



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΣΥΣΤΗΜΑΤΩΝ ΜΕΤΑΔΟΣΗΣ ΠΛΗΡΟΦΟΡΙΑΣ
ΚΑΙ ΤΕΧΝΟΛΟΓΙΑΣ ΥΛΙΚΩΝ

**Ανάπτυξη αρχιτεκτονικής ολοκληρωμένου υπολογιστικού
συστήματος για τη διαχείριση του Σακχαρώδους Διαβήτη
Τύπου 2**

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Ευγενία, Κ. Ανδρικοπούλου

Επιβλέπουσα : Κωνσταντίνα Νικήτα
Καθηγήτρια Ε.Μ.Π.

Αθήνα, Ιούλιος 2024



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΣΥΣΤΗΜΑΤΩΝ ΜΕΤΑΔΟΣΗΣ ΠΛΗΡΟΦΟΡΙΑΣ
ΚΑΙ ΤΕΧΝΟΛΟΓΙΑΣ ΥΛΙΚΩΝ

**Ανάπτυξη αρχιτεκτονικής ολοκληρωμένου υπολογιστικού
συστήματος για τη διαχείριση του Σακχαρώδους Διαβήτη
Τύπου 2**

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Ευγενία, Κ. Ανδρικοπούλου

Επιβλέπουσα : Κωνσταντίνα Νικήτα
Καθηγήτρια Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 5^η Ιουλίου 2024.

.....
Κωνσταντίνα Νικήτα
Καθηγήτρια Ε.Μ.Π.

.....
Γιώργος Στάμου
Καθηγητής Ε.Μ.Π.

.....
Αθανάσιος Βουλόδημος
Επίκουρος Καθηγητής Ε.Μ.Π.

Αθήνα, Ιούλιος 2024

.....
Ευγενία, Κ. Ανδρικοπούλου

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

Copyright © Ευγενία Ανδρικοπούλου, 2024

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

Ο Σακχαρώδης Διαβήτης Τύπου 2 είναι ο πιο συνηθισμένος τύπος διαβήτη και η σωστή αντιμετώπιση και διαχείριση βελτιώνει το βιοτικό επίπεδο των ασθενών. Η παρούσα διπλωματική εργασία αποτελεί την αρχιτεκτονική ενός ολοκληρωμένου υπολογιστικού συστήματος για τη διαχείριση του Σακχαρώδους Διαβήτη Τύπου 2. Το διαχειριστικό μέρος του συστήματος εξυπηρετεί μία εφαρμογή έξυπνου κινητού η οποία διαχειρίζεται από έναν ασθενή και μία εφαρμογή ιστού η οποία διαχειρίζεται από ένα ιατρό.

Μέσω της εφαρμογής κινητού, ο ασθενής εισάγει στο διαχειριστικό μέρος δεδομένα που σχετίζονται με τις μετρήσεις γλυκόζης και πίεσης αλλά και δεδομένα που σχετίζονται με την καθημερινότητα του και αποτελούν περιβαλλοντικούς παράγοντες που επηρεάζουν την πορεία της νόσου. Μέσω αυτής της εφαρμογής ο ασθενής διασυνδέεται με τον ιατρό του και με τις κατάλληλες άδειες που του χορηγεί, του επιτρέπει να δει τα δεδομένα που εισάγει στο σύστημα μέσω της εφαρμογής του.

Το διαχειριστικό μέρος του συστήματος παρέχει τη δυνατότητα στον ιατρό να προσθέτει ασθενείς στην προσωπική του λίστα διαχείρισης, να παρακολουθεί αναλυτικά το ιστορικό των ασθενών με τους οποίους έχει διασυνδεθεί, να καταγράφει δεδομένα που λαμβάνει από τις εξετάσεις των ασθενών του, να εισάγει ειδικές φόρμες για αυτές τις εξετάσεις και τέλος να προσθέτει εξειδικευμένες θεραπείες για κάθε ασθενή του ξεχωριστά.

Επίσης, με βάση τα δεδομένα που συλλέγει από τις δύο εφαρμογές, χρησιμοποιεί κατάλληλους αλγορίθμους για την παρουσίαση αυτών των αποτελεσμάτων ως στατιστικά δεδομένα για την ευκολότερη μελέτη και κατανόηση τους από τους χρήστες.

Τέλος, η εργασία αυτή αποτελεί μία ολοκληρωμένη πρόταση υλοποίησης ενός διαχειριστικού μέρους που θα παρέχει με ασφάλεια και ακρίβεια τις υπηρεσίες που χρειάζονται για να λειτουργήσουν με αρτιότητα εφαρμογές διαχείρισης του Σακχαρώδους Διαβήτη Τύπου 2.

Λέξεις Κλειδιά

Σακχαρώδης διαβήτης, Σακχαρώδης Διαβήτης τύπου 2, Back-End, Server, Front-End, Βάση Δεδομένων, API, endpoints, controllers, Data Transfer Objects, DTO

Abstract

Type 2 diabetes mellitus is the most common type of diabetes and proper treatment and management improves the standard of living of patients. This thesis is the management part of an integrated computer system for the management of type 2 diabetes. The administrative part of the system serves a smart mobile application which is managed by a patient and a web application which is managed by a doctor.

Through the mobile application, the patient enters data related to glucose and blood pressure measurements and data related to his/her daily life, which are environmental factors affecting the course of the disease, into the management part. Through this application, the patient interfaces with his doctor and, with the appropriate permissions granted by the doctor, allows him to view the data he enters into the system through his application.

The administrative part of the system allows the doctor to add patients to his personal management list, to monitor in detail the history of the patients with whom he is connected, to record data received from the examinations of his patients, to enter special forms for these examinations and finally to add specific treatments for each patient individually.

Also, based on the data collected from the two applications, it uses appropriate algorithms to present these results as statistical data for easier study and understanding by users.

Finally, this work is a comprehensive proposal for implementing a management part that will securely and accurately provide the services needed to run Type 2 Diabetes Mellitus management applications with integrity.

Key-Words

Diabetes mellitus, Diabetes Mellitus Type 2, Back-End, Server, Front-End, Database, API, endpoints, controllers, Data Transfer Objects, DTO

Ευχαριστίες

Ευχαριστώ θερμά την κ. Κωνσταντίνα Νικήτα, επιβλέπουσα καθηγήτρια αυτής της διπλωματικής εργασίας για την εμπιστοσύνη της αλλά και την κ. Κωνσταντία Ζαρκογιάννη για την διαρκή και αποτελεσματική βοήθεια της αλλά και για την εξαιρετική επικοινωνία και συνεργασία που είχαμε. Θέλω επιπλέον να ευχαριστήσω τους γονείς μου, Κώστα και Σοφία για τη στήριξη, τη βοήθεια και τη συμπαράσταση που μου προσφέρουν απλόχερα σε όλα τα βήματα της ζωής μου. Τέλος τα αδέρφια μου, την Ηλιάννα και τον κ. Μιχάλη που συνέβαλλαν τα μέγιστα με τις ιδέες, την κουβέντα και τη διαρκή επικοινωνία όλο αυτό το διάστημα.

Ανδρικοπούλου Ευγενία, 28 Ιουνίου 2024

Περιεχόμενα

Περίληψη	6
Λέξεις Κλειδιά	6
Abstract	9
Key-Words	9
Ευχαριστίες	11
1. Εισαγωγή	16
1.1 Αντικείμενο της Διπλωματικής	16
1.2 Οργάνωση Κειμένου	16
2. Σακχαρώδης Διαβήτης (ΣΔ)	18
2.1 Τύποι Σακχαρώδους Διαβήτη	18
2.1.1 Σακχαρώδης Διαβήτης Τύπου 1 (ΣΔΤ1)	18
2.1.2 Σακχαρώδης Διαβήτης Τύπου 2 (ΣΔΤ2)	20
2.1.3 Διαβήτης Κύησης	21
2.1.4 Άλλοι τύποι Διαβήτη	22
2.2 Συμπτώματα	23
2.3 Μέθοδοι Διάγνωσης	23
2.4 Επιπλοκές	24
2.5 Διαχείριση και Αντιμετώπιση	26
2.5.1. Διατροφή και Διατροφικές Αλλαγές	26
2.5.2. Άσκηση	26
2.5.3. Φαρμακευτική Αγωγή	26
2.5.4. Παρακολούθηση και Έλεγχος	27
2.5.5. Εκπαίδευση και Υποστήριξη	27
2.5.6. Εφαρμογές Διαχείρισης Διαβήτη	27
2.5.7. Χειρουργική Παρέμβαση	27
3. Ολοκληρωμένα συστήματα διαχείρισης του σακχαρώδους διαβήτη	29
3.1. Εισαγωγή	29
3.2. Αρχιτεκτονική ολοκληρωμένων συστημάτων διαχείρισης του Σακχαρώδους Διαβήτη	29
3.3. Ολοκληρωμένα συστήματα διαχείρισης του Σακχαρώδους Διαβήτη	35
4. Τεχνολογίες για την ανάπτυξη συστήματος διακομιστή έξυπνων εφαρμογών (Back End)	42
4.1 Εισαγωγή	42

4.2 Επίπεδο Πρόσβασης στα δεδομένα (Back – End)	42
4.2.1 Διακομιστής (Server).....	43
4.2.2 Βάση Δεδομένων	44
4.2.3 Server-Side Logic	44
4.2.4 API (Application Programming Interfaces)	45
4.2.5 Ασφάλεια.....	46
4.3 Εργαλεία και Μέθοδοι Ελέγχου και Δοκιμών του Back-End.....	47
4.3.1 Τρόποι Ελέγχου και Δοκιμών.....	47
4.3.2 Διαδικασία Δοκιμών.....	48
5. Σχεδιασμός συστήματος υποστήριξης κλινικών αποφάσεων για τη διαχείριση του σακχαρώδους διαβήτη	49
5.1 Αρχιτεκτονική του Ολοκληρωμένου Συστήματος Διαχείρισης του Σακχαρώδους Διαβήτη Τύπου 2.....	49
5.1.1 Επίπεδο παρουσίασης των δεδομένων του συστήματος (Front-End)	50
5.1.2 Επίπεδο πρόσβασης και διαχείρισης των δεδομένων του συστήματος (back – end).....	51
5.2 Τεχνολογία υλοποίησης του Back-End.....	51
5.3 Αρχιτεκτονική του Back-End.....	51
5.3.1 API.....	52
5.3.2 Κώδικας Διαχείρισης των Δεδομένων.....	55
5.3.3 Βάση Δεδομένων	55
6. Υλοποίηση συστήματος υποστήριξης κλινικών αποφάσεων για τη διαχείριση του σακχαρώδους διαβήτη	58
6.1 Αρχιτεκτονική της Βάσης Δεδομένων και ανάλυση των πινάκων οντοτήτων ..	58
6.1.1 Δεδομένα που αφορούν την εφαρμογή του ιατρού	58
6.1.2 Δεδομένα που αφορούν την εφαρμογή του ασθενούς.....	66
6.1.3 Δεδομένα που αφορούν και τις δύο εφαρμογές.....	76
6.1.4 Σταθερά δεδομένα που αφορούν το σύστημα	77
6.2 API της εφαρμογής του ασθενούς.....	82
6.2.1 AuthPatientController.....	94
6.2.2 PatientProfileController	100
6.2.3 FunctionalitiesController	107
6.2.4 MsCycleController	122
6.2.5 NotificationController	126

6.3 API της εφαρμογής του ιατρού	127
6.3.1 AuthDoctorController.....	141
6.3.2 DoctorProfileController.....	144
6.3.3 PatientController.....	146
6.3.4 PatientCheckupController	153
6.3.5 PatientFormController	167
6.3.6 PatientTreatmentController	179
6.3.7 AggregatedStatisticsController.....	186
6.3.8 MobileDataController.....	187
6.4 API για την αποστολή email στους χρήστες των δύο εφαρμογών.....	190
7. Συμπεράσματα και Μελλοντικές Επεκτάσεις.....	193
7.1 Συμπεράσματα και χρησιμότητα της εφαρμογής.....	193
7.2 Μελλοντικές Επεκτάσεις.....	194
Βιβλιογραφία	198

1. Εισαγωγή

1.1 Αντικείμενο της Διπλωματικής

Ο Σακχαρώδης Διαβήτης Τύπου 2 (ΣΔΤ2) είναι μια χρόνια πάθηση η οποία εμφανίζεται σε πολύ μεγάλο ποσοστό του πληθυσμού. Η αντιμετώπιση και διαχείριση της κρίνεται επιτακτική και χρειάζεται την αгаσθή επικοινωνία μεταξύ των ασθενών και των θεράποντων ιατρών έτσι ώστε να αποφεύγονται έγκαιρα πιθανές επικίνδυνες επιπλοκές που προκαλεί η ασθένεια αυτή. Η τεχνολογία παίζει σημαντικό ρόλο σε αυτή τη διαχείριση μέσω της χρήσης συσκευών παρακολούθησης γλυκόζης, εφαρμογών για κινητά που βοηθούν στην καταγραφή και ανάλυση δεδομένων, και προηγμένων ιατρικών συσκευών που διευκολύνουν τη χορήγηση ινσουλίνης και άλλων φαρμάκων. Αυτές οι τεχνολογικές εξελίξεις επιτρέπουν στους ασθενείς να παρακολουθούν πιο εύκολα και με ακρίβεια τα επίπεδα γλυκόζης τους και να προσαρμόζουν τον τρόπο ζωής τους καθοδηγούμενοι από τους επαγγελματίες υγείας.

Η παρούσα διπλωματική αποτελεί τον σχεδιασμό και την υλοποίηση ενός συστήματος υποστήριξης δύο εφαρμογών που αποτελούν εργαλεία της διαχείρισης του ΣΔΤ2. Το σύστημα υποστήριξης εξυπηρετεί μία εφαρμογή κινητού που χειρίζεται ένας ασθενής και μία εφαρμογή ιστού που χειρίζεται ένας ιατρός. Το σύστημα λαμβάνει, αναλύει, επεξεργάζεται και αποθηκεύει δεδομένα που σχετίζονται με τους δύο χρήστες.

Ο σχεδιασμός εξυπηρετεί την εισαγωγή δεδομένων από την εφαρμογή του κινητού που σχετίζονται με την καθημερινότητα του ασθενούς όπως είναι οι μετρήσεις γλυκόζης, οι μετρήσεις της πίεσης αλλά και συνήθειες της καθημερινής του ρουτίνας. Αν ο ασθενής είναι γένους θηλυκού μπορεί με την πρώτη εισαγωγή της ημερομηνίας έναρξης και λήξης της περιόδου του, το σύστημα υποστήριξης να παράξει εκτίμηση για τους επόμενους 12 μήνες. Τέλος προσφέρει τη δυνατότητα παρουσίασης στατιστικών οι οποίες βασίζονται στις μετρήσεις γλυκόζης που εισάγει ο ασθενής μέσω της εφαρμογής του.

Επιπλέον δίνει τη δυνατότητα στην εφαρμογή του ιστού να εμφανίζει λίστες με τα δεδομένα του ιστορικού των ασθενών που διαχειρίζεται ο κάθε ιατρός. Προσφέρει την υποστήριξη για την προβολή στατιστικών επί του συνόλου των ασθενών του ιατρού που σχετίζονται με δημογραφικά χαρακτηριστικά όπως είναι η ηλικία και το φύλο. Επίσης δίνει τη δυνατότητα στον ιατρό μέσω της εφαρμογής να προσθέτει εξετάσεις για τους ασθενείς του τις οποίες συνδέει με φόρμες ερωτήσεων – απαντήσεων και εξυπηρετεί την ανάθεση θεραπειών στους ασθενείς του.

Τέλος, παρέχει τη δυνατότητα στον ιατρό να ζητήσει άδεια από τον ασθενή έτσι ώστε ο ιατρός να μπορεί να δει δεδομένα που είναι παράγωγα αυτών που εισάγει ο ασθενής στο σύστημα μέσω της δικής του εφαρμογής.

1.2 Οργάνωση Κειμένου

1. **Εισαγωγή:** Το πρώτο κεφάλαιο παρουσιάζει το αντικείμενο της διπλωματικής εργασίας και μία συνοπτική δομή του παρόντος κειμένου.

