεις αυτές μεταφράζονται είτε σε Dart μοντέλα στο Flutter (UserModel, AchievementModel κ.λπ.), είτε σε JSON schemas που χρησιμοποιούνται για την αποθήκευση/ανάκτηση από Firestore, είτε σε Python δομές δεδομένων στο backend για την επικοινωνία με το LLM.

Τα component diagrams (διαγράμματα συστατικών) χρησιμοποιούνται για να αποδώσουν την λογική οργάνωση των συστατικών λογισμικού σε ένα σύστημα και εστιάζουν στο πώς είναι οργανωμένα τα πακέτα, υποσυστήματα και τα στοιχεία που υλοποιούν τις λειτουργίες της εφαρμογής.

Στο παρόν σύστημα, το component diagram δείχνει τον τρόπο με τον οποίο τα εσωτερικά μέρη της εφαρμογής (frontend, backend, services) αλληλοεπιδρούν μεταξύ τους και πώς διασυνδέονται με εξωτερικά APIs και υπηρεσίες τρίτων.

Ο client περιλαμβάνει όλα τα λειτουργικά κομμάτια που τρέχουν στην κινητή συσκευή. Το UI layer αποτελείται από τις οθόνες της εφαρμογής (π.χ. Welcome, Login-Signup). Οι οθόνες αυτές δεν έχουν άμεση πρόσβαση στα δεδομένα αλλά επικοινωνούν αποκλειστικά μέσω των Providers.

Οι Providers διαχειρίζονται την κατάσταση της εφαρμογής και αποτελούν τη γέφυρα μεταξύ του UI και των υπηρεσιών δεδομένων. Ο User Provider διαχειρίζεται τα στοιχεία του χρήστη, το προφίλ, το quit date και τα στατιστικά. Ο Auth Provider χειρίζεται το login και logout καθώς και την κατάσταση authentication. Οι ειδικοί providers για Leaderboard, Achievements και Daily Tips φροντίζουν τα δεδομένα που προέρχονται από το Firestore. Οι Home Screen και Progress Providers υποστηρίζουν την παρακολούθηση της προόδου στην αρχική οθόνη και την υγεία του χρήστη. Ο Notification Provider διαχειρίζεται τις push notifications και ο Chatbot Provider αναλαμβάνει την επικοινωνία με το API του chatbot.

Οι Local Services αποτελούν τις βοηθητικές υπηρεσίες που υλοποιούνται στην εφαρμογή και είναι υπεύθυνες για την επικοινωνία με το backend. Το Auth Service χειρίζεται το Firebase Authentication και το Firestore Service διαχειρίζεται την ανάγνωση και εγγραφή δεδομένων από και προς το Firestore. Ο Notification Manager χειρίζεται το Firebase Cloud Messaging. Επίσης, το Chatbot API Service στέλνει αιτήματα στο backend του chatbot. Τέλος, το Mock Data Service παρέχει εικονικά δεδομένα για testing ή development.

Στην πλευρά του backend, οι υπηρεσίες χωρίζονται σε Firebase Services και το σύστημα chatbot. Το Firebase Authentication χειρίζεται τη διαδικασία login, εγγραφών και Google

OAuth, το Firestore λειτουργεί ως βάση δεδομένων, ενώ το Cloud Messaging παρέχει push notifications. Το σύστημα chatbot περιλαμβάνει το Chatbot API Service που δέχεται αιτήματα από την εφαρμογή, τον LangChain Orchestrator που συντονίζει την επεξεργασία των ερωτήσεων του χρήστη, το Retriever Module που αναζητά σχετικά δεδομένα από τη βάση γνώσης, τη Vector Database για αποθήκευση κειμένων σε μορφή embeddings για semantic search και το LLM Model Endpoint (π.χ. OpenAI ή Groq) που δημιουργεί τις απαντήσεις.

Οι Providers επικοινωνούν με τις Local Services, οι οποίες με τη σειρά τους επικοινωνούν με το backend. Το UI μιλά μόνο με τους Providers και ποτέ απευθείας με τις υπηρεσίες. Με αυτόν τον τρόπο, η αρχιτεκτονική παραμένει καθαρή και modular. Το UI να είναι ελαφρύ και ανεξάρτητο από τις λεπτομέρειες υλοποίησης. Οι Providers διαχειρίζονται όλη τη λογική δεδομένων και τις Local Services χειρίζονται την πραγματική σύνδεση με το backend. Αυτή η δομή επιτρέπει εύκολη επέκταση της εφαρμογής και προσθήκη νέων λειτουργιών, χωρίς να επηρεάζονται τα υπόλοιπα components.

Το σύστημα έχει ένα UI layer που ενσωματώνει το chatbot. Ο χρήστης αλληλοεπιδρά με το Chat Screen, όπου στέλνει μηνύματα και λαμβάνει απαντήσεις. Ο ρόλος του HTTP Client είναι να συλλέγει το μήνυμα από τον χρήστη και να το προωθεί στο backend μέσω ασφαλούς HTTPS αιτήματος, ενώ ταυτόχρονα εμφανίζει την απάντηση στο UI.

Για να μπορεί η εφαρμογή να επικοινωνήσει με το backend, χρησιμοποιείται το ngrok ως ενδιάμεσος δημόσιος HTTPS endpoint και στείλει αιτήματα στο API Gateway του FastAPI. Το FastAPI λειτουργεί ως το πρώτο σημείο επαφής για τα εισερχόμενα αιτήματα. Το API Gateway δέχεται τα μηνύματα, ενώ ο Request Validator ελέγχει ότι το μήνυμα είναι σωστά δομημένο και περιέχει όλα τα απαραίτητα στοιχεία, όπως user ID και session token. Ο Session Manager διαχειρίζεται τις συνεδρίες των χρηστών, φορτώνοντας ή ενημερώνοντας το context από το User Memory. Με αυτόν τον τρόπο, το FastAPI οργανώνει τη ροή των δεδομένων πριν αυτά περάσουν στο κύριο σύστημα επεξεργασίας του chatbot.

Ο LangChain Orchestrator συνδυάζει διάφορα components για την παραγωγή της απάντησης. Ο Prompt Builder δημιουργεί το prompt που θα σταλεί στο LLM, ενσωματώνοντας δεδομένα χρήστη, έγγραφα από το Vector Store και το ιστορικό της συνομιλίας. Ο Conversation Manager κρατάει το context και διαχειρίζεται την αλληλουχία των μηνυμάτων. Το RAG Pipeline συνδυάζει τις ερωτήσεις του χρήστη με σχετικά κείμενα από το Vector Store, ώστε το LLM να απαντά με βάση γνώση και όχι μόνο γλωσσικά μοντέλα. Ο Fallback Handler ενεργοποιείται όταν δεν βρεθούν σχετικά αποτελέσματα, πραγματοποιώντας web search ή επιστρέφοντας μια προκαθορισμένη απάντηση. Με αυτόν τον τρόπο, ο Orchestrator εξασφαλίζει ότι η απάντηση είναι στοχευμένη, σωστή και εξατομικευμένη.

Ο Semantic Classifier ελέγχει την πρόθεση του μηνύματος και αποφασίζει αν θα περάσει στη διαδικασία RAG ή θα γίνει fallback. Το Vector Store (ChromaDB) αποθηκεύει κείμενα όπως συμβουλές και οδηγίες υγείας σε μορφή embeddings που επιτρέπουν semantic search. Το User Memory κρατά το ιστορικό της συνομιλίας, ενώ το Firebase περιέχει το προφίλ του χρήστη, τις συνήθειες καπνίσματος και την πρόοδό του, ώστε να μπορούν να προστεθούν εξατομικευμένα δεδομένα στο prompt. Εξωτερικές υπηρεσίες περιλαμβάνουν τον LLM Provider, όπως OpenRouter ή Groq, που δημιουργεί τις απαντήσεις, και το Web Search API (π.χ. Serper) για αναζήτηση πληροφοριών στο διαδίκτυο όταν τα δεδομένα του Vector Store δεν επαρκούν.

Το component diagram, βοηθά στον ορισμό των ορίων κάθε μονάδας λογισμικού και την κατανόηση των αλληλεξαρτήσεων μεταξύ modules. Επίσης, επιτρέπει τον διαχωρισμό μικροϋπηρεσιών (microservices). Τα διαγράμματα αυτά απεικονίζουν καθαρά τα συστατικά του συστήματος και τις μεταξύ τους αλληλεπιδράσεις.

5.2 Δεδομένα του Συστήματος

Το σύστημα στηρίζεται σε μία αρχιτεκτονική που συνδυάζει Flutter με Firebase Firestore για την αποθήκευση και διαχείριση δεδομένων. Η αρχιτεκτονική αυτή μπορεί να θεωρηθεί υβριδική, καθώς μέρος των δεδομένων είναι ενσωματωμένο μέσα στον κώδικα (hardcoded), ενώ τα πιο δυναμικά δεδομένα που σχετίζονται με τους χρήστες, τις ενέργειές τους και τις αλληλεπιδράσεις τους αποθηκεύονται σε συλλογές του Firestore. Με αυτόν τον τρόπο το σύστημα προσφέρει ταχύτητα της άμεσης χρήσης στατικών δεδομένων και ευελιξία που προσφέρει το cloud database της Google.

Το Firestore χρησιμοποιείται για την αποθήκευση όλων των κρίσιμων στοιχείων που αφορούν τον χρήστη και τη χρήση της εφαρμογής. Η βασική συλλογή είναι η users, όπου κάθε έγγραφο αντιστοιχεί σε έναν χρήστη και περιλαμβάνει όλα τα προσωπικά του στοιχεία. Τέτοιου είδους δεδομένα είναι τα στοιχεία που αφορούν το κάπνισμα, την πρόοδό του, αλλά και τις απαντήσεις στο αρχικό ερωτηματολόγιο. Μέσα σε αυτό το έγγραφο αποθηκεύονται επίσης πίνακες (arrays), όπως η λίστα με τους φίλους και τα ολοκληρωμένα επιτεύγματα.

Εκτός από τον πίνακα χρηστών, υπάρχουν και άλλες συλλογές. Η craving_logs αποθηκεύει τις καταγραφές του χρήστη. Αυτές περιλαμβάνουν την ένταση, το ερέθισμα που την προκάλεσε, ο τρόπος που την αντιμετώπισε, μια σύντομη περιγραφή, η χρονική στιγμή που συνέβη και το userId του χρήστη που την καταχώρησε. Η συλλογή global_chat_rooms υλοποιεί τις ομαδικές συνομιλίες, με υποσυλλογές για τα μηνύματα και τα μέλη. Έτσι, οι χρήστες μπορούν να ανταλλάσσουν μηνύματα σε πραγματικό χρόνο. Η συλλογή ratings επιτρέπει στους χρήστες να αξιολογούν την εφαρμογή, καταγράφοντας την εμπειρία τους σε σχέση με τη χρησιμότητα και τη λειτουργικότητα.

Η επιλογή του υβριδικού μοντέλου, όπου κάποια δεδομένα είναι σταθερά και άλλα αποθηκεύονται δυναμικά στο Firestore, επιτρέπει στο σύστημα να είναι και αποδοτικό και επεκτάσιμο. Τα δεδομένα που δεν αλλάζουν, όπως οι βασικές συμβουλές ή τα milestones υγείας, αποθηκεύονται στον κώδικα ώστε να είναι άμεσα διαθέσιμα. Τα προσωπικά στοιχεία του χρήστη και οι ενέργειές του, καταγράφονται στο Firestore. Με αυτόν τον τρόπο, συγχρονίζονται σε πραγματικό χρόνο και είναι διαθέσιμα από οποιαδήποτε συσκευή ο χρήστης συνδέεται.

Το ER διάγραμμα (Entity-Relationship diagram) αποτελεί βασικό εργαλείο στη σχεδίαση βάσεων δεδομένων. Τα διαγράμματα αυτά επιτρέπουν την αφηρημένη αναπαράσταση των δεδομένων ενός συστήματος, των σχέσεων τους και των βασικών περιορισμών. Το ER διάγραμμα παρέχει ένα υψηλού επιπέδου μοντέλο δεδομένων που είναι εύκολο να κατανοηθεί τόσο από τεχνικά όσο και από μη τεχνικά μέλη της ομάδας.

Στο σύστημα μας, παρόλο που χρησιμοποιείται το Firebase Firestore, το οποίο είναι NoSQL και δεν βασίζεται σε παραδοσιακή σχεσιακή λογική, η αποτύπωση ενός ER διαγράμματος

παραμένει εξαιρετικά χρήσιμη. Μας βοηθά να οργανώσουμε τις συλλογές (collections), τα έγγραφα (documents) και τα εμφωλευμένα (nested) objects.

Ένα ER διάγραμμα μοντελοποιεί το πληροφοριακό περιβάλλον ως ένα σύνολο από:

- **Entities:** που αντιστοιχούν σε αντικείμενα ή έννοιες του πραγματικού κόσμου (π.χ., Χρήστης, Επίτευγμα, Συμβουλή).
- **Attributes:** που περιγράφουν τις ιδιότητες των οντοτήτων (π.χ., username, quitDate).
- **Relationships:** που αποτυπώνουν πώς οι οντότητες συνδέονται μεταξύ τους (π.χ., ένας Χρήστης έχει Πολλαπλά Επιτεύγματα).
- **Cardinality:** που καθορίζουν τη φύση των συσχετίσεων (ένα-προς-ένα, ένα-προς-πολλά κ.λπ.).

Αυτή η αποτύπωση βοηθά σημαντικά στη σχεδίαση των Firebase Collections, καθώς και στη διατήρηση μιας καθαρής και αποδοτικής δομής δεδομένων.

5.3 Ανάλυση Υποσυστημάτων

Στην ανάλυση των διαφόρων υποσυστημάτων, έγινε χρήση των activity diagrams (διαγράμματα δραστηριοτήτων) και sequence diagrams (διαγράμματα ακολουθίας). Τα διαγράμματα σχεδιάστηκαν κατά την διάρκεια της μοντελοποίησης του συστήματος, ενώ ταυτόχρονα ενημερώνονταν κάθε φορά που άλλαζε κάτι στην ροή ή την υλοποίηση του εκάστοτε υποσυστήματος.

Τα activity diagrams αποτελούν μία από τις βασικές μορφές UML διαγραμμάτων και χρησιμοποιούνται για την αναπαράσταση της ροής πληροφορίας εντός ενός συστήματος. Εστιάζουν κυρίως στις διαδοχικές ενέργειες και τις αλληλουχίες που εκτελούνται από τους χρήστες και από το ίδιο το σύστημα. Τα διαγράμματα αυτά, απεικονίζουν τις ενέργειες που εκτελούνται σε ένα σύστημα, συνδεδεμένες με βέλη ροής που δείχνουν τη σειρά εκτέλεσής τους.

Στην παρούσα εργασία, τα activity diagrams αξιοποιούνται για να περιγράψουν τις ροές διεπαφής χρήστη. Αυτές οι ροές παρέχουν μια συνολική εικόνα της λειτουργικότητας. Επιπλέον, συνέβαλαν στην κατασκευή των UI/UX flows κατά την ανάπτυξη των οθονών. Η χρήση τους ενίσχυσε σημαντικά τον σχεδιασμό και την κατανόηση της λογικής πίσω από τις κύριες λειτουργίες του συστήματος, ενώ διευκόλυνε τη διασύνδεση μεταξύ των διεπαφών. Επίσης, αποτέλεσε σημαντικό βήμα για τη υλοποίηση των λειτουργικών απαιτήσεων, καθώς κάθε διάγραμμα αντιστοιχεί σε ένα συγκεκριμένο κομμάτι του συστήματος.

Τα sequence diagrams, χρησιμοποιούνται για την απεικόνιση της αλληλεπίδρασης μεταξύ αντικειμένων/οντοτήτων με χρονολογική σειρά. Αυτά, καταγράφουν πώς τα διάφορα μέρη ενός συστήματος συνεργάζονται για να εκτελέσουν μια συγκεκριμένη λειτουργικότητα. Έτσι, απεικονίζουν τη σειρά και τη δομή της επικοινωνίας. Η χρήση τους επιτρέπει προληπτική εντόπιση προβλημάτων (race conditions, λανθασμένη ροή, bottlenecks). Επίσης, ενισχύει τον καταμερισμό ευθυνών ανά σύστημα και υποσύστημα και παρέχει σαφή εικόνα του πότε ενεργοποιούνται οι Providers και πότε γίνεται αλλαγή κατάστασης στο UI.

Ένα Sequence Diagram αποτελείται από τα εξής στοιχεία:

- **Actors** (π.χ. Χρήστης, Mobile App, Backend Server, API Gateway).
- **Lifelines**: Κατακόρυφες γραμμές που αντιστοιχούν στην ύπαρξη της κάθε οντότητας στον χρόνο.
- **Messages**: Οριζόντια βέλη που δείχνουν τις μεθόδους ή αιτήσεις που ανταλλάσσονται μεταξύ των οντοτήτων.
- **Activation boxes**: Ορθογώνια πάνω στα lifelines που δείχνουν χρονικά διαστήματα κατά τα οποία η οντότητα εκτελεί κάποια λειτουργία.
- **Return messages**: Διακεκομμένες γραμμές που δείχνουν την επιστροφή αποτελεσμάτων ή αποκρίσεων.
- **Control structures** (alt, loop, opt): Δομές που επιτρέπουν την αναπαράσταση εναλλακτικών ή επαναλαμβανόμενων ροών.

Η απεικόνιση είναι χρονικά κάθετη και όσο προχωράμε προς τα κάτω στο διάγραμμα, τόσο πιο προχωρημένη είναι η ακολουθία γεγονότων. Στο παρόν σύστημα, τα διαγράμματα αυτά χρησιμοποιήθηκαν για την απεικόνιση των αλληλεπιδράσεων μεταξύ του frontend (Flutter), του backend, των providers και των εξωτερικών APIs. Επιπλέον, συνέβαλαν σημαντικά στην κατανόηση του πότε και πώς ανταλλάσσονται τα δεδομένα, τι είναι υπεύθυνό για κάθε λειτουργία και ποια μέρη απαιτούν βελτιστοποίηση.

5.3.1 Σύστημα Αυθεντικοποίησης

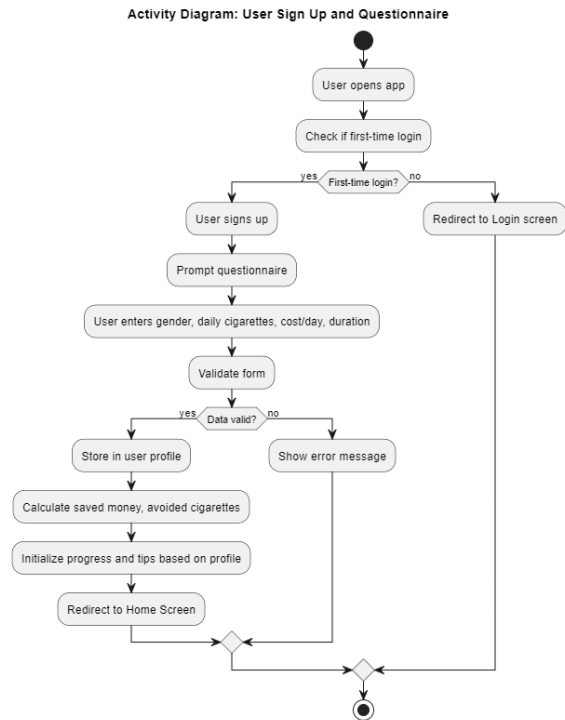
Το υποσύστημα αυθεντικοποίησης διασφαλίζει ότι κάθε χρήστης έχει μοναδική ταυτότητα εντός της πλατφόρμας. Τα προσωπικά του δεδομένα και η πρόσδοξ του παραμένουν ασφαλή και προσβάσιμα μόνο από τον ίδιο. Η αυθεντικοποίηση επιτυγχάνεται μέσω της ενσωμάτωσης του Firebase Authentication, το οποίο παρέχει ένα ασφαλές, αξιόπιστο και ευρέως χρησιμοποιούμενο σύστημα διαχείρισης ταυτοτήτων χρηστών.

Η επιλογή του Firebase Authentication βασίστηκε τόσο στην ευκολία ενσωμάτωσης με εφαρμογές Flutter όσο και στην υποστήριξη πολλών μεθόδων σύνδεσης. Συγκεκριμένα, η εφαρμογή επιτρέπει την σύνδεση με email και κωδικό πρόσβασης. Πρόκειται για τη βασική μέθοδο εγγραφής/σύνδεσης, όπου ο χρήστης δημιουργεί λογαριασμό με τα διαπιστευτήριά του. Η εφαρμογή ελέγχει την εγκυρότητα των στοιχείων αυτών και δημιουργεί session μόλις η ταυτοποίηση ολοκληρωθεί. Επίσης, παρέχεται η δυνατότητα ταχείας εγγραφής/σύνδεσης με χρήση του Google account του χρήστη. Η μέθοδος αυτή μειώνει τα εμπόδια για την αρχική είσοδο, αυξάνοντας την προσβασιμότητα.

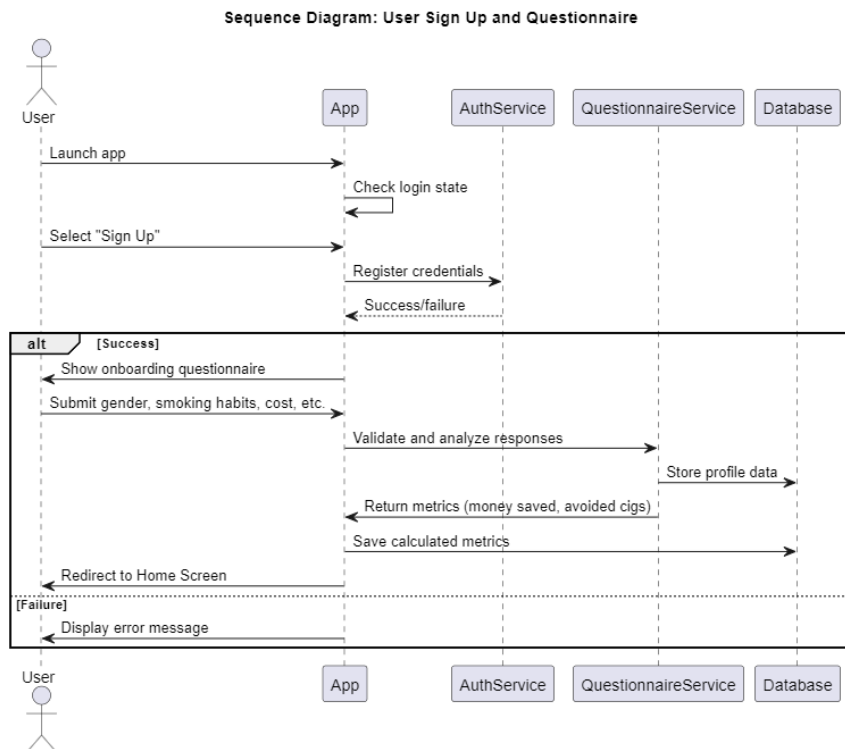
Κατά την αρχική εγγραφή, εκτός από την πιστοποίηση, η εφαρμογή συλλέγει μέσω του ερωτηματολογίου, πληροφορίες όπως ο αριθμός ημερήσιων τσιγάρων, το κόστος πακέτου, η διάρκεια του καπνίσματος και το φύλο. Αυτά τα δεδομένα αποθηκεύονται με ασφάλεια στο Cloud Firestore σε συλλογή users, με μοναδικό ID το UID που παρέχει το Firebase. Αυτό το μοντέλο εγγυάται ότι τα δεδομένα είναι μοναδικά για κάθε χρήστη, συνδεδεμένα απευθείας με τον λογαριασμό του.

Το σύστημα υποστηρίζει και αποθήκευση συνεδρίας (session). Ο χρήστης παραμένει συνδεδεμένος, ακόμη κι αν κλείσει την εφαρμογή, μέσω των μηχανισμών διατήρησης session του Firebase. Προσφέρονται λειτουργίες διαγραφής λογαριασμού και πλήρους αποσύνδεσης, τόσο από την τοπική συσκευή όσο και από τη βάση δεδομένων. Τέλος, το σύστημα προσφέρει ασφάλεια, καθώς η επικοινωνία με το backend γίνεται πάντα μέσω HTTPS. Τα δεδομένα αποθηκεύονται σε Firebase, που εφαρμόζει πολιτικές κανόνων πρόσβασης βάσει UID. Δεν επιτρέπεται η πρόσβαση σε δεδομένα άλλων χρηστών.

Η υλοποίηση του συστήματος αυθεντικοποίησης προσφέρει μια ομαλή και ασφαλή εμπειρία χρήσης. Παράλληλα, δημιουργεί τις προϋποθέσεις για μελλοντικές επεκτάσεις, όπως υποστήριξη για login μέσω Apple ID, Facebook ή ακόμα και βιομετρικών χαρακτηριστικών (όπως fingerprint login μέσω Firebase extensions).



Εικόνα 14: Activity Diagram-User Sign Up and Questionnaire



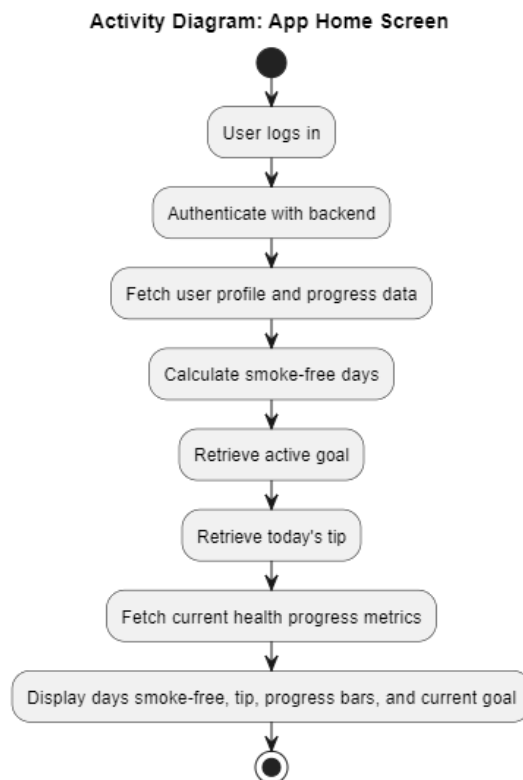
Εικόνα 15: Sequence Diagram-User Sign Up and Questionnaire

5.3.2 Προφίλ και Εξατομίκευση

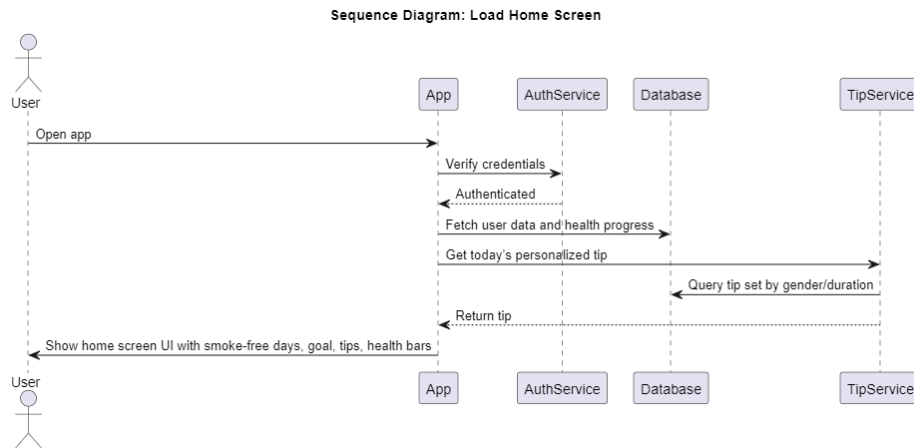
Η λειτουργικότητα προφίλ και εξατομίκευσης στην εφαρμογή παίζει καθοριστικό ρόλο στη δημιουργία μιας εμπειρίας προσαρμοσμένης στις ανάγκες, τα χαρακτηριστικά και την πρόοδο κάθε χρήστη. Η εφαρμογή δεν αντιμετωπίζει τους χρήστες ως ομοιογενή ομάδα, αλλά χρησιμοποιεί τα δεδομένα που συλλέγονται κατά την αρχική είσοδο και κατά τη διάρκεια της χρήσης για να παρέχει σχετικό περιεχόμενο και προσωποποιημένη υποστήριξη.

Κατά την πρώτη είσοδο, ο χρήστης συμπληρώνει ένα ερωτηματολόγιο, το οποίο συλλέγει βασικά χαρακτηριστικά όπως ημερομηνία διακοπής καπνίσματος, ηλικία, φύλο, αριθμός τσιγάρων ημερησίως, κόστος καπνίσματος ανά ημέρα, συνολική διάρκεια καπνίσματος (σε έτη). Ο χρήστης συμπληρώνει και άλλου είδους ερωτήσεις ψυχολογικού χαρακτήρα, για να καταλάβει το σύστημα την ρίζα του εθισμού και να προτείνει πιο στενευμένες λύσεις και απαντήσεις. Μερικές από αυτές τις ερωτήσεις είναι, Ποιος είναι ο λόγος που καπνίζεις; Τι νιώθεις όταν καπνίζεις; Ποια είναι η σχέση σου με το τσιγάρο; Κτλ. Αυτά τα δεδομένα αποθηκεύονται στο Firestore και χρησιμοποιούνται για την εξατομίκευση των υπολοίπων λειτουργιών.

Με βάση τα δεδομένα προφίλ, η εφαρμογή υπολογίζει και παρουσιάζει με δυναμικό τρόπο τις ημέρες χωρίς κάπνισμα, τα χρήματα που εξοικονομήθηκαν, τα τσιγάρα που δεν καταναλώθηκαν, υγειονομικές βελτιώσεις μέσω χρονικών επιτευγμάτων (π.χ., σε 48 ώρες αποκαθίσταται η οσμή και η γεύση). Οι πληροφορίες αυτές εμφανίζονται στην κεντρική οθόνη και στο προφίλ του χρήστη, με ενημέρωση σε πραγματικό χρόνο χάρη στη χρήση των Providers. Ο χρήστης μπορεί να δει το προφίλ του μέσω ενός drawer που εμφανίζεται στην αρχική οθόνη. Το drawer περιλαμβάνει το Username, φύλο, rank, πρόοδο προς το επόμενο επίπεδο (με progress bar), συνολικά XP. Το drawer δημιουργεί μια αίσθηση ταυτότητας μέσα στην εφαρμογή, η οποία ενισχύει τη δέσμευση του χρήστη στη διαδικασία διακοπής.



Εικόνα 16: Activity Diagram-App Home Screen



Εικόνα 17: Sequence Diagram-Load Home Screen

5.3.3 Επιτεύγματα και Υγειονομικοί Στόχοι

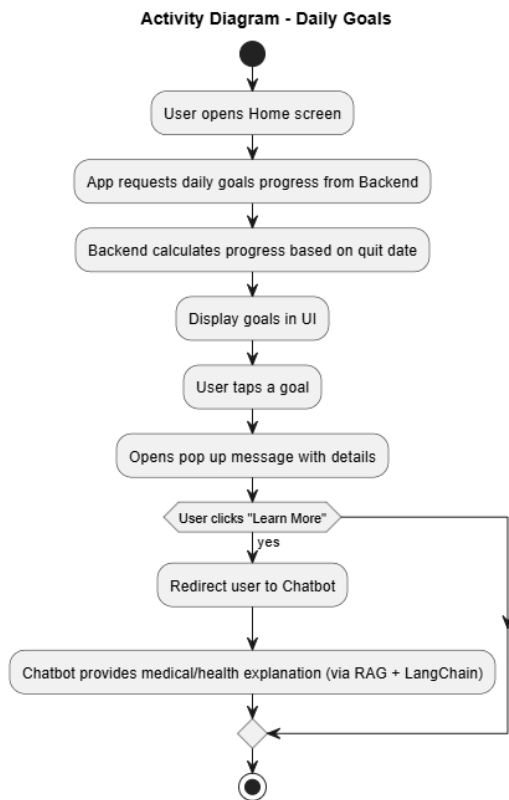
Το σύστημα επιτευγμάτων (Achievements) και εμπειρίας (XP) αποτελεί έναν από τους πιο σημαντικούς μηχανισμούς παροχής κινήτρου της εφαρμογής. Υιοθετεί στοιχεία από τον χώρο του gamification με σκοπό να ενισχύσει την ψυχολογική εμπλοκή, τη δέσμευση και την αίσθηση προόδου του χρήστη, στη δύσκολη προσπάθεια του για διακοπή του καπνίσματος. Η υλοποίηση γίνεται εξ ολοκλήρου στο Flutter με διαχείριση κατάστασης μέσω Provider και δεδομένα που αντλούνται δυναμικά από mock υπηρεσίες ή από τη βάση Firebase.

Η ύπαρξη επιτευγμάτων έχει ως κύριο στόχο την επιβράβευση της συνέπειας (π.χ. 1η ημέρα χωρίς κάπνισμα, 7 ημέρες, 30 ημέρες). Επίσης, ενισχύει την αυτοπεποίθηση μέσω οπτικής αναπαράστασης της προόδου. Το κίνητρο παραμένει μεγάλο μέσα από μετρήσιμους στόχους και την παιχνιδιοποίηση της εμπειρίας, κάνοντας την όλη διαδικασία πιο ευχάριστη και διαδραστική.

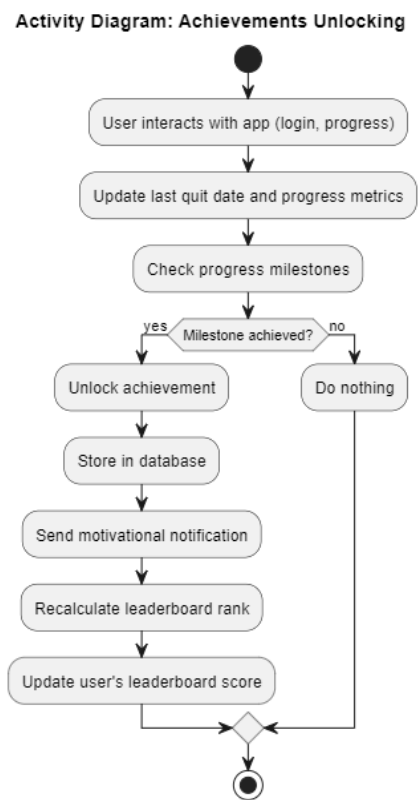
Οι υγειονομικοί στόχοι (Health Goals) έχουν σχεδιαστεί με βάση επιστημονικά τεκμηριωμένες φάσεις υγειονομικής αποκατάστασης μετά τη διακοπή του καπνίσματος. Κάθε στόχος αντιστοιχεί σε συγκεκριμένο χρονικό διάστημα από την ημερομηνία διακοπής και συνοδεύεται από τίτλο και περιγραφή (π.χ. Η καρδιά σου χτυπάει κανονικά μετά από 24 ώρες), κατάσταση ολοκλήρωσης (ολοκληρωμένο ή όχι) και ποσοστό προόδου (σε progress bar).

Η οθόνη προβάλλει όλα τα διαθέσιμα health goals του χρήστη σε μορφή λίστας. Τα ολοκληρωμένα επιτεύγματα εμφανίζονται με την ένδειξη Completed. Κάθε υγειονομικός στόχος είναι διαδραστικός, επιτρέποντας στον χρήστη να δει περισσότερες λεπτομέρειες με απλό πάτημα. Επιπλέον, στην αρχική οθόνη εμφανίζονται πάντα οι τρεις στόχοι που βρίσκονται πιο κοντά στην ολοκλήρωση, δίνοντας στο χρήστη βραχυπρόθεσμους στόχους να εστιάσει.