2. **Σακχαρώδης Διαβήτης:** Το κεφάλαιο αυτό αναπτύσσει το θεωρητικό υπόβαθρο της νόσου αναφέροντας τα είδη της, τα συμπτώματα, τις μεθόδους διάγνωσης, τις επιπλοκές και γενικές μεθόδους διαχείρισης και αντιμετώπισης
3. **Ολοκληρωμένα Συστήματα Διαχείρισης του Σακχαρώδους Διαβήτη:** Στο κεφάλαιο αυτό περιγράφονται οι απαιτήσεις ενός τέτοιου συστήματος, οι υπηρεσίες που παρέχει και τα βασικά υποσυστήματα από τα οποία αποτελείται. Τέλος γίνεται αναφορά σε τέτοιου τύπου υλοποιήσεις που υπάρχουν διαθέσιμες στην αγορά.
4. **Τεχνολογίες για την ανάπτυξη συστήματος διακομιστή έξυπνων εφαρμογών (Back-End):** Εδώ αναπτύσσονται τα υποσυστήματα που συνθέτουν το Back-End και τις διαθέσιμες τεχνολογίες για την υλοποίηση αυτών των υποσυστημάτων. Τέλος παρουσιάζονται μέθοδοι που εφαρμόζονται για τον έλεγχο της καλής λειτουργίας, της ασφάλειας και της εγκυρότητας του συστήματος.
5. **Σχεδιασμός συστήματος υποστήριξης κλινικών αποφάσεων για τη διαχείριση του Σακχαρώδους Διαβήτη:** Σε αυτό το κεφάλαιο παρουσιάζεται ο σχεδιασμός, η δομή του συστήματος Back – End και οι τεχνολογίες που χρησιμοποιήθηκαν για την υλοποίηση της παρούσας διπλωματικής εργασίας.
6. **Ανάπτυξη αρχιτεκτονικής ολοκληρωμένου υπολογιστικού συστήματος για τη διαχείριση του Σακχαρώδους Διαβήτη Τύπου 2:** Εδώ αναλύεται με λεπτομέρεια η υλοποίηση κάθε ενός από τα υποσυστήματα που συνθέτουν την αρχιτεκτονική του Back-End.
7. **Υποστήριξη για μελλοντικές επεκτάσεις του ολοκληρωμένου συστήματος, Συμπεράσματα και μελλοντική έρευνα:** Εδώ παρουσιάζονται υλοποιήσεις του Back-End που δεν χρησιμοποιήθηκαν για τις δύο εφαρμογές του συστήματος, συμπεράσματα της έρευνας που πραγματοποιήθηκε για την εκπόνηση της εργασίας και τέλος μελλοντική έρευνα σχετικά με ερωτήματα που εγείρει το παρόν κείμενο.

2. Σακχαρώδης Διαβήτης (ΣΔ)

Ο σακχαρώδης διαβήτης είναι μια χρόνια ασθένεια που εμφανίζεται είτε όταν το πάγκρεας δεν παράγει ινσουλίνη είτε όταν ο οργανισμός δεν μπορεί να χρησιμοποιήσει αποτελεσματικά την ινσουλίνη που παράγει. [1] Η ινσουλίνη είναι μια ορμόνη που παράγεται από τα βήτα κύτταρα του παγκρέατος και είναι υπεύθυνη για τη ρύθμιση της γλυκόζης στο αίμα. Η υπεργλυκαιμία, που ονομάζεται επίσης αυξημένη γλυκόζη αίματος ή αυξημένο σάκχαρο αίματος, είναι μία συνήθης συνέπεια του μη ελεγχόμενου διαβήτη και με την πάροδο του χρόνου οδηγεί σε σοβαρές βλάβες σε πολλά συστήματα του ανθρώπινου οργανισμού και ειδικά στο νευρικό και κυκλοφορικό σύστημα.

Το 2014, το 8,5% των ενηλίκων ηλικίας 18 ετών και άνω διαγνώστηκαν με διαβήτη. Το 2019, ο διαβήτης ήταν η άμεση αιτία 1,5 εκατομμυρίου θανάτων και το 48% όλων των θανάτων λόγω διαβήτη συνέβησαν πριν από την ηλικία των 70 ετών. Άλλοι 460.000 θάνατοι από νεφρική νόσο οφείλονται στον διαβήτη, ενώ η αυξημένη γλυκόζη στο αίμα προκαλεί περίπου το 20% των καρδιαγγειακών θανάτων.

Μεταξύ του 2000 και του 2019, σημειώθηκε αύξηση κατά 3% στα τυποποιημένα κατά ηλικία ποσοστά θνησιμότητας από διαβήτη. Στις χώρες με χαμηλότερο μεσαίο εισόδημα, ο δείκτης θνησιμότητας λόγω διαβήτη αυξήθηκε κατά 13%.

Αντίθετα, η πιθανότητα θανάτου από οποιαδήποτε από τις τέσσερις κύριες μη μεταδοτικές ασθένειες (καρδιαγγειακά νοσήματα, καρκίνος, χρόνιες αναπνευστικές παθήσεις ή διαβήτης) μεταξύ των ηλικιών 30 και 70 ετών μειώθηκε κατά 22% παγκοσμίως μεταξύ 2000 και 2019.

Στο κεφάλαιο αυτό θα εξετάσουμε τα συμπτώματα που υποδεικνύουν την εμφάνιση της νόσου, τους διάφορους τύπους της καθώς και τις μεθόδους διάγνωσης που χρησιμοποιούνται. Επιπλέον θα αναλύσουμε τις επιπλοκές που μπορεί να προκαλέσει αλλά και τρόπους για τη διαχείριση του.

2.1 Τύποι Σακχαρώδους Διαβήτη

Υπάρχουν τρεις βασικοί τύποι της νόσου, ο Σακχαρώδης Διαβήτης Τύπου 1, ο Σακχαρώδης Διαβήτης Τύπου 2 και ο Διαβήτης Κύησης. Επίσης υπάρχουν δύο τύποι που συναντώνται πιο σπάνια και αυτοί είναι ο Μονογονιδιακός Διαβήτης (MODY) και ο διαβήτης που προκαλείται από φάρμακα ή άλλες παθήσεις. Στις παρακάτω ενότητες θα αναπτύξουμε ξεχωριστά τον κάθε τύπο.

2.1.1 Σακχαρώδης Διαβήτης Τύπου 1 (ΣΔΤ1)

Ο διαβήτης τύπου 1 (ΣΔΤ1) είναι η πιο κοινή μορφή διαβήτη στα παιδιά, καλύπτοντας περίπου το 80% των περιπτώσεων παιδικού διαβήτη στις Ηνωμένες Πολιτείες. Η επίπτωση του ΣΔΤ1 αυξάνεται τόσο στις ΗΠΑ όσο και παγκοσμίως. Η μελέτη SEARCH for Diabetes in Youth εκτίμησε 21,7 νέες περιπτώσεις ΣΔΤ1 ανά έτος ανά 100.000 πληθυσμού μεταξύ 2002 και 2012 και μια συχνότητα 1,93 περιπτώσεων ανά 1.000 παιδιά. Ο ΣΔΤ1 είναι πιο κοινός στους μη Ισπανόφωνους λευκούς, ιδιαίτερα σε

όσους έχουν βόρεια ευρωπαϊκή καταγωγή, αλλά διαγιγνώσκεται σε όλες τις φυλές και εθνικότητες, με την αύξηση της επίπτωσης να είναι πιο έντονη στους Ισπανόφωνους στις ΗΠΑ.

Υπάρχει μεγάλη διαφορά στην επίπτωση του ΣΔΤ1 σε διάφορα μέρη του κόσμου, με την επίπτωση να κυμαίνεται από 0.1 στην Κίνα και τη Βενεζουέλα έως 36.8 στη Σαρδηνία και 36.5 στη Φινλανδία. Ο ΣΔΤ1 διαγιγνώσκεται επίσης σε ενήλικες, με περίπου το 30% των ασθενών να αναπτύσσουν συμπτώματα μετά την ηλικία των 18 ετών.

Η κύρια αιτία του T1D είναι η απώλεια των βήτα-κυττάρων που οδηγεί σε ανεπάρκεια ινσουλίνης. Περίπου το 90% των ασθενών αναπτύσσουν T1D λόγω αυτοάνοσης καταστροφής των βήτα-κυττάρων (τύπου 1A) [6], ενώ οι ασθενείς χωρίς αυτοαντισώματα κατατάσσονται ως "ιδιοπαθής T1D" ή τύπου 1B, που μπορεί να περιλαμβάνει ασθενείς με αυτοάνοσο διαβήτη χωρίς μετρήσιμες αντιδράσεις αυτοαντισωμάτων σε κοινά αυτοαντιγόνα, καθώς και ασθενείς με σπάνιες μορφές μονογονιδιακού διαβήτη. Η απουσία αυτοαντισωμάτων μπορεί να είναι πιο κοινή σε ορισμένες μη λευκές φυλετικές ομάδες. Η εμφάνιση του έχει ισχυρή γενετική βάση αλλά η ακριβής αιτία είναι πολυπαραγοντική. Πιο αναλυτικά έχουμε:

1. Γενετικοί παράγοντες

Ο ΣΔΤ1 έχει σημαντική γενετική βάση. Ο κίνδυνος εμφάνισης ΣΔΤ1 είναι αυξημένος για τους συγγενείς των ασθενών, με τα αδέρφια και τα παιδιά να έχουν σημαντικά υψηλότερες πιθανότητες εμφάνισης της νόσου. Γενετικές μεταλλάξεις που επηρεάζουν βασικά γονίδια του ανοσοποιητικού συστήματος, συνδέονται με τον ΣΔΤ1. Η περιοχή HLA στο χρωμόσωμα 6 παίζει καθοριστικό ρόλο, με αλληλότυπους όπως οι DR4-DQ8 και DR3-DQ2 να αυξάνουν σημαντικά τον κίνδυνο εμφάνισης της νόσου. Εκτός από την περιοχή HLA, το γονίδιο της ινσουλίνης (INS) και άλλα γονίδια που σχετίζονται με τη λειτουργία του ανοσοποιητικού συστήματος, όπως τα PTPN22 και CTLA-4, επηρεάζουν τον κίνδυνο ΣΔΤ1. [4], [5]

Παρά τις προόδους στη γενετική έρευνα, η εφαρμογή αυτών των γνώσεων στην πρόβλεψη και διάγνωση του T1D παραμένει πρόκληση. Τα μοντέλα πρόβλεψης που συνδυάζουν διάφορα γονίδια και πολυμορφικά στοιχεία (SNPs) έχουν προταθεί για την αξιολόγηση του κινδύνου. Επίσης, υπάρχουν διαφορές στη συχνότητα των αλληλοτύπων HLA και των γενετικών παραλλαγών μεταξύ διαφορετικών φυλετικών και εθνοτικών ομάδων, κάτι που μπορεί να επηρεάσει τη διάγνωση και την κλινική πορεία της νόσου.

2. Ανοσολογικοί παράγοντες

Η παρουσία αυτοαντισωμάτων, όπως τα GAD65, IA-2, και IAAs, υποδεικνύει την αυτοάνοση επίθεση στα β-κύτταρα. Αυτά τα αυτοαντισώματα μπορούν να εμφανιστούν χρόνια πριν την εμφάνιση των συμπτωμάτων του διαβήτη, γεγονός που δείχνει ότι η διαδικασία καταστροφής των β-κυττάρων αρχίζει πολύ νωρίτερα. [7]

Επίσης, η δυσλειτουργία των ρυθμιστικών Τ-λεμφοκυττάρων και η αυξημένη δραστηριότητα των επιθετικών Τ-λεμφοκυττάρων συμβάλλουν στην καταστροφή των β-κυττάρων. Η φλεγμονώδης αντίδραση που προκαλείται από αυτά τα κύτταρα οδηγεί στον θάνατο των β-κυττάρων και στην ανάπτυξη του διαβήτη.

3. Περιβαλλοντικοί παράγοντες

Είναι διάφοροι οι περιβαλλοντικοί παράγοντες που παίζουν ρόλο στην εμφάνιση αυτού του τύπου της νόσου. Ορισμένες ιογενείς λοιμώξεις έχουν συσχετιστεί με την έναρξη του διαβήτη τύπου 1. Ιοί όπως ο κοξάκι (Coxsackievirus) και άλλοι εντεροϊοί μπορούν να προκαλέσουν αυτοάνοση απόκριση που στοχεύει τα β-κύτταρα του παγκρέατος, οδηγώντας στην καταστροφή τους.

Αρκετά κρίσιμος παράγοντας είναι η διατροφή. Ορισμένες μελέτες έχουν δείξει ότι η διατροφή στην πρώιμη παιδική ηλικία, όπως η κατανάλωση γαλακτοκομικών προϊόντων και προϊόντων που περιέχουν γλουτένη, μπορεί να επηρεάσει τον κίνδυνο ανάπτυξης διαβήτη τύπου 1. Η διατροφή μπορεί να επηρεάσει την ανάπτυξη του ανοσοποιητικού συστήματος και την προδιάθεση για αυτοάνοσες ασθένειες, αυξάνοντας την πιθανότητα εμφάνισης της ασθένειας.

Τέλος, η έκθεση σε συγκεκριμένες τοξικές ουσίες και χημικά μπορεί να διαταράξει τη λειτουργία του παγκρέατος και να συμβάλει στην ανάπτυξη της νόσου. Αυτοί οι παράγοντες μπορεί να προκαλέσουν άμεση βλάβη στα β-κύτταρα ή να ενισχύσουν την αυτοάνοση απόκριση του οργανισμού, οδηγώντας σε φλεγμονή και καταστροφή των β-κυττάρων.

2.1.2 Σακχαρώδης Διαβήτης Τύπου 2 (ΣΔΤ2)

Όπως και στους άλλους τύπους, πρόκειται για μια χρόνια μεταβολική διαταραχή που χαρακτηρίζεται από αυξημένα επίπεδα γλυκόζης στο αίμα. Αυτό οφείλεται στην ανεπάρκεια της ινσουλίνης ή της αντίστασης των κυττάρων σε αυτή. Ο ΣΔΤ2 είναι η πιο κοινή μορφή διαβήτη, αντιπροσωπεύοντας περίπου το 90% των περιπτώσεων παγκοσμίως. Ορισμένοι από τους παράγοντες που οδηγούν στην εμφάνιση του, αναπτύσσονται παρακάτω: [9], [10]

1. Γενετική προδιάθεση

Η γενετική προδιάθεση παίζει σημαντικό ρόλο στην ανάπτυξη του ΣΔΤ2. Άτομα με οικογενειακό ιστορικό διαβήτη έχουν αυξημένο κίνδυνο εμφάνισης της νόσου. Η γενετική συμβολή περιλαμβάνει διάφορα γονίδια που επηρεάζουν τη λειτουργία των β-κυττάρων του παγκρέατος και την ευαισθησία των κυττάρων στην ινσουλίνη. [12]

2. Αντίσταση στην ινσουλίνη

Η αντίσταση στην ινσουλίνη είναι επίσης ένας βασικός παράγοντας στην ανάπτυξη της ασθένειας. Αυτή η κατάσταση συμβαίνει όταν τα κύτταρα του σώματος δεν

ανταποκρίνονται σωστά στην ινσουλίνη, με αποτέλεσμα να απαιτούνται υψηλότερα επίπεδα αυτής για τη ρύθμιση της γλυκόζης. Το πάγκρεας προσπαθεί να αντισταθμίσει την αυξημένη αντίσταση παράγοντας περισσότερη ινσουλίνη, αλλά με τον καιρό, αυτή η ικανότητα μειώνεται, οδηγώντας σε υπεργλυκαιμία.

3. Παχυσαρκία και καθιστικός τρόπος ζωής

Η παχυσαρκία, ιδιαίτερα η κεντρική (λίπος στην κοιλιακή περιοχή), συνδέεται στενά με την ανάπτυξη του. Ο καθιστικός τρόπος ζωής και η έλλειψη φυσικής δραστηριότητας συμβάλλουν στην αύξηση του βάρους και στην αντίσταση στην ινσουλίνη. Οι διατροφικές συνήθειες που περιλαμβάνουν υψηλή κατανάλωση επεξεργασμένων υδατανθράκων και ζαχάρων επίσης αυξάνουν τον κίνδυνο.

4. Διαιτητικές συνήθειες

Η υψηλή κατανάλωση επεξεργασμένων υδατανθράκων και ζαχάρων οδηγεί σε συχνές αιχμές στα επίπεδα γλυκόζης στο αίμα, προκαλώντας υπερινσουλιναιμία. Αυτή η χρόνια υπερινσουλιναιμία μπορεί να προκαλέσει αντίσταση στην ινσουλίνη, αυξάνοντας τον κίνδυνο ΣΔΤ2. Η διατροφή με χαμηλούς υδατάνθρακες έχει δείξει ότι μπορεί να βοηθήσει στη διαχείριση και ακόμα και στην αναστροφή της αντίστασης στην ινσουλίνη σε ορισμένες περιπτώσεις.