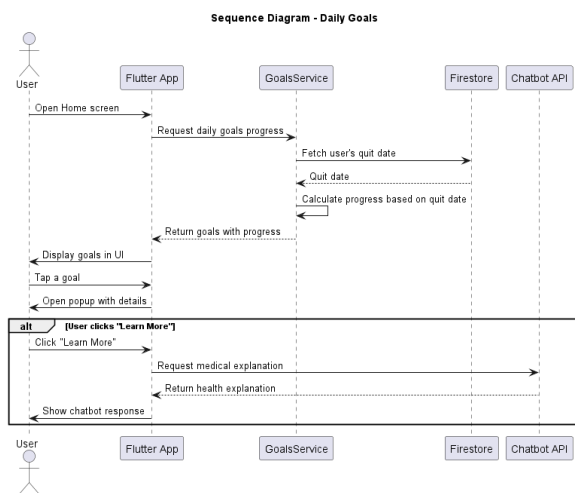
Κάθε επίτευγμα αποδίδει συγκεκριμένο αριθμό πόντων, που αθροίζονται και καθορίζουν τους συνολικούς πόντους (XP) του χρήστη, την κατάταξή του στον πίνακα κατάταξης, καθώς και την πρόοδό του στο επόμενο επίπεδο. Η πρόοδος αυτή παρουσιάζεται στο προφίλ, με progress bar και ένδειξη του πόσοι πόντοι XP απομένουν για να ανέβει επίπεδο.



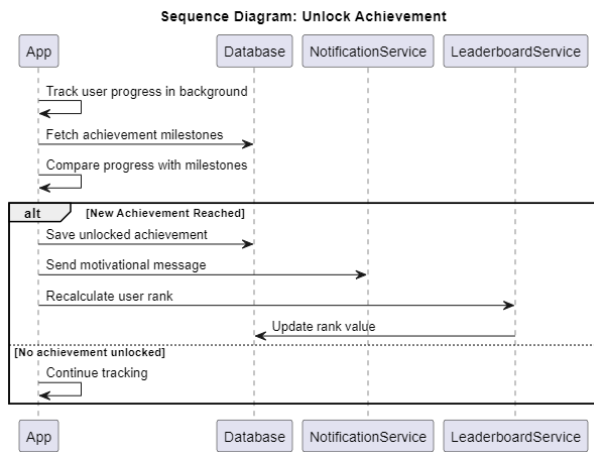
Εικόνα 18: Activity-Diagram Daily Goals



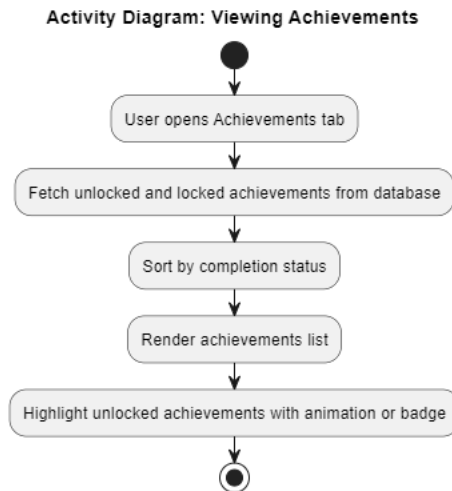
Εικόνα 19: Activity Diagram- Achievements Unlocking



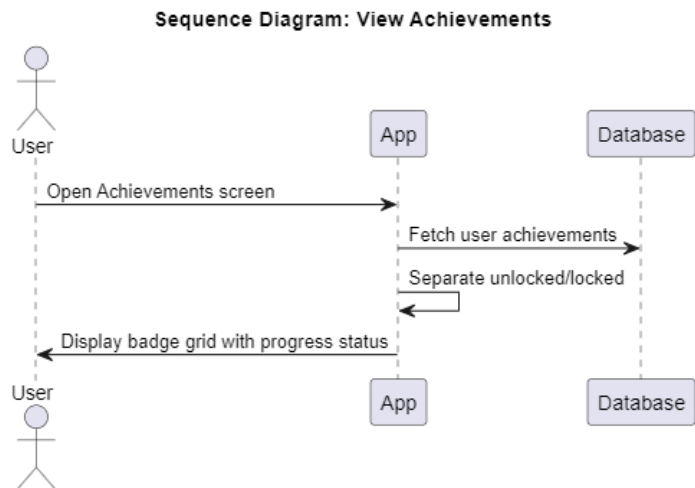
Εικόνα 20: Sequence Diagram-Daily Goals



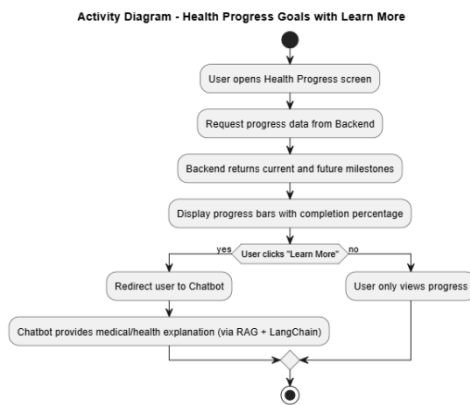
Εικόνα 21: Sequence Diagram-Unlock Achievement



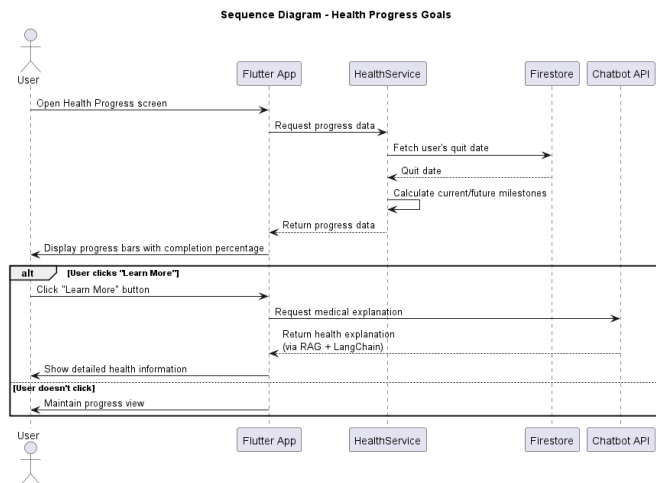
Εικόνα 22: Activity Diagram-View Achievements



Εικόνα 23: Sequence Diagram-View Achievement



Εικόνα 24: Activity Diagram-Health Goals



Εικόνα 25: Sequence Diagram-Health Goals

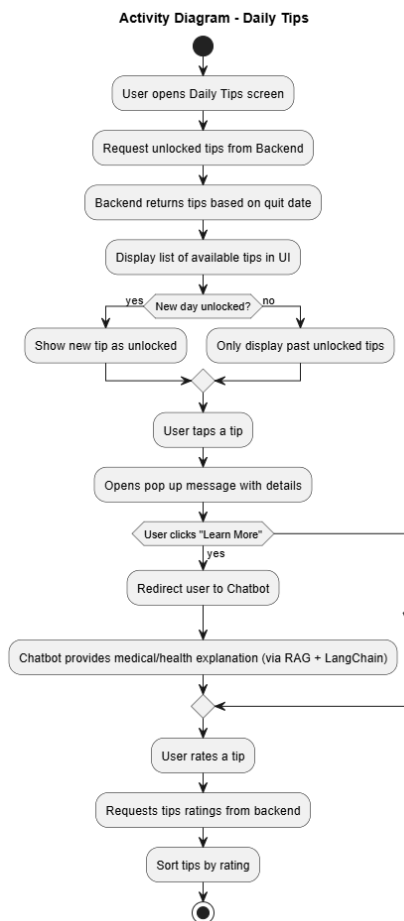
5.3.4 Υπενθυμίσεις και Ημερήσιες Συμβουλές

Οι υπενθυμίσεις και οι ημερήσιες συμβουλές αποτελούν έναν βασικό στοιχείο της εφαρμογής, με στόχο το καθημερινό κίνητρο του χρήστη, την πρόληψη υποτροπών και τη διαρκή υπενθύμιση του σκοπού του. Το υποσύστημα αυτό συνδυάζει λειτουργικά στοιχεία ψυχολογικής ενίσχυσης και καθημερινής αλληλεπίδρασης με την εφαρμογή.

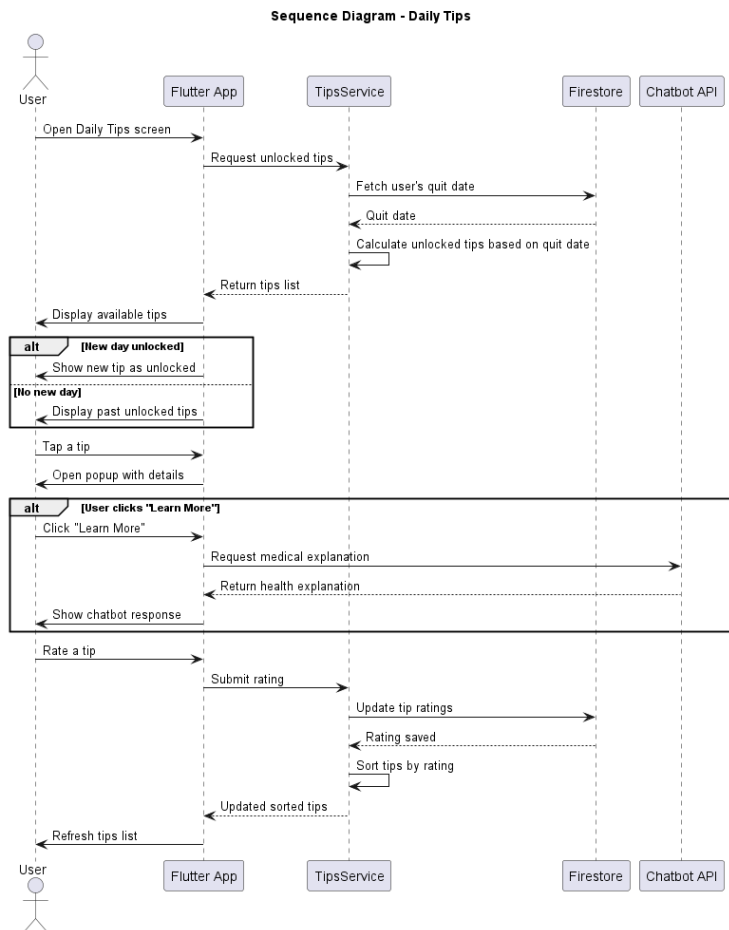
Οι υπενθυμίσεις αξιοποιούν το notification system της πλατφόρμας Android. Υπάρχει δυνατότητα για εξατομικευμένες ειδοποιήσεις ανάλογα με την πρόοδο του χρήστη (π.χ. Συμπληρώθηκαν 7 ημέρες χωρίς τσιγάρο!) και την ώρα της ημέρας (π.χ. κρίσιμες ώρες επιθυμίας για κάπνισμα). Επίσης, υπάρχουν προγραμματισμένες ειδοποιήσεις που συνεχίζουν να λειτουργούν ακόμα και όταν η εφαρμογή είναι κλειστή ή στο παρασκήνιο (background). Άλλες ενεργοποιούνται έπειτα από triggers μέσα από την εφαρμογή (π.χ. μετά την ολοκλήρωση ενός στόχου). Το σύστημα ειδοποιήσεων είναι υλοποιημένο σε συνδυασμό με Provider και τοπικό σύστημα διαχείρισης κατάστασης, ώστε να διασφαλίζεται η συνέπεια και η ακρίβεια στον χρόνο αποστολής.

Οι ημερήσιες συμβουλές υλοποιούνται ως μέρος του frontend της εφαρμογής, αντλώντας δεδομένα είτε από mock υπηρεσία είτε από το Firestore και παρουσιάζονται σε ειδική οθόνη. Κάθε συμβουλή αντιστοιχεί σε μία ημέρα χωρίς κάπνισμα (ημέρα 1 έως 30). Παρουσιάζεται με κάρτα που ξεκλειδώνει αυτόματα κάθε 24 ώρες. Επίσης, συνοδεύεται από επιστημονική τεκμηρίωση σχετικά με τρόπους αντιμετώπισης συμπτωμάτων στέρησης και τεχνικές διακοπής καπνίσματος. Επιπλέον, παραμένει ορατή και προσβάσιμη ακόμη και μετά το ξεκλείδωμα. Η λειτουργία αυτή ενισχύει την καθημερινή σύνδεση του χρήστη με την εφαρμογή, προσφέροντας σταθερά μικρά κίνητρα. Επίσης, ο χρήστης μπορεί να αξιολογήσει την κάθε συμβουλή σε ένα σύστημα 5 αστεριών, ώστε να θυμάται τι τον βοήθησε περισσότερο. Οι ξεκλειδωμένες συμβουλές ταξινομούνται ανάλογα με την αξιολόγηση του χρήστη.

Η συνέργεια μεταξύ των ημερήσιων συμβουλών και των ειδοποιήσεων αποτελεί μια μορφή ψηφιακού καθοδηγητή. Προσφέρεται καθημερινή υποστήριξη σε λογικό και συναισθηματικό επίπεδο και ενισχύει τη συνήθεια χρήσης της εφαρμογής με φυσικό τρόπο.

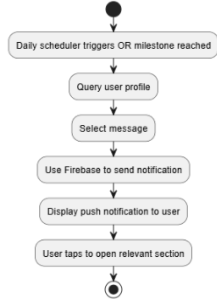


Εικόνα 26: Activity Diagram-Daily Tips



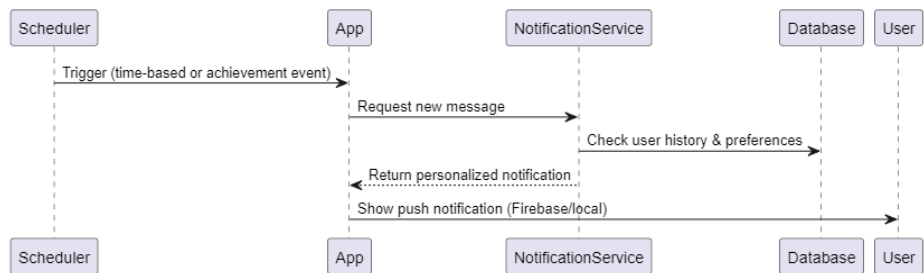
Εικόνα 27: Sequence Diagram-Daily Tips

Activity Diagram: Motivational Notifications



Εικόνα 28: Activity Diagram- Notification

Sequence Diagram: Receive Motivational Notification



Εικόνα 29: Sequence Diagram-Notifications

5.3.5 Chatbot Υποστήριξης

Το chatbot αποτελεί μία από τις πιο καινοτόμες πτυχές του συστήματος, καθώς προσφέρει στους χρήστες μια διαδραστική, φυσική συνομιλία με έναν έξυπνο agent βασισμένο σε τεχνητή νοημοσύνη. Το chatbot αυτό χρησιμοποιείται για την παροχή υποστήριξης, ψυχολογικής ενίσχυσης. Απαντά σε ερωτήσεις και δίνει χρήσιμες πληροφορίες σχετικά με τη διακοπή καπνίσματος οποιασδήποτε στιγμή, χωρίς ανθρώπινη παρέμβαση.

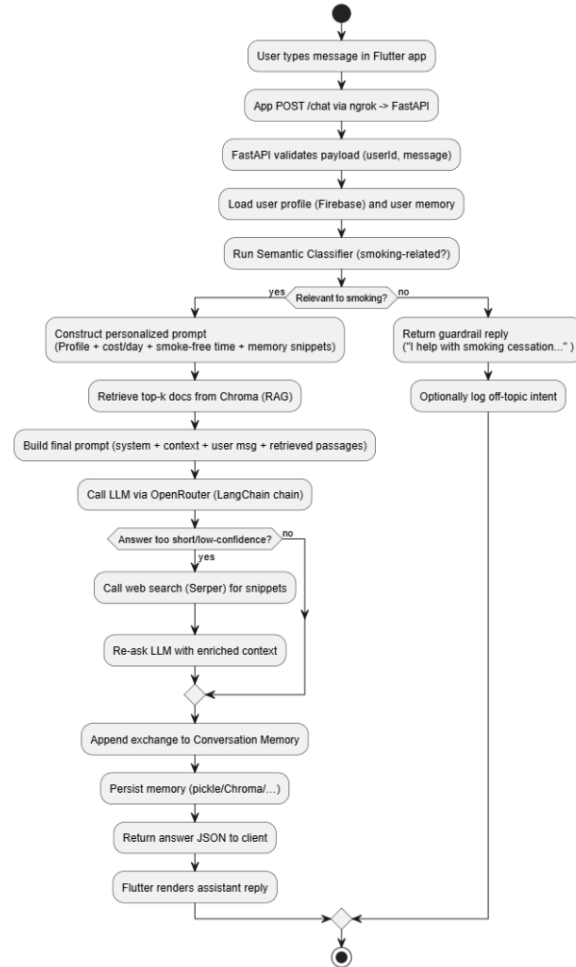
Μερικά παράδειγμα πηγών για RAG που χρησιμοποιήθηκαν είναι οδηγίες διακοπής από τον ΠΟΥ, επιστημονικά άρθρα (PubMed), εσωτερικά FAQ της εφαρμογής και οδηγίες για διαχείριση υποτροπών ή άγχους.

Backend και Ενσωμάτωση

Το Python backend φιλοξενεί την λογική του chatbot και εκθέτει FastAPI endpoints στα οποία το Flutter frontend στέλνει αιτήματα (HTTP POST με user query). Η απάντηση του chatbot επιστρέφεται και προβάλλεται στο UI με ένα messaging interface παρόμοιο με chat εφαρμογές. Η επικοινωνία είναι stateful ώστε να διατηρείται ιστορικό συνομιλίας ανά session.

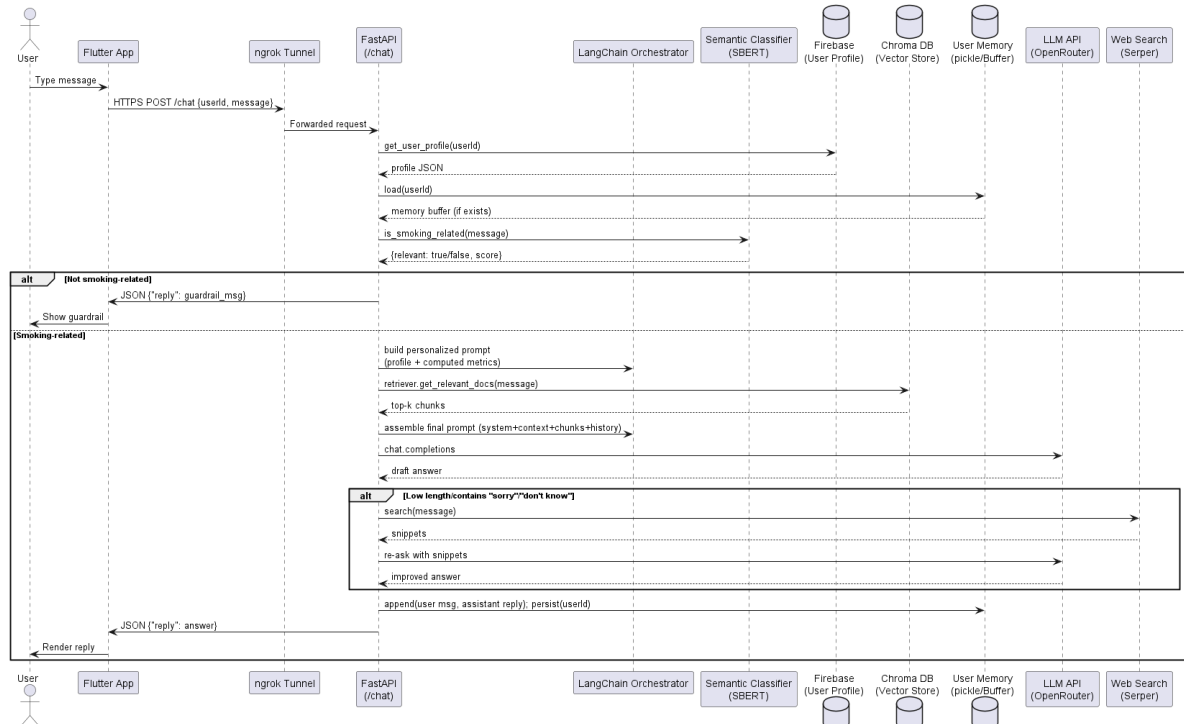
Μερικά από τα πλεονεκτήματα χρήσης του αποτελούν η συνεχής υποστήριξη χωρίς ανάγκη για ανθρώπινο προσωπικό, η εξατομίκευση απαντήσεων με βάση το ερωτηματολόγιο και οι επιστημονικά τεκμηριωμένες απαντήσεις μέσω RAG. Επιπλέον, προσφέρει καλύτερη εμπειρία onboarding για νέους χρήστες, ενώ μπορεί να λειτουργήσει ως πρώτη γραμμή υποστήριξης πριν προχωρήσει ο χρήστης σε επαγγελματική βοήθεια. Επιπροσθέτως, υποστηρίζει memory retention για προοδευτική μάθηση προφίλ χρήστη και contextual prompt injection με δεδομένα από άλλα υποσυστήματα (π.χ. cravings, επιτεύγματα, ιστορικό).

Activity Diagram - Smoking Cessation Chatbot (LangChain + RAG + Memory + Semantic Filter)



Εικόνα 30: Activity Diagram-Chatbot

Sequence Diagram - Chatbot with LangChain, RAG, Memory, Semantic Filter



Εικόνα 31: Sequence Diagram-Chatbot

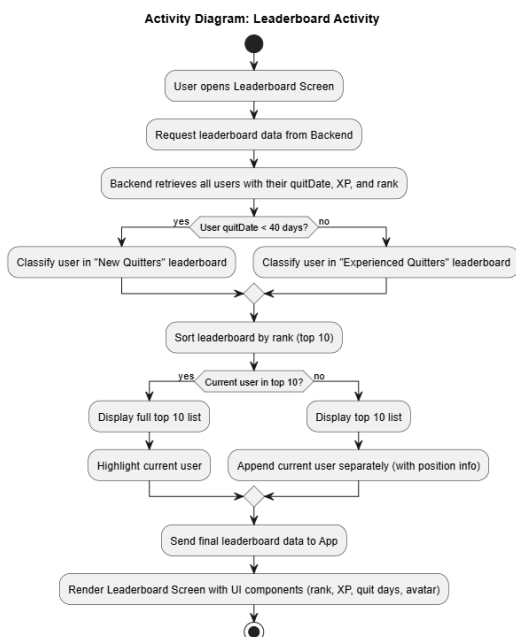
5.3.6 Πίνακας Κατάταξης και Κοινωνικές Λειτουργίες

Το υποσύστημα του Πίνακα Κατάταξης (Leaderboard) και των Κοινωνικών Λειτουργιών αποτελεί μία από τις πιο κρίσιμες λειτουργίες της εφαρμογής, καθώς ενισχύει το κίνητρο των χρηστών μέσω της κοινωνικής σύγκρισης και της διάδρασης. Στόχος του είναι να μετατρέψει τη διακοπή καπνίσματος από μια μοναχική εμπειρία σε μια συλλογική προσπάθεια, όπου οι χρήστες παρακινούνται μεταξύ τους μέσα από πρόοδο, επίδοση και επικοινωνία.

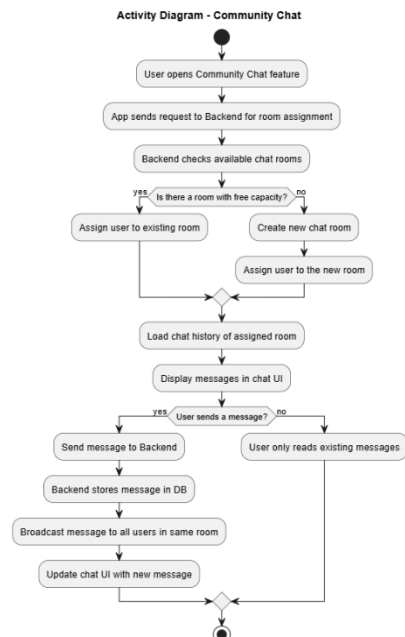
Ο πίνακας κατάταξης βασίζεται σε έναν μηχανισμό αξιολόγησης που ταξινομεί τους χρήστες με βάση τον συνολικό αριθμό ημερών χωρίς κάπνισμα (quit date). Η κατάταξη χωρίζεται σε δύο βασικές κατηγορίες, στους χρήστες με κάτω από 40 ημέρες διακοπής και στους χρήστες με 40 ή περισσότερες ημέρες. Αυτό επιτρέπει τη δίκαιη σύγκριση μεταξύ χρηστών σε διαφορετικά στάδια της διακοπής. Ο χρήστης μπορεί να δει τους 10 πρώτους χρήστες της κατάταξης της κατηγορίας του, καθώς και τη δική του θέση ακόμα κι αν δεν βρίσκεται στο top 10. Οι πληροφορίες που προβάλλονται για κάθε χρήστη περιλαμβάνουν το όνομα χρήστη, την θέση του στην κατάταξη, το επίπεδο (rank) και ημέρες αποχής.

Για να ενισχυθεί η αίσθηση κοινότητας και υποστήριξης, έχουν ενσωματωθεί λειτουργίες κοινωνικής διάδρασης όπως προσθήκη φίλων μέσω username, παρακολούθηση προόδου φίλων με δυνατότητα προβολής rank, XP και ημερών χωρίς κάπνισμα καθώς και ομαδική συνομηλία (Chatroom) για γενική επικοινωνία, ανταλλαγή εμπειριών, υποστήριξη και ενθάρρυνση μεταξύ χρηστών. Το chatroom υλοποιείται με real-time υποστήριξη.

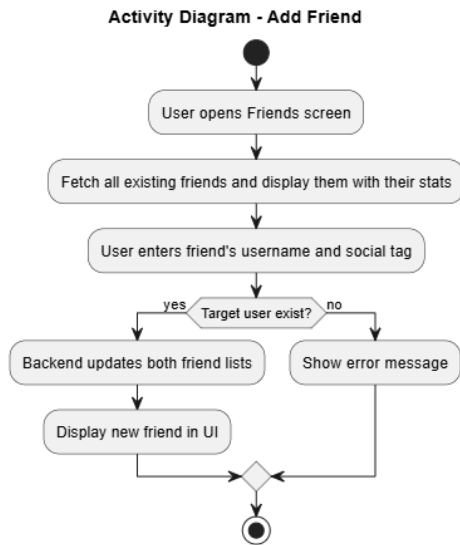
Η ύπαρξη του leaderboard και της κοινωνικής δικτύωσης παρέχει στους χρήστες κίνητρο μέσω του ανταγωνισμού (η παρουσία σε έναν πίνακα κατάταξης μπορεί να ενισχύσει την επιμονή). Επίσης, δημιουργεί την αίσθηση κοινότητας, καθώς και το γεγονός ότι οι χρήστες δεν είναι μόνοι τους στην προσπάθεια, αυξάνει την προσκόλληση και δημόσια αναγνώριση των προσπαθειών και των επιτευγμάτων. Επιπλέον, οι κοινές εμπειρίες δημιουργούν συναισθηματικούς δεσμούς και μειώνουν την πιθανότητα υποτροπής λόγω απομόνωσης ή άγχους.



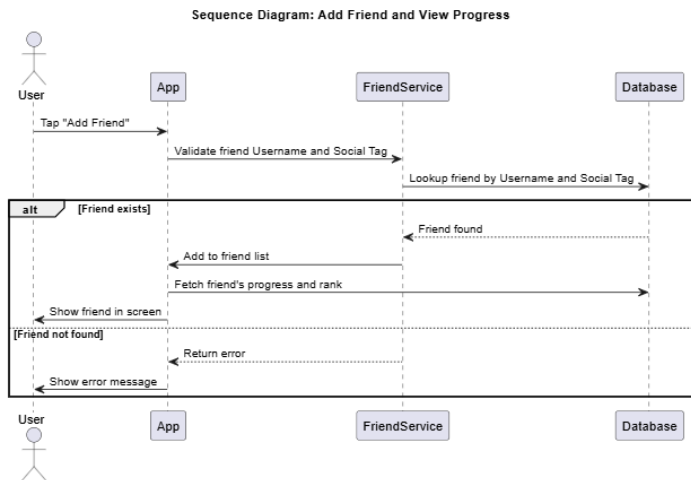
Εικόνα 32: Activity Diagram-Leaderboard



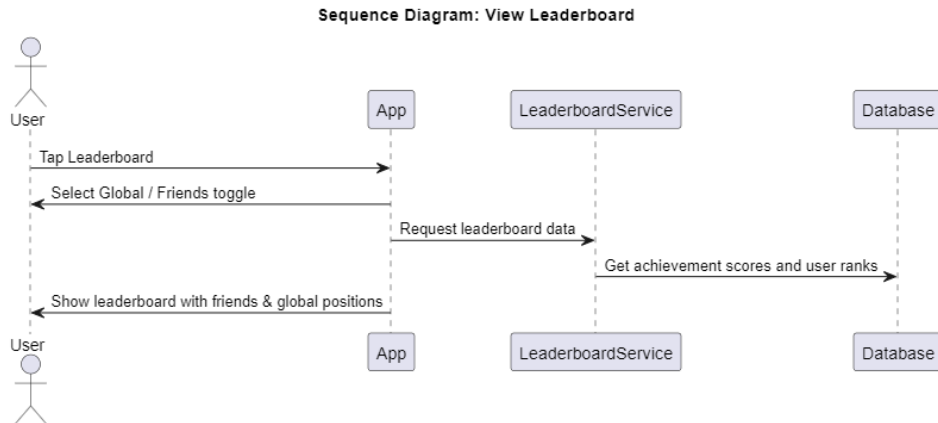
Εικόνα 33: Activity Diagram- Community Chat



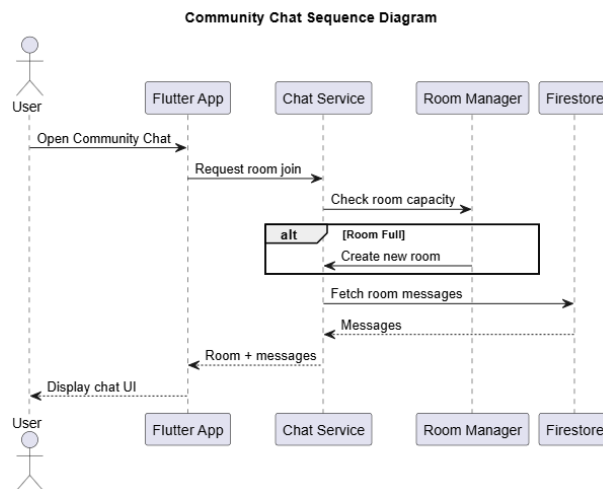
Εικόνα 34: Activity Diagram-Add Friend



Εικόνα 35: Sequence Diagram-Add Friend



Εικόνα 36: Sequence Diagram-Leaderboard



Εικόνα 37: Sequence Diagram-Community Chat

5.3.7 Αναπνευστικές Ασκήσεις και Παιχνίδια

Ένα σημαντικό υποσύστημα είναι εκείνο που εστιάζει στην απόσπαση της προσοχής του χρήστη τη στιγμή που αισθάνεται έντονη επιθυμία για κάπνισμα. Αυτό επιτυγχάνεται με δύο βασικούς τρόπους, μέσω τεχνικών αναπνοής και μέσω διασκεδαστικών mini-games, που ενισχύουν την ψυχολογική ανθεκτικότητα και την αυτοσυγκράτηση.

Αναπνευστικές Ασκήσεις

Οι αναπνευστικές ασκήσεις αποτελούν επιστημονικά τεκμηριωμένες μεθόδους μείωσης του άγχους και ελέγχου της επιθυμίας για κάπνισμα. Στην εφαρμογή, έχει σχεδιαστεί μια ειδική διεπαφή όπου ο χρήστης μπορεί να ακολουθήσει οδηγίες καθοδηγούμενης αναπνοής, βασισμένες σε μεθόδους όπως Box Breathing, 4-7-8 Breathing και Diaphragmatic Breathing.

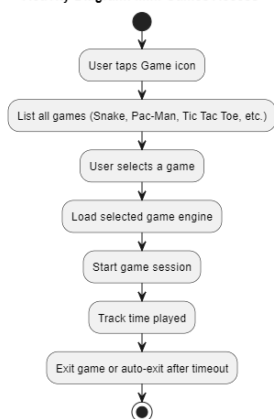
Η διεπαφή περιλαμβάνει ένα κινούμενο κύκλο που επεκτείνεται και συστέλλεται σύμφωνα με τις φάσεις εισπνοής, κράτησης και εκπνοής, προσφέροντας οπτική καθοδήγηση. Ο χρήστης καλείται να επαναλάβει τους κύκλους αναπνοής για συγκεκριμένο χρονικό διάστημα (π.χ. 2 3 λεπτά), με στόχο την αποσυμφόρηση του άγχους και την αποτροπή της υποτροπής.

Mini-games

Τα mini-games προσφέρουν άμεση και ευχάριστη απόσπαση της προσοχής, αντικαθιστώντας τη ρουτίνα του καπνίσματος με μια δραστηριότητα που κρατά τον χρήστη απασχολημένο, αποτρέποντας την επαναφορά στη συνήθεια. Τα παιχνίδια είναι ενσωματωμένα στην εφαρμογή ως ολοσέλιδες εμπειρίες, με responsive και απλή διεπαφή. Ορισμένα παραδείγματα mini-games που έχουν ήδη ενσωματωθεί περιλαμβάνουν φιδάκι (κλασικό arcade παιχνίδι), Pac-Man (lightweight έκδοση), και Flappy Bird (παιχνίδι αντανakλαστικών που προάγει τη συγκέντρωση). Τα mini-games είναι σχεδιασμένα έτσι ώστε να απαιτούν ελάχιστους πόρους συστήματος, να είναι γρήγορα προσβάσιμα και να μην απαιτούν σύνδεση στο διαδίκτυο.

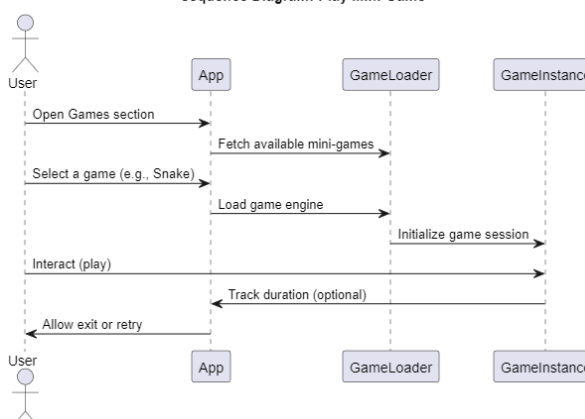
Τα mini-games και οι αναπνευστικές ασκήσεις είναι προσβάσιμα από την αρχική οθόνη, μέσω κουμπιού Beat Cravings. Αυτός ο σχεδιασμός ενθαρρύνει την άμεση χρήση σε κρίσιμες στιγμές. Με βάση τις έρευνες, η παροχή τέτοιων λειτουργιών αποσπά την προσοχή από τη σκέψη του καπνίσματος. Επιπλέον, ενισχύει τη συναισθηματική σταθερότητα και καλλιεργεί νέες συνήθειες και μηχανισμούς διαχείρισης στρες. Η προσέγγιση αυτή ενισχύει την ψυχολογική ευελιξία του χρήστη, βοηθώντας τον να αποσυνδέσει την επιθυμία από την αντίδραση (κάπνισμα) και να αποκτήσει έλεγχο.