5. Φλεγμονή και μικροβίωμα του εντέρου

Η χρόνια χαμηλού βαθμού φλεγμονή και οι αλλαγές στο μικροβίωμα του εντέρου συνδέονται επίσης με την ανάπτυξη της αντίστασης στην ινσουλίνη και του ΣΔΤ2. Η κακή διατροφή και η παχυσαρκία μπορούν να διαταράξουν την ισορροπία των βακτηρίων του εντέρου, οδηγώντας σε αυξημένη φλεγμονή και διαταραχή της σηματοδότησης της ινσουλίνης.

2.1.3 Διαβήτης Κύησης

Σε αυτή την περίπτωση, η νόσος εμφανίζεται κατά τη διάρκεια της εγκυμοσύνης σε γυναίκες που δεν είχαν διαγνωστεί πριν με διαβήτη. Συνήθως διαγιγνώσκεται στη δεύτερη ή τρίτη τριμηνία και μπορεί να αυξήσει τον κίνδυνο για την ανάπτυξη διαβήτη τύπου 2 αργότερα στη ζωή τόσο για τη μητέρα όσο και για το παιδί. Με την κατάλληλη διαχείριση, οι περισσότερες γυναίκες γεννούν υγιή μωρά.

Οι κύριες αιτίες εμφάνισης του είναι οι ορμονικές αλλαγές και η γενετική προδιάθεση. Κατά τη διάρκεια της εγκυμοσύνης, ο πλακούντας παράγει ορμόνες που μπορούν να προκαλέσουν αντίσταση στην ινσουλίνη. Αυτές οι ορμόνες περιλαμβάνουν την ανθρώπινη πλακουντιακή λακτογόνο (hPL) και τα οιστρογόνα, τα οποία μπορούν να επηρεάσουν την ικανότητα του σώματος να χρησιμοποιεί την ινσουλίνη αποτελεσματικά.

Όπως και στους άλλους τύπους, οι γυναίκες με οικογενειακό ιστορικό διαβήτη (τύπου 1, τύπου 2 ή διαβήτη κύησης) έχουν μεγαλύτερο κίνδυνο ανάπτυξης αυτού του τύπου λόγω γενετικών παραγόντων που επηρεάζουν την ευαισθησία στην ινσουλίνη και τη λειτουργία των β-κυττάρων.

Οι πιο σημαντικοί παράγοντες κινδύνου για την εμφάνιση της ασθένειας είναι οι εξής:

1. Παχυσαρκία και υπερβολικό βάρος:

Τα υπέρβαρα ή παχύσαρκα άτομα πριν από την εγκυμοσύνη έχουν αυξημένο κίνδυνο ανάπτυξης διαβήτη κύησης. Η υπερβολική συσσώρευση λίπους, ειδικά στην περιοχή της κοιλιάς, συνδέεται με αυξημένη αντίσταση στην ινσουλίνη.

2. Ηλικία:

Οι γυναίκες άνω των 25 ετών έχουν μεγαλύτερο κίνδυνο ανάπτυξης διαβήτη κύησης. Ο κίνδυνος αυξάνεται περαιτέρω με την αύξηση της ηλικίας.

3. Οικογενειακό ιστορικό διαβήτη:

Οι γυναίκες με οικογενειακό ιστορικό διαβήτη τύπου 2 ή διαβήτη κύησης έχουν μεγαλύτερο κίνδυνο ανάπτυξης διαβήτη κύησης. Οι γενετικοί παράγοντες επηρεάζουν την ευαισθησία στην ινσουλίνη και τη λειτουργία των β-κυττάρων

4. Προηγούμενος διαβήτης κύησης:

Αν μια γυναίκα είχε διαβήτη κύησης σε προηγούμενη εγκυμοσύνη, ο κίνδυνος επανεμφάνισης σε μελλοντικές εγκυμοσύνες είναι αυξημένος.

5. Σύνδρομο πολυκυστικών ωοθηκών:

Οι γυναίκες που πάσχουν από πολυκυστικές ωοθήκες, έχουν αυξημένο κίνδυνο ανάπτυξης διαβήτη κύησης λόγω των ορμονικών διαταραχών που συνδέονται με αυτήν την πάθηση.

6. Μη υγιεινός τρόπος ζωής:

Η έλλειψη σωματικής άσκησης και μια διατροφή υψηλή σε επεξεργασμένους υδατάνθρακες και ζάχαρη μπορούν να αυξήσουν τον κίνδυνο ανάπτυξης διαβήτη κύησης.

Πέρα από τους τους τρεις βασικούς τύπους που αναλύσαμε προηγουμένως, υπάρχουν και δύο δευτερεύοντες: Ο Μονογονιδιακός Διαβήτης (Monogenic Diabetes) και ο Δευτερογενής Διαβήτης (Secondary Diabetes).

2.1.4 Άλλοι τύποι Διαβήτη

Ο μονογονιδιακός διαβήτης προκαλείται από συγκεκριμένες γενετικές μεταλλάξεις που επηρεάζουν την παραγωγή και τη χρήση της ινσουλίνης. Ένας από τους πιο γνωστούς τύπους είναι ο MODY (Maturity Onset Diabetes of the Young), που εμφανίζεται σε νεότερους ενήλικες και χαρακτηρίζεται από ήπια προς μέτρια αύξηση των επιπέδων γλυκόζης στο αίμα.

Ο δευτερογενής διαβήτης προκαλείται από άλλες παθήσεις ή τη χρήση συγκεκριμένων φαρμάκων. Για παράδειγμα, η παγκρεατίτιδα, η κυστική ίνωση και η μακροχρόνια χρήση στεροειδών μπορούν να προκαλέσουν αυτόν τον τύπο διαβήτη.

2.2 Συμπτώματα

Τα συμπτώματα του διαβήτη ποικίλλουν και εξαρτώνται κυρίως από το πόσο υψηλό είναι το ποσοστό του σακχάρου στο αίμα. Είναι δυνατόν να εμφανιστούν ξαφνικά και ο χρόνος εμφάνισης τους διαφέρει ανάλογα με τον τύπο της ασθένειας. Στον σακχαρώδη διαβήτη τύπου 2 (ΣΔΤ2), τα συμπτώματα μπορεί να είναι ήπια και να χρειαστούν πολλά χρόνια για να γίνουν αντιληπτά. Στον προδιαβήτη και στον διαβήτη κύησης, τα συμπτώματα μπορεί να μην εμφανιστούν και ποτέ. Στον σακχαρώδη διαβήτη τύπου 1, τα συμπτώματα εμφανίζονται σύντομα και είναι πολύ πιο σοβαρά. Οι πιο χαρακτηριστικές ενδείξεις εμφάνισης της νόσου είναι οι εξής: [2]

- Η αίσθηση μεγάλης δίψας,
- Η ανάγκη ούρησης περισσότερο από το συνηθισμένο,
- Η απώλεια σωματικού βάρους χωρίς ο ασθενής να ακολουθεί κάποιο ειδικό πρόγραμμα διατροφής και άσκησης,
- Η θολή όραση,
- Η έντονη αίσθηση κόπωσης και αδυναμίας,
- Οι αλλαγές στη διάθεση,
- Η επιβραδυμένη επούλωση των πληγών,
- Οι εμφάνιση πολλών λοιμώξεων όπως των ούλων, του δέρματος και του κόλπου και
- Η παρουσία κετονών στα ούρα. Οι κετόνες είναι ένα υποπροϊόν της διάσπασης των μυών και του λίπους που συμβαίνει όταν δεν υπάρχει αρκετή διαθέσιμη ινσουλίνη.

2.3 Μέθοδοι Διάγνωσης

Η διάγνωση του σακχαρώδους διαβήτη πραγματοποιείται μέσω διάφορων εξετάσεων που μετρούν τα επίπεδα γλυκόζης στο αίμα. Αυτές οι εξετάσεις βοηθούν στη διάγνωση του διαβήτη τύπου 1, του διαβήτη τύπου 2 και του διαβήτη κύησης. **Οι πιο βασικές είναι οι εξής:**

1. Δοκιμασία Γλυκοζυλιωμένης Αιμοσφαιρίνης (HbA1c):

Η εξέταση HbA1c μετρά το μέσο επίπεδο γλυκόζης στο αίμα τους τελευταίους 2-3 μήνες. Μια τιμή HbA1c $\geq 6.5\%$ υποδεικνύει διαβήτη. Το πλεονέκτημα αυτής της μεθόδου είναι ότι δεν απαιτεί νηστεία και δίνει μια εικόνα του μακροπρόθεσμου ελέγχου της γλυκόζης. [13]

2. Δοκιμασία Γλυκόζης Πλάσματος Νηστείας (FPG):

Είναι μία εύκολη και γρήγορη εξέταση FPG η οποία μετρά τα επίπεδα γλυκόζης στο αίμα μετά από νηστεία 8 ωρών. Μια τιμή γλυκόζης ≥ 126 mg/dL (7.0 mmol/L) υποδεικνύει διαβήτη.

3. Δοκιμασία Ανοχής στη Γλυκόζη από το Στόμα (OGTT):

Η εξέταση OGTT μετρά τα επίπεδα γλυκόζης 2 ώρες μετά την κατανάλωση ενός ποτού με γλυκόζη. Μια τιμή γλυκόζης ≥ 200 mg/dL (11.1 mmol/L) μετά από 2 ώρες υποδεικνύει διαβήτη. Επίσης αυτή η εξέταση μπορεί να ανιχνεύσει τον διαβήτη όταν οι άλλες εξετάσεις είναι οριακές.

4. Τυχαία Δοκιμασία Γλυκόζης Πλάσματος:

Η μέτρηση γλυκόζης σε τυχαία στιγμή κατά τη διάρκεια της ημέρας. Μια τιμή γλυκόζης ≥ 200 mg/dL (11.1 mmol/L) συνοδευόμενη από συμπτώματα διαβήτη υποδεικνύει διαβήτη. Θεωρείται χρήσιμη σε επείγουσες καταστάσεις.

Πέρα από τις εξετάσεις που γίνονται για τη διάγνωση των τριών βασικών τύπων της νόσου, στην περίπτωση του ΣΔΤ1 πραγματοποιούνται επιπλέον δύο ακόμη εξετάσεις: Οι εξετάσεις Αντισωμάτων ή Αυτοάνοσες Δοκιμασίες και η Δοκιμασία C – πεπτιδίου. Οι πρώτες, είναι εξετάσεις για αντισώματα όπως τα αντισώματα αντι-νησιδίων (GAD), τα οποία είναι συνήθη στον διαβήτη τύπου 1. Η δεύτερη, μετράει την παραγωγή ινσουλίνης από το πάγκρεας. Χαμηλά επίπεδα παραγωγής υποδηλώνουν εμφάνιση του ΣΔΤ1.

Με αυτές τις εξετάσεις, οι ιατροί μπορούν να διαγνώσουν και να παρακολουθήσουν αποτελεσματικά τον σακχαρώδη διαβήτη, διασφαλίζοντας ότι οι ασθενείς λαμβάνουν την κατάλληλη θεραπεία και διαχείριση της κατάστασής τους.

2.4 Επιπλοκές

Όπως αναλύσαμε και στις προηγούμενες ενότητες του κεφαλαίου, ο διαβήτης μπορεί να προκαλέσει σοβαρές επιπλοκές αν δεν ελεγχθεί αποτελεσματικά. Οι επιπλοκές αυτές επηρεάζουν ποικίλα συστήματα του ανθρώπινου σώματος, οδηγώντας σε σημαντική μείωση της ποιότητας ζωής και αύξηση της θνησιμότητας. Οι καρδιαγγειακές παθήσεις, η νευροπάθεια, η νεφροπάθεια, η αμφιβληστροειδοπάθεια και τα διαβητικά έλκη αποτελούν μερικές μόνο από τις επιπλοκές που σχετίζονται με τον διαβήτη. Επιπλέον, καταστάσεις όπως η διαβητική κετοξέωση και το υπερωσμωτικό σύνδρομο υπεργλυκαιμίας αναδεικνύουν την ανάγκη για άμεση και συνεχή διαχείριση της πάθησης. Παρακάτω αναλύουμε ξεχωριστά την κάθε περίπτωση.

1. Καρδιαγγειακές Επιπλοκές

Ο διαβήτης αυξάνει σημαντικά τον κίνδυνο καρδιαγγειακών παθήσεων, συμπεριλαμβανομένων της στεφανιαίας νόσου, του εμφράγματος και του εγκεφαλικού. Οι διαβητικοί έχουν 2-4 φορές μεγαλύτερη πιθανότητα να αναπτύξουν καρδιοπάθειες σε σύγκριση με άτομα χωρίς διαβήτη. Η υπεργλυκαιμία μπορεί να προκαλέσει βλάβες στα αιμοφόρα αγγεία, αυξάνοντας τον κίνδυνο αθηροσκλήρωσης και θρόμβωσης.

2. Νευροπάθεια

Η διαβητική νευροπάθεια είναι μια από τις πιο κοινές επιπλοκές του διαβήτη και μπορεί να επηρεάσει έως και το 50% των ατόμων με διαβήτη. Προκαλείται από βλάβη στα νεύρα λόγω της υψηλής γλυκόζης στο αίμα, με συμπτώματα όπως πόνο, μούδιασμα και αίσθημα καύσου στα άκρα. Σε σοβαρές περιπτώσεις, μπορεί να οδηγήσει σε απώλεια αισθήσεων, αυξάνοντας τον κίνδυνο τραυματισμών και λοιμώξεων.

3. Νεφροπάθεια

Η διαβητική νεφροπάθεια ή νεφρική νόσος αποτελεί σημαντική επιπλοκή που μπορεί να οδηγήσει σε νεφρική ανεπάρκεια. Η υπεργλυκαιμία προκαλεί βλάβη στα νεφρά, επηρεάζοντας τη λειτουργία τους. Ο διαβήτης είναι η κύρια αιτία νεφρικής ανεπάρκειας, και πολλοί διαβητικοί χρειάζονται αιμοκάθαρση ή μεταμόσχευση νεφρού.

4. Αμφιβληστροειδοπάθεια

Η διαβητική αμφιβληστροειδοπάθεια προκαλείται από βλάβη στα αιμοφόρα αγγεία του αμφιβληστροειδούς και μπορεί να οδηγήσει σε απώλεια όρασης και τύφλωση. Ο διαβήτης είναι η κύρια αιτία νέας περιπτώσεων τύφλωσης στους ενήλικες. Η τακτική εξέταση των ματιών και ο έλεγχος της γλυκόζης στο αίμα μπορούν να βοηθήσουν στην πρόληψη αυτής της επιπλοκής.

5. Διαβητικά Έλκη και Λοιμώξεις

Ο διαβήτης αυξάνει τον κίνδυνο ανάπτυξης ελκών στα πόδια και λοιμώξεων, κυρίως λόγω της κακής κυκλοφορίας του αίματος και της νευροπάθειας που μειώνει την αίσθηση στα πόδια. Αυτά τα έλκη μπορούν να οδηγήσουν σε σοβαρές λοιμώξεις και σε ακραίες περιπτώσεις να απαιτήσουν ακρωτηριασμό.

6. Διαβητική Κετοξέωση και Υπερωσμωτικό Σύνδρομο [3]

Η διαβητική κετοξέωση (DKA) και το υπερωσμωτικό σύνδρομο υπεργλυκαιμίας (HHS) είναι σοβαρές και απειλητικές για τη ζωή καταστάσεις που μπορεί να εμφανιστούν σε άτομα με διαβήτη, κυρίως τύπου 1. Η DKA προκύπτει από την έλλειψη ινσουλίνης και την παρουσία κετονών στο αίμα, ενώ το HHS χαρακτηρίζεται από εξαιρετικά υψηλά επίπεδα γλυκόζης και αφυδάτωση.

Η σωστή διαχείριση του σακχαρώδους διαβήτη μέσω τακτικής παρακολούθησης της γλυκόζης, υγιεινής διατροφής, σωματικής άσκησης και φαρμακευτικής αγωγής είναι κρίσιμη για την πρόληψη αυτών των επιπλοκών.