Activity Diagram: Mini-Games Access

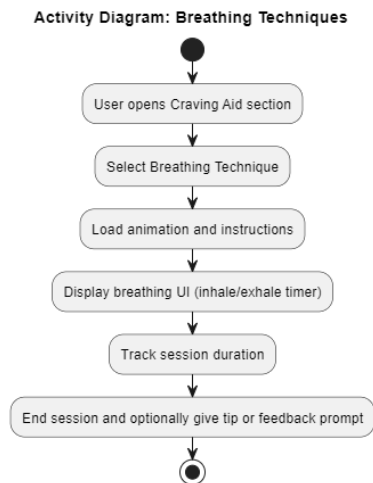


Εικόνα 38: Activity Diagram-Mini Games

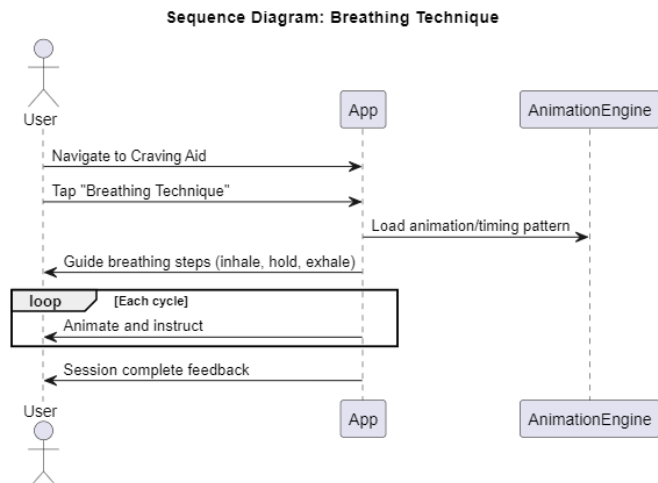
Sequence Diagram: Play Mini-Game



Εικόνα 39: Sequence Diagram-Mini Games



Εικόνα 40: Activity Diagram-Breathing Technique



Εικόνα 41: Sequence Diagram-Breathing Techniques

5.3.8 Σύστημα Καταγραφής Στερητικών Συμπτωμάτων

Ένα από τα πιο κρίσιμα στοιχεία για την υποστήριξη ενός χρήστη στην προσπάθεια του για απεξάρτηση από το κάπνισμα, είναι η παρακολούθηση και διαχείριση των στερητικών συμπτωμάτων. Τα συμπτώματα αυτά αποτελούν φυσιολογική συνέπεια της σωματικής και ψυχολογικής εξάρτησης από τη νικοτίνη. Μπορεί να περιλαμβάνουν άγχος, ευερεθιστότητα, αυξημένη όρεξη, αϋπνία, δυσκολία συγκέντρωσης, πονοκεφάλους και έντονη επιθυμία για κάπνισμα (cravings).

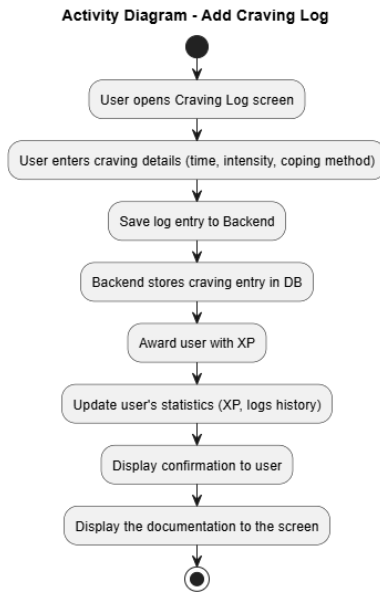
Η εφαρμογή ενσωματώνει ένα ειδικό υποσύστημα καταγραφής των παραπάνω συμπτωμάτων, επιτρέποντας στον χρήστη να σημειώνει την ένταση κάθε συμπτώματος, να προσθέτει σχόλια ή παρατηρήσεις (π.χ. τι πυροδότησε το σύμπτωμα ή πώς το αντιμετώπισε). Τέλος, καταχωρείται και η χρονική στιγμή εμφάνισης του συμπτώματος.

Η διεπαφή είναι απλή και φιλική προς τον χρήστη, με drop down menu για την ένταση της επιθυμίας του να καπνίσει και πεδίο για σχόλια. Οι καταγραφές αποθηκεύονται τοπικά και συγχρονίζονται με το προφίλ του χρήστη στη βάση δεδομένων (μέσω Firestore).

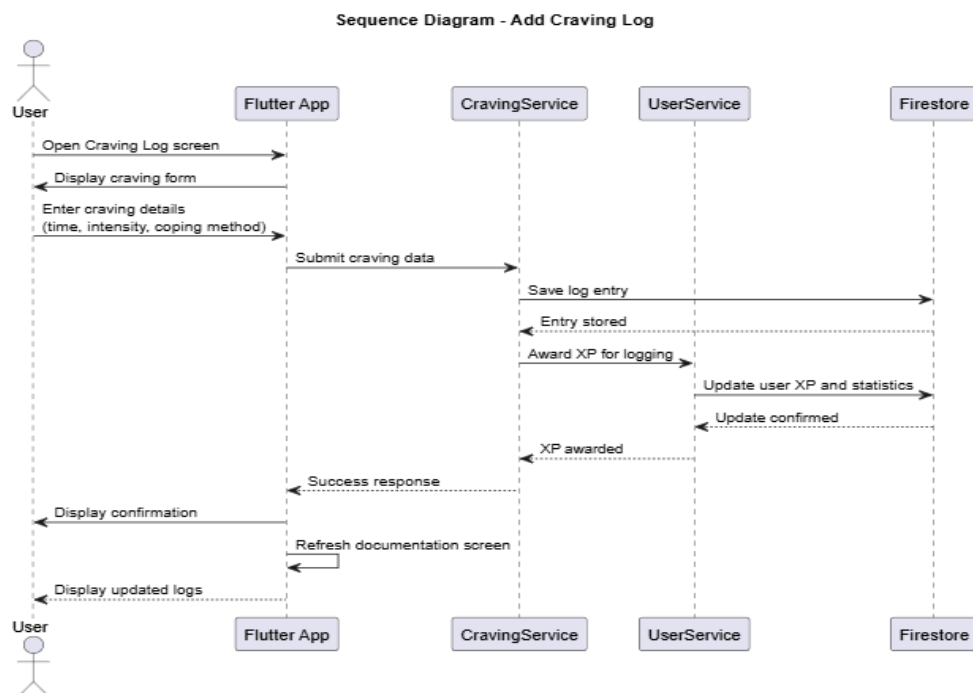
Η εφαρμογή προσφέρει δυνατότητα προβολής ιστορικού των καταγραφών, καθώς και προβολή όλων των καταγράφων όλων των χρηστών. Με αυτόν τον τρόπο, ο χρήστης παρατηρεί πως άλλοι αντιμετώπισαν επιτυχώς τα στερητικά συμπτώματα τους, παίρνει ιδέες αντιμετώπισης και τις εφαρμόζει στο μέλλον. Έτσι, χτίζεται πιο δυνατό το αίσθημα της κοινότητας των ατόμων που προσπαθούν να αντιμετωπίσουν τον εθισμό τους. Επιπλέον, έπειτα από κάθε καταγραφή επιτυχούς αντιμετώπισης συμπτωμάτων, ο χρήστης επιβραβεύεται με πόντους (XP) στο προφίλ του.

Η δυνατότητα καταγραφής των στερητικών συμπτωμάτων λειτουργεί θεραπευτικά και ενδυναμώνει τον χρήστη. Συγκεκριμένα, τον βοηθά να κατανοήσει και να παρακολουθήσει τη συμπεριφορά του σώματός του. Επίσης, του δίνει έλεγχο απέναντι στο φαινόμενο της στέρησης, ενώ παρέχει χειροπιαστή πρόοδο, όταν διαπιστώνεται μείωση στην ένταση ή τη συχνότητα. Τέλος, συμβάλλει στην εξατομικευμένη υποστήριξη, αφού μπορεί να συσχετιστεί με τα Daily Tips που τον βοηθούν. Η συνεχής καταγραφή ενισχύει το αίσθημα ότι ο χρήστης

δεν είναι μόνος και ότι η εφαρμογή παρακολουθεί, κατανοεί και βοηθά την πορεία του με ακρίβεια.



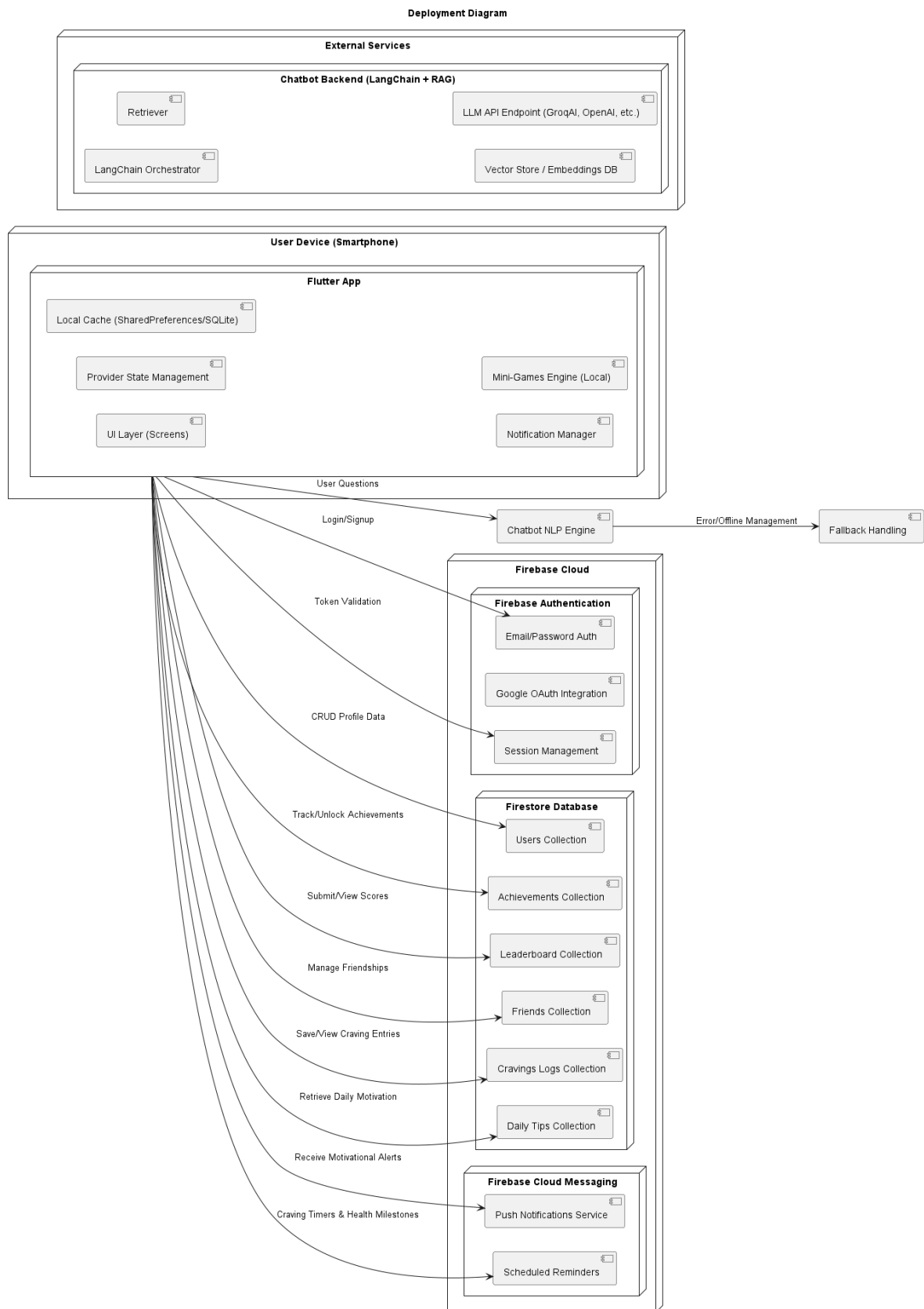
Εικόνα 42: Activity Diagram-Craving Logs



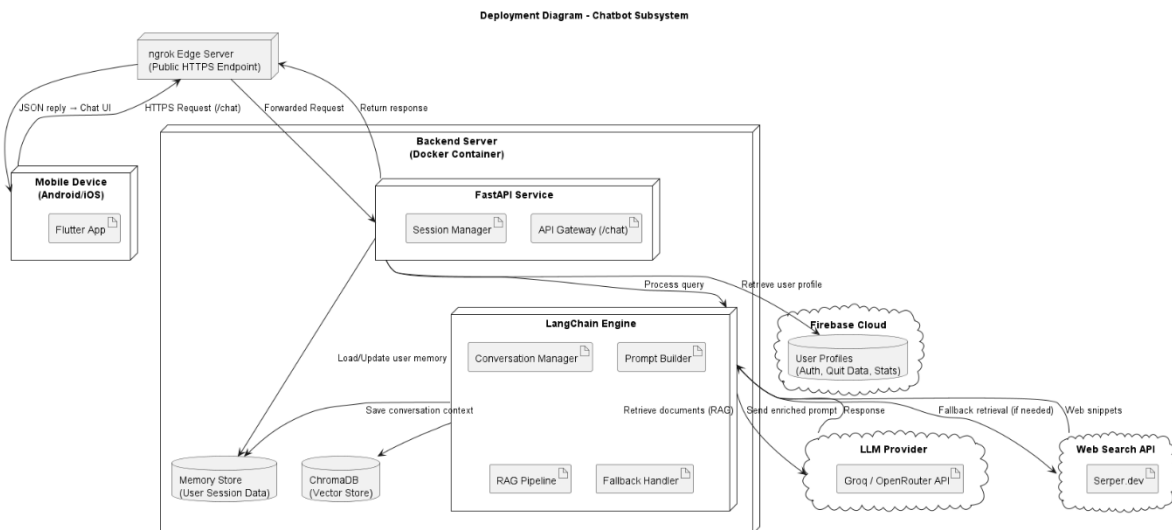
Εικόνα 43: Sequence Diagram-Craving Logs

5.4 Φυσική Αρχιτεκτονική και Διαμόρφωση του Συστήματος

Η διαμόρφωση του συστήματος και η κατανόηση των σχέσεων μεταξύ στοιχείων λογισμικού (software artifacts) και φυσικών ή εικονικών κόμβων (nodes), φαίνεται παρακάτω. Η απεικόνιση αυτή γίνεται μέσω δύο deployment διαγραμμάτων.



Εικόνα 44: Deployment Diagram



Εικόνα 45: Chatbot Deployment Diagram

Τα deployment diagrams χρησιμοποιούνται για την απεικόνιση της φυσικής αρχιτεκτονικής ενός συστήματος και δείχνει πώς αναπτύσσεται και πώς λειτουργεί το σύστημα σε πραγματικό περιβάλλον. Σε αυτό αποτυπώνονται οι συσκευές, οι υπηρεσίες και οι κόμβοι λογισμικού που συνεργάζονται για να παρέχουν όλες τις λειτουργίες της εφαρμογής.

User Device

UI Layer (Screens): Όλες οι οθόνες που συνθέτουν το γραφικό περιβάλλον της εφαρμογής. Από εδώ ο χρήστης αλληλοεπιδρά με τις λειτουργίες, βλέπει τις συμβουλές, παρακολουθεί την πρόοδό του και συμμετέχει σε chat.

Provider State Management: Το σύστημα διαχείρισης κατάστασης με Provider. Εξασφαλίζει ότι τα δεδομένα της εφαρμογής συγχρονίζονται σωστά μεταξύ των διαφόρων οθονών και υπηρεσιών.

Local Cache (SharedPreferences/SQLite): Τοπική μνήμη όπου αποθηκεύονται προσωρινά ρυθμίσεις και δεδομένα ώστε η εφαρμογή να μπορεί να λειτουργεί ακόμη και χωρίς σύνδεση στο διαδίκτυο.

Notification Manager: Διαχειρίζεται τις ειδοποιήσεις που λαμβάνει η εφαρμογή, είτε πρόκειται για προγραμματισμένα reminders είτε για push notifications από το Firebase.

Mini-Games Engine (Local): Η μηχανή που εκτελεί τα μικρά παιχνίδια που ενσωματώνονται στην εφαρμογή. Τα παιχνίδια λειτουργούν τοπικά χωρίς ανάγκη επικοινωνίας με τον server, προσφέροντας γρήγορη απόκριση και ψυχαγωγία.

Firebase Cloud

Firebase Authentication: Αναλαμβάνει την είσοδο και εγγραφή χρηστών. Υποστηρίζεται τόσο login με email και password όσο και είσοδος μέσω Google. Το Session Management διαχειρίζεται τα tokens και εξασφαλίζει ότι ο χρήστης παραμένει συνδεδεμένος με ασφάλεια.

Firestore Database: Αποτελεί τη βάση δεδομένων της εφαρμογής. Εδώ αποθηκεύονται όλα τα δυναμικά δεδομένα, όπως τα στοιχεία χρηστών, οι φίλοι τους, τα επιτεύγματα που έχουν

κατακτήσει, τα ημερολόγια cravings και οι καθημερινές συμβουλές. Η δομή είναι σε μορφή συλλογών και εγγράφων, κάτι που ταιριάζει με το μοντέλο δεδομένων της εφαρμογής.

Firebase Cloud Messaging: Χρησιμοποιείται για να στέλνει ειδοποιήσεις στους χρήστες. Υπάρχει υποστήριξη τόσο για push notifications, που στέλνονται σε πραγματικό χρόνο, όσο και για προγραμματισμένα reminders που θυμίζουν στον χρήστη milestones ή τον παρακινούν να ξεπεράσει μια δύσκολη στιγμή.

External Services

Chatbot Backend : Εδώ υλοποιείται η λειτουργία της συνομιλίας με έναν έξυπνο βοηθό. Το σύστημα βασίζεται σε LangChain, που οργανώνει τη ροή της συζήτησης με RAG.

Vector Store / Embeddings DB: Αποθηκεύονται αναπαραστάσεις κειμένων σε διανυσματική μορφή για να μπορεί το σύστημα να βρίσκει σχετικές πληροφορίες.

Ngrok Edge Server: Ο ρόλος του ngrok είναι να λειτουργεί ως ενδιάμεσος ασφαλής server. Το ngrok αναλαμβάνει να προωθήσει το αίτημα του χρήστη από το κινητό προς τον backend server.

FastAPI Service: λειτουργεί ως API Gateway, παρέχοντας το endpoint /chat για την επικοινωνία με την εφαρμογή. Το **Session Manager** τακτοποιεί τον χρήστη και φορτώνει τα δεδομένα της συνεδρίας του. Επίσης, επικοινωνεί με το Firebase για την ανάκτηση στοιχείων του χρήστη.

LangChain Engine: παρακολουθεί και διαχειρίζεται τον διάλογο με τον χρήστη. Ο **Conversation Manager** κρατάει το ιστορικό των συνομιλιών και διαχειρίζεται τη ροή του διαλόγου. Ο **Prompt Builder** δημιουργεί τα prompts που αποστέλλονται στο LLM, εμπλουτίζοντάς τα με δεδομένα από τη βάση γνώσης. Το **RAG Pipeline** συνδυάζει τις γενετικές δυνατότητες του LLM με εξωτερικά έγγραφα που ανακτώνται από τον vector store, ώστε οι απαντήσεις να βασίζονται σε πραγματική γνώση και όχι μόνο σε γλωσσικά μοντέλα. Ο **Fallback Handler** καλύπτει περιπτώσεις όπου το LLM δεν μπορεί να δώσει επαρκή απάντηση, απευθυνόμενος σε μια εξωτερική μηχανή αναζήτησης για επιπλέον πληροφορίες.

ChromaDB (Vector Store) : αποθηκεύει έγγραφα σε διανυσματική μορφή (embeddings), επιτρέποντας στο chatbot να πραγματοποιεί έξυπνη αναζήτηση περιεχομένου και να ανακτά πληροφορίες σχετικές με το ερώτημα του χρήστη.

Memory Store (User Session Data): διατηρεί τα δεδομένα της συνεδρίας και το ιστορικό των συνομιλιών.

LLM Provider: φιλοξενεί τα LLMs, τα οποία λαμβάνουν τα εμπλουτισμένα prompts από το backend και επιστρέφουν απαντήσεις σε φυσική γλώσσα.

Web Search API (Serper): λειτουργεί ως fallback πηγή πληροφόρησης, παρέχοντας αποσπάσματα από το διαδίκτυο όταν το LLM δεν γνωρίζει την απάντηση ή απαιτούνται επικαιροποιημένα δεδομένα.

6. Υλοποίηση της Εφαρμογής

Η διαδικασία κατασκευής των οθονών στην εφαρμογή ακολούθησε μια οργανωμένη και συστηματική προσέγγιση, με βασικό άξονα την καθαρή δομή, τη δυνατότητα επαναχρησιμοποίησης στοιχείων και τη μελλοντική ευκολία συντήρησης. Σε κάθε περίπτωση, η ανάπτυξη ξεκινούσε με τον σχεδιασμό της εκάστοτε οθόνης, σύμφωνα με τις λειτουργικές απαιτήσεις και τη γενική αισθητική του έργου. Καθοριζόταν εξαρχής το περιεχόμενο που έπρεπε να εμφανίζεται, οι αλληλεπιδράσεις που έπρεπε να υποστηρίζονται, καθώς και ο σκοπός που εξυπηρετούσε η οθόνη ως κομμάτι του συνόλου της εφαρμογής.

Η τελική φάση περιλάμβανε την οπτική και λειτουργική δοκιμή κάθε οθόνης, με εκτέλεση της εφαρμογής σε φυσική συσκευή, ώστε να επαληθευτεί η ακριβής απόδοση των mock δεδομένων και η ομαλή ροή εμπειρίας για τον χρήστη.

Η υλοποίηση των οθονών με αυτό τον τρόπο εξασφάλισε ένα αρθρωτό, επεκτάσιμο και εύχρηστο περιβάλλον διεπαφής, ενώ τα mock δεδομένα επέτρεψαν τον πλήρη έλεγχο των σεναρίων χρήσης ακόμα και πριν την ολοκλήρωση της σύνδεσης με Firebase και Firestore. Ανάλογα με τις ανάγκες κάθε σελίδας (στατική ή δυναμική αλλαγή), χρησιμοποιήθηκε το κατάλληλο widget τύπου.

Η οργάνωση των αρχείων της εφαρμογής διαδραματίζει καθοριστικό ρόλο στη συντήρηση, την επεκτασιμότητα και τη ανάπτυξη ενός έργου. Το σύστημα ακολουθεί μία καθαρή, αρθρωτή δομή βασισμένη σε βέλτιστες πρακτικές του οικοσυστήματος και της αρχιτεκτονικής MVVM (Model-View-ViewModel).

6.1 Ενσωμάτωση Firebase και Firestore

Η ενσωμάτωση του Firebase και της Firestore στην εφαρμογή αποτέλεσε κρίσιμο βήμα για την παροχή δυνατοτήτων authentication, αποθήκευσης δεδομένων χρηστών, leaderboard, ειδοποιήσεων και γενικότερα τη σύνδεση του frontend με ένα cloud-based backend σύστημα.

Η διαδικασία ξεκίνησε με την αρχική παραμετροποίηση του Firebase Console. Δημιουργήθηκε ένα νέο project, στο οποίο συνδέθηκε η Android έκδοση της εφαρμογής μέσω του αρχείου google-services.json, το οποίο προστέθηκε στον φάκελο android/app/ της εφαρμογής. Παράλληλα, έγινε η απαραίτητη ρύθμιση των dependencies στα αρχεία build.gradle, στο επίπεδο του project και στο module, ώστε να επιτραπεί η χρήση των Firebase SDKs.

Στο επίπεδο του Flutter, χρησιμοποιήθηκαν κυρίως τα πακέτα `firebase_core`, `firebase_auth`, `cloud_firestore`, `firebase_storage` και `firebase_messaging`. Η αρχικοποίηση του Firebase πραγματοποιήθηκε στο `main.dart`, όπου πριν εκκινήθει η εφαρμογή, προστέθηκε η εντολή `await Firebase.initializeApp()`, διασφαλίζοντας ότι όλες οι Firebase υπηρεσίες είναι έτοιμες πριν αρχίσει η ροή του UI.

Στη συνέχεια, δημιουργήθηκε μία υπηρεσία `FirestoreService`, υπεύθυνη για την επικοινωνία με τη βάση δεδομένων Firestore. Αυτή η υπηρεσία περιλάμβανε μεθόδους για την ανάκτηση,

δημιουργία και ενημέρωση εγγράφων, οργανωμένων ανά collection, όπως users, ratings, global_chat_rooms, global_chat_messages, και carving_logs. Για παράδειγμα, η μέθοδος `getUserById(String uid)` αναζητά το έγγραφο με το συγκεκριμένο UID και επέστρεφε ένα αντικείμενο τύπου `UserModel`, το οποίο είχε οριστεί στο αρχείο `models/user_model.dart`.

Η αυθεντικοποίηση χρηστών επιτεύχθηκε μέσω του `firebase_auth`, υποστηρίζοντας τόσο εγγραφή και σύνδεση με email/κωδικό όσο και Google Sign-In. Ολόκληρη η ροή authentication διαχειρίστηκε μέσω `AuthService`, όπου υλοποιήθηκαν συναρτήσεις όπως `signInWithEmailAndPassword`, `registerWithEmailAndPassword`, και `signInWithGoogle`. Μετά τη σύνδεση, η εφαρμογή παίρνει το uid του χρήστη και το χρησιμοποιεί για να ανακτήσει ή να δημιουργήσει το προφίλ του στο Firestore.

Κατά την εγγραφή, επιπλέον δεδομένα όπως το φύλο, οι ημέρες χωρίς κάπνισμα, το κόστος τσιγάρων κτλ., αποθηκεύονταν στη βάση για μελλοντική χρήση. Αυτά τα δεδομένα συνθέταν ένα custom user model, το οποίο χρησιμοποιήθηκε καθολικά στην εφαρμογή. Το Firestore χρησιμοποιήθηκε επίσης για την αποθήκευση της προόδου του χρήστη στα επιτεύγματα. Όταν ο χρήστης πλησιάζει στην ολοκλήρωση ενός στόχου, η πρόοδος ενημερώνεται δυναμικά μέσω μεθόδων της υπηρεσίας `AchievementProvider`, ενώ το UI αντανακλούσε αυτές τις αλλαγές με χρήση `StreamBuilder`.

Συνολικά, η ενσωμάτωση του Firebase και του Firestore προσέφερε μία αποδοτική βάση δεδομένων με δυνατότητες real-time ενημέρωσης, επιτρέποντας στην εφαρμογή να χειριστεί κρίσιμες λειτουργίες με ασφάλεια και αξιοπιστία, χωρίς την ανάγκη για τοπικό server ή custom backend.

6.2 Υλοποίηση Chatbot

Το chatbot υλοποιήθηκε σε Python, βασισμένο σε LLM και εκτέθηκε προσωρινά στο διαδίκτυο μέσω της υπηρεσίας Ngrok για απευθείας επικοινωνία με την εφαρμογή Flutter. Η βασικές τεχνολογίες για την Python πλευρά περιλαμβάνει FastAPI ως microframework για τον web server, το Langchain για τον χειρισμό των LLM αλληλεπιδράσεων και OpenRouter API για την πρόσβαση σε μοντέλα όπως το Groq και Deepseek. Το FastAPI περιλάμβανε μία βασική POST διαδρομή, π.χ. `/chat`, η οποία δέχεται τα ερωτήματα του χρήστη από την εφαρμογή σε μορφή JSON, τα προωθεί στο LLM, και επιστρέφει την απάντηση σε μορφή JSON.

```
20. @app.post("/chat/")
21. async def chat(request: ChatRequest):
22.     logger.info(f"Received message from user {request.userId}: {request.message}")
23.     try:
24.         reply = str(run_chat_pipeline(request.userId, request.message)) # ☒ force plain string
25.         logger.info(f"Reply generated for user {request.userId}: {reply}")
26.         logger.info(f"Reply type: {type(reply)}")
27.         return {"reply": reply}
28.     except Exception as e:
29.         logger.exception("Error in /chat/ endpoint")
30.         raise HTTPException(status_code=500, detail=str(e))
```

Κώδικας 1: FastAPI Endpoint

Για να επιτρέψουμε την απομακρυσμένη πρόσβαση σε αυτό το τοπικά φιλοξενούμενο chatbot από την Flutter εφαρμογή, χρησιμοποιήθηκε το εργαλείο Ngrok. Το Ngrok δημιουργεί ένα προσωρινό HTTPS URL το οποίο συνδέεται με το τοπικό FastAPI server, επιτρέποντας την επικοινωνία μεταξύ του Android app και του backend. Αξίζει να σημειωθεί ότι το Ngrok

παρέχει προσωρινά URLs, επομένως η συγκεκριμένη λύση χρησιμοποιήθηκε για σκοπούς ανάπτυξης και testing. Σε τελικό περιβάλλον παραγωγής, το chatbot backend μπορεί να αναπτυχθεί σε πλατφόρμες όπως Render ή AWS.

Στη Flutter πλευρά, δημιουργήθηκε ένας ChatProvider με μέθοδο που στέλνει το μήνυμα του χρήστη στο FastAPI endpoint και λαμβάνει την απάντηση. Ο χρήστης μπορεί να εισάγει το μήνυμά του σε μια φόρμα εντός του ChatbotScreen, να το στείλει, και να δει την απάντηση εμφανιζόμενη στο UI. Η επικοινωνία είναι απλή και λειτουργεί με async λογική ώστε να διατηρείται η απόκριση της εφαρμογής.

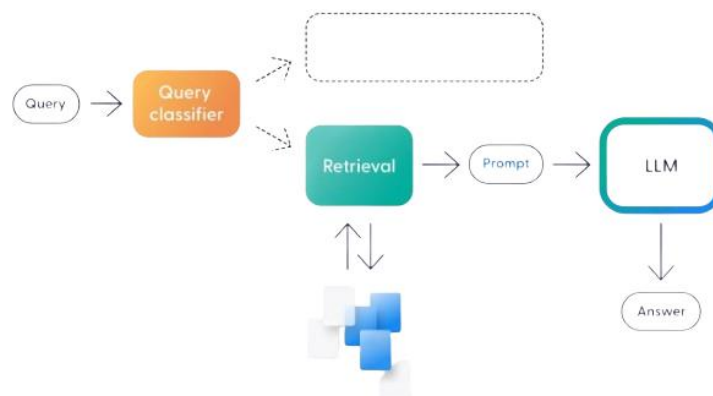
Η αρχιτεκτονική αυτής της λύσης επιτρέπει την αποσύνδεση του LLM chatbot από την εφαρμογή, διατηρώντας ευελιξία στην επιλογή μοντέλων, παρακολούθηση στατιστικών χρήσης και συνεχή βελτίωση της εμπειρίας συνομιλίας ανεξάρτητα από τις εκδόσεις του συστήματος.

Το σύστημα ξεκινάει ελέγχοντας αν η ερώτηση του χρήστη σχετίζεται με την κατάργηση του καπνίσματος. Χρησιμοποιεί ένα Sentence Transformer για να συγκρίνει σημασιολογικά το μήνυμα με μια προκαθορισμένη λίστα ερωτήσεων σχετικών με το κάπνισμα. Αν η ομοιότητα είναι κάτω από ένα κατώφλι (Threshold), το chatbot απαντάει ότι είναι διαθέσιμο μόνο για ερωτήσεις σχετικές με την κατάργηση του καπνίσματος, αποτρέποντας απρόσφορες απαντήσεις.

```
1. # Load semantic smoking classifier model
2. smoking_model = SentenceTransformer('all-MiniLM-L6-v2')
3.
4. # Define example smoking-related prompts
5. smoking_examples = [
6.     "How do I quit smoking?",
7.     "What are the health risks of cigarettes?",
8.     "Tips to manage cravings",
9.     "What happens to my lungs after quitting?",
10.    "What helps with nicotine withdrawal?",
11.    "How long does withdrawal last?",
12.    "Should I use nicotine patches or gum?",
13.    "How do I stay motivated to quit?",
14.    "What are benefits of quitting smoking?"
15. ]
16. smoking_embeddings = smoking_model.encode(smoking_examples, convert_to_tensor=True)
17.
18. def is_smoking_related_semantic(message: str, threshold: float = 0.7) -> bool:
19.     query_embedding = smoking_model.encode(message, convert_to_tensor=True)
20.     similarity_scores = util.cos_sim(query_embedding, smoking_embeddings)
21.     max_score = similarity_scores.max().item()
22.     logger.info(f"Semantic similarity score: {max_score:.3f}")
23.     return max_score >= threshold
24.
25.
```

Κώδικας 2: Semantic Classifier

Η κεντρική ροή καθοδηγείται από τη συνάρτηση `run_chat_pipeline(userId: str, message: str)`, η οποία ακολουθεί τα εξής βήματα. Αρχικά, φιλτράρει το θέμα (Semantic Filtering). Χρησιμοποιεί το μοντέλο all-MiniLM-L6-v2 από τη βιβλιοθήκη sentence-transformers, κωδικοποιεί το μήνυμα του χρήστη και το συγκρίνει με μια προ-κωδικοποιημένη λίστα `smoking_examples` (ερωτήσεις σχετικές με το κάπνισμα) υπολογίζοντας το cosine similarity. Αν το `max_score` από τις ομοιότητες είναι μικρότερο από ένα κατώφλι (`threshold=0.6`), το μήνυμα θεωρείται off-topic. Έπειτα, το chatbot επιστρέφει μια τυποποιημένη απάντηση, αποφεύγοντας να δαπανήσει πόρους (tokens, API calls) σε μη σχετικές ερωτήσεις.



Εικόνα 46: Ροή RAG με Classifier

Πριν απαντήσει, το chatbot φορτώνει το προφίλ του χρήστη από τη βάση δεδομένων. Αυτό το προφίλ περιλαμβάνει λεπτομερή στοιχεία όπως η ημερομηνία κατάργησης, ο αριθμός τσιγάρων, το κόστος, τα αίτια και τα συναισθήματα σχετικά με το κάπνισμα κτλ. Αυτά τα δεδομένα δομούνται σε ένα λεπτομερές πλαίσιο (prompt) που τροφοδοτείται στο μοντέλο τεχνητής νοημοσύνης. Αυτό του επιτρέπει να παρέχει εξατομικευμένες συμβουλές και να σχεδιάζει συγκεκριμένες στρατηγικές αντιμετώπισης βασισμένες στην κατάσταση του χρήστη.

```

1. # Structured user prompt for better LLM results
2.     personalized_context = f""""You are helping a user quit smoking.
3.
4.     User Profile:
5.     - Name: {user_profile.get("username", "Unknown")}
6.     - Gender: {user_profile.get("gender", "unspecified")}
7.     - Cigarettes per day: {cigarettes_per_day} cigarettes/day
8.     - Cigarettes per pack: {cigarettes_per_pack} cigarettes/pack
9.     - Cost per pack: {cost_per_pack} $
10.    - Cost per day: {cost_per_day:.2f} $
11.    - Smoking Duration: {user_profile.get("smokingYears", 0)} years
12.    - Quit Date: {quit_timestamp if quit_timestamp else "Not provided"}
13.    - Smoke-free time: {smoke_free_time_str}
14.    - Completed Achievements: {"", ".join(user_profile.get("completedAchievements", []))}
15.    - Reason of smoking: {user_profile.get("whySmoke", "Unknown")}
16.    - Feelings while smoking: {user_profile.get("feelWhenSmoking", "Unknown")}
17.    - Type of smoker: {user_profile.get("typeOfSmoker", "Unknown")}
18.    - Reason of quitting: {user_profile.get("whyQuit", "Unknown")}
19.    - Methods of quitting tried/ if tried: {user_profile.get("triedQuitMethods", "Unknown")}
20.    - What do cigarettes mean to the user emotionally: {user_profile.get("emotionalMeaning", "Unknown")}
21.    - In what situations the user crave cigarettes: {user_profile.get("cravingSituations", "Unknown")}
22.    - How confident is the user in quitting: {user_profile.get("confidenceLevel", "Unknown")}
23.    - Do the user live/work with other smokers? : {user_profile.get("smokingEnvironment", "Unknown")}
24.    - The users biggest fear about quitting: {user_profile.get("biggestFear", "Unknown")}
25.    - The users biggest motivation to stay smoke-free: {user_profile.get("biggestMotivation", "Unknown")}
26.
27.    Use this information to personalize your answers and motivation strategies.
28.    """"
29.
30.    full_prompt = personalized_context + "\n\nUser Message: " + message
31.

```

Κώδικας 3: Contextual Prompt Injection

Για την εξατομίκευση καλεί τη συνάρτηση `get_user_profile(userId)` (που ανακτά δεδομένα από το Firestore) για να φορτώσει το προφίλ του χρήστη. Έπειτα, δομεί ένα εκτενές, εξατομικευμένο prompt, που περιλαμβάνει δεδομένα όπως `cigarettesPerDay`, `costPerPack`, `quitDate`, `smokingYears`, `whyQuit`, `emotionalMeaning`, `biggestFear`, `biggestMotivation`, κ.λπ. Αυτό το prompt προσθέτει context στο αίτημα του χρήστη, επιτρέποντας στο LLM να δώσει πιο συγκεκριμένες και προσωπικές απαντήσεις.

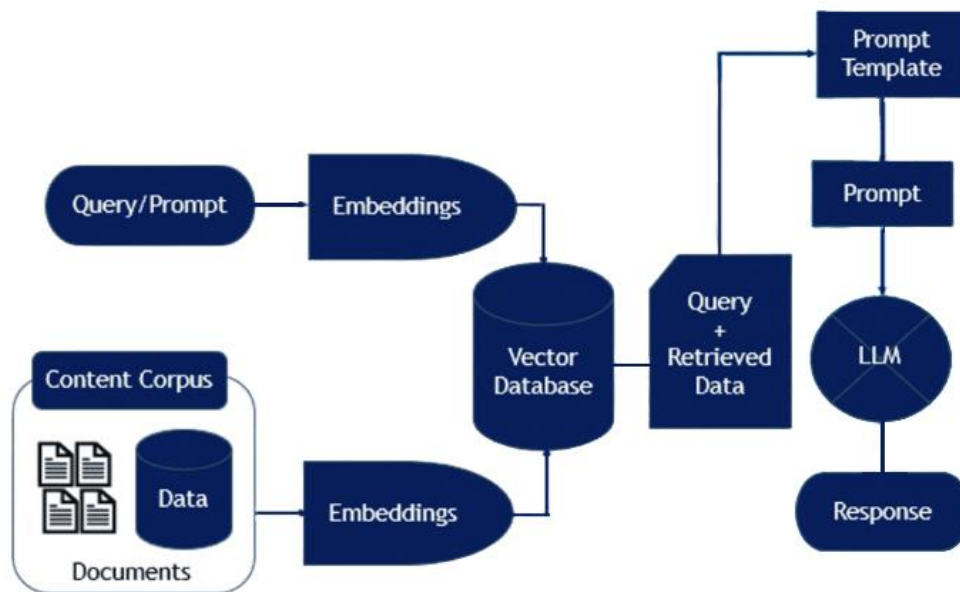
```

1. def load_qa_chain():
2.     global _qa_chain
3.     if _qa_chain is None:
4.         logger.info("Initializing QA chain...")
5.         embedding = HuggingFaceEmbeddings(model_name="paraphrase-MiniLM-L3-v2")
6.         vectorstore = Chroma(persist_directory="./memory", embedding_function=embedding)
7.
8.         llm = ChatOpenAI(
9.             model_name="deepseek/deepseek-chat",
10.            ...
11.        )
12.
13.        _qa_chain = ConversationalRetrievalChain.from_llm(
14.            llm=llm,
15.            retriever=vectorstore.as_retriever(),
16.            memory=ConversationBufferMemory(memory_key="chat_history", return_messages=True)
17.        )
18.        logger.info("QA chain initialized.")
19.    return _qa_chain

```

Κώδικας 4: Retriever

Για να θυμάται το ιστορικό της συζήτησης με τον κάθε χρήστη, το σύστημα χρησιμοποιεί ένα μηχανισμό μνήμης (ConversationBufferMemory). Αυτή η μνήμη αποθηκεύεται τοπικά για κάθε χρήστη και φορτώνεται σε κάθε αλληλεπίδραση, επιτρέποντας στο chatbot να έχει μια συνεκτική και συνεχή συζήτηση που βασίζεται σε προηγούμενες ερωτήσεις και απαντήσεις. Το ιστορικό της συζήτησης (qa_chain.memory) για τον χρήστη αποθηκεύεται τοπικά ως ένα pickle αρχείο (./memory/{userId}.pkl) μετά από κάθε αλληλεπίδραση. Αυτό επιτρέπει τη συνέχιση της συζήτησης σε επόμενα sessions.



Εικόνα 47: Αρχιτεκτονική του Chatbot με RAG και Contextual Prompt Injection

Το DeepSeek χρησιμοποιήθηκε μέσω του OpenRouter. Το openai_api_base παρακάμπτει τα official OpenAI endpoints και δείχνει στο OpenRouter. Τα HuggingFaceEmbeddings με το paraphrase-MiniLM-L3-v2 μοντέλο χρησιμοποιούνται για να μετατρέψουν τα έγγραφα βάσης γνώσης σε διανύσματα και να τα αποθηκεύσουν σε μια τοπική βάση Chroma (persist_directory="./memory"). Αυτό καθιστά το σύστημα αυτόνομο και φθινό στη λειτουργία, χωρίς την ανάγκη για server vector database. Η μνήμη της συζήτησης δεν είναι απλώς μια προσωρινή λίστα. Είναι ένα ConversationBufferMemory αντικείμενο του LangChain που διατηρεί την ακολουθία των μηνυμάτων (ανθρώπου και AI) ως μια λίστα από αντικείμενα μηνυμάτων (e.g., HumanMessage, AIMessage). Αυτά μπορούν να σειριοποιηθούν (με pickle) και να φορτωθούν ξανά. Η load_qa_chain() χρησιμοποιεί ένα global variable (_qa_chain) για να φορτώνει την αλυσίδα μόνο μία φορά κατά τη διάρκεια ζωής του server (singleton pattern). Έτσι βελτιώνεται η απόδοση αποφεύγοντας τις

επαναφορτώσεις των βαρέων μοντέλων και της βάσης δεδομένων για κάθε αίτημα. Ο κώδικας περιλαμβάνει `logger.info()` και `logger.exception()` statements, τα οποία παρέχουν ορατότητα σε κάθε στάδιο της διεργασίας (semantic score, φόρτωση προφίλ, κ.λπ.) και βοηθούν στον εντοπισμό σφαλμάτων.

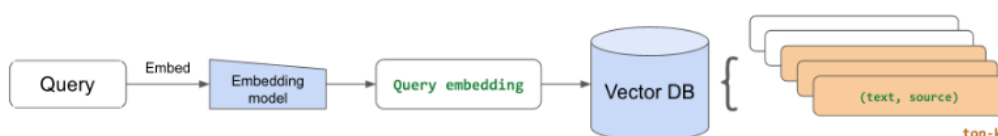
6.3 Ανάκτηση Γνώσης με RAG

Για να βελτιωθεί η ποιότητα των απαντήσεων του chatbot και να προσφέρεται εξατομικευμένη, σχετική υποστήριξη στους χρήστες, εφαρμόστηκε η τεχνική RAG (Retrieval-Augmented Generation). Η προσέγγιση RAG ενισχύει την απάντηση του LLM με πληροφορίες που ανακτώνται σε πραγματικό χρόνο από μια προ επεξεργασμένη βάση γνώσης αντί να βασίζεται αποκλειστικά στη μνήμη του μοντέλου. Στο συγκεκριμένο σύστημα, η βάση γνώσης αποτελείται από άρθρα, επιστημονικές μελέτες και εγχειρίδια σχετικά με τη διακοπή του καπνίσματος, την ψυχολογική υποστήριξη, στρατηγικές αντιμετώπισης του άγχους, καθώς και απαντήσεις σε συχνές απορίες. Αυτά τα κείμενα οργανώθηκαν και ενσωματώθηκαν ως documents στο Langchain framework.

Για την επεξεργασία αυτών των δεδομένων χρησιμοποιήθηκαν οι παρακάτω μηχανισμοί. Τα αρχεία γνώσης χωρίστηκαν σε chunks (π.χ. 500 tokens), για κάθε chunk δημιουργήθηκε ένα vector embedding μέσω μοντέλων όπως HuggingFaceEmbeddings και όλα τα embeddings αποθηκεύτηκαν σε έναν vector store (ChromaDB). Κατά την εκτέλεση του backend όταν ο χρήστης στέλνει ένα ερώτημα, το Langchain αναλύει την πρόθεση του. Έπειτα, το ερώτημα μετατρέπεται σε embedding, το σύστημα αναζητά τα πιο σχετικά chunks από το vector store και τέλος το LLM συνθέτει απάντηση βασισμένη τόσο στο ερώτημα όσο και στα ανακτημένα κομμάτια γνώσης. Η λύση αυτή είναι ιδιαίτερα χρήσιμη όταν το chatbot καλείται να απαντήσει σε επιστημονικά ακριβείς ερωτήσεις ή να παραθέσει συγκεκριμένες στρατηγικές βασισμένες σε σύγχρονη βιβλιογραφία, χωρίς να κινδυνεύει από ψευδείς απαντήσεις των LLM.

Ανάκτηση και Παραγωγή Απάντησης (Retrieval και Generation):

Αρχικά φορτώνεται μια βιβλιοθήκη εγγράφων (vector store) από το `./memory` directory χρησιμοποιώντας τα HuggingFaceEmbeddings και τη Chroma DB. Έτσι, δημιουργείται μια ConversationalRetrievalChain από το LangChain. Αυτή η αλυσίδα χρησιμοποιεί το ChatOpenAI LLM, με ρύθμιση για το `deepseek/deepseek-chat` μοντέλο, προσαρμοσμένο για χρήση μέσω του OpenRouter API. Έπειτα, ενσωματώνεται ένα ConversationBufferMemory για να διατηρεί το ιστορικό της συζήτησης για τον συγκεκριμένο χρήστη. Επιπλέον πραγματοποιείται αναδρομική ανάκτηση (retrieval) από τη vector store βάση του prompt και του ιστορικού και στη συνέχεια να περνούν τα σχετικά αποσπάσματα στο LLM για να παραγάγει την τελική απάντηση.



Εικόνα 48: Vector DB Retriever

Εφεδρικός Μηχανισμός (Web Search Fallback):

Αν η αρχική απάντηση από το LLM είναι πολύ σύντομη (< 20 χαρακτήρες) ή περιέχει φράσεις όπως *i don't know*, ενεργοποιείται ένας εφεδρικός μηχανισμός. Το σύστημα καλεί τη `query_web_search_api(message)`, η οποία στέλνει το ερώτημα στο Serper API (μια υπηρεσία αναζήτησης Google) για να ανακτήσει snippets από αποτελέσματα αναζήτησης. Τα snippets αυτά τροφοδοτούνται ξανά στο LLM (αυτή τη φορά απευθείας με `.predict()`, χωρίς την αλυσίδα ανάκτησης) μαζί με το προσωπικό context, ώστε να δημιουργηθεί μια νέα, πιο ενημερωμένη απάντηση.

```
1. def query_web_search_api(query: str):
2.     api_key = os.getenv("SERPER_API_KEY")
13.     ...
14.     try:
15.         response = requests.post(url, headers=headers, json=data)
16.         if response.status_code == 200:
17.             json_data = response.json()
18.             snippets = [result.get("snippet", "") for result in json_data.get("organic", [])]
19.             return " ".join(snippets)
20.         else:
21.             logger.error(f"Serper API error {response.status_code}: {response.text}")
22.     except Exception:
23.         logger.exception("Web search API exception")
25.     return None
```

Κώδικας 5: Fallback Search

Τέλος, πρέπει να σημειωθεί ότι οι απαντήσεις του LLM είναι σε Markdown μορφή, οπότε στην τελική απάντηση στο UI, το κείμενο μετατράπηκε σε κανονικό format με όλες τις σωστές μορφολογίες (bold, italics κτλ.).

Διαχείριση Μνήμης Συνομιλίας

Το `user_memory.py` διαχειρίζεται την μνήμη συνομιλίας μεταξύ του χρήστη και του LLM, επιτρέποντας στο chatbot να έχει αντίληψη του διαλόγου και να απαντά με βάση το ιστορικό. Χρησιμοποιείται για την αποθήκευση όλων των μηνυμάτων της συνομιλίας (user και AI). Όταν γίνεται μια κλήση API στο chatbot, επιστρέφει όλο το ιστορικό συνομιλίας, χρήσιμο για είσοδο στο LLM (π.χ. κατά το RetrievalQA). Η μνήμη υλοποιήθηκε ως ένα σύστημα που συνδυάζει persistent local storage με ένα mechanism περικοπής (pruning) για να διατηρεί τις συζητήσεις εστιασμένες και να ελέγχει το κόστος. Ορίζονται συναρτήσεις για τη φόρτωση (`get_user_memory`) και τη συντήρηση (`prune_memory`) της μνήμης συζήτησης για κάθε χρήστη. Επίσης, η μνήμη αποθηκεύεται τοπικά στο δίσκο για να επιβιώνει σε επανεκκινήσεις του server και «κλαδεύεται» για να αποφεύγεται η υπερχειλίση του πλαισίου (context window) του LLM.

Η μνήμη για κάθε χρήστη αποθηκεύεται σε ένα ξεχωριστό αρχείο στον φάκελο `./memory/` με βάση το `user_id` (`./memory/{user_id}.pkl`). Αυτό επιτρέπει την απομόνωση των συζητήσεων μεταξύ διαφορετικών χρηστών. Για την αποθήκευση και φόρτωση χρησιμοποιείται η βιβλιοθήκη `pickle`. Το `pickle` είναι ένα πρωτόκολλο serialization της Python που μπορεί να μετατρέψει ένα αντικείμενο της μνήμης (που είναι μια σύνθετη δομή δεδομένων) σε μια ακολουθία bytes για εγγραφή στο δίσκο και αντίστροφα. Αυτό είναι απλό και αποτελεσματικό για πρωτότυπα, αλλά μπορεί να θέλει προσοχή σε production (π.χ., ασφάλεια, συμβατότητα εκδόσεων).

Ο τύπος μνήμης που χρησιμοποιείται είναι το `ConversationBufferMemory` του `LangChain`. Αυτό το object δεν είναι μια απλή λίστα, αφού διατηρεί εσωτερικά ένα `ChatMemory` object, το οποίο περιέχει μια λίστα από αντικείμενα `Message` (π.χ., `HumanMessage`, `AIMessage`). Κάθε `Message` περιέχει το περιεχόμενο και συχνά `metadata`. Η παράμετρος `return_messages=True` διασφαλίζει ότι όταν αυτή η μνήμη τροφοδοτηθεί στην αλυσίδα, θα

περάσει η πλήρης λίστα των αντικειμένων μηνυμάτων, η οποία είναι η μορφή που αναμένουν τα Chat Models.

```
1. def prune_memory(memory):
2.     if hasattr(memory, "chat_memory"):
3.         messages = memory.chat_memory.messages
4.         if len(messages) > MAX_MEMORY_LENGTH:
5.             logger.info(f"Pruning memory to last {MAX_MEMORY_LENGTH} messages")
6.             memory.chat_memory.messages = messages[-MAX_MEMORY_LENGTH:]
```

Κώδικας 6: Memory Pruning

Επιπλέον, η συνάρτηση `prune_memory` εφαρμόζει έναν απλό κανόνα περικοπής και διατηρεί μόνο τα τελευταία `MAX_MEMORY_LENGTH` μηνύματα. Ο μηχανισμός λειτουργεί απευθείας πάνω στη λίστα `memory.chat_memory.messages`. Αν το μήκος της λίστας υπερβαίνει το όριο (`MAX_MEMORY_LENGTH = 10`), αντικαθιστά την υπάρχουσα λίστα με μια νέα που περιέχει μόνο τα `MAX_MEMORY_LENGTH` τελευταία στοιχεία (`messages[-MAX_MEMORY_LENGTH:]`). Αυτό είναι κρίσιμα σημαντικό για δύο λόγους. Αρχικά, για την διαχείριση πλαισίου (context window). Τα LLMs έχουν ένα hard limit στο πόσα tokens μπορούν να δεχτούν ως είσοδο (context window). Αν η συζήτηση γίνει πολύ μεγάλη, θα ξεπεραστεί αυτό το όριο. Η περικοπή το αποτρέπει αυτό. Επιπλέον, βοηθάει το μοντέλο να εστιάσει στην πιο πρόσφατη συζήτηση, μειώνοντας τον θόρυβο από πολύ παλιές αλληλεπιδράσεις που μπορεί να μην είναι πλέον σχετικές.

Η συνάρτηση `get_user_memory` πρώτα ελέγχει αν υπάρχει το αρχείο μνήμης για τον συγκεκριμένο χρήστη (`os.path.exists(memory_path)`). Αν υπάρχει, το φορτώνει (`pickle.load(f)`). Έπειτα, εφαρμόζει τον μηχανισμό περικοπής (`prune_memory(memory)`) για να διασφαλίσει ότι είναι σε βιώσιμο μέγεθος και το επιστρέφει. Αν δεν υπάρχει ή αν η φόρτωση αποτύχει (π.χ., corrupt file ή από ασύμβατη έκδοση), η συνάρτηση δημιουργεί και επιστρέφει ένα νέο, κενό αντικείμενο `ConversationBufferMemory`. Αυτό εξασφαλίζει ότι το chatbot έχει πάντα μια έγκυρη μνήμη με την οποία μπορεί να δουλέψει.

Αυτό το αρχείο είναι κρίσιμο για τη φυσική, συνεχόμενη ροή της συζήτησης, επιτρέποντας στην εφαρμογή να συμπεριφέρεται όχι ως μηχανή ερωταπαντήσεων, αλλά ως διαλογικό bot.

Εισαγωγή και Επεξεργασία Πηγών Γνώσης

Το `ingest_documents.py` είναι υπεύθυνο για τη μετατροπή εξωτερικών εγγράφων (PDF, TXT, DOCX, κ.λπ.) σε vector embeddings και την αποθήκευσή τους στον vector store, ώστε να μπορούν να ανακτηθούν κατά τη διάρκεια της συνομιλίας μέσω RAG. Υλοποιήθηκε ως ένα αυτόνομο script προ επεξεργασίας που φορτώνει γνώση στο σύστημα RAG του chatbot. Σκοπός του είναι να μετατρέψει ένα σύνολο από αρχεία PDF σχετικά με την κατάργηση του καπνίσματος σε μια μορφή που μπορεί να χρησιμοποιηθεί για αποτελεσματική ανάκτηση πληροφοριών.



Εικόνα 49: Δημιουργία Vector DB

Η διαδικασία φόρτωσης εγγράφων (loading) ξεκινάει σαρώνοντας έναν προκαθορισμένο φάκελο (`./documents`), για να βρει όλα τα αρχεία με κατάληξη `.pdf`. Τα αρχεία ταξινομούνται αλφαβητικά, για να διασφαλιστεί μια σειρά επεξεργασίας. Για κάθε PDF αρχείο,

χρησιμοποιείται ένας PyPDFLoader από τη βιβλιοθήκη LangChain. Αυτός ο loader διαβάζει το PDF και εξάγει το περιεχόμενό του ανά σελίδα. Κάθε σελίδα μετατρέπεται σε ένα αντικείμενο Document του LangChain, το οποίο περιέχει το κείμενο και metadata (όπως τον αριθμό σελίδας και την πηγή). Όλα τα Document objects από όλα τα PDF συγκεντρώνονται σε μια ενιαία λίστα.

```
1. text_splitter = RecursiveCharacterTextSplitter(chunk_size=1000, chunk_overlap=200)
2. split_docs = text_splitter.split_documents(all_docs)
3.
4. print(f"✅ Split into {len(split_docs)} chunks.")
5.
6. vectorstore = Chroma.from_documents(split_docs, embedding, persist_directory=VECTORSTORE_DIR)
7.
8. print("✅ Documents embedded and stored in Chroma.")
9.
```

Κώδικας 7: Document Splitting

Ο διαχωρισμός κειμένου (splitting) γίνεται, γιατί το κείμενο από τις σελίδες των PDF είναι συνήθως πολύ μεγάλο για να το χειριστεί το μοντέλο (embedding model) και ο μηχανισμός ανάκτησης. Για αυτό το λόγο, χρησιμοποιείται ένας RecursiveCharacterTextSplitter. Αυτός ο splitter χωρίζει τα μεγάλα κείμενα σε μικρότερα τμήματα (chunks). Οι παράμετροι chunk_size=1000 και chunk_overlap=200 σημαίνουν ότι κάθε τμήμα θα έχει περίπου 1000 χαρακτήρες, ενώ δύο διαδοχικά τμήματα θα έχουν επικάλυψη 200 χαρακτήρων. Η επικάλυψη είναι κρίσιμα σημαντική. Εξασφαλίζει ότι η σημασιολογική σύνδεση μεταξύ των ιδεών στα όρια των τμημάτων δεν χάνεται, βελτιώνοντας την ποιότητα της ανάκτησης.

Η δημιουργία vector store (embedding και storage) αποτελεί το τελευταίο και πιο σημαντικό βήμα και είναι η μετατροπή των τμημάτων κειμένου σε μια μορφή που μπορεί να αναζητηθεί σημασιολογικά. Για αυτό χρησιμοποιείται το μοντέλο paraphrase-MiniLM-L3-v2 μέσω της κλάσης HuggingFaceEmbeddings. Αυτό το μοντέλο μετατρέπει κάθε τμήμα κειμένου σε μία διανυσματική αναπαράσταση (embedding), όπου σημασιολογικά παρόμοιο κείμενο αντιστοιχεί σε κοντινά διανύσματα. Τα διανύσματα αυτά, μαζί με τα αρχικά τμήματα κειμένου, αποθηκεύονται σε μια τοπική βάση δεδομένων vectorstore χρησιμοποιώντας τη Chroma.from_documents(). Η παράμετρος persist_directory="./memory" οδηγεί τη Chroma να αποθηκεύσει όλα τα δεδομένα (τα διανύσματα και τα κείμενα) στον υποφάκελο ./memory του δίσκου. Αυτή η βάση δεδομένων θα χρησιμοποιηθεί αργότερα από το κύριο chatbot (run_chat_pipeline). Όταν ο χρήστης κάνει μια ερώτηση, η ερώτηση επίσης μετατρέπεται σε embedding και η Chroma βρίσκει και επιστρέφει τα πιο σημασιολογικά παρόμοια τμήματα κειμένου από αυτή την αποθηκευμένη βάση γνώσης. Αυτά τροφοδοτούνται στο LLM ως context για να δημιουργήσει την απάντηση.

Όλα αυτά μαζί ακολουθούν την διαδικασία για να εκτελεστεί το Chatbot σωστά με τα επιθυμητά αποτελέσματα. Το ingest_documents.py εκτελείται μόνο μία φορά για να δημιουργήσει το vectorstore. Έπειτα, το αρχείο αυτό δημιουργεί τη βάση γνώσης του Chatbot. Κατά την πραγματική συνομιλία το ερώτημα του χρήστη γίνεται embedding, γίνεται αναζήτηση για σχετικά chunks, στην συνέχεια οι απαντήσεις δημιουργούνται με LLM και memory και context από τα chunks και τέλος το user_memory.py φροντίζει, ώστε ο διάλογος να έχει συνέχεια και μνήμη.

6.4 Διαχείριση Ειδοποιήσεων

Η λειτουργία ειδοποιήσεων αποτελεί βασικό πυλώνα της εφαρμογής, καθώς προσφέρει συνεχή υποστήριξη και παρακίνηση στον χρήστη μέσω τακτικών υπενθυμίσεων, καθημερινών συμβουλών και επιβραβεύσεων. Για την υλοποίηση, χρησιμοποιήθηκε η βιβλιοθήκη `flutter_local_notifications`

Η υλοποίηση των ειδοποιήσεων βασίζεται σε ένα συνδυασμό τοπικών ειδοποιήσεων (scheduled notifications) και Firebase Cloud Messaging (FCM) για push notifications. Ο κώδικας ορίζει μια κλάση `NotificationProvider` που λειτουργεί ως κεντρικός διαχειριστής. Κατά την αρχικοποίηση, το σύστημα ζητά τα απαραίτητα δικαιώματα από τον χρήστη. Σε συσκευές Android 13 και νεότερες, ζητάει επιπλέον το δικαίωμα `SCHEDULE_EXACT_ALARM` μέσω μιας ειδικής οθόνης ρυθμίσεων του συστήματος. Αυτό είναι απαραίτητο για τον ακριβή προγραμματισμό ειδοποιήσεων. Γίνεται χρήση του `SharedPreferences` για να αποθηκευτεί ότι τα δικαιώματα έχουν ζητηθεί και να μην ενοχλείται ο χρήστης ξανά. Για να αποφευχθεί ο επαναπρογραμματισμός και η επαναποστολή τους, αποθηκεύεται ένα flag στο `SharedPreferences` για κάθε μία που έχει ήδη προγραμματιστεί.

Στην υλοποίηση χρησιμοποιείται το πακέτο `flutter_local_notifications`. Αυτό επιτρέπει την αποστολή τοπικών ειδοποιήσεων σε Android και iOS συσκευές, σε συνδυασμό με το `timezone` για ακριβή χρονισμό. Η ρύθμιση και η αρχικοποίηση της βιβλιοθήκης γίνεται στην εκκίνηση της εφαρμογής (`main.dart`), εξασφαλίζοντας ότι οι ειδοποιήσεις είναι διαθέσιμες άμεσα. Η μέθοδος `zonedSchedule` χρησιμοποιείται με `AndroidScheduleMode.exactAllowWhileIdle` για να εξασφαλιστεί ότι οι ειδοποιήσεις θα ενεργοποιηθούν ακριβώς στην ώρα τους, ακόμα και αν η συσκευή βρίσκεται σε κατάσταση αναμονής.

Το `NotificationProvider` περιέχει μια μέθοδο `resetScheduledNotifications`. Αυτή η μέθοδος είναι κρίσιμη για την εφαρμογή, καθώς επιτρέπει την ακύρωση όλων των υπαρχόντων προγραμματισμένων ειδοποιήσεων και τον επαναπρογραμματισμό τους από την αρχή. Αυτό είναι απαραίτητο σε περίπτωση που ο χρήστης ενημερώσει την ημερομηνία που έκοψε το κάπνισμα, διαγράψει και ξαναδημιουργήσει το προφίλ του ή για την αντιμετώπιση σφαλμάτων.

Υπάρχει και ο μηχανισμός `checkAttributeNotifications` που ελέγχει milestones που βασίζονται σε χαρακτηριστικά, όπως το ποσό των χρημάτων που έχει εξοικονομήσει ο χρήστης και ενεργοποιεί σχετικές ειδοποιήσεις. Αποφεύγει την επανάληψη της ίδιας ειδοποίησης χρησιμοποιώντας flags. Το provider ακούει για εισερχόμενη push notifications όταν η εφαρμογή είναι στο foreground (`FirebaseMessaging.onMessage`) και τις εμφανίζει αμέσως στον χρήστη χρησιμοποιώντας τοπικές ειδοποιήσεις.

Οι ειδοποιήσεις λειτουργούν σωστά σε όλες τις Android συσκευές και προσαρμόζονται με βάση την κατάσταση της εφαρμογής (ανοιχτή ή κλειστή), ενώ κατάλληλος χειρισμός επιτρέπει μελλοντικά και ειδοποιήσεις push μέσω Firebase Cloud Messaging.

6.5 Υλοποίηση Επιτευγμάτων και Πίνακα Κατάταξης

Επιτεύγματα

Η προσέγγιση που ακολουθήθηκε για την τεχνική υλοποίηση βασίστηκε στον Provider. Συγκεκριμένα, δημιουργήθηκε μία κλάση `AchievementsProvider`, η οποία διατηρεί τη λίστα επιτευγμάτων ανά χρήστη, ενημερώνει την πρόοδό τους και αποθηκεύει την κατάσταση σε `Firestore`. Αυτό επιτρέπει το συγχρονισμό της προόδου σε πολλαπλές συσκευές. Το σύστημα επιτευγμάτων υλοποιήθηκε ως ένα `ChangeNotifier` που λειτουργεί σε συνδυασμό με το `UserProvider`. Η βασική του λογική είναι να παρακολουθεί τις αλλαγές στα δεδομένα του χρήστη (όπως οι ημέρες χωρίς τσιγάρο, τα χρήματα που εξοικονομήθηκαν, κ.λπ.) και να υπολογίζει αυτόματα την πρόοδο προς τα διάφορα milestones, να ανιχνεύει ολοκληρωμένα επιτεύγματα και να απονέμει ανταμοιβές (XP).

Ο πυρήνας του συστήματος είναι η μέθοδος `_generateAchievements`. Αυτή η μέθοδος δημιουργεί μια λίστα με όλα τα πιθανά επιτεύγματα. Κάθε `Achievement` ορίζεται από έναν τίτλο, μια περιγραφή, μια εικόνα, την τρέχουσα πρόοδο (μια τιμή από 0.0 έως 1.0), τα XP που απονέμονται και μια κατάσταση (`isAwarded`) που δείχνει αν το επίτευγμα έχει ήδη απονεμηθεί. Η πρόοδος υπολογίζεται δυναμικά βασισμένη στα πραγματικά δεδομένα του χρήστη από το `UserProvider`.

Η διαδικασία ανίχνευσης νέων ολοκληρωμένων επιτευγμάτων γίνεται στην μέθοδο `_calculateAchievements`. Αυτή η μέθοδος καλείται αυτόματα όταν τα δεδομένα του χρήστη αλλάζουν, χρησιμοποιώντας ένα μηχανισμό `debounce` (καθυστέρηση 30 δευτερολέπτων) για να αποφύγει υπερβολικούς υπολογισμούς. Όταν καλείται δημιουργεί μια νέα λίστα επιτευγμάτων με τα πιο πρόσφατα δεδομένα. Έπειτα, εντοπίζει επιτεύγματα των οποίων η πρόοδος είναι ≥ 1.0 αλλά δεν έχουν γίνει `flagged` ως `isAwarded` ούτε στο τοπικό `Set` ούτε στο `UserProvider`. Για κάθε νέο επίτευγμα, προσθέτει τον τίτλο του σε μια λίστα, υπολογίζει το συνολικό XP και ενημερώνει με μια μοναδική `batch operation` τη βάση δεδομένων. Έπειτα, προσθέτει τα επιτεύγματα στη λίστα των ολοκληρωμένων του χρήστη και πιστώνει τα XP. Τέλος, ενημερώνει και το τοπικό `UserProvider` για την αλλαγή στα XP και στα ολοκληρωμένα επιτεύγματα, διασφαλίζοντας συγχρονισμό.

Το `AchievementsProvider` φορτώνει αρχικά την λίστα των ήδη ολοκληρωμένων επιτευγμάτων από το `Firestore` (`_loadAwardedAchievements`). Αυτή η λίστα αποθηκεύεται σε ένα τοπικό `Set<String>` `_awardedAchievements` και χρησιμοποιείται μαζί με την αντίστοιχη λίστα στο `UserProvider` για να καθορίσει την τιμή του `isAwarded` για κάθε επίτευγμα, αποφεύγοντας διπλότυπα. Η κλάση χρησιμοποιεί ένα `Lock` (`_calculationLock`) για να διασφαλίσει ότι ο υπολογισμός των επιτευγμάτων δεν θα τρέξει παράλληλα πολλές φορές, αποτρέποντας `race conditions`. Περιέχει μεθόδους για επαναφορά (`resetAchievements`) όλων των επιτευγμάτων, που καθαρίζει τόσο το τοπικό state όσο και τα δεδομένα στη βάση.

Πίνακας Κατάταξης

Παράλληλα, το σύστημα κατάταξης (`leaderboard`) λειτουργεί με βάση τους πόντους (XP) που συγκεντρώνει κάθε χρήστης καθώς επιτυγχάνει στόχους. Οι χρήστες κατατάσσονται σύμφωνα με το επίπεδο τους (`Rank`), το οποίο ανανεώνεται δυναμικά. Τεχνικά, το `leaderboard` υλοποιήθηκε μέσω της κλάσης `LeaderboardProvider`. Αυτή φορτώνει δεδομένα (XP και `rank`) από το `Firestore`, ταξινομεί τους χρήστες και εμφανίζει τους κορυφαίους δέκα. Αν ο τρέχων χρήστης δεν περιλαμβάνεται στους πρώτους δέκα, εμφανίζεται σε ξεχωριστή γραμμή στο κάτω μέρος της λίστας ώστε να βλέπει τη δική του πρόοδο και να ενισχύεται η εμπλοκή του με την εφαρμογή.

Το υποσύστημα υλοποιήθηκε ως ένα απλό και αποτελεσματικό σύστημα κατάταξης που βασίζεται στο υπολογισμένο `userRank` κάθε χρήστη. Αρχικά, φορτώνει όλους τους χρήστες από τη βάση δεδομένων, τους φιλτράρει, τους ταξινομεί και επιλέγει τους κορυφαίους για προβολή. Πριν από την ταξινόμηση, εφαρμόζεται ένα φίλτρο στους χρήστες. Ο κώδικας ελέγχει αν ο τρέχων χρήστης είναι νέος (έχει λιγότερες από 40 ημέρες απεξάρτησης) και φιλτράρει τη συνολική λίστα ώστε να περιλαμβάνει μόνο άλλους νέους χρήστες (`userDays < 40`). Αυτό δημιουργεί μια πιο συγκρίσιμη και δίκαιη ομάδα.

Ένας σημαντικός μηχανισμός είναι η διασφάλιση ότι ο τρέχων χρήστης θα εμφανίζεται πάντα στο `leaderboard`, ακόμα και αν η βαθμολογία του (`userRank`) δεν τον τοποθετεί στην πρώτη δεκάδα. Αν μετά τη δημιουργία της λίστας των κορυφαίων 10 διαπιστωθεί ότι ο τρέχων χρήστης δεν βρίσκεται μέσα σε αυτήν (`!isInTop10`), τότε προστίθεται στο τέλος της λίστας `_topUsers`. Η θέση του στην τελική λίστα αντιστοιχεί στην πραγματική του θέση στην ολική ταξινομημένη λίστα, δίνοντας μια ξεκάθαρη εικόνα της απόδοσής του έναντι όλων των άλλων.

6.6 Παιχνίδια, Αναπνοές, Συμβουλές

Mini-Games

Η προσέγγιση των `mini-games` έχει στόχο να προσφέρει έναν απλό, ψυχαγωγικό μηχανισμό, να δράσει ως εργαλείο απόσπασης προσοχής όταν ο χρήστης βιώνει έντονα στερητικά συμπτώματα. Τα `mini-games` είναι μικρά 2D παιχνίδια, όπως το φιδάκι το `Pac Man`, απλά `reflex-based` παιχνίδια, τα οποία έχουν ενσωματωθεί ως αυτόνομες οθόνες με εύκολη πρόσβαση από την αρχική σελίδα της εφαρμογής. Η υλοποίησή τους έγινε με χρήση `Flutter widgets` και `custom rendering`, καθώς και με απευθείας ενσωμάτωση ήδη υπαρχουσών υλοποιήσεων που βρίσκονται στο διαδίκτυο. Τα παιχνίδια αυτά έχουν περιορισμένη διάρκεια ή παρτίδες ώστε να μην κουράζουν και ενεργοποιούνται είτε αυτόβουλα από τον χρήστη.