2.5 Διαχείριση και Αντιμετώπιση

Η διαχείριση και αντιμετώπιση του σακχαρώδη διαβήτη περιλαμβάνει μια σειρά από μεθόδους που μπορούν να προσαρμοστούν ανάλογα με τον τύπο και τις ατομικές ανάγκες του ασθενούς. Ακολουθούν οι κύριες μέθοδοι διαχείρισης, με αναφορά στον τύπο διαβήτη όπου απαιτείται:

2.5.1. Διατροφή και Διατροφικές Αλλαγές

Η υγιεινή διατροφή είναι θεμελιώδης για τη διαχείριση όλων των τύπων διαβήτη. Οι διατροφικές αλλαγές βοηθούν στη ρύθμιση των επιπέδων γλυκόζης στο αίμα και μπορούν να περιλαμβάνουν: [11]

- **Μείωση της πρόσληψης ζάχαρης και επεξεργασμένων υδατανθράκων:** Ιδιαίτερα σημαντική για τον διαβήτη τύπου 2.
- **Κατανάλωση πλούσιων σε φυτικές ίνες τροφίμων:** Βοηθά στη σταθεροποίηση των επιπέδων γλυκόζης στο αίμα.
- **Μείωση των μερίδων και έλεγχος των θερμίδων:** Μπορεί να συμβάλει στην απώλεια βάρους, μειώνοντας τον κίνδυνο εμφάνισης διαβήτη τύπου 2.
- **Συχνά γεύματα μικρότερης ποσότητας:** Βοηθούν στη διατήρηση σταθερών επιπέδων γλυκόζης.

2.5.2. Άσκηση

Η τακτική σωματική άσκηση βοηθά στη βελτίωση της ευαισθησίας στην ινσουλίνη και στη διαχείριση των επιπέδων γλυκόζης στο αίμα. Συνιστάται για όλους τους τύπους. Οι πιο αποτελεσματικές μέθοδοι άσκησης είναι η αερόβια, όπως το περπάτημα, το τρέξιμο, η ποδηλασία, και οι ασκήσεις ενδυνάμωσης, όπως οι ασκήσεις με βάρη με λάστιχα και άλλα. [9], [10]

2.5.3. Φαρμακευτική Αγωγή

Διαβήτης Τύπου 1:

- **Ινσουλίνη:** Απαραίτητη για την επιβίωση των ατόμων με διαβήτη τύπου 1. Μορφές ινσουλίνης περιλαμβάνουν ταχείας δράσης, μακράς δράσης και μεσαίας διάρκειας δράσης.

Διαβήτης Τύπου 2:

- **Μετφορμίνη:** Συνήθως το πρώτο φάρμακο που χορηγείται για τον διαβήτη τύπου 2.
- **Σουλφονουλουρίες:** Βοηθούν στην αύξηση της παραγωγής ινσουλίνης.
- **Θειαζολιδινεδιόνες:** Βελτιώνουν την ευαισθησία στην ινσουλίνη.

- **GLP-1 αγωνιστές:** Αυξάνουν την παραγωγή ινσουλίνης και μειώνουν την όρεξη.
- **SGLT2 αναστολείς:** Βοηθούν το σώμα να αποβάλει γλυκόζη μέσω των ούρων.
- **Ινσουλίνη:** Μπορεί να χρειαστεί σε προχωρημένες περιπτώσεις.

2.5.4. Παρακολούθηση και Έλεγχος

Η τακτική παρακολούθηση των επιπέδων γλυκόζης στο αίμα είναι κρίσιμη για τη διαχείριση του διαβήτη. Αυτό περιλαμβάνει:

- **Συνεχή παρακολούθηση γλυκόζης (CGM):** Παρέχει συνεχή δεδομένα για τα επίπεδα γλυκόζης, χρήσιμα κυρίως για τον διαβήτη τύπου 1.
- **Μέτρηση γλυκόζης με γλυκομετρητή:** Συνιστάται για όλους τους τύπους διαβήτη.

2.5.5. Εκπαίδευση και Υποστήριξη

Η εκπαίδευση των ασθενών για τη διαχείριση του διαβήτη είναι κρίσιμη. Περιλαμβάνει:

- **Εκπαιδευτικά προγράμματα:** Που βοηθούν τους ασθενείς να κατανοήσουν τη διατροφή, την άσκηση και την αυτοπαρακολούθηση.
- **Ομάδες υποστήριξης:** Παρέχουν ψυχολογική υποστήριξη και συμβουλές.

2.5.6. Εφαρμογές Διαχείρισης Διαβήτη

Οι εφαρμογές κινητών τηλεφώνων μπορούν να βοηθήσουν στη διαχείριση του διαβήτη παρέχοντας εργαλεία για την παρακολούθηση των επιπέδων γλυκόζης, τη διατροφή και την άσκηση. Μερικές δημοφιλείς εφαρμογές περιλαμβάνουν:

- **mySugr:** Παρέχει καταγραφή των επιπέδων γλυκόζης, ινσουλίνης και πρόσληψης τροφής.
- **Glucose Buddy:** Επιτρέπει την παρακολούθηση των επιπέδων γλυκόζης, την καταγραφή της διατροφής και της άσκησης.
- **Diabetes:** Προσφέρει λεπτομερείς αναλύσεις και προβλέψεις για τα επίπεδα γλυκόζης.
- **Glooko:** Συνδέεται με πολλούς γλυκομετρητές και παρέχει αναφορές και αναλύσεις.

2.5.7. Χειρουργική Παρέμβαση

Σε ορισμένες περιπτώσεις διαβήτη τύπου 2, μπορεί να συνιστάται βαριατρική χειρουργική για απώλεια βάρους, η οποία μπορεί να βοηθήσει στη βελτίωση του ελέγχου της γλυκόζης.

Με αυτές τις μεθόδους διαχείρισης και θεραπείας, οι ασθενείς με σακχαρώδη διαβήτη μπορούν να επιτύχουν καλύτερο έλεγχο των επιπέδων γλυκόζης και να μειώσουν τον κίνδυνο επιπλοκών.

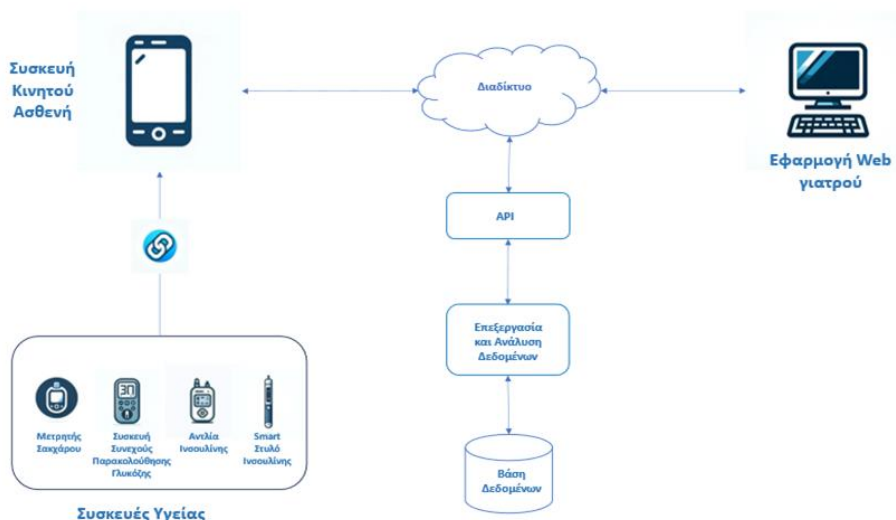
3. Ολοκληρωμένα συστήματα διαχείρισης του σακχαρώδους διαβήτη

3.1. Εισαγωγή

Τα ολοκληρωμένα συστήματα διαχείρισης του σακχαρώδους διαβήτη είναι προηγμένα εργαλεία που συνδυάζουν διάφορες τεχνολογίες με σκοπό την παρακολούθηση και διαχείριση των επιπέδων γλυκόζης στο αίμα των ατόμων που πάσχουν από τη νόσο. Αυτά τα συστήματα περιλαμβάνουν συσκευές συνεχούς παρακολούθησης γλυκόζης (CGM), αντλίες ινσουλίνης και λογισμικό διαχείρισης δεδομένων που επιτρέπουν στους ασθενείς και στους επαγγελματίες υγείας να λαμβάνουν σε πραγματικό χρόνο δεδομένα και ειδοποιήσεις. Ο στόχος τους είναι η βελτίωση της γλυκαιμικής ρύθμισης, η μείωση των επεισοδίων υπογλυκαιμίας και υπεργλυκαιμίας, και η γενική βελτίωση της ποιότητας ζωής των ατόμων με διαβήτη, επιτρέποντάς τους να διαχειρίζονται πιο αποτελεσματικά την κατάστασή τους και να μειώνουν τις επιπλοκές που σχετίζονται με τη νόσο. [16], [17], [18] Πέρα από τις μετρήσεις των επιπέδων γλυκόζης, τα συστήματα αυτά καταγράφουν και επεξεργάζονται και άλλα σημαντικά δεδομένα όπως διατροφικές πληροφορίες, φυσικές δραστηριότητες, δόσεις ινσουλίνης, συμπτώματα και άλλες παρατηρήσεις υγείας, καθώς και αποτελέσματα ιατρικών εξετάσεων. Τα δεδομένα αυτά, επεξεργάζονται και αναλύονται μέσω προηγμένων αλγορίθμων και τεχνητής νοημοσύνης, παρέχοντας ολοκληρωμένες και εξατομικευμένες συστάσεις για τη διαχείριση του διαβήτη. Τέλος, είναι πολύ χρήσιμα τεχνολογικά μέσα και για το προσωπικό υγείας και κυρίως τους ιατρούς, διότι διευκολύνουν τη λήψη αποφάσεων και τη δημιουργία εξατομικευμένων θεραπευτικών σχεδίων βασισμένων σε πληθώρα πραγματικών δεδομένων, βελτιώνοντας τελικά τα κλινικά αποτελέσματα και μειώνοντας τον κίνδυνο επιπλοκών. [20], [32]

3.2. Αρχιτεκτονική ολοκληρωμένων συστημάτων διαχείρισης του Σακχαρώδους Διαβήτη

Στο παρόν κεφάλαιο παρουσιάζονται ενδεικτικά μερικά από τα πιο γνωστά συστήματα διαχείρισης του σακχαρώδους διαβήτη. Τα περισσότερα συστήματα, με μικρές διαφοροποιήσεις, έχουν περίπου την ίδια αρχιτεκτονική λειτουργίας. Ένα πρότυπο σχήμα ενός τέτοιου είδους συστήματος φαίνεται παρακάτω:



Σχήμα 3.1 1. Αρχιτεκτονική ολοκληρωμένου συστήματος διαχείρισης Σακχαρώδους Διαβήτη

Ένα τυπικό τέτοιου είδους σύστημα αποτελείται :

1. Συσκευές του ασθενούς που σχετίζονται με τον διαβήτη:

Μετρητές γλυκόζης αίματος: Οι παραδοσιακοί μετρητές γλυκόζης, γνωστοί και ως μετρητές γλυκόζης αίματος (Blood Glucose Meters), είναι φορητές συσκευές που χρησιμοποιούνται για την άμεση μέτρηση των επιπέδων γλυκόζης στο αίμα. Η διαδικασία μέτρησης απαιτεί τη λήψη μιας μικρής σταγόνας αίματος, συνήθως από το δάχτυλο, η οποία τοποθετείται σε μια ειδική ταινία μέτρησης. Η ταινία εισάγεται στον μετρητή, ο οποίος αναλύει το αίμα και εμφανίζει τα επίπεδα γλυκόζης σε λίγα δευτερόλεπτα. Αυτή η μέθοδος είναι ευρέως διαδεδομένη λόγω της ακρίβειας και της αξιοπιστίας της, αλλά απαιτεί πολλαπλές μετρήσεις κατά τη διάρκεια της ημέρας για την παρακολούθηση των επιπέδων γλυκόζης. Παρόλο που οι παραδοσιακοί μετρητές δεν παρέχουν συνεχή δεδομένα, αποτελούν ένα βασικό εργαλείο για την αυτοδιαχείριση του διαβήτη και βοηθούν τα άτομα να προσαρμόζουν τη διατροφή, τη φαρμακευτική αγωγή και τον τρόπο ζωής τους για καλύτερο έλεγχο της γλυκόζης.



Εικόνα 1: Μετρητής γλυκόζης αίματος

Συσκευές Συνεχούς καταγραφής γλυκόζης (CGM): Οι μετρητές γλυκόζης συνεχούς καταγραφής (Continuous Glucose Monitors - CGM) είναι καινοτόμες συσκευές που επιτρέπουν την συνεχή παρακολούθηση των επιπέδων γλυκόζης στο αίμα. Αυτές οι συσκευές χρησιμοποιούν έναν μικρό αισθητήρα που τοποθετείται κάτω από το δέρμα και μετρά τη γλυκόζη στο μεσοκυττάριο υγρό. Ο αισθητήρας στέλνει δεδομένα σε έναν αναμεταδότη, ο οποίος τα προβάλλει σε μία έξυπνη συσκευή ή σε κάποια εφαρμογή κινητού ή ιστού. Οι μετρητές CGM παρέχουν δεδομένα σε πραγματικό χρόνο, ειδοποιήσεις για υπερβολικά υψηλά ή χαμηλά επίπεδα γλυκόζης και ιστορικά δεδομένα για ανάλυση. Αυτό επιτρέπει στα άτομα με διαβήτη να διαχειρίζονται καλύτερα την

υγεία τους, βελτιώνοντας τον έλεγχο της γλυκόζης και μειώνοντας τον κίνδυνο επιπλοκών.



Εικόνα 3: Συσκευή συνεχούς καταγραφής γλυκόζης - CGM



Εικόνα 2: Εφαρμογές κινητού και ρολογιού που συγχρονίζονται με CGM

Αντλίες Ινσουλίνης: Οι αντλίες ινσουλίνης είναι προηγμένες ιατρικές συσκευές που χρησιμοποιούνται για τη συνεχή χορήγηση ινσουλίνης σε άτομα με διαβήτη, προσφέροντας μια πιο φυσική προσέγγιση στη διαχείριση της γλυκόζης. Η συσκευή αποτελείται από μια μικρή αντλία που φοριέται εξωτερικά και συνδέεται με ένα λεπτό σωληνάκι που εισάγεται υποδόρια. Η αντλία προγραμματίζεται για να παρέχει βασική ινσουλίνη σε συνεχή ροή καθ' όλη τη διάρκεια της ημέρας και μπορεί να προσαρμοστεί για να παρέχει επιπλέον δόσεις ινσουλίνης (bolus) πριν από τα γεύματα ή όταν τα επίπεδα γλυκόζης είναι υψηλά. Αυτός ο τρόπος χορήγησης επιτρέπει καλύτερη ρύθμιση των επιπέδων γλυκόζης, μειώνοντας τις διακυμάνσεις και τις πιθανότητες υπογλυκαιμίας. Οι αντλίες ινσουλίνης προσφέρουν μεγαλύτερη ευελιξία και άνεση σε σύγκριση με τις παραδοσιακές ενέσεις ινσουλίνης, βοηθώντας τα άτομα με διαβήτη να διατηρούν έναν πιο σταθερό και υγιή τρόπο ζωής. [19]



Εικόνα 4: Αντλία ινσουλίνης

Έξυπνα στυλό Ινσουλίνης: Τα έξυπνα στυλό ινσουλίνης είναι προηγμένες συσκευές χορήγησης ινσουλίνης που προσφέρουν βελτιωμένη ακρίβεια και ευκολία στη διαχείριση του σακχαρώδη διαβήτη. Αυτά τα στυλό διαθέτουν ενσωματωμένη τεχνολογία που καταγράφει και αποθηκεύει δεδομένα σχετικά με τις δόσεις ινσουλίνης, συμπεριλαμβανομένων της ποσότητας, της ώρας και της ημερομηνίας κάθε ένεσης. Μέσω Bluetooth ή άλλων ασύρματων συνδέσεων, τα δεδομένα μεταφέρονται σε ειδικές εφαρμογές σε smartphones ή άλλα έξυπνα συστήματα. Οι εφαρμογές αυτές επιτρέπουν στους χρήστες και στους ιατρούς να παρακολουθούν τις δόσεις ινσουλίνης σε συνδυασμό με άλλα δεδομένα, όπως τα επίπεδα γλυκόζης, τη διατροφή και τη φυσική δραστηριότητα, προσφέροντας μια ολιστική εικόνα της διαχείρισης του διαβήτη. Τα έξυπνα στυλό μπορούν επίσης να παρέχουν ειδοποιήσεις και υπενθυμίσεις για τις επόμενες δόσεις, βοηθώντας τους χρήστες να αποφεύγουν την υπερδοσολογία ή την παράλειψη δόσεων. Με αυτόν τον τρόπο, οι έξυπνες αυτές συσκευές συμβάλλουν σημαντικά στη βελτίωση του ελέγχου της γλυκόζης και στην προαγωγή της υγείας των ατόμων με διαβήτη.



Εικόνα 5: Έξυπνο στυλό ινσουλίνης

2. Εφαρμογή Κινητού:

Αυτές οι εφαρμογές επιτρέπουν την ανάλυση και παρουσίαση δεδομένων όπως τα επίπεδα γλυκόζης στο αίμα, την πρόσληψη τροφής, τη φυσική δραστηριότητα και τις δόσεις ινσουλίνης. Τα δεδομένα μεταφέρονται από τους μετρητές γλυκόζης, τις αντλίες ινσουλίνης και άλλες σχετικές συσκευές όπως παρουσιάστηκαν παραπάνω, στις εφαρμογές μέσω Bluetooth ή άλλων ασύρματων τεχνολογιών. Οι ιατροί έχουν τη δυνατότητα να παρακολουθούν τα δεδομένα σε πραγματικό χρόνο ή σε καθορισμένα χρονικά διαστήματα, επιτρέποντάς τους να κάνουν ενημερωμένες προσαρμογές στη θεραπευτική αγωγή και να παρέχουν πιο στοχευμένες συμβουλές στους ασθενείς. Επιπλέον, οι εφαρμογές αυτές μπορούν να στέλνουν ειδοποιήσεις για κρίσιμες αλλαγές στα επίπεδα γλυκόζης, προωθώντας την άμεση παρέμβαση και βελτιώνοντας την

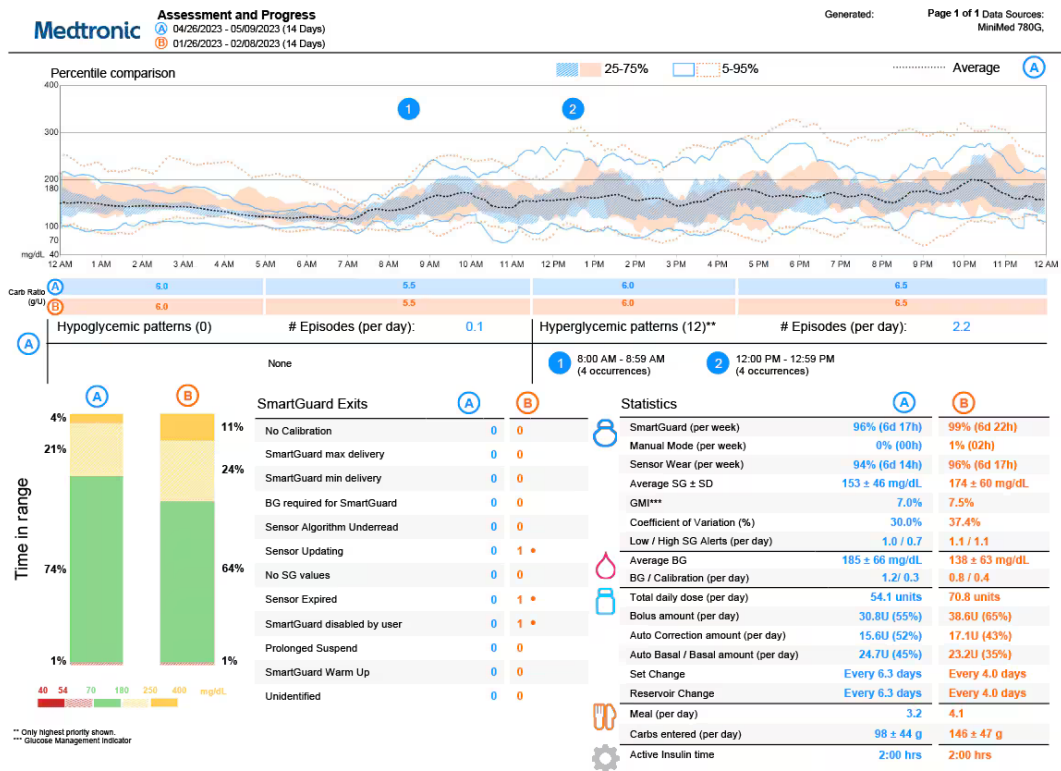
πρόληψη επιπλοκών. Με αυτόν τον τρόπο, οι ψηφιακές εφαρμογές ενισχύουν τη συνεργασία μεταξύ γιατρών και ασθενών και συμβάλλουν στην αποτελεσματικότερη διαχείριση του διαβήτη.



Εικόνα 6: Εφαρμογή κινητού για τη διαχείριση του Σακχαρώδους Διαβήτη

3. Εφαρμογή Ιστού Ιατρού

Είναι η διεπαφή μεταξύ ιατρού και διαχειριστικού μέρους του ολοκληρωμένου συστήματος. Αυτές οι εφαρμογές επιτρέπουν την ανάλυση και παρουσίαση δεδομένων όπως τα επίπεδα γλυκόζης στο αίμα, την πρόσληψη τροφής, τη φυσική δραστηριότητα και τις δόσεις ινσουλίνης. Τα δεδομένα μεταφέρονται από τους μετρητές γλυκόζης, τις αντλίες ινσουλίνης και άλλες σχετικές συσκευές όπως παρουσιάστηκαν παραπάνω, στις εφαρμογές μέσω Bluetooth ή άλλων ασύρματων τεχνολογιών. Οι ιατροί έχουν τη δυνατότητα να παρακολουθούν τα δεδομένα σε πραγματικό χρόνο ή σε καθορισμένα χρονικά διαστήματα, επιτρέποντάς τους να κάνουν ενημερωμένες προσαρμογές στη θεραπευτική αγωγή και να παρέχουν πιο στοχευμένες συμβουλές στους ασθενείς. Επιπλέον, οι εφαρμογές αυτές μπορούν να στέλνουν ειδοποιήσεις για κρίσιμες αλλαγές στα επίπεδα γλυκόζης, προωθώντας την άμεση παρέμβαση και βελτιώνοντας την πρόληψη επιπλοκών. Με αυτόν τον τρόπο, οι ψηφιακές εφαρμογές ενισχύουν τη συνεργασία μεταξύ γιατρών και ασθενών και συμβάλλουν στην αποτελεσματικότερη διαχείριση του διαβήτη.



Εικόνα 7: Εφαρμογή ιστού για τη διαχείριση του Σακχαρώδους Διαβήτη

4. Διαδίκτυο

Η εφαρμογή του κινητού και του ιστού επικοινωνούν μέσω του διαδικτύου με τον Server του συστήματος.

5. API

Είναι μία ή περισσότερες διεπαφές μεταξύ διαδικτύου και συστήματος επεξεργασίας, ανάλυσης και αποθήκευσης δεδομένων.

6. Επεξεργασία και Ανάλυση Δεδομένων

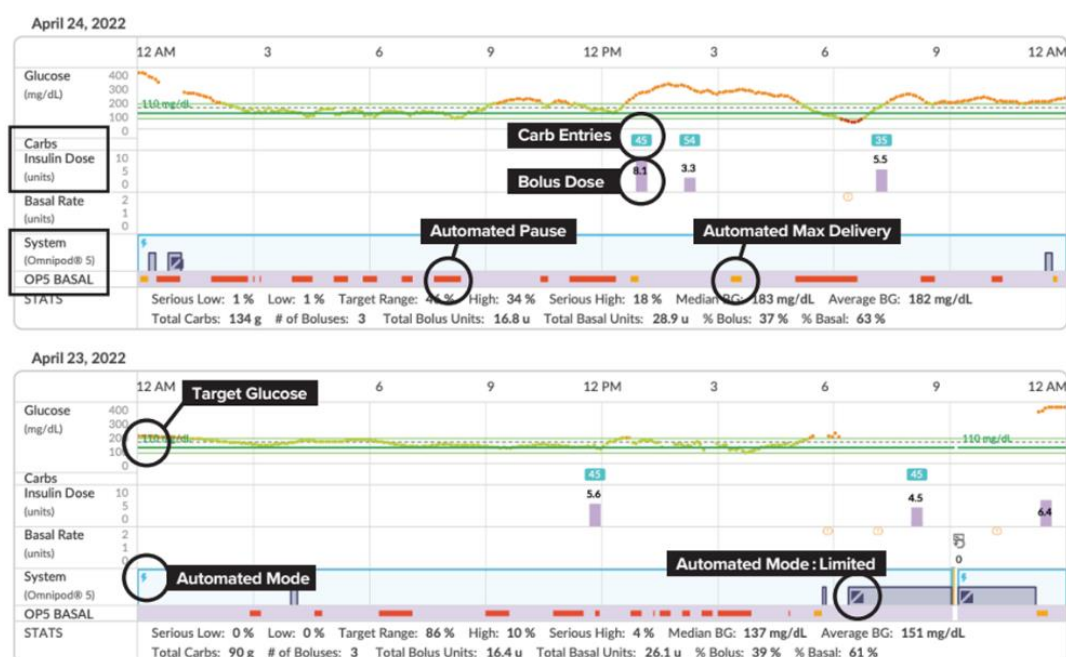
Είναι μία σειρά διεργασιών του συστήματος η οποία χρησιμοποιεί προηγμένους αλγορίθμους με στόχο να επεξεργαστεί, να αναλύσει και να αποθηκεύσει δεδομένα που σχετίζονται με τον ασθενή.

7. Βάση Δεδομένων

Εδώ αποθηκεύονται με ασφάλεια όλα τα δεδομένα που ρέουν στο σύστημα.

3.3. Ολοκληρωμένα συστήματα διαχείρισης του Σακχαρώδους Διαβήτη

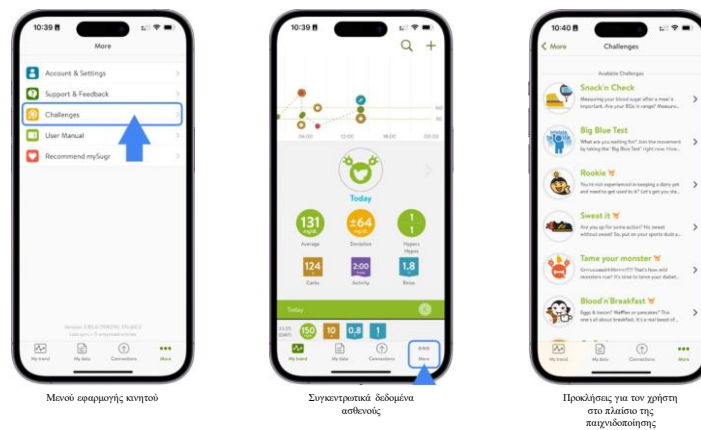
1. **Glooko** : Το Glooko είναι μια προηγμένη ψηφιακή πλατφόρμα διαχείρισης του διαβήτη, σχεδιασμένη για να διευκολύνει τη ζωή των ασθενών και των ιατρών. Η πλατφόρμα επιτρέπει τη συγχώνευση δεδομένων από διάφορες συσκευές παρακολούθησης γλυκόζης, αντλίες ινσουλίνης και έξυπνα στυλό σε μία ενιαία εφαρμογή. Μέσω του Glooko, οι ασθενείς μπορούν να καταγράφουν και να παρακολουθούν τα επίπεδα γλυκόζης τους, τις δόσεις ινσουλίνης, τη διατροφή και τη φυσική τους δραστηριότητα, προσφέροντας μια ολοκληρωμένη εικόνα της υγείας τους. Οι ιατροί, από την πλευρά τους, έχουν τη δυνατότητα να παρακολουθούν αυτά τα δεδομένα σε πραγματικό χρόνο, επιτρέποντας την ανάλυση και την προσαρμογή της θεραπείας με μεγαλύτερη ακρίβεια. Η πλατφόρμα προάγει τη συνεχή επικοινωνία μεταξύ ασθενών και ιατρών μέσω της αποστολής μηνυμάτων και ειδοποιήσεων, διευκολύνοντας την άμεση παρέμβαση και υποστήριξη. Με αυτό τον τρόπο, το Glooko συμβάλλει σημαντικά στη βελτίωση της διαχείρισης του διαβήτη και κατ' επέκταση της υγείας των ασθενών.



Εικόνα 8: Συγκεντρωτικά δεδομένα ασθενούς στο portal του Glooko

2. **MySugr** : Το MySugr είναι μια εφαρμογή κινητού για τη διαχείριση του διαβήτη. Είναι σχεδιασμένη για να απλοποιεί τη ζωή των ασθενών με μια διασκεδαστική και φιλική προς τον χρήστη προσέγγιση. Προσφέρει εύκολη καταγραφή δεδομένων και λεπτομερείς αναφορές, παρακολουθώντας τα επίπεδα σακχάρου στο αίμα, τις μετρήσεις υδατανθράκων και τη διαχείριση ινσουλίνης ενώ ταυτόχρονα προσφέρει εκτίμηση για τα επίπεδα της γλυκόζης (γλυκοζυλιωμένη αιμοσφαιρίνη). Η εφαρμογή ενσωματώνει στοιχεία παιχνιδιοποίησης όπως προκλήσεις και κίνητρα, με στόχο την κλιμακούμενη δέσμευση του ασθενούς στην παρακολούθηση της νόσου του, και την ένταξη

της στην καθημερινή του ρουτίνα. Αν και υποστηρίζει τον συγχρονισμό με πολλούς μετρητές γλυκόζης και αντλίες ινσουλίνης, το MySugr επικεντρώνεται περισσότερο στη βελτίωση της καθημερινής εμπειρίας των ασθενών παρά στην επαγγελματική ανάλυση δεδομένων και την προσαρμογή θεραπείας που παρέχει το Glooko.



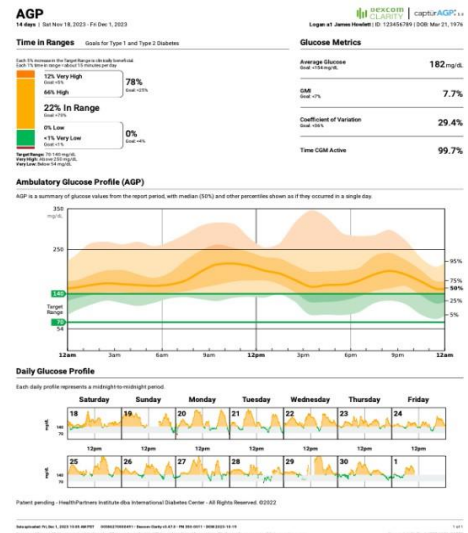
Εικόνα 9: Εφαρμογή κινητού MySugr

3. Dexcom CLARITY : Το Dexcom CLARITY είναι μια πλατφόρμα σχεδιασμένη για τη διαχείριση του διαβήτη, η οποία χρησιμοποιείται σε συνδυασμό με τα συστήματα συνεχούς καταγραφής γλυκόζης (CGM) της Dexcom. Η πλατφόρμα αυτή επιτρέπει στους χρήστες να παρακολουθούν και να αναλύουν τα δεδομένα της γλυκόζης τους σε πραγματικό χρόνο, παρέχοντας λεπτομερείς αναφορές και γραφήματα. Μέσω της εφαρμογής τους, οι ασθενείς μπορούν να παρακολουθούν τις τάσεις και τα μοτίβα της γλυκόζης τους, να βλέπουν τις διακυμάνσεις και να κατανοούν καλύτερα πώς η διατροφή, η άσκηση και η θεραπεία επηρεάζουν τα επίπεδα γλυκόζης τους.

Επιπλέον, η πλατφόρμα παρέχει στους ιατρούς εργαλεία για την παρακολούθηση των δεδομένων των ασθενών τους από απόσταση, επιτρέποντας την εξατομικευμένη προσαρμογή της θεραπείας και τη βελτίωση του ελέγχου του διαβήτη. Αυτό το ολοκληρωμένο σύστημα μπορεί να στέλνει ειδοποιήσεις και αναφορές απευθείας στους επαγγελματίες υγείας, διευκολύνοντας την επικοινωνία και τη συνεργασία μεταξύ ασθενών και ιατρών.