Breathing Techniques

Η οθόνη αναπνευστικών ασκήσεων (`Breathing Techniques`) υλοποιήθηκε για να καθοδηγεί τον χρήστη σε ήρεμες, ρυθμισμένες αναπνοές διάρκειας 2 3 λεπτών. Η υλοποίηση της βασίζεται σε ένα επαναχρησιμοποιήσιμο `widget` (`BreathingExerciseWidget`) το οποίο ελέγχει έναν χρονομετρητή και καθοδηγεί τον χρήστη μέσα από μια προκαθορισμένη ακολουθία φάσεων.

Η λογική του `widget` ορίζεται από την ιδιότητα `phaseDurations`. Αυτή είναι μια λίστα ακεραίων που καθορίζει τη διάρκεια κάθε φάσης της ασκήσεως αναπνοής σε δευτερόλεπτα. Για παράδειγμα, `[4, 7, 8, 0]` σημαίνει: 4 δευτερόλεπτα Εισπνοή, 7 δευτερόλεπτα Κράτηση, 8 δευτερόλεπτα Εκπνοή και 0 δευτερόλεπτα Κράτηση μετά την εκπνοή. Η μηδενική διάρκεια στην τελευταία φάση υποδηλώνει ότι ο κύκλος ξεκινάει άμεσα ξανά από την αρχή μετά την ολοκλήρωση της εκπνοής, δημιουργώντας έναν συνεχή ρυθμό.

Η τεχνική αναπνοής υλοποιήθηκε ως ένα `stateful widget` που χρησιμοποιεί κινούμενες εικόνες, χρονομέτρηση και πολυαισθητηριακά σήματα (ήχος, δόνηση) για να καθοδηγήσει τον χρήστη μέσα από έναν προκαθορισμένο ρυθμό αναπνοής. Βασίζεται σε διακριτές φάσεις (π.χ., εισπνοή, κράτηση, εκπνοή) με `customizable` διάρκεια.

Το widget δέχεται μια λίστα με τις διάρκειες κάθε φάσης (phaseDurations). Για κάθε φάση ένας AnimationController ελέγχει ένα χρονικό διάστημα ίσο με τη διάρκειά της. Μια Tween animation μεταβάλλει το μέγεθος (scale) ενός κύκλου από μια αρχική τιμή σε μια τελική. Αυτή η κίνηση προσομοιώνει οπτικά την αναπνοή, αφού ο κύκλος διευρύνεται κατά την εισπνοή και συρρικνώνεται κατά την εκπνοή. Η μετάβαση από τη μια φάση στην επόμενη είναι αυτόματη. Όταν ο AnimationController ολοκληρώνεται (status== AnimationStatus.completed), εκτελείται η μέθοδος _nextPhase που ενημερώνει τον δείκτη της τρέχουσας φάσης και ξεκινάει τον επόμενο κύκλο animation.

Παράλληλα με το animation, ένα Timer με περιοδικότητα 1 δευτερολέπτου ενημερώνει ένα _countdown που εμφανίζεται ως αντίστροφη μέτρηση για τον χρήστη. Για να ενισχυθεί η προσοχή, εμπλουτίστηκε η αλλαγή φάσης με ένα ηχητικό (_playBeep()) και μια σύντομη δόνηση (_vibrate()). Αυτά ειδοποιούν τον χρήστη να αλλάξει τη φάση της αναπνοής του, ακόμα και αν δεν κοιτάζει την οθόνη. Αυτό επιτρέπει στον χρήστη να κλείσει τα μάτια και να επικεντρωθεί αποκλειστικά στη διαδικασία της αναπνοής.

Daily Tips

Η λειτουργία βασίζεται στη μέθοδο _calculateUnlockedTips. Αυτή υπολογίζει τον αριθμό των συμβουλών που έχει ξεκλειδώσει ο χρήστης με βάση τις ημέρες που έχουν περάσει από το quitDate. Ο υπολογισμός γίνεται ως DateTime.now().difference(quitDate).inDays + 1. Το +1 εξασφαλίζει ότι την πρώτη ημέρα ο χρήστης θα έχει ξεκλειδώσει την πρώτη συμβουλή. Ο αριθμός αυτός περιορίζεται, ώστε να μην ξεπερνά το συνολικό πλήθος των διαθέσιμων συμβουλών. Όλες οι συμβουλές φορτώνονται από μια στατική πηγή δεδομένων (MockDataService). Όμως, μόνο ο αριθμός τους που αντιστοιχεί στις ημέρες αποχής θεωρείται ξεκλειδωμένος και είναι ορατός για τον χρήστη.

Ο χρήστης έχει τη δυνατότητα να αξιολογήσει κάθε συμβουλή. Η αξιολόγηση (rating) αποθηκεύεται τοπικά στη συσκευή χρησιμοποιώντας το SharedPreferences. Το κλειδί για την αποθήκευση είναι μοναδικό για κάθε συμβουλή (tipRating_ \${tip.title}). Όταν φορτώνονται οι συμβουλές, η μέθοδος _loadTipRatings παίρνει αυτές τις αξιολογήσεις και τις αντιστοιχεί στο αντίστοιχο Tip. Η λίστα των συμβουλών ταξινομείται βάσει της αξιολόγησης (tipsSortedByRating) για να προβάλλονται οι πιο χρήσιμες συμβουλές πρώτες.

Το DailyTipsProvider ακούει για αλλαγές στα δεδομένα του UserProvider (π.χ., εάν ο χρήστης ενημερώσει την ημερομηνία κατάργησης). Όταν εντοπιστεί μια τέτοια αλλαγή, καλείται ξανά η _calculateUnlockedTips, γεγονός που εγγυάται ότι ο αριθμός των ξεκλειδωμένων συμβουλών είναι πάντα συγχρονισμένος με την τρέχουσα πρόοδο του χρήστη.

6.7 Φίλοι, Συνομιλίες, Καταγραφές και Λοιπά

Κάθε χρήστη έχει τη δυνατότητα να δημιουργήσει ένα Social Tag, το οποίο είναι ένας τετραψήφιος κωδικός (π.χ. #1234). Αυτό το tag λειτουργεί ως το δημόσιο αναγνωριστικό του προφίλ. Για να προσθέσει κάποιος έναν φίλο, πρέπει να γνωρίζει και να εισάγει ακριβώς και τα δύο στοιχεία (username και social tag) του άλλου χρήστη.

Η διαδικασία προσθήκης φίλου είναι αμοιβαία. Όταν ο χρήστης Α βρει και προσπαθήσει να προσθέσει τον χρήστη Β χρησιμοποιώντας το σωστό username και social tag, το σύστημα εκτελεί μια συναλλαγή (transaction) στη βάση δεδομένων Firestore. Αυτή η

συναλλαγή προσθέτει τον χαρακτηριστικό αναγνωριστικό (UID) του χρήστη B στη λίστα friendsList του χρήστη A και ταυτόχρονα προσθέτει τον UID του χρήστη A στη λίστα friendsList του χρήστη B. Αυτό δημιουργεί μια αμοιβαία σύνδεση και οι δύο χρήστες εμφανίζονται στις λίστες φίλων ο ένας του άλλου.

Η οθόνη Friends χρησιμοποιεί έναν διπλό StreamBuilder για να εμφανίζει πραγματικό χρόνο την τρέχουσα λίστα φίλων. Ο πρώτος stream παρακολουθεί το έγγραφο του τρέχοντα χρήστη για να πάρει τη λίστα των UIDs των φίλων του (friendsList). Ο δεύτερος stream, μόλις η πρώτη λίστα είναι διαθέσιμη, πραγματοποιεί ένα whereIn ερώτημα στη συλλογή users για να φέρει τα πλήρη προφίλ όλων των χρηστών των οποίων τα UID βρίσκονται στη λίστα φίλων. Αυτό επιτρέπει την προβολή πληροφοριών όπως το username, ο rank, οι ημέρες απεξάρτησης και τα χρήματα που εξοικονομήθηκαν για κάθε φίλο.

Community Chat

Το community chat υλοποιήθηκε ως ένα απλό και αποτελεσματικό σύστημα συλλογικής συζήτησης όπου όλοι οι χρήστες της εφαρμογής μπορούν να συμμετέχουν και να βλέπουν τα μηνύματα όλων των άλλων σε πραγματικό χρόνο.

Η εφαρμογή δεν χρησιμοποιεί μια ενιαία συλλογή, αλλά ακολουθεί μια ιεραρχία. Μια συλλογή global_chat_rooms, όπου κάθε έγγραφο αντιπροσωπεύει ένα δωμάτιο. Κάθε δωμάτιο έχει την δική του υποσυλλογή messages (για τα μηνύματα) και μια υποσυλλογή members (για τα μέλη του δωματίου).

Η μέθοδος sendGlobalChatMessage είναι υπεύθυνη για την αποστολή των μηνυμάτων. Δέχεται το αναγνωριστικό και το όνομα του χρήστη που στέλνει, καθώς και το κείμενο του μηνύματος και δημιουργεί ένα νέο έγγραφο στη συλλογή global_chat. Αυτό το έγγραφο περιέχει τα πεδία senderId, senderName, message και ένα timestamp.

Έπειτα, η μέθοδος getGlobalChatMessagesStream επιστρέφει ένα QuerySnapshot stream. Αυτό το stream πάει στη συλλογή global_chat, ταξινομεί τα μηνύματα κατά το timestamp (σε φθίνουσα σειρά για να εμφανίζονται τα νεότερα από κάτω) και παρέχει μια ροή σε πραγματικό χρόνο. Το StreamBuilder στο Flutter ακούει αυτή την ροή και επανασχεδιάζει αυτόματα τη λίστα κάθε φορά που προστίθεται ένα νέο μήνυμα.

Με τη μέθοδος assignUserToChatRoom καθορίζεται σε ποιο δωμάτιο θα μπει ο χρήστης. Πρώτα ελέγχει αν ο χρήστης είναι ήδη μέλος σε οποιοδήποτε από τα υπάρχοντα δωμάτια. Αν ναι τον βάζει σε αυτό. Αν ο χρήστης είναι νέος, ψάχνει να βρει το πρώτο διαθέσιμο δωμάτιο που έχει λιγότερα από 10 μέλη (membersSnapshot.size < 10). Ο στόχος είναι να γεμίσει πρώτα τα υπάρχοντα δωμάτια. Μόνο αν όλα τα υπάρχοντα δωμάτια έχουν 10 μέλη, δημιουργεί ένα νέο δωμάτιο και προσθέτει εκεί τον χρήστη.

Όταν ένας χρήστης στέλνει ένα μήνυμα (sendGlobalChatMessage), αυτό αποθηκεύεται αποκλειστικά στην υποσυλλογή messages του δωματίου στο οποίο έχει εκχωρηθεί. Αντίστοιχα, όταν ακούει για μηνύματα (getGlobalChatMessagesStream), ακούει μόνο στο stream του συγκεκριμένου δωματίου του. Αυτό σημαίνει ότι ο χρήστης βλέπει μόνο τα μηνύματα των άλλων χρηστών που βρίσκονται στο ίδιο δωμάτιο μαζί του, όχι όλης της πλατφόρμας. Τέλος, τα μηνύματα εμφανίζονται σε μια ListView.builder.

Craving Logs

Τα δεδομένα των craving logs προέρχονται από μια συλλογή στο Firestore που ονομάζεται `craving_logs`. Κάθε έγγραφο σε αυτή τη συλλογή αντιστοιχεί σε έναν συγκεκριμένο χρήστη μέσω του πεδίου `userId`. Αυτό εξασφαλίζει ότι κάθε χρήστης βλέπει αποκλειστικά τα δικά του καταγεγραμμένα logs. Το screen χρησιμοποιεί ένα `StreamBuilder` που ακούει σε πραγματικό χρόνο (`snapshots()`) για ερωτήματα στη συλλογή `craving_logs`. Το ερώτημα φιλτράρει τα έγγραφα ώστε να εμφανίζονται μόνο αυτά που ανήκουν στον τρέχοντα χρήστη (`where('userId', isEqualTo: userId)`). Αυτό σημαίνει ότι κάθε φορά που ο χρήστης προσθέτει ένα νέο log από το `/addDocs` screen, η λίστα ενημερώνεται αμέσως και αυτόματα χωρίς την ανάγκη για `refresh`.

Δημιουργία Λογαριασμού

Η δημιουργία λογαριασμού έγινε με δύο επιλογές, απλή εγγραφή με email και κωδικό ή είσοδο με Google μέσω του `firebase_auth`. Για την εγγραφή, χρησιμοποιείται η μέθοδος `signUpWithEmail`, όπου καλεί την αντίστοιχη μέθοδο του Firebase Auth (`createUserWithEmailAndPassword`). Η μέθοδος αυτή δημιουργεί έναν νέο λογαριασμό στο Firebase, με το συγκεκριμένο email και κωδικό. Για τη σύνδεση, η `signInWithEmail` καλεί τη `signInWithEmailAndPassword` του Firebase, η οποία επαληθεύει τα διαπιστευτήρια και επιστρέφει το αντικείμενο του χρήστη. Σε περίπτωση σφάλματος σε οποιαδήποτε από τις δύο μεθόδους, αυτό καταγράφεται στην κονσόλα και το `exception` προωθείται στο UI για να το δει ο χρήστης (π.χ., εμφάνιση μηνύματος λάθους). Το Firebase αναλαμβάνει να επαληθεύσει ότι το email είναι σωστά μορφοποιημένο, ότι ο λογαριασμός υπάρχει και ότι ο κωδικός πρόσβασης ταιριάζει. Η διαχείριση κωδικών (`hashing`, `comparison`) γίνεται εξ ολοκλήρου από το Firebase, γεγονός που αφαιρεί οποιοδήποτε `security burden` από την εφαρμογή. Η επιτυχής σύνδεση επιστρέφει το `User` object.

Η είσοδος με λογαριασμό Google γίνεται με τη μέθοδο `signInWithGoogle`. Έτσι, χρησιμοποιείται το πακέτο `google_sign_in` για να ξεκινήσει τη σύνδεση. Αρχικά, ζητάει από τον χρήστη να επιλέξει ένα λογαριασμό Google. Αν ο χρήστης ακυρώσει τη διαδικασία, η μέθοδος επιστρέφει `null`. Εάν επιλεγεί λογαριασμός, η εφαρμογή λαμβάνει τα διαπιστευτήρια (`access token` και `id token`) από το Google. Στη συνέχεια, αυτά τα token χρησιμοποιούνται για τη δημιουργία ενός `GoogleAuthProvider credential` object. Αυτό τελικά περνάει στη μέθοδο `signInWithCredential` του Firebase Auth για να πραγματοποιηθεί η σύνδεση. Ολόκληρη αυτή η διαδικασία επιτρέπει μια ασφαλή και χωρίς πρόσθετη διαχείριση κωδικών σύνδεση, ενσωματώνοντας άμεσα τον λογαριασμό Google του χρήστη με το Firebase Authentication.

Διαγραφή Λογαριασμού

Η διαγραφή λογαριασμού υλοποιήθηκε ως μια επιλογή στην οθόνη ρυθμίσεων, η οποία εμφανίζει πρώτα ένα μήνυμα επιβεβαίωσης. Η λειτουργία διαγραφής λογαριασμού υλοποιήθηκε ως μια διαδικασία που εξασφαλίζει τη πλήρη και μόνιμη αφαίρεση όλων των δεδομένων του χρήστη τόσο από το Authentication όσο και από το Firestore. Η λογική είναι να μην αφήνει ίχνη του χρήστη στο σύστημα.

Το πρώτο και κρίσιμο βήμα είναι η διαγραφή του εγγράφου του χρήστη από τη συλλογή `users` στη Firestore. Χρησιμοποιείται το `uid` του τρέχοντα χρήστη (που λαμβάνεται είτε από το `UserProvider` είτε απευθείας από το `FirebaseAuth.instance.currentUser`) ως αναγνωριστικό για να βρεθεί και να διαγραφεί το έγγραφο `users/<uid>`. Αυτό το βήμα διαγράφει ολοκληρωτικά το προφίλ και όλα τα δεδομένα που σχετίζονται με τον χρήστη (`quitDate`, `cigarettesPerDay`, `achievements`, κ.λπ.) από τη βάση δεδομένων. Αμέσως μετά, η διαδικασία

προχωρά στη διαγραφή του ίδιου του λογαριασμού από το Firebase Authentication με την κλήση `user.delete()`. Αυτό το βήμα αφαιρεί τον χρήστη από τη λίστα των εξουσιοδοτημένων χρηστών, καταστρέφοντας μόνιμα τα διαπιστευτήριά του (email/κωδικός ή σύνδεση Google) και απενεργοποιώντας οποιαδήποτε μελλοντική σύνδεση.

Μετά την επιτυχή ολοκλήρωση και των δύο βημάτων, ο χρήστης αποσυνδέεται (`signOut()`) για να καθαρίσει οποιαδήποτε τοπική κατάσταση και στη συνέχεια ανακατευθύνεται στην αρχική οθόνη, με το ιστορικό πλοήγησης να διαγράφεται πλήρως. Η διαδικασία περιλαμβάνει `exception handling` για συγκεκριμένα σφάλματα, όπως το `requires-recent-login`, το οποίο απαιτεί από τον χρήστη να επαληθεύσει ξανά την ταυτότητά του πριν από μια τόσο ευαίσθητη ενέργεια, διασφαλίζοντας ένα επιπλέον επίπεδο ασφάλειας.

Restart Progress

Η λειτουργία επανεκκίνησης προόδου (`restart progress`) προσφέρει στον χρήστη τη δυνατότητα να ξεκινήσει από το μηδέν χωρίς να διαγράψει τον λογαριασμό του. Με την επιλογή αυτή, η εφαρμογή καθαρίζει τα πεδία `quitDate`, `xp`, `healthGoals`, `cravingLogs`, `achievements`. Έπειτα, επαναφέρει τον χρήστη στο στάδιο του αρχικού ερωτηματολογίου. Αυτό επιτρέπει στους χρήστες που έχουν υποτροπιάσει να συνεχίσουν χωρίς το ψυχολογικό βάρος του παρελθόντος.

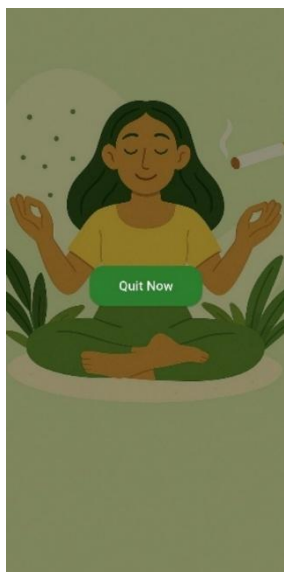
Η διαδικασία ξεκινάει με την κλήση της μεθόδου `_restartApp`. Αυτή η μέθοδος είναι υπεύθυνη για τον συντονισμό τριών κρίσιμων λειτουργιών. Αρχικά, καλεί το `resetScheduledNotifications` στο `NotificationProvider`. Αυτό ακυρώνει όλες τις προγραμματισμένες ειδοποιήσεις (`milestones`, `daily tips`) που σχετίζονταν με την παλιά ημερομηνία διακοπής καπνίσματος (`quit date`) και ετοιμάζει το σύστημα για έναν νέο προγραμματισμό με βάση τη νέα ημερομηνία που θα οριστεί. Έπειτα, καλεί το `resetAchievements` στο `AchievementsProvider`. Αυτή η μέθοδος διαγράφει τοπικά και στη βάση δεδομένων `Firestore` όλα τα `XP` που είχε κερδίσει ο χρήστης, μηδενίζοντας την πρόοδό στα `achievements`. Μετά την ολοκλήρωση των επαναφορών, η εφαρμογή ανακατευθύνεται στην οθόνη ερωτηματολογίου. Έπειτα, διαγράφεται όλο το προηγούμενο ιστορικό πλοήγησης. Αυτό έχει το αποτέλεσμα να ξεκινάει η εφαρμογή από την αρχή, σαν να εγκαθίσταται για πρώτη φορά, αναγκάζοντας τον χρήστη να περάσει ξανά από τη διαδικασία ρύθμισης του προφίλ του, να ορίσει μια νέα ημερομηνία κατάργησης και να ξεκινήσει από το μηδέν.

Η λειτουργία αυτή δεν διαγράφει απλώς μεμονωμένα δεδομένα. Είναι μια ελεγχόμενη επαναφορά ολόκληρου του `state` της εφαρμογής, που αφορά πολλά `providers` (`User`, `Notifications`, `Achievements`) και διασφαλίζει ότι όλα τα μέρη της εφαρμογής είναι συγχρονισμένα και έτοιμα για μια νέα αρχή, χωρίς κανένα αποτύπωμα της προηγούμενης χρήσης.

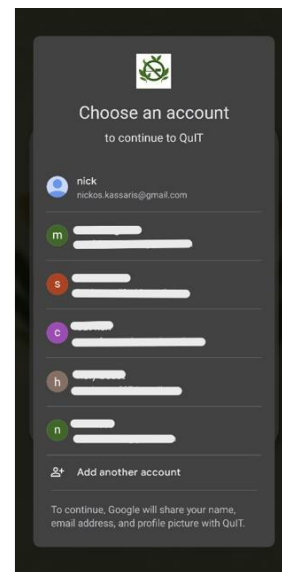
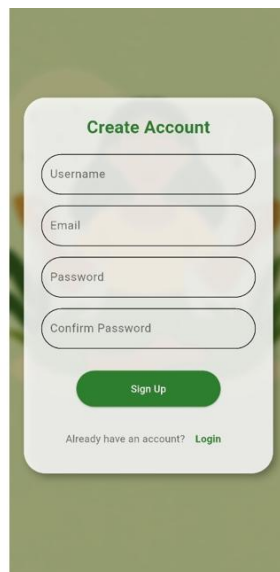
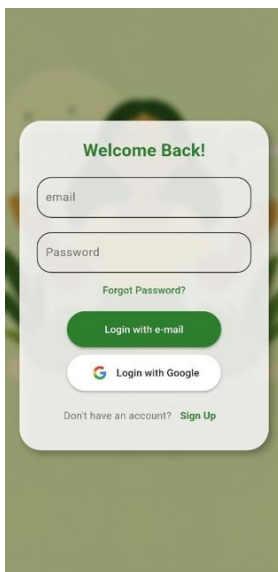
7. Παρουσίαση και Λειτουργία του Συστήματος

Στο παρόν κεφάλαιο παρουσιάζονται οι βασικές οθόνες της εφαρμογής και αναλύεται η λειτουργικότητά τους. Για κάθε οθόνη παρατίθεται περιγραφή του σκοπού της, των δεδομένων που προβάλλει, καθώς και του τρόπου με τον οποίο συμβάλλει στη γενικότερη εμπειρία του χρήστη.

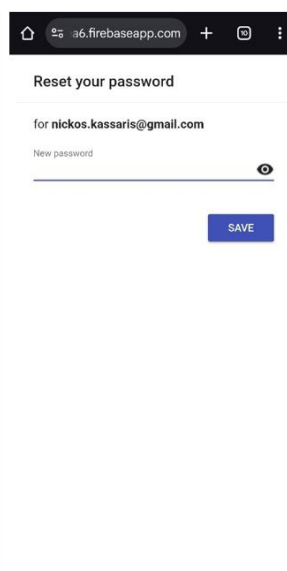
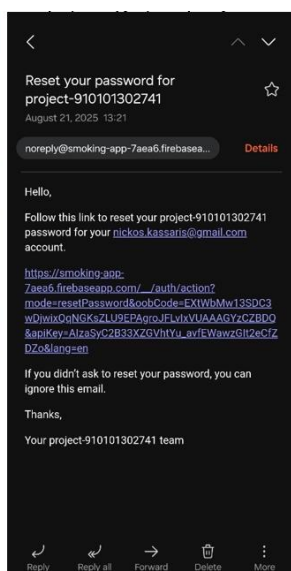
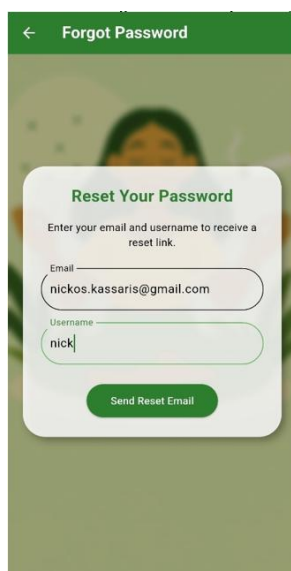
Η πρώτη επαφή του χρήστη με την εφαρμογή πραγματοποιείται μέσω της οθόνης εκκίνησης.



Εικόνα 50: Οθόνη Εισαγωγής



Εικόνα 51: Σύνδεση/Εγγραφή Χρήστη

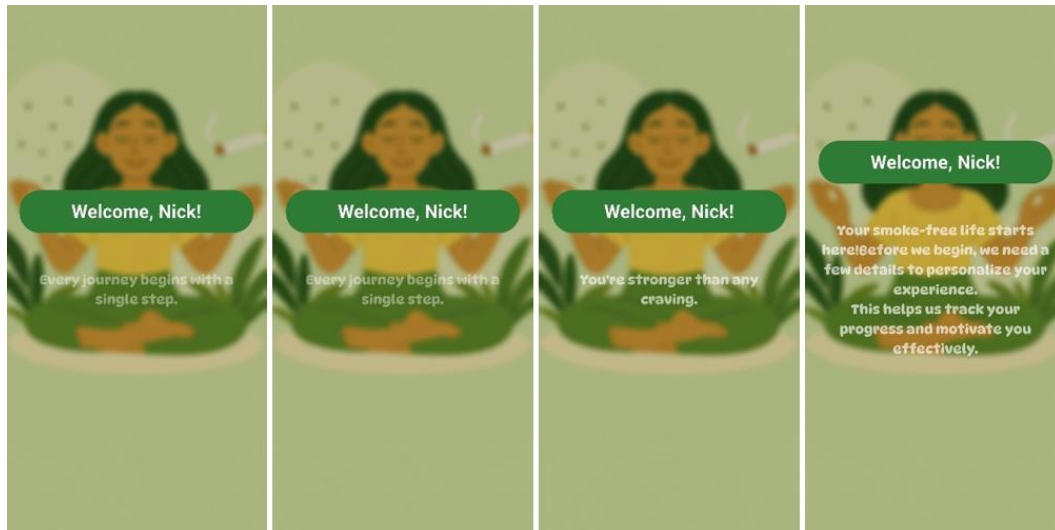


Εικόνα 52: Ανάκτηση Κωδικού Πρόσβασης

Αμέσως μετά, ο χρήστης μεταφέρεται στην οθόνη σύνδεσης/ εγγραφής. Στην περίπτωση σύνδεσης, ο χρήστης εισάγει το email και τον κωδικό πρόσβασης που έχει ήδη καταχωρήσει ή εναλλακτικά χρησιμοποιεί την επιλογή σύνδεσης με Google. Στην περίπτωση εγγραφής, του ζητούνται τα απαραίτητα στοιχεία ταυτοποίησης, ενώ με την ολοκλήρωση της διαδικασίας ο λογαριασμός του δημιουργείται στο backend της εφαρμογής. Η

λειτουργικότητα αυτή αξιοποιεί το Firebase Authentication για την ασφαλή διαχείριση των διαπιστευτηρίων και τη διασφάλιση ομαλής εμπειρίας.

Σε περίπτωση που ο χρήστης ξεχάσει τον κωδικό πρόσβασής του, υπάρχει επιλογή (forgot password?) στην οθόνη σύνδεσης (login) , για την ανάκτηση του με αποστολή συνδέσμου στο email του. Αυτή τη διαδικασία αλλαγής κωδικού πρόσβασης και αποστολή email, την χειρίζεται εξ ολοκλήρου το firebase.



Εικόνα 53: Εισαγωγικά Μηνύματα Έπειτα από την Πρώτη Σύνδεση του Χρήστη

Αφού συνδεθεί για την πρώτη μόνο φορά, ο χρήστης μεταφέρεται στην οθόνη υποδοχής (Welcome Screen). Στο σημείο αυτό εμφανίζονται κάποια ενθαρρυντικά μηνύματα με animations. Αυτό προσφέρει μια σύντομη εισαγωγική εμπειρία πριν την πλήρη φόρτωση του συστήματος. Η διάρκεια εμφάνισης είναι μερικά δευτερόλεπτα, ώστε να δώσει την αίσθηση ταυτότητας και επαγγελματισμού, χωρίς να καθυστερεί υπερβολικά τη μετάβαση στη λειτουργική χρήση.

Εικόνα 54: Ερωτηματολόγιο Πρώτης Σύνδεσης

Με την πρώτη επιτυχημένη είσοδο, ο χρήστης οδηγείται στο ερωτηματολόγιο. Σε αυτήν την οθόνη συλλέγονται κρίσιμες πληροφορίες, όπως ο αριθμός τσιγάρων που καπνίζει

ημερησίως, το οικονομικό κόστος, το φύλο, καθώς και αρκετές άλλες ερωτήσεις ψυχολογικού περιεχομένου. Τα δεδομένα αυτά αποθηκεύονται ώστε να διαμορφώνουν εξατομικευμένες συμβουλές, προσωποποιημένους υπολογισμούς για χρήματα που εξοικονομούνται και τσιγάρα που αποφεύγονται, καθώς και εξατομικευμένες απαντήσεις από το chatbot. Η οθόνη αυτή αποτελεί κεντρικό στοιχείο της αρχιτεκτονικής της εφαρμογής, καθώς θέτει τις βάσεις για την προσωποποίηση της εμπειρίας κάθε χρήστη.

Η ακολουθία των παραπάνω οθονών συνθέτει τη βασική ροή εισόδου στην εφαρμογή. Ο συνδυασμός της εισαγωγικής παρουσίασης, της ασφαλούς ταυτοποίησης και της αρχικής συλλογής δεδομένων συμβάλλει στη δημιουργία μιας ολοκληρωμένης και ομαλής πρώτης επαφής με το σύστημα.

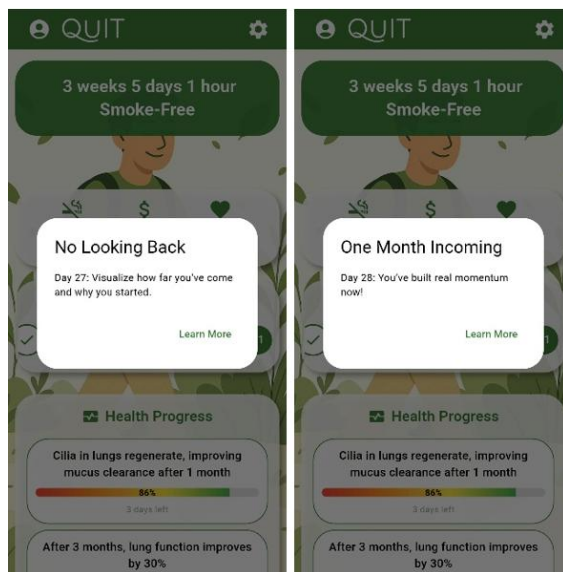


Εικόνα 55: Αρχική Οθόνη

Η αρχική οθόνη λειτουργεί ως κεντρικό σημείο αναφοράς του χρήστη. Σε αυτήν παρουσιάζεται η συνολική πρόοδος του από την ημέρα που επέλεξε να διακόψει το κάπνισμα. Στην αρχική οθόνη φαίνονται οι στόχοι (Daily Goals), οι οποίοι αποτελούν βασικό κίνητρο για τον χρήστη στην προσπάθειά του να διακόψει το κάπνισμα. Η σχεδίαση της οθόνης δίνει προτεραιότητα στην άμεση ενημέρωση σχετικά με την πορεία του προγράμματος, προσφέροντας σαφή και οπτικοποιημένα δεδομένα.

Κεντρικό στοιχείο είναι η εμφάνιση μίας ενότητας με τις τρεις σημαντικότερες βελτιώσεις στην υγεία που δεν έχουν ακόμη ολοκληρωθεί. Για κάθε στόχο παρουσιάζεται μια γραμμή προόδου (progress bar), η οποία ενημερώνεται δυναμικά με βάση την ημερομηνία που έχει δηλώσει ο χρήστης ως ημέρα διακοπής. Έτσι, μπορεί να δει με ευκολία πόσο κοντά βρίσκεται στην επίτευξη του στόχου, όπως για παράδειγμα την αποκατάσταση της κυκλοφορίας του αίματος ή τη βελτίωση της αναπνευστικής λειτουργίας.

Η δομή της οθόνης έχει μελετηθεί ώστε να μην παρουσιάζει υπερβολική πληροφορία, αλλά να παρακινεί μέσω μικρών και μετρήσιμων βημάτων. Κάθε μπάρα προόδου είναι συνδεδεμένη με το σύστημα επιτευγμάτων, ώστε με την ολοκλήρωση ενός στόχου ο χρήστης να ενημερώνεται, ενισχύοντας την αίσθηση προόδου και αυξάνοντας την πιθανότητα διατήρησης του νέου τρόπου ζωής του.



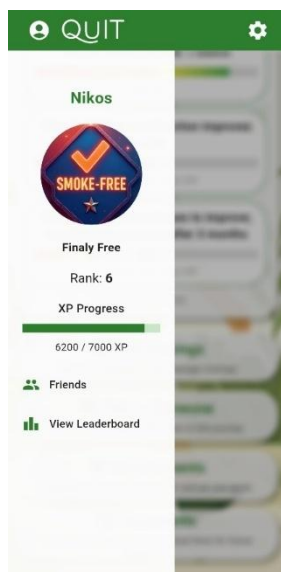
Εικόνα 56: Καθημερινοί Στόχοι (Daily Goals)

Η αρχική οθόνη λειτουργεί επίσης ως κεντρικός κόμβος πλοήγησης, καθώς κάθε daily goal μπορεί να πατηθεί. Πατώντας πάνω σε ένα στόχο, εμφανίζεται ένα παράθυρο με λεπτομέρειες. Κάθε στόχος αντιστοιχεί σε μια μέρα. Όταν η μέρα περάσει ο στόχος δεν μπορεί να πατηθεί πλέον και έχει την κατάλληλη σήμανση ολοκλήρωσης. Η διαδραστικότητα αυτή ενισχύει την αίσθηση ελέγχου και ενθαρρύνει τον χρήστη να παρακολουθεί ενεργά την πρόοδό του. Πατώντας το κουμπί Learn More ο χρήστης ανακατευθύνεται στην οθόνη του chatbot όπου θα του απαντηθούν λεπτομέρειες σχετικά με τον κάθε στόχο της εκάστοτε ημέρας, καθώς και άλλα μηνύματα και απορίες.

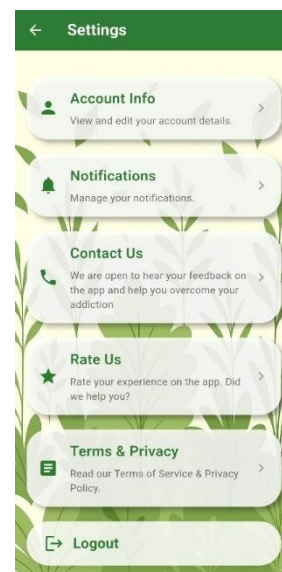
Με τον τρόπο αυτό, η ενσωμάτωση των daily goals στην αρχική οθόνη δεν αποτελεί απλώς μια αισθητική ή πληροφοριακή επιλογή, αλλά έναν κρίσιμο μηχανισμό που παρακινεί τον χρήστη καθημερινά, ενισχύοντας το κίνητρό του και δημιουργώντας μια διαρκή αλληλεπίδραση με την εφαρμογή. Παράλληλα, η αρχική σελίδα λειτουργεί ως πύλη για πρόσβαση σε άλλες βασικές λειτουργίες της εφαρμογής.