Εφαρμογή ασθενούς Dexcom CLARITY



Παρουσίαση δεδομένων γλυκόζης ασθενούς στην εφαρμογή ιατρού Dexcom CLARITY

Εικόνα 10: Οθόνες από τις εφαρμογές ασθενούς και ιατρού του συστήματος Dexcom-Clarity

- 4. LibreView:** Το LibreView είναι ένα ασφαλές και δωρεάν σύστημα διαχείρισης του διαβήτη σε περιβάλλον cloud. Έχει σχεδιαστεί για να βοηθά τους ανθρώπους με διαβήτη και τους θεράποντες ιατρούς τους στη διαχείριση της νόσου. Η πλατφόρμα αυτή συνδέεται με τους αισθητήρες συνεχούς παρακολούθησης γλυκόζης, όπως το FreeStyle Libre, επιτρέποντας την ασύρματη μεταφορά δεδομένων γλυκόζης σε ασφαλή διαδικτυακή τοποθεσία. Οι χρήστες μπορούν να αποκτούν λεπτομερείς αναφορές και αναλύσεις που διευκολύνουν την παρακολούθηση των επιπέδων γλυκόζης στο αίμα και την αναγνώριση μοτίβων και τάσεων για αυτή, συμβάλλοντας έτσι στη βελτιστοποίηση της θεραπείας και στη λήψη καλύτερων αποφάσεων για την υγεία τους.

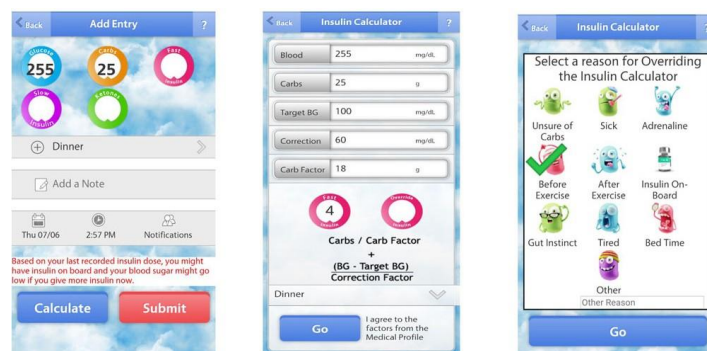


Εικόνα 11: Εφαρμογή ιστού LibreView



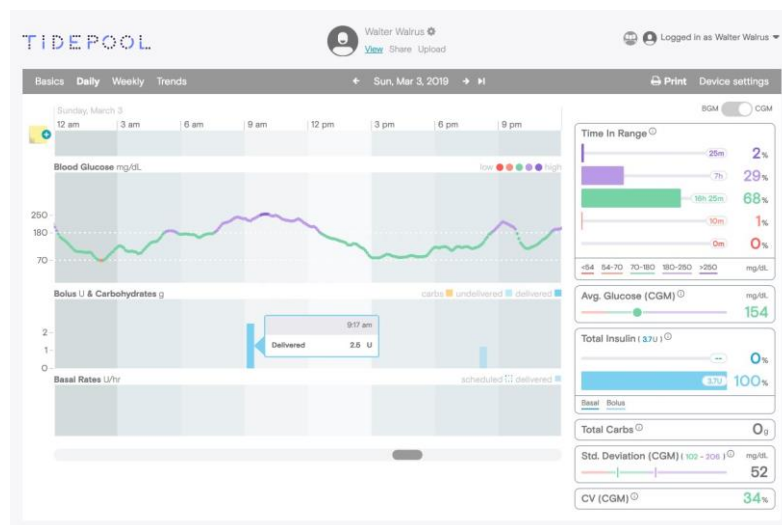
Εικόνα 12: Εφαρμογή κινητού

5. **BlueLoop:** Το BlueLoop είναι μια εφαρμογή κινητού, η οποία αναπτύχθηκε από το πρόγραμμα Diabetes Online Community (DOC) της οργάνωσης College Diabetes Network (CDN) και έχει στόχο τη διαχείριση του νεανικού διαβήτη (Σακχαρώδης Διαβήτης Τύπου 2). Είναι σχεδιασμένη τόσο για τα άτομα με διαβήτη όσο και για τους φροντιστές τους καθώς η εφαρμογή επιτρέπει την καταγραφή και παρακολούθηση σημαντικών δεδομένων υγείας, όπως τα επίπεδα γλυκόζης στο αίμα, τις δόσεις ινσουλίνης, τις διατροφικές συνήθειες και τη σωματική δραστηριότητα του ασθενούς. Επίσης διευκολύνει την αποθήκευση και την οργάνωση αυτών των πληροφοριών σε ένα κεντρικό σημείο, επιτρέποντας την εύκολη πρόσβαση και ανασκόπηση από τον χρήστη και τους θεράποντες του.



Εικόνα 13: Εφαρμογή κινητού BlueLoop

6. **Tidepool:** Το Tidepool είναι μια πλατφόρμα διαχείρισης δεδομένων για άτομα με διαβήτη, σχεδιασμένη για να βοηθά στη συγκέντρωση, στην ανάλυση και στην παρακολούθηση τους. Η πλατφόρμα επιτρέπει στους χρήστες να συλλέγουν δεδομένα από διάφορες συσκευές, όπως οι αντλίες ινσουλίνης, τις συσκευές CGM και τους μετρητές γλυκόζης αίματος, και να τα συγκεντρώνουν σε ένα κεντρικό σημείο. Μέσω αυτής της πλατφόρμας, οι χρήστες λαμβάνουν λεπτομερείς αναφορές και γραφήματα που βοηθούν στην αναγνώριση μοτίβων και τάσεων στα επίπεδα γλυκόζης, βελτιώνοντας έτσι τη λήψη αποφάσεων σχετικά με τη διαχείριση της νόσου. Το Tidepool προσφέρει επίσης τη δυνατότητα κοινής χρήσης αυτών των δεδομένων με τους ιατρούς τους, διευκολύνοντας τη συνεργασία και την εξατομικευμένη φροντίδα.

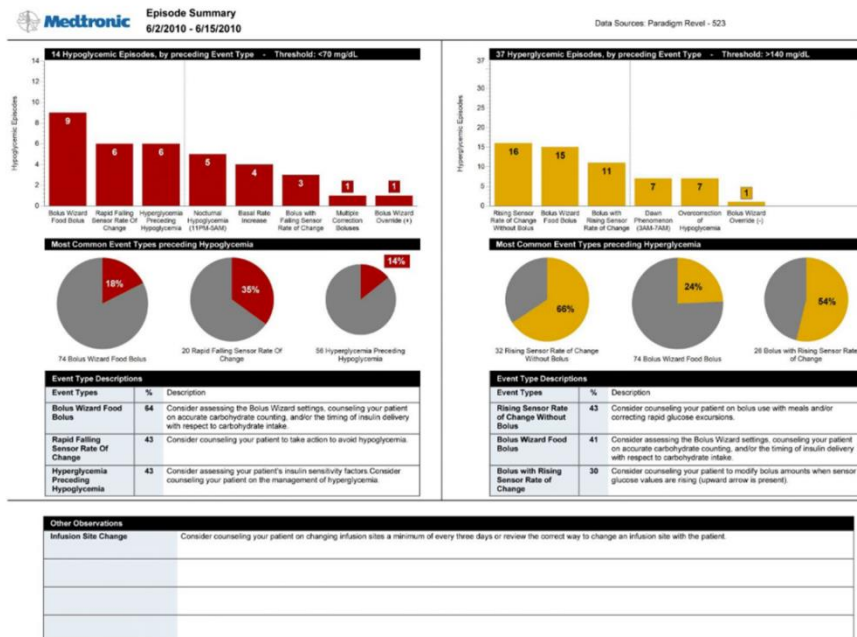


Εικόνα 14: Εφαρμογή Tidepool

7. **CareLink:** Το CareLink είναι μια πλατφόρμα διαχείρισης του διαβήτη που αναπτύχθηκε από τη Medtronic και σχεδιάστηκε για να βοηθά άτομα με διαβήτη αλλά και τους επαγγελματίες υγείας τους στη διαχείριση της νόσου μέσω της ανάλυσης δεδομένων. Η πλατφόρμα επιτρέπει στους χρήστες να μεταφορτώνουν δεδομένα από τις αντλίες ινσουλίνης και τα συστήματα συνεχούς παρακολούθησης γλυκόζης (CGMs) της Medtronic σε ένα ασφαλές, διαδικτυακό περιβάλλον. Αυτά τα δεδομένα μπορούν να μετατραπούν σε λεπτομερείς αναφορές και γραφήματα που διευκολύνουν την αναγνώριση μοτίβων και τάσεων στα επίπεδα γλυκόζης και τις δόσεις ινσουλίνης. Μέσω του CareLink, οι επαγγελματίες υγείας μπορούν να αποκτούν πρόσβαση σε αυτά τα δεδομένα, επιτρέποντας την καλύτερη παρακολούθηση και προσαρμογή της θεραπείας με βάση τις ανάγκες του κάθε ασθενούς, ενισχύοντας έτσι την αποτελεσματικότητα της διαχείρισης του διαβήτη.

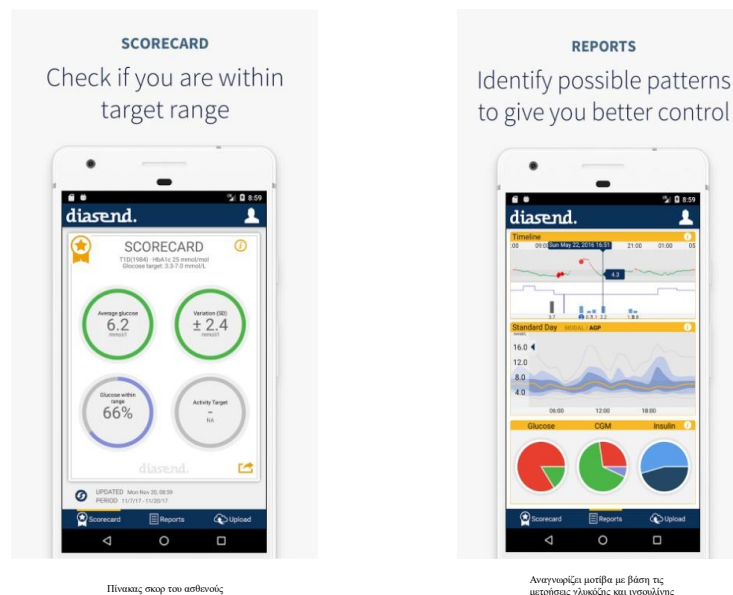


Εικόνα 15: Οθόνη παρακολούθησης ασθενών του CareLink - Εφαρμογή ιατρού

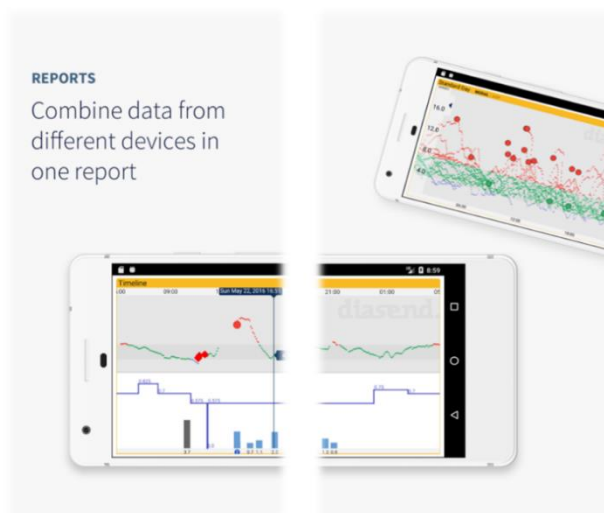


Εικόνα 16: Οθόνη από την εφαρμογή του ασθενούς που παρουσιάζει τα χαρακτηριστικά ενός επεισοδίου

8. **Diasend:** Το Diasend είναι μια ολοκληρωμένη πλατφόρμα διαχείρισης του διαβήτη που επιτρέπει στους ασθενείς και τους επαγγελματίες υγείας να συλλέγουν, να αναλύουν και να μοιράζονται δεδομένα από διάφορες συσκευές υγείας. Συγκεκριμένα, η πλατφόρμα συλλέγει δεδομένα από αντλίες ινσουλίνης, μετρητές γλυκόζης αίματος και συσκευές συνεχούς παρακολούθησης γλυκόζης (CGMs). Συγκεντρώνοντας όλα τα δεδομένα σε ένα ενιαίο, εύχρηστο περιβάλλον, μέσω του Diasend, οι χρήστες μπορούν να δημιουργούν λεπτομερείς αναφορές και γραφήματα που απεικονίζουν τα επίπεδα γλυκόζης, τη χρήση ινσουλίνης και άλλες σχετικές παραμέτρους, διευκολύνοντας την παρακολούθηση και τη διαχείριση της νόσου. Η πλατφόρμα προσφέρει επίσης τη δυνατότητα ασφαλούς κοινής χρήσης αυτών των δεδομένων με τους επαγγελματίες υγείας, επιτρέποντας την πιο αποτελεσματική και εξατομικευμένη προσαρμογή της θεραπείας, με στόχο τη βελτίωση της ποιότητας ζωής των ατόμων με διαβήτη.



Εικόνα 18: Οθόνη σκορ και οθόνη ανάλυσης αποτελεσμάτων της εφαρμογής κινητού του Diasend

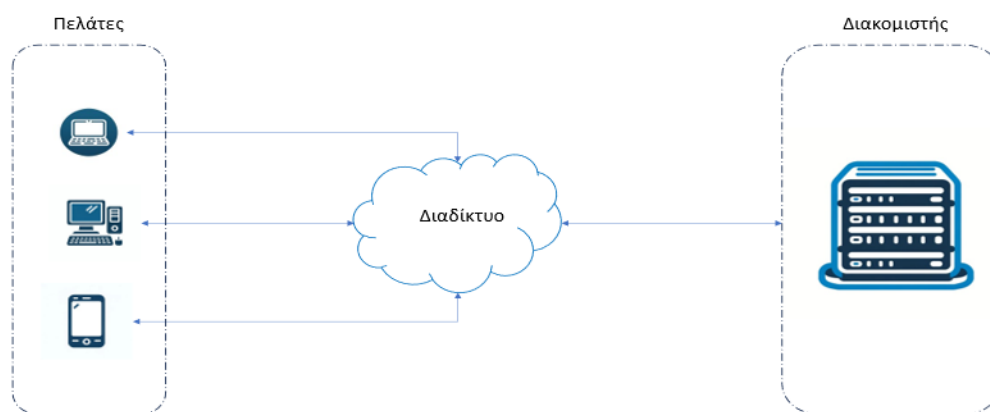


Εικόνα 17: Οθόνη με αναφορές από την εφαρμογή κινητού του Diasend

4. Τεχνολογίες για την ανάπτυξη συστήματος διακομιστή έξυπνων εφαρμογών (Back End)

4.1 Εισαγωγή

Το πιο διαδεδομένο μοντέλο αρχιτεκτονικής δικτύου για τον σχεδιασμό εφαρμογών και συστημάτων, είναι το μοντέλο πελάτη – διακομιστή (client – server model). Χρησιμοποιείται τόσο σε μικρής όσο και σε μεγάλης κλίμακας περιβάλλοντα και αυτό διότι εμφανίζει σημαντικά πλεονεκτήματα έναντι άλλων αρχιτεκτονικών. Προσφέρει την κεντροποίηση των πόρων και των δεδομένων, διευκολύνοντας τη διαχείριση, την ενημέρωση και την εφαρμογή ισχυρών μέτρων ασφαλείας. Επίσης προσθέτοντας διακομιστές ή ενισχύοντας τους υπάρχοντες το φορτίο των αιτημάτων μοιράζεται με τον βέλτιστο τρόπο έτσι ώστε να αποφευχθεί η κατάρρευση του συστήματος. Όλα αυτά τα χαρακτηριστικά διασφαλίζουν υψηλή και αξιόπιστη διαθεσιμότητα υπηρεσιών στον τελικό χρήστη. Μία τέτοια αρχιτεκτονική φαίνεται παρακάτω:



Σχήμα 4.1: Αρχιτεκτονική Μοντέλου Πελάτη – Διακομιστή

Όπως φαίνεται στο παραπάνω σχήμα, το σύστημα αποτελείται από δύο επίπεδα, το επίπεδο του πελάτη ή αλλιώς το επίπεδο παρουσίασης των δεδομένων, και το επίπεδο του διακομιστή ή επίπεδο πρόσβασης στα δεδομένα του συστήματος, είτε αυτά αντλούνται απευθείας από τη βάση, είτε προσφέρονται στο πρώτο επίπεδο κατόπιν διεργασιών επεξεργασίας και ανάλυσης. Στη μηχανική λογισμικού το επίπεδο παρουσίασης ονομάζεται Front-End ενώ το επίπεδο πρόσβασης στα δεδομένα ονομάζεται Back-End. Στο μοντέλο client – server, το Front-End είναι ο client και το Back-End είναι ο server.

4.2 Επίπεδο Πρόσβασης στα δεδομένα (Back – End)

Ο όρος "Back-End" αναφέρεται γενικά στο επίπεδο ενός ολοκληρωμένου συστήματος που δεν είναι προσβάσιμο στους τελικούς χρήστες. Περιλαμβάνει την υποδομή και τη λογική από την πλευρά του διακομιστή που εξασφαλίζουν τη λειτουργικότητα και την απόδοση της εφαρμογής. Αυτό το μέρος του συστήματος είναι υπεύθυνο για τη διαχείριση όλου του πλάνου λογικών διεργασιών της εφαρμογής, την αποθήκευση και

την ανάκτηση δεδομένων, και την εξασφάλιση της ασφάλειας και της απόδοσης του. Παρέχει τα δεδομένα και τις υπηρεσίες που χρειάζεται η διεπαφή του χρήστη, δηλαδή το επίπεδο παρουσίασης (Front-End) για να λειτουργεί αποτελεσματικά. Στην ουσία, το Back-End είναι ο κινητήρας που οδηγεί τις βασικές λειτουργίες της εφαρμογής, παρέχοντας μια απρόσκοπτη εμπειρία στον τελικό χρήστη.

Τα πιο σημαντικά στοιχεία ενός Back-End είναι ο διακομιστής (server), η βάση δεδομένων, ο κώδικας που υλοποιεί τη λογική στην πλευρά του διακομιστή, τα API, το ενδιάμεσο λογισμικό (middleware) και οι μηχανισμοί ασφαλείας. Αυτά τα στοιχεία συνεργάζονται για να εξασφαλίσουν την ομαλή λειτουργία, την επεκτασιμότητα και την ασφάλεια των εφαρμογών του Front-End. Στη συνέχεια αναλύονται τα παραπάνω.

4.2.1 Διακομιστής (Server)

Ο διακομιστής είναι ένα ισχυρό υπολογιστικό σύστημα το οποίο παρέχει πόρους, δεδομένα, υπηρεσίες ή εφαρμογές σε άλλους υπολογιστές, γνωστούς ως πελάτες (clients). Αποτελεί την πλατφόρμα στην οποία εκτελείται και είναι διαθέσιμη μία εφαρμογή στους χρήστες. Μπορεί να είναι ένας φυσικός ή εικονικός υπολογιστής ο οποίος λειτουργεί στο νέφος (cloud), παρέχοντας τους πόρους που είναι απαραίτητοι για τη λειτουργία του λογισμικού Back-End. Επιπλέον οι διακομιστές είναι υπεύθυνοι για τη διαχείριση των ηλεκτρονικών μηνυμάτων και την αποθήκευση δεδομένων. Έχουν σχεδιαστεί για να διαχειρίζονται πολλαπλά αιτήματα ταυτόχρονα, εξασφαλίζοντας ότι τα συστήματα πελατών μπορούν να έχουν πρόσβαση στους απαραίτητους πόρους αποτελεσματικά και αξιόπιστα.

Είναι ισχυρά εξοπλισμένοι έτσι ώστε να διαχειρίζονται βαριά φορτία εργασίας και να διατηρούν υψηλή διαθεσιμότητα και απόδοση. Εκτός από τους λειτουργικούς τους ρόλους, οι διακομιστές είναι ζωτικής σημασίας για την ασφάλεια και την επεκτασιμότητα των συστημάτων πληροφορικής. Συχνά εφοδιάζονται με προηγμένα μέτρα ασφαλείας, όπως τείχη προστασίας (firewalls), κρυπτογράφηση και συστήματα ανίχνευσης εισβολών, για την προστασία ευαίσθητων δεδομένων και την αποτροπή μη εξουσιοδοτημένης πρόσβασης.

Η τακτική συντήρηση και οι ενημερώσεις είναι απαραίτητες για τη διατήρηση της ομαλής και ασφαλούς λειτουργίας τους. Η επεκτασιμότητα είναι επίσης σημαντική διότι επιτρέπει στους διακομιστές να διαχειρίζονται αυξανόμενους όγκους δεδομένων και κίνησης καθώς αυξάνεται η ζήτηση. Τεχνικές όπως η εξισορρόπηση φορτίου (load balancing), η εικονικοποίηση (virtualization) και η υπολογιστική νέφος (cloud computing) βοηθούν στην κατανομή των φορτίων εργασίας αποτελεσματικά και παρέχουν την ευελιξία που απαιτείται για τη δυναμική επέκταση των πόρων.

Με τη βελτιστοποίηση της ασφάλειας και της επεκτασιμότητας, οι διακομιστές παίζουν καθοριστικό ρόλο στη διατήρηση της αξιοπιστίας και της αποτελεσματικότητας των ψηφιακών λειτουργιών σε διάφορους τομείς.

4.2.2 Βάση Δεδομένων

Οι βάσεις δεδομένων είναι οργανωμένες συλλογές δομημένων πληροφοριών ή δεδομένων που αποθηκεύονται ηλεκτρονικά και διαχειρίζονται από ένα Σύστημα Διαχείρισης Βάσεων Δεδομένων (DBMS). Η βασική τους λειτουργία είναι η αποθήκευση, ανάκτηση και διαχείριση μεγάλων όγκων δεδομένων με αποτελεσματικό τρόπο. Υπάρχουν διάφοροι τύποι βάσεων, όπως οι σχεσιακές (π.χ. MySQL, PostgreSQL), οι NoSQL (π.χ. MongoDB, Cassandra), οι γραφικές (π.χ. Neo4j), οι βάσεις δεδομένων σε πραγματικό χρόνο (π.χ. Firebase). Οι σχεσιακές βάσεις χρησιμοποιούν τη γλώσσα SQL για τη διαχείριση των δεδομένων, διασφαλίζοντας την ακεραιότητα τους μέσω των ιδιοτήτων ACID (Ατομικότητα, Συνοχή, Απομόνωση, Ανθεκτικότητα). Οι NoSQL βάσεις δεδομένων παρέχουν μεγαλύτερη ευελιξία για την επεξεργασία μη δομημένων δεδομένων και είναι κατάλληλες για εφαρμογές μεγάλων δεδομένων και διαδικτυακές εφαρμογές που λειτουργούν σε πραγματικό χρόνο.

Στην επιστήμη της πληροφορικής, οι βάσεις δεδομένων διαδραματίζουν κρίσιμο ρόλο στο back end συστημάτων, καθώς προσφέρουν την υποδομή για τη διαχείριση και αποθήκευση δεδομένων που χρησιμοποιούν οι εφαρμογές. Μέσω προηγμένων αλγορίθμων και τεχνικών, οι βάσεις εξασφαλίζουν την ασφάλεια, την ακεραιότητα και τη διαθεσιμότητα των δεδομένων. Τεχνικές όπως η αναπαραγωγή και η δημιουργία αντιγράφων ασφαλείας εγγυώνται την απρόσκοπτη λειτουργία των βάσεων ακόμη και σε περιπτώσεις βλαβών. Οι σύγχρονες βάσεις υλοποιούνται ψηφιακά, ελέγχονται από DBMS και αποτελούν αντικείμενο επιστημονικής και τεχνικής μελέτης. Η πρόοδος στις τεχνολογίες βάσεων δεδομένων, όπως οι βάσεις δεδομένων στη μνήμη και οι κατακευματισμένες βάσεις δεδομένων, ενισχύει την απόδοση και επεκτασιμότητα, υποστηρίζοντας τις αυξανόμενες απαιτήσεις των σύγχρονων εφαρμογών και υπηρεσιών.

4.2.3 Server-Side Logic

Ο όρος server side logic, αναφέρεται στον κώδικα που εκτελείται στον διακομιστή για τη διαχείριση των αιτημάτων και των δεδομένων των χρηστών πριν επιστραφούν στον πελάτη. Αυτός ο κώδικας περιλαμβάνει λειτουργίες όπως η επεξεργασία φορμών, η επαλήθευση δεδομένων, η διαδικασία πιστοποίησης ταυτότητας των χρηστών και η διασύνδεση με βάσεις για την αποθήκευση και την ανάκτηση των απαραίτητων πληροφοριών. Η λογική με τη οποία έχει εφοδιαστεί ο διακομιστής είναι κρίσιμη για την ασφάλεια και την αποτελεσματικότητα της εφαρμογής, καθώς επιτρέπει την κεντρική διαχείριση των δεδομένων και διασφαλίζει ότι οι χρήστες αλληλοεπιδρούν με αξιόπιστα και με δεδομένα που ελέγχονται.

Συνήθως γράφεται σε γλώσσες προγραμματισμού, όπως η C#, η Java, η Python, η Ruby ή η JavaScript (Node.js), χρησιμοποιώντας τυποποιημένα περιβάλλοντα εργασίας (frameworks) που παρέχουν δομή και βοηθητικά εργαλεία για τη δημιουργία αξιόπιστων εφαρμογών. Η χρήση server side logic μειώνει το φορτίο στον πελάτη και αυξάνει την ασφάλεια, καθώς οι ευαίσθητες λειτουργίες και τα δεδομένα δεν είναι άμεσα προσβάσιμα στον τελικό χρήστη.

Παράλληλα μειώνει τον όγκο των δεδομένων που μετακινούνται μέσω του δικτύου, μεταφέροντας μόνο τις άμεσα απαραίτητες και ταυτόχρονα επεξεργασμένες πληροφορίες (αποτελέσματα) που εξυπηρετούν το κάθε αίτημα. Με την υλοποίηση αυτής της λογικής στον διακομιστή, οι εφαρμογές μπορούν να παραμένουν ευέλικτες, ασφαλείς και επεκτάσιμες, προσφέροντας μια συνεπή και αξιόπιστη εμπειρία χρήστη.

4.2.4 API (Application Programming Interfaces)

Τα API (Application Programming Interfaces) είναι ένα σύνολο κανόνων και πρωτοκόλλων τα οποία επιτρέπουν σε διαφορετικές εφαρμογές λογισμικού να επικοινωνούν μεταξύ τους. Καθορίζουν τις μεθόδους και τη μορφή των δεδομένων που μπορούν να χρησιμοποιήσουν οι εφαρμογές για να υποβάλλουν αιτήματα και να ανταλλάξουν πληροφορίες.

Βασικό στοιχείο των API είναι τα endpoints. Τα endpoints είναι συγκεκριμένες διευθύνσεις URL ή διαδρομές μέσα σε ένα API οι οποίες καθορίζουν τους «προσβάσιμους δρόμους» σε συγκεκριμένες λειτουργίες ή δεδομένα μιας εφαρμογής. Κάθε endpoint αναφέρεται σε μία συγκεκριμένη λειτουργία ή πηγή δεδομένων ή οποία εκτελείται ή ανακτάται καλώντας μέσω του endpoint μία μέθοδο HTTP. Το HTTP είναι το βασικό πρωτόκολλο που χρησιμοποιείται για τη μεταφορά δεδομένων στον παγκόσμιο ιστό.

Λειτουργεί σε ένα μοντέλο client – server όπου ο client στέλνει ένα αίτημα στον server και εκείνος με τη σειρά του επιστρέφει την απάντηση. Το HTTP ορίζει μια σειρά από μεθόδους που εκτελούν συγκεκριμένες εντολές για ένα συγκεκριμένο πόρο.

Οι πιο συνηθισμένες μέθοδοι είναι :

- η GET, η οποία ζητά δεδομένα από έναν διακομιστή,
- η POST, η οποία υποβάλλει δεδομένα προς επεξεργασία,
- η PUT, η οποία ενημερώνει έναν υπάρχοντα πόρο με νέα δεδομένα και
- η DELETE, η οποία αφαιρεί τον καθορισμένο πόρο.

Αυτές οι μέθοδοι ολοκληρώνουν την επικοινωνία μεταξύ API και εφαρμογής του front – end.

Όπως αναφέρθηκε ήδη παραπάνω, κάθε endpoint στην ουσία διαβιβάζει ένα αίτημα του client στον server. Κάθε φορά που εκτελείται ένα αίτημα μέσω μίας HTTP μεθόδου, ο server επιστρέφει μία απάντηση. Αυτή η απάντηση έχει συγκεκριμένη μορφή και μπορεί να περιλαμβάνει την πληροφορία όπως έρχεται από τη βάση ή φιλτραρισμένη με βάση τους κανόνες που έχουμε ορίσει εμείς. Επίσης οι μέθοδοι HTTP POST και HTTP PUT εκτελούν αιτήματα τα οποία έχουν ως είσοδο κάποια δεδομένα. Τα δεδομένα που μεταφέρονται είτε ως αίτημα είτε ως απάντηση, έχουν συγκεκριμένη μορφή και δομή έτσι ώστε διαφορετικά συστήματα να μπορούν να επικοινωνούν μεταξύ τους με ευκολία. Οι πιο συνηθισμένες μορφές είναι οι JSON (JavaScript Object Notation) και XML (eXtensible Markup Language).

Τα API είναι ζωτικής σημασίας για τη λειτουργία των συστημάτων και την ανάπτυξη πολυσύνθετων εφαρμογών. Επιτρέπουν στους προγραμματιστές να ενσωματώνουν λειτουργίες από τρίτες πηγές, όπως συστήματα πληρωμών, υπηρεσίες χαρτογράφησης και πλατφόρμες κοινωνικών δικτύων, βελτιώνοντας έτσι την εμπειρία του χρήστη και μειώνοντας τον χρόνο ανάπτυξης. Επιπλέον, τα API διευκολύνουν τη συντήρηση και την αναβάθμιση των εφαρμογών, καθώς οι αλλαγές μπορούν να γίνονται μεμονωμένα σε κάθε τμήμα χωρίς να επηρεάζονται τα υπόλοιπα.

Εκτός από την ευελιξία και την αποδοτικότητα, τα API είναι υπεύθυνα για την ασφάλεια των δεδομένων. Μέσω μηχανισμών όπως η πιστοποίηση (authentication) και η εξουσιοδότηση (authorization), τα API μπορούν να προστατεύουν ευαίσθητες πληροφορίες και να διασφαλίζουν ότι μόνο εξουσιοδοτημένοι χρήστες έχουν πρόσβαση σε συγκεκριμένες λειτουργίες και δεδομένα.

Συνολικά, τα API αποτελούν έναν αναπόσπαστο μέρος της σύγχρονης ανάπτυξης λογισμικού, προωθώντας την καινοτομία και επιτρέποντας τη δημιουργία εξελιγμένων και διασυνδεδεμένων εφαρμογών. Η χρήση τους είναι κρίσιμη για την αποτελεσματική και ασφαλή επικοινωνία μεταξύ διαφορετικών συστημάτων, καθιστώντας τα ένα ισχυρό εργαλείο για προγραμματιστές και επιχειρήσεις.

4.2.5 Ασφάλεια

Η ασφάλεια ενός Back-End συστήματος αποτελεί ζωτικό συστατικό για την προστασία των δεδομένων και την ακεραιότητα των εφαρμογών που υποστηρίζει. Ένα από τα βασικότερα μέτρα είναι η κρυπτογράφηση δεδομένων, τόσο κατά τη μεταφορά με χρήση SSL/TLS, όσο και σε κατάσταση ηρεμίας, για την αποτροπή υποκλοπής και μη εξουσιοδοτημένης πρόσβασης. Επιπλέον, η χρήση μηχανισμών πιστοποίησης και εξουσιοδότησης διασφαλίζει ότι μόνο οι εξουσιοδοτημένοι χρήστες έχουν πρόσβαση σε ευαίσθητες πληροφορίες και λειτουργίες.

Τα firewalls και τα συστήματα ανίχνευσης εισβολών προσθέτουν επιπλέον επίπεδα προστασίας, αποτρέποντας κακόβουλες επιθέσεις και ανιχνεύοντας ύποπτη δραστηριότητα. Οι τακτικοί έλεγχοι ασφαλείας και οι ενημερώσεις λογισμικού είναι απαραίτητοι για την αντιμετώπιση νέων και εξελισσόμενων απειλών, διασφαλίζοντας ότι τα συστήματα παραμένουν ασφαλή και ενημερωμένα.