Εικόνα 59: Οθόνη Στατιστικών



Εικόνα 58: Drawer Αρχικής Οθόνης



Εικόνα 57: Οθόνη Ρυθμίσεων

Οθόνη στατιστικών που εμφανίζεται όταν ο χρήστης πατάει στα προσωπικά στατιστικά του (money saved, cigarettes avoided, life gained) στην αρχική οθόνη. Drawer που εμφανίζεται όταν ο χρήστης πατάει πάνω αριστερά στο εικονίδιο προφίλ που εμφανίζει την πρόοδο του, επίπεδο (rank), πόντοι (XP), όνομα και εικονίδιο επιπέδου, καθώς και κουμπιά που τον κατευθύνουν στο κοινωνικό μέρος του συστήματος (friends και leaderboard). Τέλος, οθόνη ρυθμίσεων συστήματος πατώντας το κουμπί με το γρανάζι πάνω δεξιά στην αρχική οθόνη. Η οθόνη ρυθμίσεων συγκεντρώνει τις βασικές λειτουργίες διαχείρισης λογαριασμού, όπως διαγραφή ή επανεκκίνηση της προόδου. Επίσης, προσφέρει επιλογές για αξιολόγηση, εμφάνιση όρων χρήσης και δυνατότητα αποσύνδεσης.



Εικόνα 60: Οθόνη Προόδου Στόχων Υγείας

συνοδεύεται από μία μπάρα προόδου, η οποία προσαρμόζεται αυτόματα με βάση την ημερομηνία διακοπής που έχει καταχωρηθεί στο προφίλ. Έτσι, η εφαρμογή υπολογίζει σε πραγματικό χρόνο πόσο κοντά βρίσκεται ο χρήστης στην πλήρη ολοκλήρωση του εκάστοτε στόχου. Επιπλέον, κάθε φορά που εισέρχεται στην οθόνη ο χρήστης γίνεται αυτόματη κύλιση (auto scroll), στο πρώτο μη ολοκληρωμένο επίτευγμα.

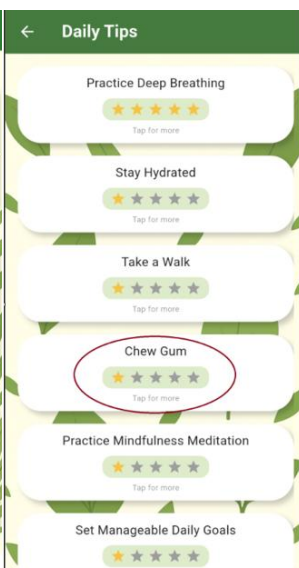
Για παράδειγμα, αν ένας στόχος αφορά την αποκατάσταση της κυκλοφορίας του αίματος μέσα σε τρεις μήνες, η γραμμή προόδου θα δείχνει τι ποσοστό ολοκλήρωσης αντιπροσωπεύουν οι μέρες που έχουν περάσει και πόσο απομένει για την ολοκλήρωση του. Στην πλήρη επίτευξη ενός milestone, η εφαρμογή όχι μόνο ενημερώνει οπτικά με την ολοκλήρωση της μπάρας αλλά ενεργοποιεί και μία σχετική επιβράβευση, όπως μια ειδοποίηση με ένα μήνυμα συγχαρητηρίων που ενισχύει ψυχολογικά τον χρήστη. Η οθόνη δεν περιορίζεται μόνο σε στατικές πληροφορίες, αλλά έχει διαδραστικό χαρακτήρα. Ο χρήστης μπορεί να πατήσει σε κάθε milestone Learn More, για να δει αναλυτική περιγραφή σχετικά με την βελτίωση της υγείας του, την επιστημονική της βάση και τα αναμενόμενα αποτελέσματα (η απάντηση δίνεται από το chatbot και λαμβάνει υπόψιν το προφίλ του χρήστη εξατομικεύοντας την εμπειρία του). Αυτή η προσέγγιση όχι μόνο ενημερώνει αλλά και εκπαιδεύει, ενισχύοντας τη συνειδητοποίηση των οφελών που συνεπάγεται η διακοπή του καπνίσματος.

Η οθόνη Health Progress αποτελεί μία από τις σημαντικότερες ενότητες της εφαρμογής, καθώς αποτυπώνει με τρόπο παραστατικό και δυναμικό τις βελτιώσεις της υγείας του χρήστη μετά τη διακοπή του καπνίσματος. Ο σχεδιασμός της δίνει έμφαση στην επιστημονική τεκμηρίωση και στη σταδιακή ανάδειξη των θετικών αποτελεσμάτων, ώστε ο χρήστης να έχει σαφή εικόνα των οφελών που αποκομίζει σε βάθος χρόνου.

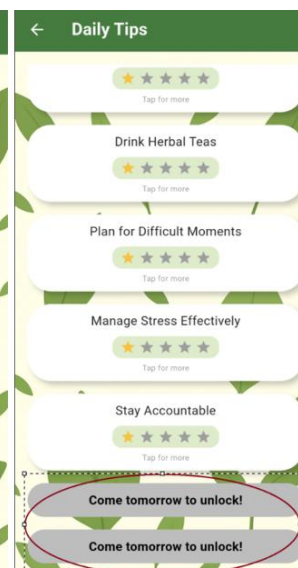
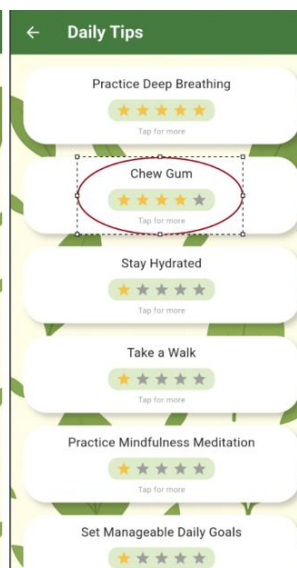
Με την είσοδο στην οθόνη, ο χρήστης αντικρίζει μια λίστα από υγειονομικά milestones, όπως για παράδειγμα την αποκατάσταση της λειτουργίας της καρδιάς, τη βελτίωση της αναπνευστικής ικανότητας ή τη μείωση του κινδύνου εμφάνισης ασθενειών. Κάθε milestone



Εικόνα 61: Beat Cravings Οθόνη

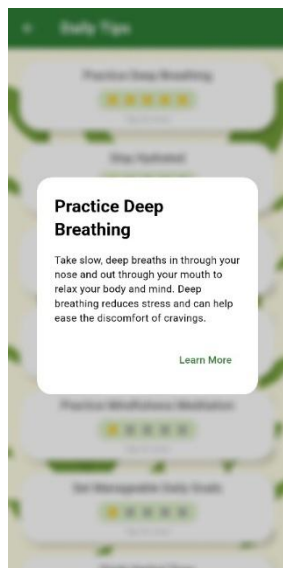


Εικόνα 62: Οθόνη Ημερήσιων Συμβουλών (Daily Tips)



Η οθόνη για την αντιμετώπιση στερητικών συμπτωμάτων (Beat Cravings) προσφέρει άμεσες λύσεις στις στιγμές έντονης επιθυμίας για κάπνισμα. Περιλαμβάνει αναπνευστικές ασκήσεις με οπτικοποιήσεις, καθώς και απλά διαδραστικά παιχνίδια, τα οποία αποσπούν την προσοχή και βοηθούν στη μείωση του άγχους. Η ύπαρξη αυτής της οθόνης αποτελεί κρίσιμο εργαλείο διαχείρισης δύσκολων καταστάσεων.

Με το άνοιγμα της οθόνης Daily Tips, ο χρήστης συναντά μια λίστα με σύντομα αλλά στοχευμένα μηνύματα, τα οποία ξεκλειδώνονται σε καθημερινή βάση. Η σχεδίαση είναι τέτοια ώστε η εφαρμογή να εμφανίζει μόνο ένα νέο tip ανά ημέρα, το οποίο παραμένει διαθέσιμο και προστίθεται στην ήδη υπάρχουσα λίστα. Με αυτόν τον τρόπο διαμορφώνεται ένα προσωπικό αρχείο, στο οποίο ο χρήστης μπορεί να επιστρέφει όποτε χρειάζεται βοήθεια. Τα tips μπορεί να περιλαμβάνουν πρακτικές συμβουλές, όπως τεχνικές αναπνοής για τον έλεγχο των cravings. Σε άλλες περιπτώσεις μπορεί να είναι πιο ψυχολογικά μηνύματα, που στοχεύουν στην τόνωση της αυτοπεποίθησης και στη δημιουργία αίσθησης προόδου, όπως για παράδειγμα η υπενθύμιση ότι ο χρήστης έχει ήδη ξεπεράσει ένα κρίσιμο χρονικό σημείο χωρίς τσιγάρο.



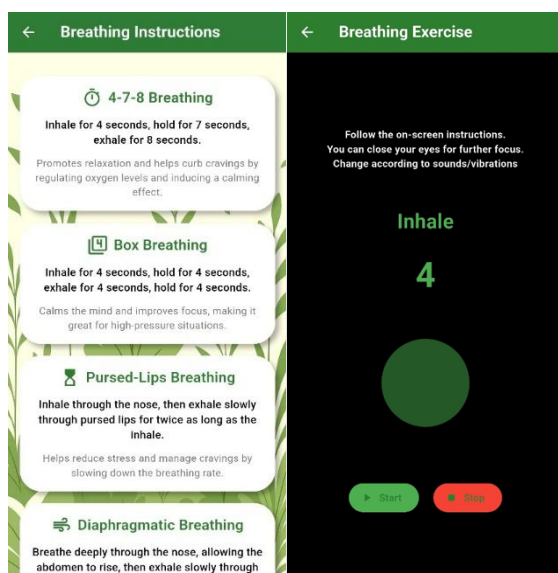
Εικόνα 63: Αναδυόμενο Παράθυρο με Λεπτομέρειες Ημερήσιας Συμβουλής

Η διεπαφή έχει δομηθεί με τρόπο απλό και ευχάριστο, ώστε τα tips να προβάλλονται με έμφαση και να είναι ευανάγνωστα. Κάθε νέο tip συνοδεύεται από ένα σύστημα αξιολόγησης που αντικατοπτρίζει ποσό βοηθητικό είναι για τον χρήστη η εκάστοτε συμβουλή. Όσο μεγαλύτερη η αξιολόγηση, τόσο πιο πάνω εμφανίζεται στην λίστα.

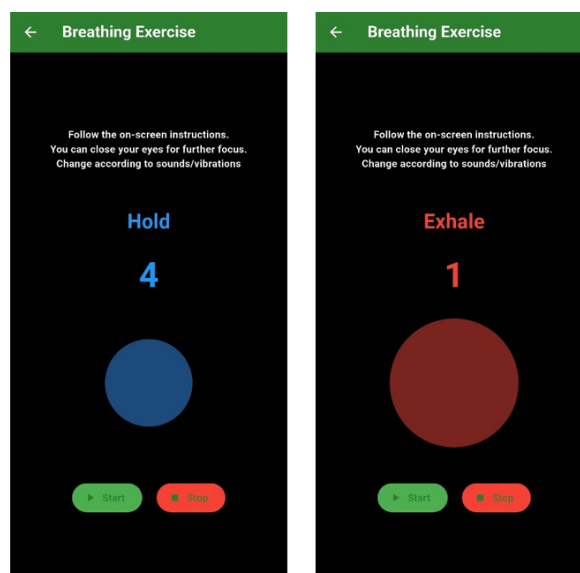
Παράλληλα, αν πατήσει ο χρήστης πάνω σε κάθε συμβουλή, εμφανίζονται αναλυτικές λεπτομέρειες και επεξήγηση της αποτελεσματικότητας του, επιστημονικά τεκμηριωμένα. Πατώντας το κουμπί Learn More, ανακατευθύνεται στο chatbot, όπου του απαντά για την συγκεκριμένη συμβουλή, δίνοντας περεταίρω λεπτομέρειες και επεξηγήσεις, καθώς του απαντά και ό,τι άλλη απορία έχει.

Συνολικά, η οθόνη Daily Tips αποτελεί μια από τις πιο ανθρώπινες πτυχές της εφαρμογής, καθώς προσφέρει καθημερινή συναισθηματική υποστήριξη. Πρόκειται για μια λειτουργία που δεν περιορίζεται μόνο στην παροχή πληροφορίας, αλλά ενδυναμώνει τον χρήστη σε καθημερινό επίπεδο, δημιουργώντας αίσθηση συνέπειας και ενισχύοντας την προσπάθειά του.

Η οθόνη τεχνικών αναπνοής (Breathing Techniques) είναι ένα εργαλείο που βοηθά τον χρήστη να διαχειριστεί τις έντονες επιθυμίες για κάπνισμα μέσα από καθοδηγούμενες ασκήσεις. Ο χρήστης βλέπει στην οθόνη μια οπτικοποιημένη καθοδήγηση. Ένας κύκλος που μεγαλώνει και μικραίνει υποδεικνύοντας πότε πρέπει να εισπνεύσει και πότε να εκπνεύσει. Συνοδεύεται από μια σύντομη δόνηση και έναν ήχο σε κάθε αλλαγή της κατάστασης αναπνοής, ώστε ο χρήστης να μπορεί να τις εκτελεί χωρίς να κοιτά την οθόνη ή ακόμα και με κλειστά τα ματιά του, σε στυλ διαλογισμού, για μεγαλύτερη συγκέντρωση. Η διάρκεια και το μοτίβο της άσκησης είναι ρυθμισμένα ώστε να προκαλούν χαλάρωση, μειώνοντας το στρες που συχνά συνοδεύει την προσπάθεια απεξάρτησης. Παράλληλα, η οθόνη παρέχει και σύντομες οδηγίες κειμένου, ώστε ο χρήστης να κατανοεί τη διαδικασία, ακόμη κι αν δεν έχει προηγούμενη εμπειρία με τεχνικές αναπνοής.

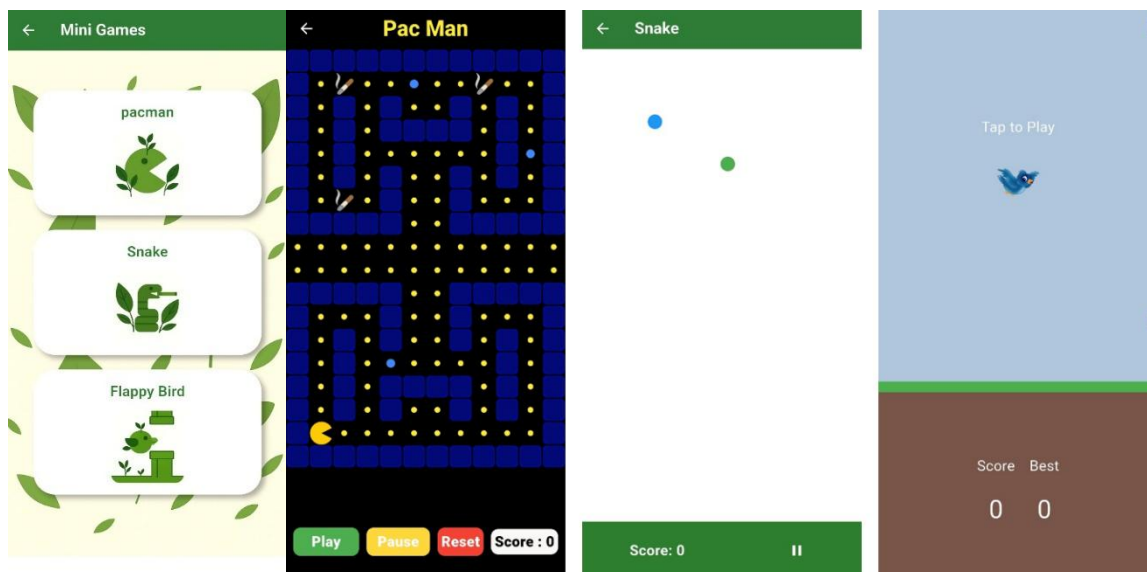


Εικόνα 64: Αναπνευστικές Ασκήσεις



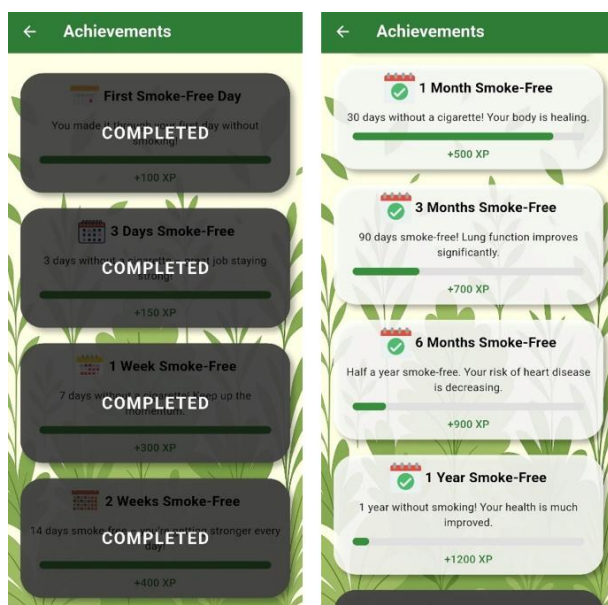
Εικόνα 65: Οθόνη Καθοδήγησης Αναπνευστικών Ασκήσεων

Η οθόνη Mini Games προσφέρει στον χρήστη μια συλλογή από απλά και κλασικά παιχνίδια, όπως Snake, Pac-Man ή Flappy Bird, τα οποία λειτουργούν ως μέσο απόσπασης της προσοχής όταν εμφανίζεται έντονη επιθυμία για κάπνισμα. Ο στόχος δεν είναι η ψυχαγωγία σε βάθος, αλλά η γρήγορη και αποτελεσματική ενασχόληση με κάτι που σταματά τη σκέψη γύρω από το τσιγάρο.



Εικόνα 66: Οθόνη Παιχνιδιών

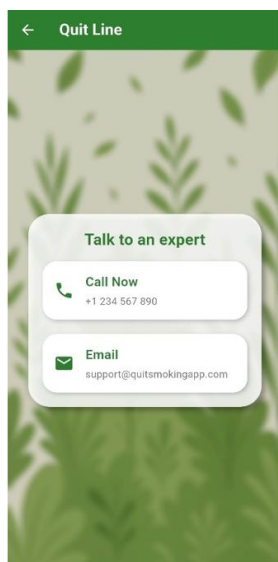
Εικόνα 67: Παιχνίδια (Pac Man, Φιδάκι, Flappy Bird)



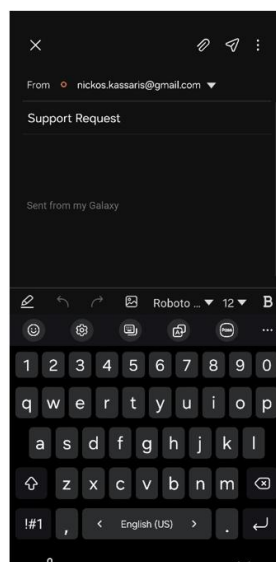
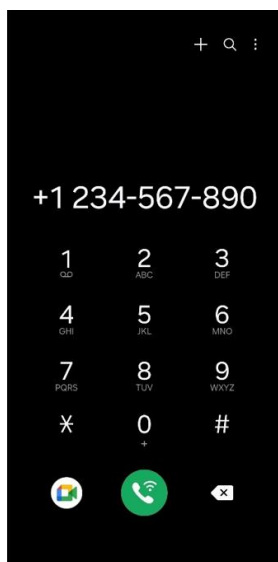
Εικόνα 68: Επιτεύγματα Ολοκληρωμένα (αριστερά) και Προς Ολοκλήρωση (Δεξιά)

Η οθόνη των επιτευγμάτων προσφέρει αναλυτική απεικόνιση των στόχων που έχει πετύχει ή βρίσκεται σε διαδικασία επίτευξης ο χρήστης. Κάθε επίτευγμα εμφανίζεται με τίτλο, περιγραφή, εικονίδιο και μπάρα προόδου, ενώ όταν ολοκληρώνεται, ξεχωρίζει οπτικά με ειδική σήμανση. Η λειτουργικότητα αυτή ενισχύει το αίσθημα επιβράβευσης και δέσμευσης στη διαδικασία διακοπής. Τα επιτεύγματα, έχουν σχεδιαστεί έτσι ώστε να συνδυάζουν τόσο την ψυχολογική ενίσχυση όσο και την τεχνική υλοποίηση που απαιτείται για την παρακολούθηση της προόδου.

Η οθόνη Talk to Someone έχει σχεδιαστεί ως το σημείο όπου ο χρήστης μπορεί να αναζητήσει άμεση υποστήριξη και να νιώσει ότι δεν είναι μόνος του στη διαδικασία διακοπής του καπνίσματος. Η φιλοσοφία της βασίζεται στην παροχή τριών τρόπων επικοινωνίας, οι οποίοι καλύπτουν διαφορετικές ανάγκες. Αυτοί αποτελούν την άμεση και εξατομικευμένη υποστήριξη από γιατρό, την άνεση και αμεσότητα ενός chatbot που λειτουργεί ως γρήγορη βοήθεια και τη συμμετοχή σε ένα ομαδικό περιβάλλον συνομιλίας με άλλα άτομα που βρίσκονται στην ίδια πορεία. Κατά την είσοδο στην οθόνη, ο χρήστης αντικρίζει τρεις εμφανείς επιλογές, καθεμία με το δικό της εικονίδιο και συνοδευτική περιγραφή.



Εικόνα 69: Οθόνη Επικοινωνίας με Ειδικό και Οθόνη Ανακατεύθυνσης με Αυτόματη Συμπλήρωση



Εικόνα 70: Οθόνη Επικοινωνίας

Η πρώτη επιλογή αφορά την επικοινωνία με γιατρό. Η δεύτερη επιλογή οδηγεί στο chatbot, το οποίο αποτελεί έναν ενδιάμεσο μηχανισμό υποστήριξης. Η τρίτη επιλογή αφορά το community chat, έναν χώρο όπου οι χρήστες της εφαρμογής μπορούν να συζητήσουν μεταξύ τους, να μοιραστούν εμπειρίες και να ενθαρρύνουν ο ένας τον άλλον. Η επιλογή επικοινωνίας με γιατρό καλύπτει το σενάριο όπου ο χρήστης χρειάζεται άμεση επαγγελματική υποστήριξη. Η υλοποίηση στηρίζεται σε μια διεπαφή που δίνει στον χρήστη δυνατότητα είτε να στείλει γραπτό μήνυμα μέσω ηλεκτρονικού ταχυδρομείου, είτε να καλέσει μια γραμμή υποστήριξης. Όταν ο χρήστης επιλέξει, είτε επικοινωνία μέσω τηλεφώνου, είτε μέσω email, ανακατευθύνεται αυτόματα με το πάτημα του αντιστοίχου κουμπιού με συμπληρωμένα τα στοιχεία (αριθμός τηλεφώνου γραμμής υποστήριξης ή διεύθυνση ηλεκτρονικού ταχυδρομείου) στην προεπιλεγμένη εφαρμογή.



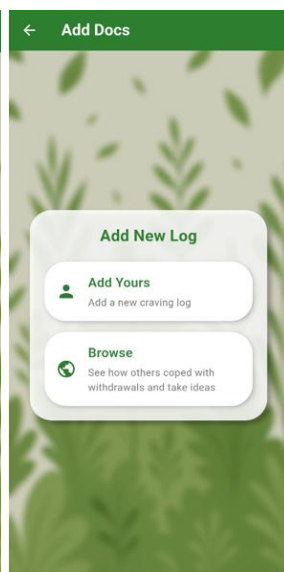
Η τρίτη επιλογή, το community chat, αποτελεί τον χώρο κοινωνικής υποστήριξης. Εδώ, οι χρήστες συζητούν μεταξύ τους σε πραγματικό χρόνο, ενισχύοντας το αίσθημα ότι αποτελούν μέλος μιας κοινότητας. Από πλευράς UI, κάθε μήνυμα συνοδεύεται από το όνομα χρήστη, ώστε η εμπειρία να είναι πιο προσωπική. Επίσης κάθε δωμάτιο (chatroom), αποτελείται από συγκεκριμένο αριθμό ατόμων, ώστε να διασφαλιστεί μια τάξη στην συνομιλία, να μην υπάρχουν υπερβολικά μηνύματα και δημιουργηθεί αναρχία.



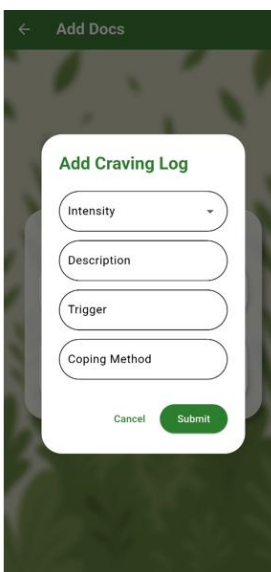
Εικόνα 71: Οθόνη Ομαδικής Συνομιλίας



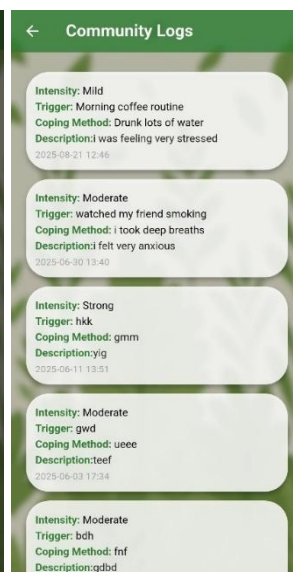
Εικόνα 72: Λίστα με τις καταγραφές



Εικόνα 73: Προσθήκη Νέας Καταγραφής



Εικόνα 74: Πλοήγηση στις Καταγραφές Άλλων Χρηστών



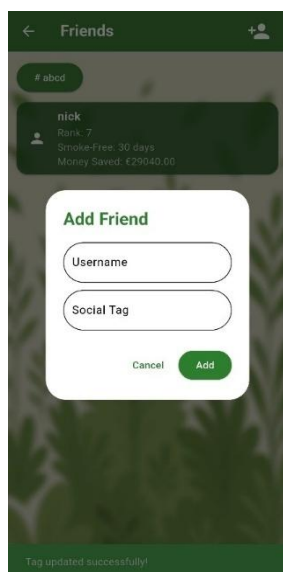
Η οθόνη Craving Logs λειτουργεί ως κεντρικός κόμβος προσωπικής τεκμηρίωσης των στερητικών συμπτωμάτων (cravings) και των στρατηγικών αντιμετώπισής τους, αλλά και ως σημείο πλοήγησης σε ανώνυμα αρχεία άλλων χρηστών, με στόχο τη μάθηση από πραγματικά παραδείγματα. Στην αρχική εμφανίζεται μια καθαρή λίστα με όλα τα καταγεγραμμένα επεισόδια. Κάθε στοιχείο της λίστας έχει την ημερομηνία, την ένταση που δήλωσε ο χρήστης, το ερέθισμα (στρες, καφές, κοινωνική περίσταση, οσμή καπνού) και τη στρατηγική που εφάρμοσε για την αντιμετώπιση (ασκήσεις αναπνοής, σύντομος περίπατος, τσίχλα νικοτίνης, απομάκρυνση από trigger, σύνδεση με φίλο/γιατρό/chatbot), καθώς και μια σύντομη περιγραφή. Πατώντας στο εικονίδιο «+» πάνω δεξιά, ο χρήστης έχει την επιλογή πλοήγησης σε καταγραφές άλλων χρηστών ανώνυμα ή να κάνει μια νέα καταγραφή, όπου ανοίγει λεπτομερής κάρτα με επιπλέον πεδία. Η καταχώριση νέου craving γίνεται μέσα από φόρμα που ελαχιστοποιεί την τριβή. Με μία κίνηση επιλέγει ένταση, αναλύει το ερέθισμα (trigger), την περιγραφή και τον τρόπο αντιμετώπισης, ενώ πεδία όπως η ώρα καταγραφής (timestamp) συμπληρώνονται αυτόματα.

Δίπλα στις προσωπικές καταγραφές βρίσκεται η ενότητα πλοήγησης σε καταγραφές άλλων χρηστών, η οποία είναι αυστηρά ανώνυμη. Η προβολή υλοποιείται ως Discover feed με άπειρο κύλιση. Για κάθε εγγραφή προβάλλεται, η ένταση, το trigger και η εφαρμοσμένη στρατηγική, συνοδευμένα από σύντομη αφήγηση για το τι λειτούργησε. Ιδιαίτερη βαρύτητα δίνεται στην προστασία δεδομένων. Οι προσωπικές καταγραφές προστατεύονται από κανόνες Firestore που περιορίζουν την ανάγνωση/εγγραφή στον κάτοχο του userId.

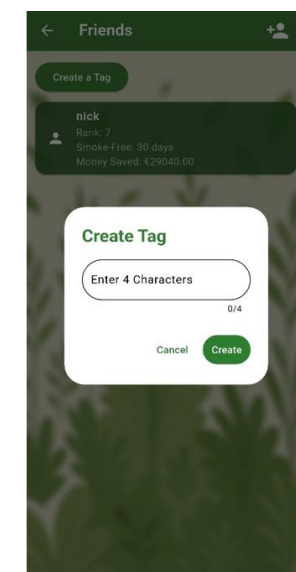
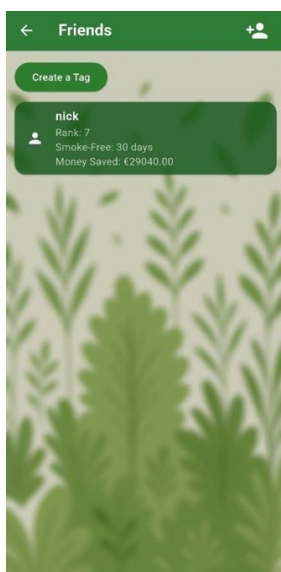
Η οθόνη φίλων έχει ως στόχο να υποστηρίξει την κοινωνική διάσταση της εφαρμογής, ενισχύοντας τα κίνητρα του χρήστη μέσα από τη δικτύωση με άλλους ανθρώπους που βρίσκονται στην ίδια πορεία διακοπής του καπνίσματος. Στην οθόνη αυτή ο χρήστης μπορεί να δει μια λίστα με τους φίλους του, όπου εμφανίζονται βασικές πληροφορίες όπως το όνομα χρήστη, οι μέρες αποχής από το κάπνισμα, τα χρήματα που έχει εξοικονομήσει και το επίπεδο του (rank).

Ο χρήστης έχει τη δυνατότητα να προσθέσει νέους φίλους με βάση το username και έναν τετραψήφιο αριθμό που ορίζει εκείνος (προεπιλεγμένος είναι το 0000). Η λειτουργία αυτή

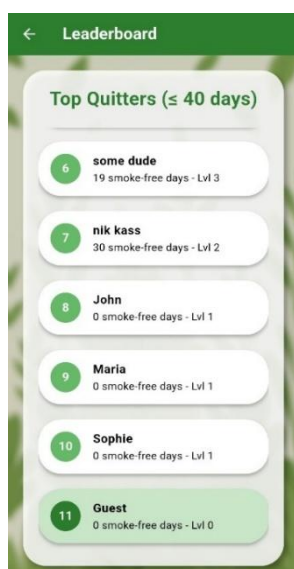
ενισχύει το αίσθημα κοινότητας, καθώς επιτρέπει στους χρήστες να παρακολουθούν φίλους που βρίσκονται σε παρόμοιο στάδιο, προσφέροντας έμπνευση και παρακίνηση.



Εικόνα 75: Προσθήκη Φίλου



Εικόνα 76: Δημιουργία Social Tag



Εικόνα 77: Χρήστης με Κατάταξη εκτός δεκάδας

Η οθόνη Leaderboard αποτελεί τον βασικό μηχανισμό κοινωνικής παρακίνησης της εφαρμογής, αφού συγκεντρώνει και εμφανίζει την κατάταξη των χρηστών με βάση την πρόοδο τους στη διακοπή του καπνίσματος και τα επιτεύγματα που έχουν ολοκληρώσει.

Στην αρχική της μορφή, η οθόνη παρουσιάζει έναν δυναμικό πίνακα κατάταξης, στον οποίο οι χρήστες εμφανίζονται ταξινομημένοι σύμφωνα με το επίπεδό τους, το οποίο αυξάνεται από την ολοκλήρωση επιτευγμάτων. Ο πίνακας αυτός εμφανίζει χρήστες με quit date μικρότερο των 40 ημερών, ώστε να διασφαλίζεται δίκαιη σύγκριση ανάμεσα σε νέους και πιο έμπειρους συμμετέχοντες.

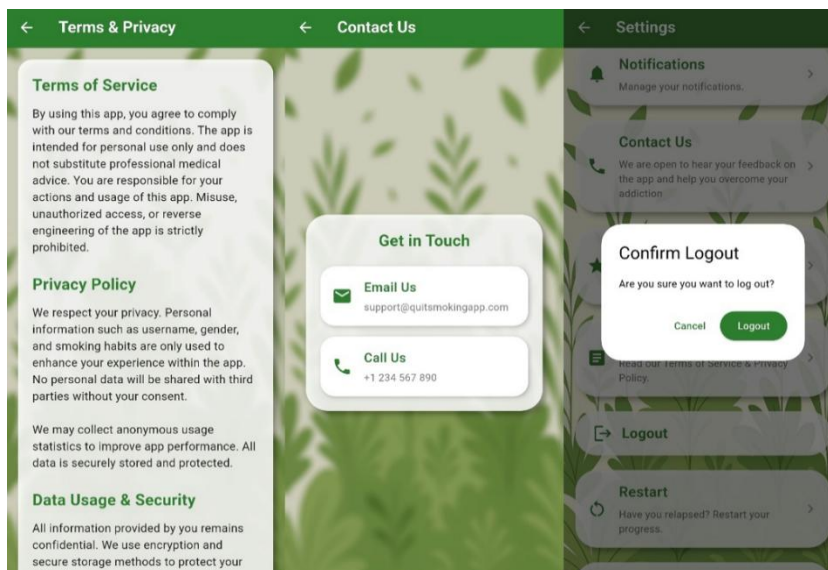
Κάθε θέση στον πίνακα περιλαμβάνει το username, τον αριθμό των ημερών αποχής του χρήστη και το τρέχον επίπεδο. Εάν ο χρήστης δεν βρίσκεται στους πρώτους δέκα της κατάταξης, το σύστημα φροντίζει να προβάλλει σε ξεχωριστή ενότητα τη δική του θέση, ώστε να διατηρείται το κίνητρο βελτίωσης. Με τον τρόπο αυτό δημιουργείται ένα κλίμα υγιούς ανταγωνισμού και ταυτόχρονα έμπνευσης, καθώς οι νεότεροι χρήστες μπορούν να παραδειγματιστούν από τις επιτυχίες των πιο προχωρημένων.

Στην οθόνη των ρυθμίσεων της εφαρμογής υπάρχουν τα εξής:

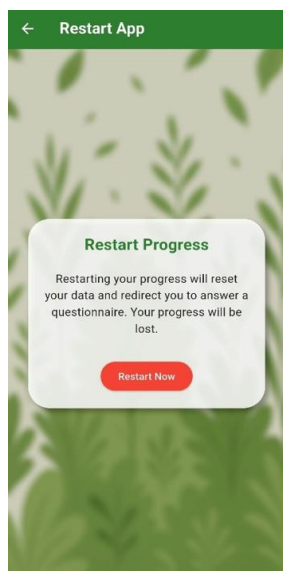
Η οθόνη με τα στοιχεία του λογαριασμού, καθώς και το κουμπί με την ενεργοποίηση/απενεργοποίηση των ειδοποιήσεων, που ανακατευθύνει τον χρήστη στις ρυθμίσεις της εφαρμογής του τηλεφώνου. Επιπλέον, όταν ο χρήστης εισέρχεται για πρώτη φορά στην εφαρμογή του εμφανίζεται στην αρχική οθόνη ένα αναδυόμενο παράθυρο από το λειτουργικό σύστημα του τηλεφώνου για αποδοχή ή απόρριψη αδειάς των ειδοποιήσεων του συστήματος. Οθόνη Όρων και Προϋποθέσεων. Οθόνη επικοινωνίας με ανακατεύθυνση του χρήστη στο τηλέφωνο ή email με αυτόματη συμπλήρωση των πληροφοριών επικοινωνίας. Κουμπί αποσύνδεσης, με αναδυόμενο παράθυρο για επιβεβαίωση.



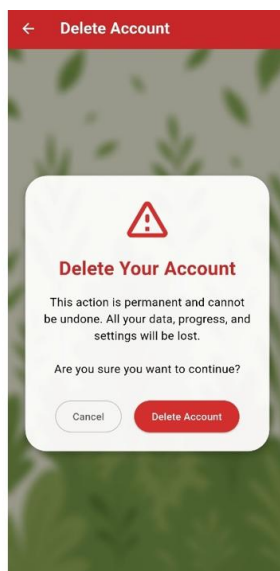
Εικόνα 78: Πληροφορίες Λογαριασμού



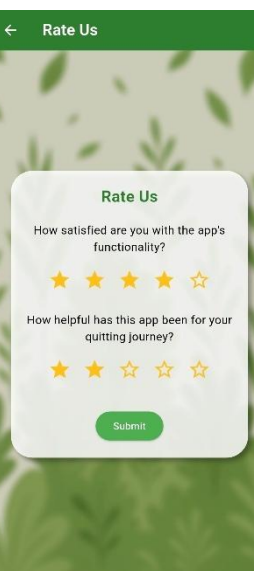
Εικόνα 79: Όροι Χρήσης, Καρτέλα Επικοινωνίας, Αποσύνδεση Λογαριασμού



Εικόνα 80: Επανεκκίνηση Προόδου και Διαγραφή Λογαριασμού



Εικόνα 81: Αξιολόγηση Εφαρμογής



Εικόνα 82: Ειδοποιήσεις

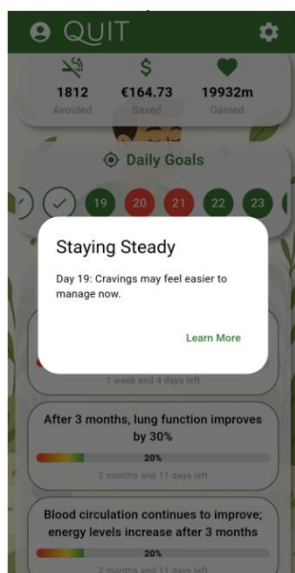
Οθόνη επανεκκίνησης προόδου σε περίπτωση υποτροπής του χρήστη. Πατώντας το κουμπί restart now, ο χρήστης ανακατευθύνεται πάλι στην οθόνη ερωτηματολογίου και ξανά συμπληρώνει τα πεδία. Η πρόοδος του διαγράφεται και τα επιτεύγματα, βραβεία, πόντοι κτλ. αρχικοποιούνται.

Οθόνη διαγραφής λογαριασμού. Όλα τα δεδομένα του χρήστη διαγράφονται από το firebase, καθώς και η σύνδεση του από το firebase console. Επιπλέον, ο χρήστης για λόγους ασφαλείας του firebase, πρέπει να έχει συνδεθεί πρόσφατα, για να του επιτραπεί να διαγράψει τον λογαριασμό του.

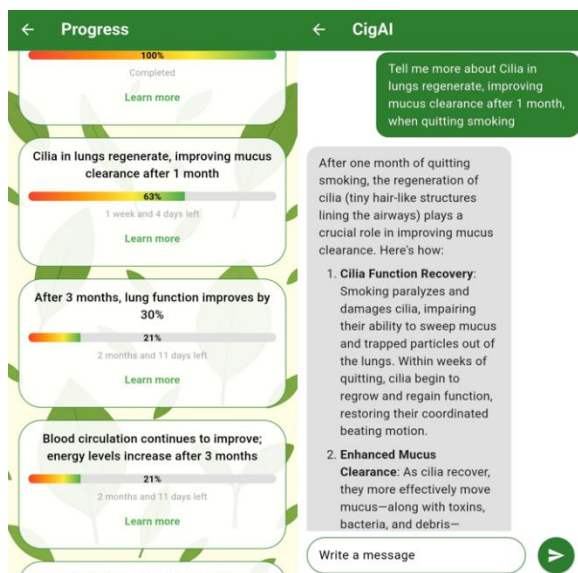
Στην οθόνη αξιολόγησης εμπειρίας χρήστη και αποτελεσματικότητας της εφαρμογής. Οι αξιολογήσεις αποθηκεύονται ανώνυμα στο firebase, όπου έπειτα μπορούν να χρησιμοποιηθούν για βελτίωση του συστήματος. Όταν ο χρήστης υποβάλει την αξιολόγηση, δεν του επιτρέπεται να υποβάλει νέα, αλλά του εμφανίζεται ένας σύνδεσμος για το Google

Play Store (όταν η εφαρμογή είναι διαθέσιμη για δημόσια χρήση), όπου ο χρήστης μπορεί εκεί να αφήσει μια πιο αναλυτική κριτική. Επιπλέον, παρουσιάζονται και μερικές από τις ειδοποιήσεις της εφαρμογής (αρκετές μαζί για λόγους επίδειξης).

Το chatbot αποτελεί τη γρήγορη εναλλακτική όταν ο χρήστης θέλει άμεση απάντηση. Το chatbot απαντά σε θέματα όπως στρατηγικές αντιμετώπισης στερητικών, συμβουλές για χαλάρωση ή ενημέρωση για τα οφέλη της αποχής από το κάπνισμα. Για παράδειγμα βλέπουμε ότι αν πατήσει ο χρήστης το κουμπί Learn More στο αναδυόμενο παράθυρο με τους καθημερινούς στόχους, ανακατευθύνεται στην οθόνη του chatbot, με αυτόματα συμπληρωμένη την ερώτηση καθώς και άλλα βοηθητικά system prompts που δεν φαίνονται. Έπειτα αν ο χρήστης θέλει μια πιο λεπτομερή και τεκμηριωμένη οπτική του στόχου, τότε το chatbot του εμφανίζει την απάντηση αντλώντας πληροφορίες αρχικά από την επιστημονικά τεκμηριωμένη βάση γνώσης μέσω RAG και έπειτα από το διαδίκτυο. Έτσι καταλήγει ο χρήστης με μια πλήρως λεπτομερή απάντηση που θα τον ικανοποιήσει και θα του δώσει τα σωστά κίνητρα και συμβουλές για να πετύχει τον στόχο του.

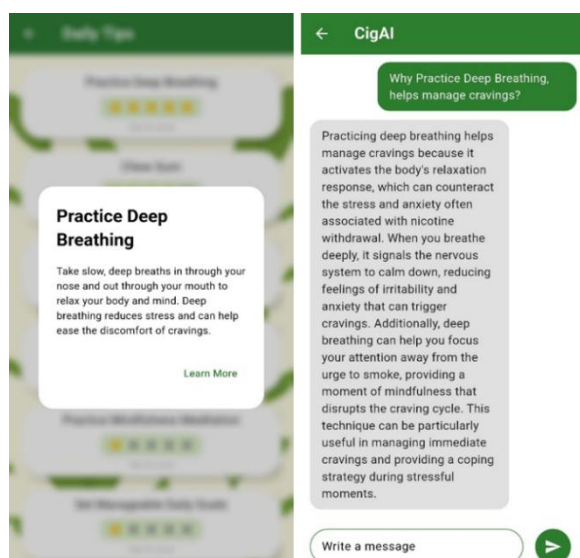


Εικόνα 83: Chatbot στους ημερήσιους στόχους (Daily Goals)



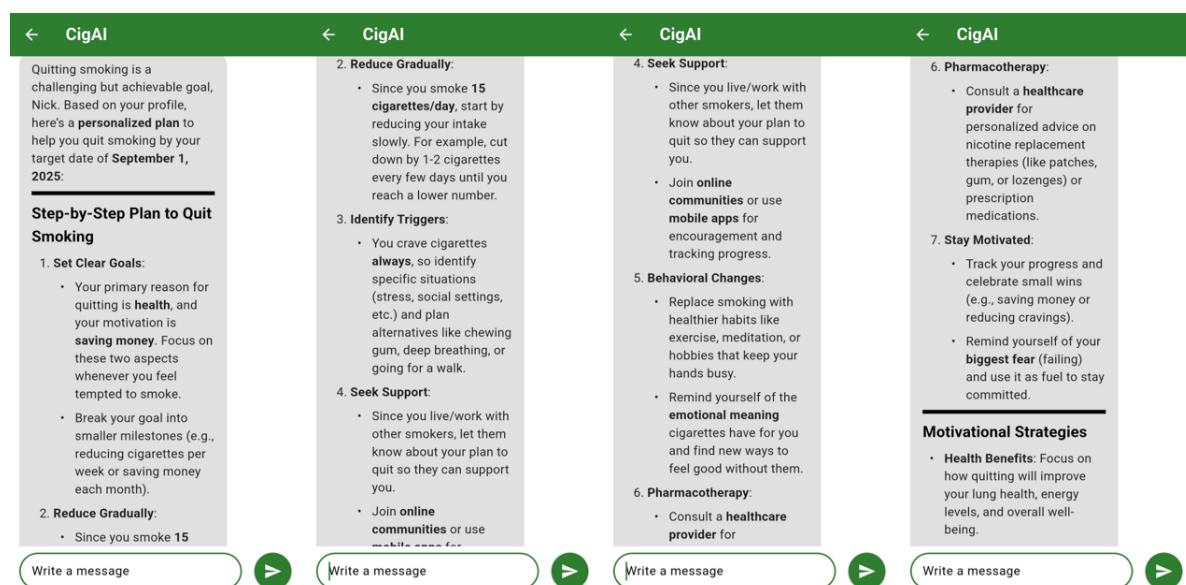
Εικόνα 84: Chatbot στην πρόοδο της υγείας και η απαντήσεις του

Αν ο χρήστης θελήσει να έχει μια πιο εμπειριστωμένη ανάλυση για την πρόοδο της υγείας του, τότε πατώντας το κουμπί Learn More στους υγειονομικούς στόχους, ανακατευθύνεται στην οθόνη του Chatbot, όπου του δίνει λεπτομέρειες επιστημονικά τεκμηριωμένες. Έτσι, αντί να ακολουθεί τυφλά ότι του εμφανίζεται στην εφαρμογή, βλέποντας τις λεπτομέρειες και τα επιστημονικά στοιχεία, η πειθώς του αυξάνεται και συνεπώς και το κίνητρό του.



Εικόνα 85: Chatbot στις ημερήσιες συμβουλές

Στις ημερήσιες συμβουλές, με ένα απλό πάτημα στο κουμπί Learn More, ο χρήστης μπορεί να μάθει λεπτομέρειες για την συγκεκριμένη συμβουλή. Τέτοιες λεπτομέρειες αφορούν το πως να την εφαρμόσει σωστά, σε τι ακριβώς και πότε θα του είναι χρήσιμη, καθώς και για ποιον λόγο λειτουργεί.



Εικόνα 86: Εξατομικευμένη Απάντηση του Chatbot

Ο χρήστης ωστόσο, μπορεί να χρησιμοποιήσει το chatbot κάνοντας του δικιές του ερωτήσεις σε ότι αφορά την διακοπή καπνίσματος. Εδώ βλέπουμε ότι η εμπειρία είναι προσωποποιημένη, αφού αντλεί δεδομένα από τις προηγούμενες συνομιλίες (έχει υλοποιηθεί σύστημα μνήμης), καθώς και από το ερωτηματολόγιο που συμπληρώνει ο χρήστης στην αρχή. Στο παράδειγμα ο χρήστης είχε δηλώσει ότι καπνίζει 15 τσιγάρα την ημέρα, ότι καπνίζει με πολύ μεγάλη συχνότητα (πάντα), ότι ζει και εργάζεται με άλλους καπνιστές κτλ. και το chatbot, εκτός ότι του το αναφέρει, του δίνει απάντηση σχετική με αυτές τις πληροφορίες.



Τέλος, έχει δοθεί που μεγάλη έμφαση και στον τύπο ερωτήσεων που θα μπορεί να γίνεται και να απαντά επιτυχώς το chatbot. Αν οι ερωτήσεις δεν είναι σχετικές με το κάπνισμα και την διαδικασία διακοπής του, τότε το εμφανίζεται σχετικό μήνυμα που ενημερώνει τον χρήστη. Επιπλέον, αν δεν μπορεί να δώσει απάντηση για τον οποιονδήποτε λόγο, π.χ. κλειστός server ή κάποια άλλη βλάβη στο σύστημα, τότε εμφανίζεται πάλι σχετικό μήνυμα στην οθόνη του χρήστη.

*Εικόνα 87: Διαχείριση
ερωτήσεων και λαθών
Chatbot*

Η παρουσίαση των παραπάνω οθονών αναδεικνύει τη άρτια αρχιτεκτονική της εφαρμογής, η οποία ενοποιεί την προσωπική παρακολούθηση, την ψυχολογική υποστήριξη, τα κίνητρα μέσω επιβράβευσης και την κοινωνική αλληλεπίδραση. Κάθε επιμέρους λειτουργία συμβάλλει στην ενίσχυση της θέλησης του χρήστη να διακόψει το κάπνισμα και να διατηρήσει αυτή την συνήθεια του σε βάθος χρόνου.

8. Αξιολόγηση και Συμπεράσματα

8.1 Απόδοση Chatbot και Συστήματος

Η αξιολόγηση ενός chatbot που βασίζεται σε μεγάλα γλωσσικά μοντέλα είναι κρίσιμο βήμα. Χρειάζεται να εξετάσουμε την ποιότητα των απαντήσεων, την ακρίβεια και την αντικειμενικότητα. Η αξιολόγηση αποκτά ακόμη μεγαλύτερη σημασία όταν το chatbot αφορά την υγεία, όπως στην περίπτωση της παρούσας διπλωματικής εργασίας. Πρέπει να δίνει σωστές και ασφαλείς πληροφορίες για να βοηθά ουσιαστικά τον χρήστη.

8.1.1 Αξιολόγηση LLM και RAG

Η αξιολόγηση των μεγάλων γλωσσικών μοντέλων αποτελεί σήμερα ένα από τα πιο κρίσιμα ζητήματα στην έρευνα και την ανάπτυξη τεχνητής νοημοσύνης. Τα LLMs έχουν φτάσει σε εντυπωσιακό επίπεδο κατανόησης και παραγωγής φυσικής γλώσσας. Ταυτόχρονα, συνοδεύονται από σοβαρές προκλήσεις αξιοπιστίας, ασφάλειας και της πρακτικής τους χρησιμότητας. Η αξιολόγηση είναι απαραίτητη για την αντιμετώπιση αυτών των προκλήσεων και να εντοπίσουμε τις αδυναμίες του συστήματος, για να διασφαλίσουμε ότι το chatbot μπορεί να χρησιμοποιηθεί με ασφάλεια και αποτελεσματικότητα.

Η αξιολόγηση ενός LLM γίνεται διότι τα παραδοσιακά μέτρα ακρίβειας που χρησιμοποιούνται σε κλασικά μοντέλα μηχανικής μάθησης δεν είναι αρκετά. Αυτό συμβαίνει, αφού τα παραγόμενα αποτελέσματα είναι συνήθως ανοιχτού τύπου, με ερωτήσεις που έχουν αόριστες απαντήσεις [49]. Έτσι, χρειάζεται ένα πιο “χαλαρό” πλαίσιο αξιολόγησης που να συνδυάζει ποσοτικές και ποιοτικές μεθόδους.

Ένα σημαντικό κριτήριο αξιολόγησης είναι η πιστότητα (faithfulness). Αυτό το κριτήριο αφορά το κατά πόσο η απάντηση του μοντέλου βασίζεται σε πραγματικές πληροφορίες και δεν παράγει ψευδείς ή ανύπαρκτες. Η πιστότητα είναι ιδιαίτερη σημαντική σε περιπτώσεις όπου η ακρίβεια της πληροφορίας είναι κρίσιμη, όπως στην υγεία, στη νομική ή στην εκπαίδευση. Επιπλέον, ένα άλλο κριτήριο αξιολόγησης είναι η συνάφεια (relevance). Το κριτήριο αυτό δείχνει αν η απάντηση ανταποκρίνεται στην ερώτηση που έθεσε ο χρήστης. Μια απάντηση μπορεί να είναι αληθής αλλά άσχετη, επηρεάζοντας έτσι την ποιότητα της αλληλεπίδρασης.

Ένα ακόμα κριτήριο αξιολόγησης είναι η χρησιμότητα (helpfulness). Η απάντηση εκτός από σωστή, πρέπει να είναι και χρήσιμη για τον χρήστη, να μπορεί να τον καθοδηγήσει να λάβει μια απόφαση ή να εκτελέσει μια ενέργεια. Στην περίπτωση μας αυτό σημαίνει το chatbot να δίνει απαντήσεις που δεν είναι γενικές, αλλά εφαρμόσιμες. Στο κριτήριο αυτό περιλαμβάνεται και η ασφάλεια, δηλαδή η αποφυγή προκαταλήψεων και επικίνδυνων συμβουλών. Ένα LLM μπορεί να είναι τεχνικά άψογο, αλλά αν προτείνει επικίνδυνες πρακτικές ή αναπαράγει στερεότυπα, η αξιολόγηση θα πρέπει να το καταγράψει αρνητικά.

Η μεθοδολογία αξιολόγησης μπορεί να είναι ανθρώπινη, αυτόματη ή υβριδική. Η ανθρώπινη αξιολόγηση αποτελεί το πρότυπο, αφού οι ειδικοί ή οι χρήστες μπορούν να εκτιμήσουν καλύτερα την ποιότητα σε σχέση με τα συμφραζόμενα. Όμως, είναι ακριβή και χρονοβόρα. Η αυτόματη αξιολόγηση και πιο συγκεκριμένα η μέθοδος LLM-as-a-Judge, χρησιμοποιεί

άλλα γλωσσικά μοντέλα ως κριτές που βαθμολογούν απαντήσεις με βάση συγκεκριμένα κριτήρια. Αυτή η μέθοδος είναι γρήγορη και scalable. Παρόλα αυτά, υπάρχει κίνδυνος μεροληψίας και αστάθειας. Το υβριδικό μοντέλο αξιολόγησης συνδυάζει και τις δύο μεθόδους, προκειμένου να υπάρχει ταχύτητα και έλεγχος ποιότητας.

Αξιολόγηση RAG συστημάτων

Εκτός από τις κλασικές μετρικές, υπάρχουν και πιο τεχνικές αξιολογήσεις που αφορούν τη λειτουργία του συστήματος RAG. Σε αυτά τα συστήματα η ποιότητα εξαρτάται από τον retriever και τον generator. Η αξιολόγηση της ανάκτησης περιλαμβάνει μετρικές όπως recall, MRR και hit rate, καθώς και μετρικές faithfulness για την παραγωγή. Έτσι, μπορούμε να ξεχωρίσουμε αν ένα σφάλμα προήλθε από την επιλογή λάθος εγγράφων ή από λανθασμένη χρήση των σωστών εγγράφων.

Η αξιολόγηση του RAG, είναι μια αρκετά απαιτητική διαδικασία. Αυτό συμβαίνει, διότι τέτοια συστήματα δεν περιορίζονται μόνο στη γλωσσική ικανότητα ενός μοντέλου αλλά αποτελούνται από αρκετά στάδια. Αρχικά γίνεται ανάκτηση πληροφοριών από μία εξωτερική βάση γνώσης και έπειτα γίνεται η παραγωγή φυσικής γλώσσας που βασίζεται σε αυτές τις πληροφορίες. Έτσι, η αξιολόγηση ενός RAG εξετάζει την ποιότητα της ανάκτησης αλλά και της τελικής απάντησης.

Οι βασικές μετρικές που χρησιμοποιούνται για την ανάκτηση είναι η ακρίβεια (precision), η ανάκληση και η κάλυψη του περιεχομένου. Η ακρίβεια δείχνει αν τα κείμενα που επέστρεψε το σύστημα είναι σχετικά με την ερώτηση, ενώ η ανάκληση αφορά το αν το σύστημα κατάφερε να βρει όλα τα σχετικά αποσπάσματα που υπήρχαν στη βάση. Η κάλυψη αναφέρεται στο αν οι απαντήσεις περιλαμβάνουν όλες τις κρίσιμες πληροφορίες που χρειάζεται ο χρήστης.

Για την παραγωγή, η αξιολόγηση βασίζεται σε μετρικές όπως η faithfulness, δηλαδή το κατά πόσο η απάντηση μένει πιστή στις πηγές που ανακτήθηκαν, τη συνάφεια (relevancy) και τη χρησιμότητα για τον τελικό χρήστη. Ένα RAG μπορεί να φέρει σωστά έγγραφα αλλά το LLM να συνθέσει λάθος πληροφορίες (hallucination). Έτσι, η αξιολόγηση πρέπει να ελέγχει ότι οι παραγόμενες απαντήσεις όντως βασίζονται στο ανακτημένο περιεχόμενο και δεν προσθέτουν παραπλανητικά δεδομένα.

Η αξιολόγηση των συστημάτων RAG βασίζεται στην παραδοσιακή αξιολόγηση της πληροφόρησης, με μετρικές που βασίζονται σε ground truth δεδομένα. Αυτή συνδυάζεται με την αξιολόγηση της φυσικής γλώσσας, με μετρικές πιο υποκειμενικές που μετρούν τη σαφήνεια και την ευχρηστία. Αυτό καθιστά την αξιολόγηση περίπλοκη αλλά και πιο αντιπροσωπευτική της πραγματικής εμπειρίας του χρήστη.

LLM-as-a-Judge

Μια από τις πιο καινοτόμες προσεγγίσεις που έχουν αναπτυχθεί τα τελευταία χρόνια είναι η χρήση μεγάλων γλωσσικών μοντέλων ως αυτόματων κριτών (LLM-as-a-Judge). Η βασική ιδέα είναι ότι το ίδιο το LLM μπορεί να αξιολογήσει και να βαθμολογεί τις απαντήσεις ενός άλλου μοντέλου με βάση καθορισμένα κριτήρια. Αντί να απαιτείται ένας μεγάλος αριθμός ανθρώπινων αξιολογητών, ένα LLM μπορεί να αναλάβει να δώσει αντικειμενική άποψη για την ποιότητα, την ακρίβεια, τη συνάφεια και τη σαφήνεια μιας απάντησης.

Η μέθοδος αυτή έχει χρησιμοποιηθεί σε διάφορα benchmarks, όπου τα LLMs αξιολογούν και συγκρίνουν απαντήσεις διαφορετικών μοντέλων σε πραγματικές ερωτήσεις. Το πλεονέκτημα είναι ότι η διαδικασία γίνεται ταχύτερη, πιο scalable και μετρήσιμη. Ένα LLM μπορεί να ακολουθήσει ένα συγκεκριμένο πλαίσιο αξιολόγησης και να αποδώσει βαθμούς σε πολλές μετρικές χωρίς να επηρεάζεται από κόπωση ή προκαταλήψεις που συναντώνται σε ανθρώπους.

Ωστόσο, η χρήση του LLM-as-a-Judge δεν είναι χωρίς προκλήσεις. Το LLM μπορεί να παρουσιάσει τις ίδιες προκαταλήψεις με το μοντέλο που αξιολογεί, ενώ υπάρχει πάντα η πιθανότητα να βαθμολογήσει με τρόπο ασυνεπή. Για να περιοριστεί αυτό, χρησιμοποιούνται prompts που καθορίζουν ξεκάθαρα τα κριτήρια αξιολόγησης και ζητούν από το LLM να αιτιολογήσει τις αποφάσεις του. Έτσι, η διαδικασία γίνεται πιο αντικειμενική και προσεγγίζει τις ανθρώπινες αξιολογήσεις.

Η συγκεκριμένη μέθοδος γίνεται συνεχώς πιο δημοφιλής, γιατί επιτρέπει σε ερευνητές και μηχανικούς να αξιολογούν συστήματα σε μεγάλη κλίμακα, κάτι που παλαιότερα θα απαιτούσε τεράστιο κόστος και χρόνο. Σε συνδυασμό με εργαλεία που ενσωματώνουν αυτή τη μέθοδο και προσφέρουν δομημένα dashboards, η αξιολόγηση γίνεται πιο πρακτική και εύχρηστη.

Τα τελευταία χρόνια έχουν αναπτυχθεί εξειδικευμένα εργαλεία για την αξιολόγηση LLMs, όπως το TruLens και το DeepEval. Αυτά επιτρέπουν τη συστηματική καταγραφή, τον έλεγχο και την αυτοματοποίηση της διαδικασίας. Προσφέρουν έτοιμες μετρικές και ενσωμάτωση σε αρκετά frameworks. Έτσι κάνουν την αξιολόγηση κομμάτι της ροής ανάπτυξης, κάτι που ενισχύει την αξιοπιστία.

Η αξιολόγηση των LLM απαιτεί συνδυασμό μεθόδων, πολλά κριτήρια και εργαλεία που μπορούν να αποδώσουν ποσοτικά και ποιοτικά αποτελέσματα. Ο στόχος είναι να υπάρχει ισορροπία μεταξύ ταχύτητας, ακρίβειας και αξιοπιστίας, ώστε τα μοντέλα να μπορούν να αξιοποιηθούν με ασφάλεια και αποτελεσματικότητα στην πράξη.

8.1.2 Μέθοδοι Αξιολόγησης

Υπάρχουν διαφορετικοί τρόποι αξιολόγησης. Ο συνηθισμένος είναι η ανθρώπινη αξιολόγηση. Ειδικοί ή χρήστες διαβάζουν τις απαντήσεις και τις βαθμολογούν. Αυτή η μέθοδος είναι αξιόπιστη αλλά χρονοβόρα. Μια πιο σύγχρονη μέθοδος είναι το LLM-as-a-Judge. Σε αυτήν τη μέθοδο χρησιμοποιούμε άλλο μεγάλο γλωσσικό μοντέλο για να βαθμολογήσει τις απαντήσεις. Έτσι μπορούμε να έχουμε γρήγορη και συστηματική αξιολόγηση.

TruLens

Το TruLens είναι μια πλατφόρμα αξιολόγησης και παρακολούθησης LLM εφαρμογών. Προσφέρει εργαλεία καταγραφής και ανάλυσης στις εκτελέσεις του chatbot. Προσφέρει feedback functions που μπορούν να προσαρμοστούν στις ανάγκες του χρήστη. Με αυτές ελέγχεται η πιστότητα, η ακρίβεια κτλ. Το TruLens συνδέεται εύκολα με συστήματα όπως το LangChain και μπορεί να ενσωματωθεί σε πολύπλοκα pipelines. Δίνει επίσης δυνατότητα καταγραφής της διαδρομής του ερώτηματος μέσα στο σύστημα (tracing). Αυτό βοηθά στο debugging και στην κατανόηση των σφαλμάτων. Στο σύστημά μας το TruLens βοήθησε να μετρηθεί αν οι απαντήσεις του chatbot ήταν πιστές στις πηγές.

Επιπλέον, το TruLens ενσωματώνεται απευθείας στη ροή εκτέλεσης ενός RAG. Η χρήση του έγινε μέσα από Jupyter notebook, ακολουθώντας το Quickstart που παρέχεται στη βιβλιοθήκη. Ουσιαστικά γίνεται pipeline wrap του chatbot με τα εργαλεία που προσφέρει, έτσι ώστε να συλλέγονται αυτόματα δεδομένα απόδοσης και αξιοπιστίας.

Επίσης, δίνει τη δυνατότητα να οριστούν custom evaluators, αλλά και να χρησιμοποιηθούν έτοιμες μετρικές, όπως faithfulness και relevance. Ένα πλεονέκτημα της βιβλιοθήκης είναι ότι επιτρέπει την οπτικοποίηση των αποτελεσμάτων σε πίνακες και dashboards, κάνοντας πιο εύκολη την κατανόηση της συνολικής ποιότητας.

Η ενσωμάτωση στο notebook έγινε με απλό τρόπο. Ορίζεται πρώτα το retrieval pipeline του chatbot. Αυτή είναι η διαδικασία με την οποία λαμβάνεται το ερώτημα του χρήστη, αναζητούνται σχετικά έγγραφα και παράγεται η τελική απάντηση. Στη συνέχεια το pipeline αυτό τυλίγεται με την κλάση που παρέχει το TruLens για logging και αξιολόγηση. Έπειτα, κάθε φορά που εκτελείται ένα ερώτημα, τα δεδομένα καταγράφονται αυτόματα και αξιολογούνται με τις μετρικές που έχουν οριστεί. Έτσι, χωρίς να χρειάζεται κάποια πολύπλοκη διαδικασία, αποκτάμε αναλυτική εικόνα για το πώς αποδίδει το chatbot σε πραγματικές συνθήκες χρήσης.

DeepEval

Το DeepEval είναι μια βιβλιοθήκη για unit tests σε LLMs. Προσφέρει ένα πλαίσιο ειδικά για γλωσσικά μοντέλα και φέρνει συστηματικότητα στην αξιολόγηση. Τα τεστ που αναπτύσσονται τρέχουν αυτόματα και ελέγχουν αν το chatbot πληροί συγκεκριμένα κριτήρια. Το DeepEval υποστηρίζει μετρικές όπως G-Eval αλλά και πιο κλασικές, βασισμένες σε LLM-as-a-Judge. Σημαντικό χαρακτηριστικό του είναι ότι επιτρέπει reproducible tests, δηλαδή τεστ που μπορούν να επαναληφθούν με συνέπεια. Στη περίπτωση μας, το DeepEval χρησιμοποιήθηκε για να στηθεί ένα σύνολο από τεστ και να ελέγχεται κάθε φορά αν το chatbot παραμένει συνεπές και αξιόπιστο.

Η διαδικασία αξιολόγησης αποτελείται από δυο βήματα. Αρχικά, δημιουργήθηκαν τα δεδομένα καταγραφής της αλληλεπίδρασης χρήστη και chatbot, ενώ στην συνέχεια τα δεδομένα αυτά αναλύθηκαν μέσω των μετρικών της βιβλιοθήκης.

Η δημιουργία των logs υλοποιήθηκε στο αρχείο create_logs.py. Αυτό διαβάζει τα αρχεία μνήμης, τα οποία είχαν αποθηκευτεί σε μορφή pickle. Έπειτα, εξάγει από αυτά τις πληροφορίες που είναι απαραίτητες για την αξιολόγηση. Για κάθε αποθηκευμένο session εντοπίζονται οι ερωτήσεις των χρηστών, οι απαντήσεις του chatbot, καθώς και τα αποσπάσματα που είχαν ανακτηθεί από τη βάση γνώσης από το RAG. Τα δεδομένα αυτά αποθηκεύονται σε ένα αρχείο logs.jsonl (JSON lines). Κάθε γραμμή του αρχείου αντιπροσωπεύει μία αλληλεπίδραση, με τα αντίστοιχα πεδία input, response και retrieved_context. Έτσι, εξασφαλίζεται ότι οι αξιολογήσεις γίνονται σε ένα τυποποιημένο σύνολο δεδομένων.

Για την αξιολόγηση του chatbot υλοποιήθηκε μια ξεχωριστή διαδικασία, στο αρχείο evaluate_with_deepeval.py, για την επεξεργασία των αρχείων καταγραφής και την αυτόματη παραγωγή μετρικών. Η διαδικασία αυτή περιλαμβάνει τη φόρτωση των δεδομένων και έπειτα την εκτέλεση των ίδιων των μετρικών.

Αρχικά, παραμετροποιείται η πρόσβαση στο API της OpenAI, καθώς το DeepEval αξιοποιεί ένα LLM για να εκτελέσει την αξιολόγηση. Ορίστηκε ως βασικό μοντέλο το gpt-4o-mini, το οποίο χρησιμοποιήθηκε για την υπολογιστική εκτίμηση των μετρικών. Στη συνέχεια καθορίστηκαν τα μονοπάτια εισόδου και εξόδου, με το αρχείο logs.jsonl να περιέχει τις αλληλεπιδράσεις του chatbot και το αρχείο deepeval_report.csv να αποθηκεύει τα αποτελέσματα της αξιολόγησης.

Η κεντρική συνάρτηση του κώδικα (evaluate_log_file), διάβαζει γραμμή-γραμμή τα δεδομένα από το αρχείο JSON. Για κάθε αλληλεπίδραση δημιουργείται ένα αντικείμενο τύπου LLMTestCase, το οποίο περιλάμβανε το αρχικό ερώτημα του χρήστη, την απάντηση του chatbot, το ανακτημένο περιεχόμενο (retrieved context). Με αυτόν τον τρόπο, κάθε test case περιέγραφε πλήρως το πλαίσιο της αξιολόγησης.

Οι μετρικές που εφαρμόστηκαν ήταν τρεις και σχετίζονται άμεσα με την αξιολόγηση ενός RAG. Η πρώτη ήταν η Answer Relevancy Metric, που ελέγχει αν η απάντηση είναι σχετική με το ερώτημα του χρήστη. Η δεύτερη ήταν η Faithfulness Metric, η οποία εξετάζει αν η απάντηση είναι σχετική με το ανακτημένο περιεχόμενο και δεν περιέχει σύνθετες πληροφορίες. Η τρίτη ήταν η Contextual Precision Metric, που μετρά την ακρίβεια με την οποία το chatbot αξιοποιεί το διαθέσιμο υλικό από τη βάση γνώσης. Κάθε μία από αυτές τις μετρικές υπολογιζόταν με τη βοήθεια του GPT μοντέλου, το οποίο παρείχε αριθμητικά σκορ και σύντομες εξηγήσεις.

Τα αποτελέσματα αποθηκεύονταν σε dictionary list και στο τέλος εξάγονταν σε αρχείο CSV. Έτσι δημιουργείται ένας πίνακας που για κάθε ερώτημα εμφάνιζε τις επιμέρους βαθμολογίες. Με αυτόν τον τρόπο ήταν εφικτή η συστηματική ανάλυση της απόδοσης του chatbot σε κλίμακα και η ανίχνευση πιθανών αδυναμιών.

Στην περίπτωση της διπλωματικής, το TruLens χρησιμοποιήθηκε συμπληρωματικά με το DeepEval, ώστε να υπάρχει μια πιο σφαιρική άποψη για την ποιότητα των απαντήσεων. Τα δύο εργαλεία δεν κάνουν το ίδιο πράγμα, αλλά συμπληρώνουν το ένα το άλλο. Το TruLens επικεντρώνεται στην παρακολούθηση, στην καταγραφή και στην αναλυτική κατανόηση της πορείας μιας απάντησης. Το DeepEval επικεντρώνεται στην αυτοματοποίηση και στην αναπαραγωγή τεστ. Το πρώτο δείχνει γιατί μια απάντηση μπορεί να είναι προβληματική. Το δεύτερο δείχνει αν η απάντηση περνάει συγκεκριμένα κριτήρια κάθε φορά. Ο συνδυασμός τους έδωσε πολύ καλή εικόνα για το chatbot.

8.1.3 Αποτελέσματα της Αξιολόγησης

Η αξιολόγηση του chatbot πραγματοποιήθηκε με δύο διαφορετικά εργαλεία, το DeepEval και το TruLens. Τα αποτελέσματα που προέκυψαν αποτυπώνουν την ποιότητα των απαντήσεων που παρήγαγε το σύστημα.

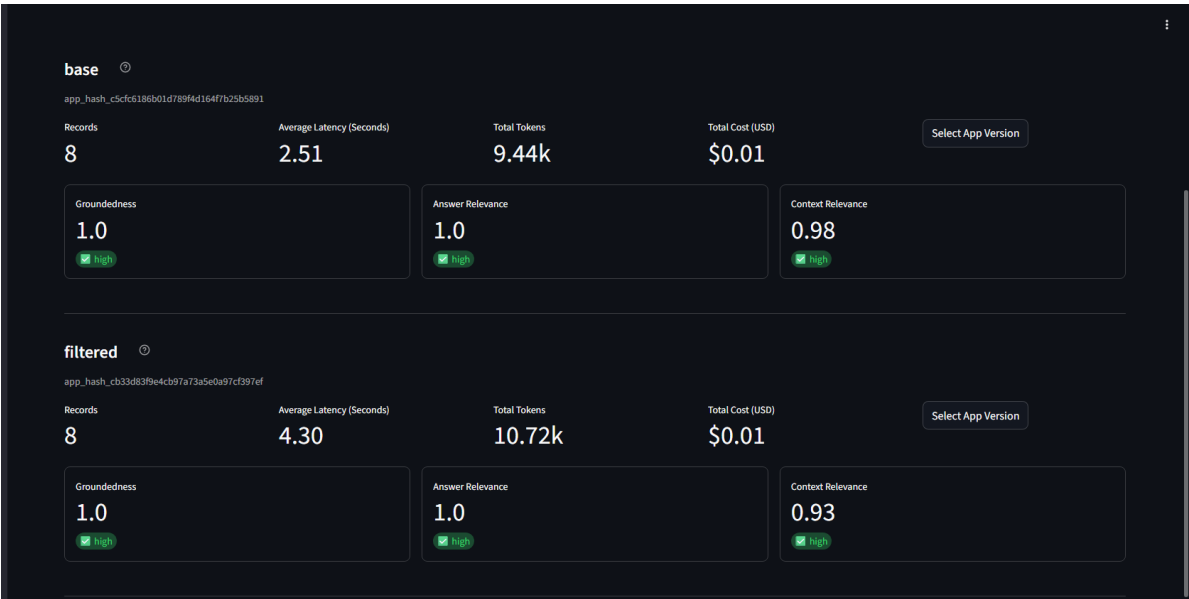
Στην περίπτωση του DeepEval, τα δεδομένα των logs μετατράπηκαν σε αρχείο CSV που περιείχε για κάθε είσοδο του χρήστη τις αντίστοιχες βαθμολογίες. Οι μετρικές που εφαρμόστηκαν ήταν το Answer Relevancy, το Faithfulness και το Contextual Precision. Οι βαθμολογίες ήταν κανονικοποιημένες στην κλίμακα [0, 1], με την τιμή 1 να αντιστοιχεί σε τέλειο σκορ. Τα αποτελέσματα έδειξαν ότι το chatbot κατάφερε να πετύχει σχεδόν το τέλειο σε όλες τις μετρικές. Για παράδειγμα, στην ερώτηση “how to quit smoking?” η απάντηση αξιολογήθηκε με 1.0 τόσο στη συνάφεια όσο και στην πιστότητα, κάτι που δείχνει ότι το

σύστημα έδωσε μια σχετική και πλήρως τεκμηριωμένη απάντηση. Στην ερώτηση “Last time i tried quitting after 2.5 month i had very intense cravings for a long time” οι βαθμολογίες ήταν και πάλι 1.0 σε όλες τις μετρικές, αποδεικνύοντας ότι το chatbot παρέμεινε εντός του θέματος και αξιοποίησε κατάλληλα το πλαίσιο για να διαμορφώσει την απάντησή του. Τα συνολικά αποτελέσματα επιβεβαιώνουν ότι το σύστημα είχε εξαιρετική συνέπεια και υψηλή ακρίβεια, αφού σε όλες σχεδόν τις περιπτώσεις οι βαθμολογίες άγγιζαν το απόλυτο.

App Version	Cost (App)	Cost Currency	Number of Events	Groundedness	Answer Relevance	Context Relevance	Record Metadata
filtered	190110000000000002	USD	15	0.987453454	1	1	0
filtered	06384	USD	15	1	1	0.9	0
filtered	165960000000000002	USD	15	0.9	1	1	0
filtered	275560000000000004	USD	15	1	1	1	0

Εικόνα 88: Αποτελέσματα Αξιολόγησης στο TruLens Dashboard

Όσον αφορά το TruLens, τα αποτελέσματα εμφάνισαν επίσης πολύ υψηλές τιμές, όμως σε ορισμένες περιπτώσεις δεν ήταν απολύτως τέλειες. Οι βαθμολογίες παρέμειναν κοντά στο 1 με μικρές αποκλίσεις. Αυτό συνέβη, διότι η γνώση βάσης που είχε ενσωματωθεί στο notebook ήταν περιορισμένη, γεγονός που δυσκόλευε την παραγωγή απολύτως πλήρων απαντήσεων. Επιπλέον, για να ενσωματωθεί το TruLens στην αρχιτεκτονική του chatbot, έγιναν μικρές αλλαγές στο pipeline του RAG, με αποτέλεσμα να διαφοροποιείται ελαφρώς η διαδικασία ανάκτησης και σύνθεσης απαντήσεων. Αυτές οι τεχνικές διαφοροποιήσεις εξηγούν γιατί οι αξιολογήσεις του TruLens δεν ήταν απόλυτα ίδιες με αυτές του DeepEval.



Εικόνα 89: TruLens Dashboard

Τα δύο εργαλεία κατέδειξαν ότι το chatbot λειτουργεί με υψηλή ποιότητα και συνέπεια. Η αξιολόγηση μέσω DeepEval, με βαθμολογίες ίσες με το 1.0, απέδειξε ότι το σύστημα ανταποκρίνεται με τον καλύτερο δυνατό τρόπο. Η συμπληρωματική χρήση του TruLens προσέφερε μια πιο διαφορετική εικόνα. Το chatbot αποδίδει όταν ενσωματώνεται σε διαφορετικές αρχιτεκτονικές και υπό συνθήκες που προσεγγίζουν την πραγματική χρήση. Η μικρή διαφορά στις βαθμολογίες δεν υποδηλώνει πρόβλημα, αλλά ενισχύει την αξιοπιστία της συνολικής αξιολόγησης, καθώς δείχνει ότι τα αποτελέσματα είναι σταθερά υψηλά παρά τις διαφοροποιήσεις στο περιβάλλον εκτέλεσης.

Παρ' όλα αυτά τα δεδομένα αξιολόγησης μπορεί να μην καλύπτουν όλες τις περιπτώσεις. Το LLM-as-a-Judge μπορεί να μην είναι αντικειμενικό. Οι πραγματικοί χρήστες μπορεί να κάνουν πιο δύσκολες ή πιο απρόβλεπτες ερωτήσεις. Για αυτό χρειάζεται στο μέλλον και ανθρώπινη αξιολόγηση, αλλά και δοκιμές με πραγματικούς καπνιστές που θα χρησιμοποιούν το chatbot στην πράξη.

Η αξιολόγηση είναι απαραίτητη για την ανάπτυξη ενός αξιόπιστου chatbot. Με τα εργαλεία αξιολόγησης μπορούμε να έχουμε μια πρώτη, πολύ καλή εικόνα της απόδοσης. Στο μέλλον χρειάζεται να προστεθούν και άλλες μέθοδοι αξιολόγησης, όπως ανθρώπινες μελέτες και μετρήσεις πραγματικών κλινικών αποτελεσμάτων.

8.2 Ανάλυση της Εμπειρίας Χρήστη

Σε αυτό το κεφάλαιο θα αξιολογηθεί η ανάλυση της εμπειρίας χρήστη (UX), όπως αυτή διαμορφώθηκε κατά τη χρήση της εφαρμογής από πραγματικούς χρήστες. Η προσέγγιση που ακολουθήθηκε βασίστηκε σε συνδυασμό παρατήρησης, συνεντεύξεων και καταγραφής της αλληλεπίδρασης του χρήστη με τα διάφορα στοιχεία της εφαρμογής. Στόχος ήταν να εντοπιστούν τόσο τα ισχυρά σημεία της εμπειρίας όσο και πιθανά σημεία βελτίωσης.

Η γενική εντύπωση των χρηστών υπήρξε ιδιαίτερα θετική, με το περιβάλλον χρήσης να χαρακτηρίζεται καθαρό, μοντέρνο και εύκολα προσβάσιμο. Το Flutter βοήθησε σε αυτό, καθώς επέτρεψε την υλοποίηση ενός UI που προσαρμόζεται σε διαφορετικά μεγέθη οθόνης χωρίς αστοχίες στην απόδοση. Το πράσινο θέμα σε συνδυασμό με τις στρογγυλεμένες γωνίες, το καθιστά φιλικό προς τον χρήστη και προσφέρει ένα καθησυχαστικό περιβάλλον. Αυτό είναι ιδιαίτερα σημαντικό για μια εφαρμογή που στοχεύει στην αλλαγή συμπεριφοράς όπως η διακοπή του καπνίσματος.

Η αρχική εμπειρία αξιολογήθηκε ως λειτουργική και κατατοπιστική. Το ερωτηματολόγιο, έκανε τους χρήστες να νιώσουν ότι η εφαρμογή είναι εξατομικευμένη στις ανάγκες τους. Η εμφάνιση των στατιστικών, (π.χ. χρήματα που έχουν εξοικονομηθεί, υγεία που έχει ανακτηθεί) δημιούργησε το αίσθημα προόδου. Τα επιτεύγματα προσθέτουν ένα στοιχείο παιχνιδιοποίησης που κρίθηκε αποτελεσματικό στην αύξηση της δέσμευσης.

Στην αρχική οθόνη οι χρήστες ανέφεραν ότι οι σημαντικές πληροφορίες (διακοπής, ημέρες χωρίς κάπνισμα, βασικοί στόχοι) ήταν άμεσα ορατές. Οι μπάρες προόδου, η οθόνη επιτευγμάτων και το leaderboard βελτίωσαν σημαντικά την αίσθηση προόδου. Η δυνατότητα προσθήκης φίλων καθώς και η πρόσβαση σε ομαδική συνομιλία ενίσχυσε το κοινωνικό στοιχείο της εφαρμογής, κάτι το οποίο άρεσε ιδιαίτερα στους χρήστες.

Η ενότητα με τις αναπνοές και τις ασκήσεις χαλάρωσης προσέφερε μια άμεση διαχείρισης του στρες. Η ύπαρξη οδηγιών, καθοδήγησης (animation κύκλου εισπνοής/εκπνοής) δημιούργησε θετική ψυχολογία στους χρήστες. Παράλληλα, η προσθήκη μικρών παιχνιδιών λειτουργούσε ως μηχανισμός απόσπασης προσοχής που αποδείχθηκε αποτελεσματικός σε στιγμές έντονης επιθυμίας.

Η πλοήγηση στην εφαρμογή παρέμεινε ομαλή και χωρίς κολλήματα. Οι χρήστες δεν συνάντησαν εμπόδια στη μετάβαση μεταξύ διαφορετικών σελίδων. Επιπλέον, η χρήση εικονιδίων με τίτλο και περιγραφή ενίσχυσε τη σαφήνεια κάθε λειτουργίας, περιορίζοντας την ανάγκη για εκμάθηση ή εξωτερική καθοδήγηση.

Συμπερασματικά, η UX σχεδίαση της εφαρμογής χαρακτηρίστηκε από εργονομία και εύκολη χρήση. Αυτά τα στοιχεία αυξάνουν αρκετά τη αλληλεπίδραση του χρήστη με την εφαρμογή. Το συνεχές feedback, η οπτικοποίηση της προόδου και η ενσωμάτωση κοινωνικών και χαρακτηριστικών παιχνιδιοποίησης, καθώς και ψηφιακού βοηθού, καθιστούν την εμπειρία συνολικά θετική και ευχάριστη.

8.3 Περιορισμοί του Συστήματος

Παρά την ευρεία λειτουργικότητα και τη θετική εμπειρία χρήστη που προσφέρει το σύστημα, υπάρχουν ορισμένοι περιορισμοί που εντοπίστηκαν κατά τη διάρκεια της ανάπτυξης, των δοκιμών και της αξιολόγησης. Αυτοί οι περιορισμοί καθορίζουν τα σημεία στα οποία θα μπορούσε να υπάρξει βελτίωση ή επέκταση στο μέλλον.

Πίνακας 7: Απαιτήσεις Συστήματος που Υλοποιήθηκαν

Τύπος Απαίτησης	Περιγραφή	Υλοποίηση στο Σύστημα
Λειτουργική	Δημιουργία και διαχείριση λογαριασμού και αυθεντικοποίηση χρηστών μέσω Firebase Authentication	Υλοποιήθηκε με email/password και Google login, με ασφάλεια μέσω Firebase UID
Λειτουργική	Ερωτηματολόγιο με προσωπικά δεδομένα χρήστη (π.χ. φύλο, τσιγάρα/ημέρα, κόστος)	Ενσωματώθηκε στην οθόνη First Login Questionnaire και αποθηκεύεται στο Firestore
Λειτουργική	Υπολογισμός και εμφάνιση εξοικονομημένων χρημάτων και αποφευγμένων τσιγάρων	Υλοποιήθηκε με βάση τα δεδομένα του χρήστη και δυναμική ενημέρωση
Λειτουργική	Προβολή health goals με progress bars	Ολοκληρώθηκε στην αρχική οθόνη με δυναμική απεικόνιση των 3 πιο κοντινών στόχων και σε ξεχωριστή οθόνη
Λειτουργική	Οθόνη επιτευγμάτων με XP	Υλοποιήθηκε με mockup service και δυναμικό rendering, με ένδειξη Completed
Λειτουργική	Σύστημα leaderboard	Υλοποιήθηκε με Provider και Firestore integration, εμφανίζει top 10 και τον χρήστη
Λειτουργική	Προσθήκη φίλων	Υλοποιήθηκε με αμοιβαία αποδοχή φιλίας
Λειτουργική	Community chat με αυτόματη δημιουργία νέων δωματίων όταν γεμίσει ένα	Υλοποιήθηκε με Firestore real-time chat και δυναμική δόμηση δωματίων
Λειτουργική	Chatbot με LangChain, RAG, memory και semantics για σχετικά ερωτήματα	Υλοποιήθηκε μέσω FastAPI endpoint με Ngrok tunnel και ενσωμάτωση στο Flutter app
Λειτουργική	Καταγραφή cravings (ένταση, συνθήκες, τρόπος διαχείρισης) και αποθήκευση τους σε logs	Υλοποιήθηκε με Firestore documents και προβολή σε προσωπικά και κοινά logs
Λειτουργική	Daily tips με μηχανισμό σταδιακού ξεκλειδώματος (μία συμβουλή ανά ημέρα)	Υλοποιήθηκε με provider και Firestore, οι παλαιότερες συμβουλές παραμένουν προσβάσιμες
Λειτουργική	Αναπνευστικές ασκήσεις με οδηγίες και animation	Υλοποιήθηκε σε ξεχωριστή οθόνη, με animation που καθοδηγεί τον χρήστη σε κύκλους εισπνοής εκπνοής
Λειτουργική	Mini-games ως εργαλείο διάσπασης προσοχής από cravings	Ενσωματώθηκαν τρία μικρά παιχνίδια (Snake, Pac-Man, Flappy Bird)
Λειτουργική	Διαχείριση ειδοποιήσεων	Υλοποιήθηκε με Provider, push notifications στο παρασκήνιο και ενεργοποίηση από επιτεύγματα και συμβουλές

Λειτουργική	Ενότητα Ρυθμίσεων με διαγραφή λογαριασμού, logout, restart progress	Όλες οι λειτουργίες ολοκληρώθηκαν μέσω Settings Screen και συνδέθηκαν με Firebase Authentication και Firestore
Μη Λειτουργική	Ασφάλεια δεδομένων με αυστηρούς κανόνες πρόσβασης	Εφαρμόστηκαν Firestore Security Rules ώστε κάθε χρήστης να έχει πρόσβαση μόνο στα δικά του δεδομένα
Μη Λειτουργική	Δυνατότητα πλήρους διαγραφής δεδομένων	Ο χρήστης μπορεί να διαγράψει τον λογαριασμό του, με αποτέλεσμα την οριστική διαγραφή όλων των δεδομένων του από το Firestore
Μη Λειτουργική	Υψηλή απόδοση και ταχύτητα απόκρισης σε πραγματικό χρόνο	Επιτεύχθηκε με χρήση Firestore real-time listeners για live ενημερώσεις
Μη Λειτουργική	Σταθερότητα συστήματος και ανοχή σε σφάλματα	Χειρισμός exceptions στο Flutter
Μη Λειτουργική	Ελκυστικό και προσβάσιμο UI/UX	Χρησιμοποιήθηκαν Flutter widgets με consistent theming, animations και responsive σχεδίαση για διαφορετικά μεγέθη οθόνης
Μη Λειτουργική	Διαθεσιμότητα εφαρμογής σε Android συσκευές	Το τελικό build υλοποιήθηκε και δοκιμάστηκε σε Android, με πλήρη λειτουργικότητα
Μη Λειτουργική	Επεκτασιμότητα για μελλοντικές λειτουργίες	Η αρχιτεκτονική με Providers και modular κώδικα επιτρέπει εύκολη προσθήκη νέων οθονών ή λειτουργιών
Μη Λειτουργική	Συμβατότητα με εξωτερικές υπηρεσίες (π.χ. LLM APIs για chatbot)	Το σύστημα διασυνδέεται επιτυχώς με OpenRouter API και LangChain, μέσω FastAPI backend και Ngrok tunneling
Μη Λειτουργική	Διασφάλιση συνέπειας δεδομένων σε πραγματικό χρόνο	Η χρήση Firestore real-time sync εξασφαλίζει ότι οι αλλαγές (π.χ. leaderboard, chat, cravings) συγχρονίζονται αμέσως
Μη Λειτουργική	Δυνατότητα offline πρόσβασης σε κρίσιμες λειτουργίες	Υλοποιήθηκε caching σε Firestore ώστε βασικές πληροφορίες (tips, progress) να είναι διαθέσιμες και χωρίς σύνδεση
Μη Λειτουργική	Καταγραφή σφαλμάτων και παρακολούθηση της κατάστασης του συστήματος	Ενσωματώθηκε μηχανισμός logging σε επίπεδο backend (FastAPI) και crash reporting στο Flutter (Firebase Crashlytics)

Αρχικά, σημαντικός περιορισμός είναι η εξάρτηση της εφαρμογής από σύνδεση στο διαδίκτυο. Παρότι πολλές λειτουργίες (όπως εμφάνιση επιτευγμάτων, leaderboard, αλληλεπίδραση με chatbot, craving logs και χρήση του community chat) απαιτούν online λειτουργία για συγχρονισμό με το Firestore και τις υποδομές του backend, η απουσία offline λειτουργίας περιορίζει τη χρηστικότητα της εφαρμογής αφού δεν έχει υλοποιηθεί fallback μηχανισμός chatbot.

Επιπλέον, η υλοποίηση του chatbot βασίζεται σε έναν τοπικό server με tunneling. Αν και αυτή η λύση λειτούργησε για σκοπούς επίδειξης και ανάπτυξης, δεν είναι κατάλληλη για παραγωγική χρήση λόγω περιορισμένης αξιοπιστίας και διαθεσιμότητας. Σε περίπτωση αποσύνδεσης του tunnel, οι χρήστες δεν μπορούν να έχουν πρόσβαση στο chatbot.

Όσον αφορά την εξατομίκευση, παρόλο που η εφαρμογή προσαρμόζει ορισμένα δεδομένα, δεν υφίσταται προσαρμογή περιεχομένου στην ανάλυση μοτίβων cravings βάσει ώρας ή συναισθηματικής κατάστασης, αν δεν του το αναφέρει ο χρήστης.

Όσον αφορά την προστασία προσωπικών δεδομένων, έχουν ληφθεί βασικά μέτρα (ασφαλής είσοδος, αποθήκευση δεδομένων στο Firestore με περιορισμένη πρόσβαση). Παρόλα αυτά στην εφαρμογή δεν έχει υλοποιηθεί συγκατάθεση βάσει GDPR, το οποίο επιτρέπει στους χρήστες να ελέγχουν ποιες πληροφορίες συλλέγονται και αποθηκεύονται.

Τέλος, η εφαρμογή δεν έχει ακόμα υποβληθεί σε εκτεταμένα performance tests για όλες τις αναλύσεις και συσκευές, γεγονός που περιορίζει την προετοιμασία για μελλοντική κυκλοφορία στο Play Store ή το App Store.

Εν κατακλείδι, οι περιορισμοί της εφαρμογής αφορούν κυρίως ζητήματα συνδεσιμότητας, ασφάλειας και προσωποποίησης. Παρ' όλα αυτά, η αρχιτεκτονική του συστήματος έχει σχεδιαστεί με τέτοιο τρόπο ώστε να επιτρέπει μελλοντικές επεκτάσεις, κάνοντας δυνατή την αντιμετώπιση των περισσότερων από αυτούς τους περιορισμούς με αναβαθμίσεις και ενημερώσεις.

8.4 Συμπεράσματα και Προτάσεις για Μελλοντική Επέκταση

Η παρούσα εργασία είχε ως στόχο τη σχεδίαση, ανάπτυξη και αξιολόγηση μιας καινοτόμας εφαρμογής υποστήριξης για διακοπή του καπνίσματος, αξιοποιώντας σύγχρονες τεχνολογίες Flutter, Firebase, LLMs και Langchain. Μέσα από τη διαδικασία υλοποίησης, ενσωματώθηκαν πολλές λειτουργίες που καλύπτουν τις ανάγκες ενός χρήστη που επιθυμεί να απαλλαγεί από τη βλαβερή αυτή συνήθεια, ενισχύοντας τη δέσμευσή του με στοχευμένες ειδοποιήσεις, σύστημα επιτευγμάτων και κοινωνική διάδραση.

Η συνολική απόδοση του συστήματος αποδείχθηκε ικανοποιητική, με σταθερή λειτουργία σε όλες τις βασικές ροές και με σωστή διαχείριση των αιτημάτων προς το backend. Ο χρήστης έχει τη δυνατότητα να βλέπει την πρόοδό του σε πραγματικό χρόνο, να δέχεται κίνητρα και υποστήριξη μέσα από έναν ψηφιακό βοηθό, να συμμετέχει σε ένα κοινωνικό περιβάλλον χρηστών με παρόμοιους στόχους, αλλά και να αντιμετωπίζει αποτελεσματικά στιγμές έντονης επιθυμίας για κάπνισμα με αναπνευστικές ασκήσεις και mini-games.

Η υλοποίηση βασίστηκε σε δοκιμασμένες αρχιτεκτονικές, με έμφαση στη modular σχεδίαση και στη χρήση mock δεδομένων για την πρώτη φάση, ενώ η χρήση Providers στο Flutter εξασφάλισε αποτελεσματική διαχείριση κατάστασης και καθαρό διαχωρισμό της λογικής από την παρουσίαση. Το chatbot βασισμένο σε Langchain και RAG τεχνικές έδειξε ικανοποιητικά αποτελέσματα στην υποστήριξη του χρήστη, παρά τους περιορισμούς που υπάρχουν στο fallback επίπεδο ή στις περιπτώσεις ασυνήθιστων ερωτημάτων.

Παρά τις επιτυχίες της εφαρμογής, υπήρξαν και περιορισμοί. Η χρήση του ngrok για προσωρινό hosting του backend δεν προσφέρει λύση παραγωγής. Επιπλέον, κάποιες λειτουργίες, όπως το σύστημα φίλων και τα cravings logs, απαιτούν περαιτέρω εμβάθυνση για να καλύψουν πλήρως τη δυναμική κοινωνικής αλληλεπίδρασης. Ορισμένα σφάλματα UI/UX σε διαφορετικές αναλύσεις συσκευών ενδέχεται να επηρεάσουν την εμπειρία χρήστη.

Για μελλοντική επέκταση, προτείνεται η διασύνδεση της εφαρμογής με wearable συσκευές για αυτόματη καταγραφή βιομετρικών δεδομένων. Το σύστημα επιτευγμάτων μπορεί να συνδεθεί με εικονικές ανταμοιβές. Το chatbot μπορεί να εξελιχθεί με fine-tuned μοντέλα ειδικά για στήριξη σε εθισμούς, ενώ θα ήταν ενδιαφέρον να προστεθεί καταγραφή συναισθηματικής κατάστασης ώστε να ανιχνεύονται μοτίβα συμπεριφοράς. Προτείνεται δημιουργία συστήματος πλήρους εξατομίκευσης στόχων και δημιουργία αποστολών (missions), από τον AI agent. Τέλος, το σύστημα φιλίας μπορεί να εμπλουτιστεί αρκετά, με προβολή επιτευγμάτων των φίλων, δωμάτιο συνομιλίας μεταξύ τους, καθώς και εισαγωγή κοινών στόχων και αλληλοπροκλήσεων.

Συνολικά, η εφαρμογή αποδεικνύει την αποτελεσματικότητα του συνδυασμού τεχνολογίας και ψυχολογικών εργαλείων στη διαδικασία απεξάρτησης από το κάπνισμα, θέτοντας τις βάσεις για μια πλήρως εξατομικευμένη, διαδραστική και επιστημονικά τεκμηριωμένη παρέμβαση.

Αρχεία Συστήματος

Ο πηγαίος κώδικας μαζί με όλα τα αρχεία που δημιουργήθηκαν και χρησιμοποιήθηκαν για την ανάπτυξη και αξιολόγηση του συστήματος και της διπλωματικής εργασίας, βρίσκονται σε GitHub repository στον παρακάτω σύνδεσμο.

<https://github.com/ntua-el19188/Smoking-Cessation-App>

Παραπομπές

- [1] Wikipedia, "History of Tobacco," 2025. [Online]. Available: https://en.wikipedia.org/wiki/History_of_tobacco.
- [2] Wikipedia, "Health effects of tobacco.," 2025. [Online]. Available: https://en.wikipedia.org/wiki/Health_effects_of_tobacco.
- [3] W. H. Organization., "Tobacco: the tobacco epidemic and disease burden," 25 June 2025. [Online]. Available: <https://www.who.int/news-room/fact-sheets/detail/tobacco?>.
- [4] ΕΛΣΤΑΤ, "Έρευνα για τις συνήθειες καπνίσματος στην Ελλάδα," 2023. [Online]. Available: https://www.statistics.gr/el/home?_com_liferay_portal_search_web_portlet_SearchPortlet_INSTANCE_3_formDate=1756656113460&p_p_id=com_liferay_portal_search_web_portlet_SearchPortlet_INSTANCE_3&p_p_lifecycle=0&p_p_state=maximized&p_p_mode=view&_com_liferay_p.
- [5] Eurostat, "Smoking habits – Statistics Explained," 2022. [Online]. Available: https://ec.europa.eu/eurostat/statistics-explained/index.php?title=Smoking_statistics.
- [6] A. D. & M. C. Yilmaz, "EuroNews," 2025. [Online]. Available: <https://gr.euronews.com/my-europe/2025/01/10/poies-xwres-ee-exoun-ta-upsilotera-pososta-kapnistwn>.
- [7] Wikipedia, "Health effects of tobacco," [Online]. Available: https://en.wikipedia.org/wiki/Health_effects_of_tobacco.
- [8] W. H. Organization, "Protecting people from tobacco smoke," [Online]. Available: <https://www.who.int/activities/protecting-people-from-tobacco-smoke>.
- [9] P. Vitória, S. E. Pereira, G. Muinos, H. D. Vries and M. L. Lima, "Parents modelling, peer influence and peer selection impact on adolescent smoking behavior: A longitudinal study in two age cohorts.," *Addictive behaviors*, vol. 100, p. 106131, January 2020.
- [10] Wikipedia, "Neuronal acetylcholine receptor subunit alpha-5," [Online]. Available: https://en.wikipedia.org/wiki/Neuronal_acetylcholine_receptor_subunit_alpha-5.
- [11] W. H. Organization, "Tobacco: Health consequences," 2020. [Online]. Available: <https://www.who.int/news-room/fact-sheets/detail/tobacco>.
- [12] N. C. Institute, "Harms of cigarette smoking and health benefits of quitting," 2017. [Online]. Available: <https://www.cancer.gov/about-cancer/causes-prevention/risk/tobacco/cessation-fact-sheet>.
- [13] B. D. e. a. Lushniak, "The health consequences of smoking—50 years of progress: A report of the Surgeon General," 2014. [Online]. Available: <https://stacks.cdc.gov/view/cdc/21569>.
- [14] N. R. e. a. Anthonisen, "Effects of Smoking Intervention and the Use of an Inhaled Anticholinergic Bronchodilator on the Rate of Decline of FEV1: The Lung Health Study," *JAMA*, vol. 272, pp. 1497-1505, November 1994.
- [15] C. e. a. Willi, "Active Smoking and the Risk of Type 2 DiabetesA Systematic Review and Meta-analysis," *JAMA*, vol. 298, pp. 2654-2664, December 2007.
- [16] G. e. a. Taylor, "Change in mental health after smoking cessation: systematic review and meta-analysis," *BMJ*, vol. 348, 2014.
- [17] W. H. Organization, "The economics of tobacco and tobacco control," 2017. [Online]. Available: <https://www.who.int/publications/i/item/9789241512250?>.
- [18] D. e. a. Johnson, "Gamification for health and wellbeing: A systematic review of the literature," *Internet Interventions*, vol. 6, pp. 89-106, 2016.
- [19] L. C. e. a. Abrams, "iPhone Apps for Smoking Cessation: A Content Analysis," *American Journal of Preventive Medicine*, vol. 40, p. 279–285, March 2011.
- [20] M. H. B. C. R. A. Whittaker and Y. Gu, "Mobile phone-based interventions for smoking cessation," *Cochrane Database of Systematic Reviews*, 2016.
- [21] K. W. Bold, K. A. Garrison, A. DeLucia, M. Horvath, M. Nguyen, E. Camacho and J. Torous, "Smartphone Apps for Smoking Cessation: Systematic Framework for App Review and Analysis," *Journal of Medical Internet Research*, vol. 25, p. e45183, July 2023.

- [22] T. L. Webb, J. Joseph, L. Yardley and S. Michie, "Using the Internet to Promote Health Behavior Change: A Systematic Review and Meta-analysis of the Impact of Theoretical Basis, Use of Behavior Change Techniques, and Mode of Delivery on Efficacy," *Journal of Medical Internet Research*, vol. 12, p. e4, February 2010.
- [23] O. Perski, D. Crane, E. Beard and J. Brown, "Does the addition of a supportive chatbot promote user engagement with a smoking cessation app? An experimental study," *DIGITAL HEALTH*, vol. 5, January 2019.
- [24] C. Free, G. Phillips, L. Watson, L. Galli, L. Felix, P. Edwards, V. Patel and A. Haines, "The Effectiveness of Mobile-Health Technologies to Improve Health Care Service Delivery Processes: A Systematic Review and Meta-Analysis," *PLOS Medicine*, vol. 10, pp. 1-26, January 2013.
- [25] J. B. Bricker, K. E. Mull, J. A. Kientz, R. Vilardaga, L. D. Mercer, K. J. Akioka and J. L. Heffner, "Randomized, controlled pilot trial of a smartphone app for smoking cessation using acceptance and commitment therapy," *Drug and Alcohol Dependence*, vol. 143, pp. 87-94, 2014.
- [26] J. Kwakernaak, J. A. H. Eekhof, M. W. M. De Waal, E. A. M. Barenbrug and N. H. Chavannes, "Patients' Use of the Internet to Find Reliable Medical Information About Minor Ailments: Vignette-Based Experimental Study," *Journal of Medical Internet Research*, vol. 21, p. e12278, November 2019.
- [27] e. a. Bontemps-Jones, "A Content Analysis of Popular Smartphone Apps for Smoking Cessation," *American Journal of Preventive Medicine*, vol. 45, p. 732–736, December 2013.
- [28] B. L. Haskins, D. Lesperance, P. Gibbons and E. D. Boudreaux, "A systematic review of smartphone applications for smoking cessation," *Translational Behavioral Medicine*, vol. 7, pp. 292-299, May 2017.
- [29] e. a. Fang, "Effectiveness of eHealth Smoking Cessation Interventions: Systematic Review and Meta-Analysis," *J Med Internet Res*, vol. 25, p. e45111, 28 July 2023.
- [30] A. Aggarwal, C. C. Tam, D. Wu, X. Li and S. Qiao, "Artificial Intelligence–Based Chatbots for Promoting Health Behavioral Changes: Systematic Review," *Journal of Medical Internet Research*, vol. 25, p. e40789, February 2023.
- [31] L. He, E. Basar, E. Krahmer, R. Wiers and M. Antheunis, "Effectiveness and User Experience of a Smoking Cessation Chatbot: Mixed Methods Study Comparing Motivational Interviewing and Confrontational Counseling," *Journal of Medical Internet Research*, vol. 26, p. e53134, August 2024.
- [32] A. F. M. Arabi and e. a. Koyuturk, *Habit Coach: Customising RAG-based chatbots to support behavior change*, 2024.
- [33] J. S. White and e. a. Salem, "Developing a Game (Inner Dragon) Within a Leading Smartphone App for Smoking Cessation: Design and Feasibility Evaluation Study," *JMIR Serious Games*, vol. 11, p. e46602, August 2023.
- [34] e. a. DeLaughter, "Crave-Out: A Distraction/Motivation Mobile Game to Assist in Smoking Cessation," *JMIR Serious Games*, vol. 4, p. e3, May 2016.
- [35] N. B. Baskerville, L. L. Struik and D. Dash, "Crush the Crave: Development and Formative Evaluation of a Smartphone App for Smoking Cessation," *JMIR mHealth and uHealth*, vol. 6, p. e52, March 2018.
- [36] Wikipedia, "Flutter (software)," [Online]. Available: https://en.wikipedia.org/wiki/Flutter_%28software%29.
- [37] Wikipedia, "Dart (programming language)," [Online]. Available: [https://en.wikipedia.org/wiki/Dart_\(programming_language\)](https://en.wikipedia.org/wiki/Dart_(programming_language)).
- [38] Wikipedia, "Firebase," [Online]. Available: <https://en.wikipedia.org/wiki/Firebase>.
- [39] Wikipedia, "Large language model," [Online]. Available: https://en.wikipedia.org/wiki/Large_language_model.
- [40] Wikipedia, "LangChain," [Online]. Available: <https://en.wikipedia.org/wiki/LangChain>.
- [41] Wikipedia, "FastAPI," [Online]. Available: <https://en.wikipedia.org/wiki/FastAPI>.
- [42] Wikipedia, "Android Studio," [Online]. Available: https://en.wikipedia.org/wiki/Android_Studio.
- [43] Wikipedia, "Unified Modeling Language," [Online]. Available: https://en.wikipedia.org/wiki/Unified_Modeling_Language.
- [44] Wikipedia, "PlantUML," [Online]. Available: <https://en.wikipedia.org/wiki/PlantUML>.

- [45] N. L. R. F. T. Y. D. S. Y. X. E. I. Y. B. D. C. W. D. H. S. C. A. M. P. F. Ziwei Ji, "Survey of Hallucination in Natural Language Generation," Available: <https://arxiv.org/abs/2202.03629>.
- [46] P. Central, "Developing a Game (Inner Dragon) Within a Leading Smartphone App for Smoking Cessation: Design and Feasibility Evaluation Study," 2023. Available: <https://pmc.ncbi.nlm.nih.gov/articles/PMC10457699/>.
- [47] J. o. M. I. Research, "Crush the Crave: Development and Formative Evaluation of a Smartphone App for Smoking Cessation," 2018. Available: <https://mhealth.jmir.org/2018/3/e52>.