Η διαχείριση των κλειδιών κρυπτογράφησης και η χρήση ασφαλών πρωτοκόλλων για την αποθήκευση και μεταφορά αυτών των κλειδιών είναι επίσης κρίσιμα στοιχεία. Επιπλέον, η εφαρμογή πολιτικών για την τακτική αλλαγή κωδικών πρόσβασης και η εκπαίδευση των χρηστών σχετικά με τις βέλτιστες πρακτικές ασφαλείας συμβάλλουν στην ενίσχυση της συνολικής ασφαλείας του συστήματος.

Η υλοποίηση μέτρων εφεδρείας και ανάκαμψης από καταστροφές διασφαλίζει ότι τα δεδομένα μπορούν να ανακτηθούν και οι υπηρεσίες να αποκατασταθούν γρήγορα σε περίπτωση βλάβης ή απώλειας δεδομένων. Αυτές οι πρακτικές περιλαμβάνουν τη δημιουργία τακτικών αντιγράφων ασφαλείας και τη διατήρηση εφεδρικών υποδομών που μπορούν να ενεργοποιηθούν σε περίπτωση έκτακτης ανάγκης.

Τέλος, η συνεχής παρακολούθηση και αξιολόγηση των μέτρων ασφαλείας μέσω διαδικασιών όπως οι δοκιμές διείσδυσης (penetration testing) και οι αναλύσεις κινδύνου, επιτρέπουν την έγκαιρη αναγνώριση και αντιμετώπιση αδυναμιών στο σύστημα. Η ολοκληρωμένη προσέγγιση στην ασφάλεια ενός Back-End συστήματος διασφαλίζει την εμπιστευτικότητα, την ακεραιότητα και τη διαθεσιμότητα των δεδομένων και των εφαρμογών, προστατεύοντας τόσο την επιχείρηση όσο και τους χρήστες της από απειλές και παραβιάσεις ασφαλείας.

4.3 Εργαλεία και Μέθοδοι Ελέγχου και Δοκιμών του Back-End

Ο έλεγχος και οι δοκιμές είναι κρίσιμα για τη διασφάλιση της αξιοπιστίας, της ασφαλείας και της απόδοσης όχι μόνο του Back-End αλλά και της λειτουργίας του συνολικού συστήματος δηλαδή Front-End & Back-End. Υπάρχουν διάφοροι μέθοδοι και εργαλεία που μπορούν να χρησιμοποιηθούν για αυτήν τη διαδικασία.

4.3.1 Τρόποι Ελέγχου και Δοκιμών

1. Δοκιμές Μονάδας (Unit Testing):

Ελέγχουν την ορθή λειτουργία μεμονωμένων μονάδων κώδικα. Τα εργαλεία όπως το JUnit για Java και το pytest για Python είναι ευρέως διαδεδομένα. Συνήθως πραγματοποιούνται από τον μηχανικό που έχει αναπτύξει τη συγκεκριμένη μονάδα προκειμένου να επιβεβαιώσει ότι έχει ολοκληρώσει την ανάπτυξη με βάση τις προδιαγραφές.

2. Δοκιμές Ενοποίησης (Integration Testing):

Βεβαιώνουν ότι οι διαφορετικές μονάδες του κώδικα λειτουργούν σωστά μαζί. Χρησιμοποιούνται εργαλεία όπως το Selenium και το Postman για δοκιμές API.

3. Λειτουργικές Δοκιμές (Functional Testing):

Εξασφαλίζουν ότι οι λειτουργίες της εφαρμογής εκτελούνται όπως αναμένεται. Τα εργαλεία που χρησιμοποιούνται περιλαμβάνουν το FitNesse και το Cucumber.

4. Δοκιμές Απόδοσης (Performance Testing):

Μετρούν την ταχύτητα, την ανταπόκριση και τη σταθερότητα της εφαρμογής υπό διαφορετικά φορτία. Εργαλεία όπως το Apache JMeter και το LoadRunner είναι δημοφιλή για αυτό το σκοπό.

5. Δοκιμές Ασφαλείας (Security Testing):

Εντοπίζουν ευπάθειες που μπορεί να εκμεταλλευτούν κακόβουλοι χρήστες και λογισμικά. Χρησιμοποιούνται εργαλεία όπως το OWASP ZAP και το Burp Suite.

6. Δοκιμές Καπνού (Smoke Testing):

Βεβαιώνουν ότι οι βασικές λειτουργίες της εφαρμογής λειτουργούν σωστά μετά από μια νέα έκδοση ή αλλαγή. Αυτές οι δοκιμές είναι συνήθως αυτοματοποιημένες και γρήγορες.

4.3.2 Διαδικασία Δοκιμών

1. **Σχεδιασμός Δοκιμών:** Καθορισμός των περιπτώσεων δοκιμών και των σεναρίων που θα ελεγχθούν.
2. **Ανάπτυξη Δοκιμών:** Αναλυτική καταγραφή των δοκιμών με βάση τον σχεδιασμό.
3. **Εκτέλεση Δοκιμών:** Εκτέλεση των δοκιμών σε περιβάλλον ανάπτυξης ή εξειδικευμένο περιβάλλον δοκιμών.
4. **Ανάλυση Αποτελεσμάτων:** Εξέταση των αποτελεσμάτων για ανίχνευση σφαλμάτων ή προβλημάτων.
5. **Διόρθωση Σφαλμάτων:** Επιδιόρθωση των σφαλμάτων και επανεκτέλεση των δοκιμών.

Οι σωστές και πλήρεις δοκιμές του Back-End διασφαλίζουν την αξιοπιστία και την αποδοτικότητα της εφαρμογής, προλαμβάνοντας πιθανά προβλήματα πριν αυτά φτάσουν στον τελικό χρήστη. Η συνεχής βελτίωση των διαδικασιών και των εργαλείων δοκιμών είναι απαραίτητη για την διατήρηση υψηλών επιπέδων ποιότητας και ασφάλειας στην ανάπτυξη λογισμικού.

5. Σχεδιασμός συστήματος υποστήριξης κλινικών αποφάσεων για τη διαχείριση του σακχαρώδους διαβήτη

Σε αυτό το κεφάλαιο παρουσιάζεται το σύστημα πρόσβασης και διαχείρισης των αιτημάτων και δεδομένων για την υποστήριξη δύο εφαρμογών που αφορούν τη διαχείριση του Σακχαρώδους Διαβήτη τύπου 2.

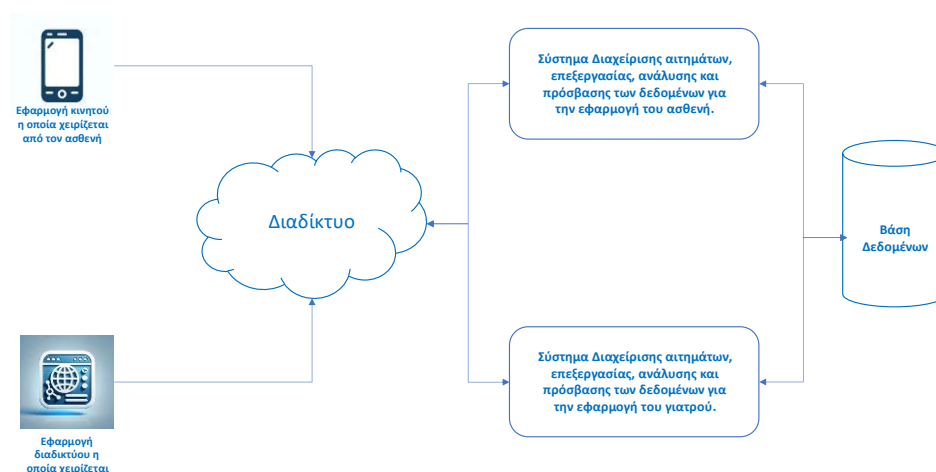
5.1 Αρχιτεκτονική του Ολοκληρωμένου Συστήματος Διαχείρισης του Σακχαρώδους Διαβήτη Τύπου 2

Όπως αναφέρθηκε στο τρίτο κεφάλαιο, κάθε αντίστοιχου τύπου σύστημα, έχει ως στόχο τη διαχείριση των δεδομένων ασθενών που νοσούν από τον ΣΔ. Το σύστημα μπορεί να χωριστεί με βάση τις εξής διαστάσεις:

- A. Τη διαχείριση των δεδομένων ανάλογα με τον ρόλο του χρήστη:
 1. Οργάνωση και καταγραφή της καθημερινότητας του ασθενούς και
 2. Επίβλεψη της συνολικής εικόνας του από τον γιατρό.

- B. Τα επίπεδα παρουσίασης και διαχείρισης:
 1. Επίπεδο παρουσίασης των δεδομένων (Front-End) και
 2. Επίπεδο πρόσβασης και διαχείρισης των δεδομένων (Back-End).

Το γενικό σχήμα της αρχιτεκτονικής του ολοκληρωμένου συστήματος που εξυπηρετεί η παρούσα διπλωματική εργασία φαίνεται στο παρακάτω σχήμα:



Σχήμα 5.1: Αρχιτεκτονική Ολοκληρωμένου Συστήματος Διαχείρισης Σακχαρώδους Διαβήτη Τύπου 2

5.1.1 Επίπεδο παρουσίασης των δεδομένων του συστήματος (Front-End)

Εφαρμογή έξυπνου κινητού – Εφαρμογή του ασθενούς

Όπως φαίνεται στο σχήμα 5.1 το Front-End του συγκεκριμένου συστήματος ικανοποιεί τον διαχωρισμό που κάναμε σχετικά με την διαχείριση του ΣΔΤ2. Έτσι στο Front-End έχουμε μία εφαρμογή έξυπνου κινητού (mobile application) η οποία χρησιμοποιείται από τον ασθενή και μία εφαρμογή διαδικτύου (web application) η οποία χρησιμοποιείται από τον γιατρό.

Στην εφαρμογή του ο ασθενής, έχει τη δυνατότητα να καταγράφει τις καθημερινές του συνήθειες. Υπάρχουν λίστες έτοιμων πιθανών απαντήσεων οι οποίες επιλέγονται από τον χρήστη μέσω διάφορων στοιχείων όπως κουμπάκια επικύρωσης εισαγωγής (radio button groups) και κουτάκια επιλογής (select boxes). Ο ασθενής θα πρέπει να εισάγει καθημερινά τα γεύματα που καταναλώνει μέσα στη μέρα, την άσκηση και τη διάθεση του.

Επίσης πέρα από τις καθημερινές του συνήθειες, εισάγει δεδομένα που σχετίζονται με την κλινική του εικόνα όπως τη μέτρηση της γλυκόζης, την αρτηριακή του πίεση, το βάρος του, το αν νοσεί από κάποια άλλη ασθένεια και τη φαρμακευτική αγωγή που παίρνει καθημερινά.

Με βάση τις μετρήσεις της γλυκόζης που εισάγει, μπορεί να δει την εξέλιξη της εικόνας του με τη μορφή γραφικών παραστάσεων που παρουσιάζουν αποτελέσματα σε βάθος χρόνου μίας εβδομάδας, 15 ημερών και ενός μήνα.

Εφαρμογή διαδικτύου – Εφαρμογή του ιατρού

Ο ιατρός μέσω της εφαρμογής του μπορεί να παρακολουθεί τόσο την τρέχουσα εικόνα των ασθενών του όσο και το ιστορικό τους. Κάθε ασθενής έχει μία καρτέλα πληροφοριών που αφορούν κάποια βασικά προσωπικά δεδομένα, στοιχεία του ιστορικού του, καθώς και κλινικά και εργαστηριακά αποτελέσματα. Ο ιατρός έχει τη δυνατότητα να προσθέτει ασθενείς στη λίστα του με βάση το ΑΜΚΑ τους αλλά και να δημιουργεί καρτέλες νέων ασθενών.

Για κάθε ασθενή έχει τη δυνατότητα να προσθέτει τα αποτελέσματα των εξετάσεων που πραγματοποίησε, να βλέπει ξεχωριστά κάθε εξέταση και να δημιουργεί φόρμες ερωτήσεων – απαντήσεων για κάθε μία από αυτές. Μέσω υποβολής αιτήματος στον ασθενή που έχει αναλάβει τη διαχείριση και θεραπεία του, μπορεί να βλέπει σε μορφή γραφικών παραστάσεων τις μετρήσεις γλυκόζης που εισάγει ο ασθενής στο σύστημα μέσω της εφαρμογής του κινητού.

Τέλος μπορεί να δει κάποια στατιστικά επί του συνόλου των ασθενών του, που αφορούν την ηλικία, το φύλο και το κάπνισμα σε μορφή γραφημάτων.

Οι δύο εφαρμογές του Front-End επικοινωνούν μέσω του διαδικτύου (Internet) με το Back-End του συστήματος.

5.1.2 Επίπεδο πρόσβασης και διαχείρισης των δεδομένων του συστήματος (back – end)

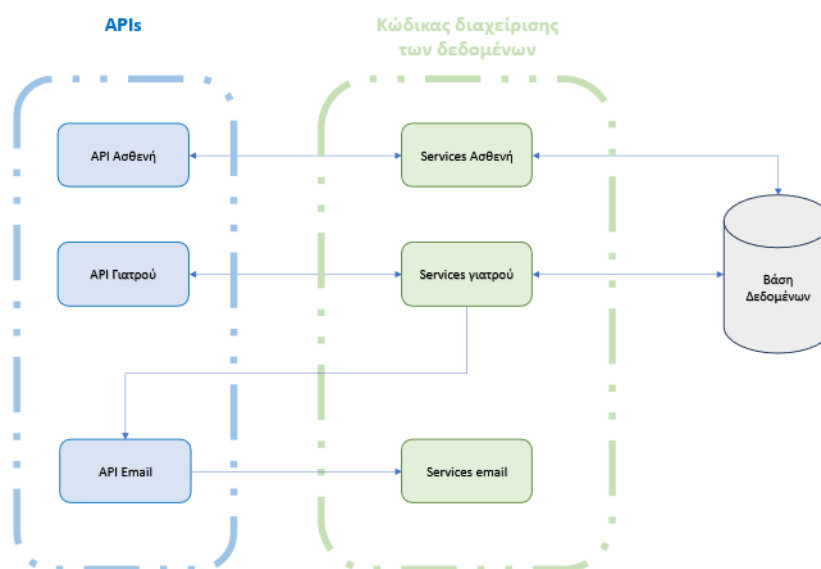
Το Back-End του συστήματος αποτελείται από το κομμάτι διαχείρισης και επεξεργασίας των αιτημάτων και των δεδομένων και τη βάση δεδομένων. Η διαχείριση των δεδομένων χωρίζεται σε δύο μέρη όπου το ένα εξυπηρετεί την εφαρμογή του ασθενούς και το άλλο την εφαρμογή του ιατρού. Τα δύο αυτά μέρη επικοινωνούν με τη βάση και αντλούν δεδομένα τόσο από ξεχωριστούς όσο και από κοινούς πίνακες.

5.2 Τεχνολογία υλοποίησης του Back-End

Η γλώσσα που χρησιμοποιήθηκε είναι η C# και η υλοποίηση του Back-End έγινε με χρήση στοιχείων που προσφέρονται από το .NET Core framework. Πρόκειται για ένα, ανοιχτού κώδικα framework το οποίο έχει σχεδιαστεί από τη Microsoft για την ανάπτυξη εφαρμογών ιστού και κινητού αλλά και διάφορων άλλων υπηρεσιών που βασίζονται κυρίως σε τεχνολογίες cloud. Ως cross platform χαρακτηρίζεται η ικανότητα ενός λογισμικού να εκτελείται σε διαφορετικά λειτουργικά συστήματα και υπολογιστικά περιβάλλοντα. Αυτή η ικανότητα διευκολύνει τους επαγγελματίες της πληροφορικής να σχεδιάζουν και να υλοποιούν κώδικα ο οποίος δε θα χρειάζεται επεμβάσεις ή και ανακατασκευή για να μπορεί να εγκατασταθεί και να λειτουργήσει σε Windows, Linux, macOS, Android, iOS κ.α. Αυτό γίνεται εφικτό μέσω ενός περιβάλλοντος που είναι υπεύθυνο να εκτελεί τέτοιου τύπου κώδικα και μιας σειράς βιβλιοθηκών που έχουν σχεδιαστεί έτσι ώστε να είναι συμβατές με πολλαπλά λειτουργικά συστήματα.

5.3 Αρχιτεκτονική του Back-End

Όπως φαίνεται στο παρακάτω σχήμα, το Back-End του συστήματος αποτελείται από τρία βασικά μέρη. Τα API, τον κώδικα που είναι υπεύθυνος για την άντληση, την επεξεργασία, την ανάλυση και την αποθήκευση δεδομένων, και τέλος τη βάση δεδομένων.

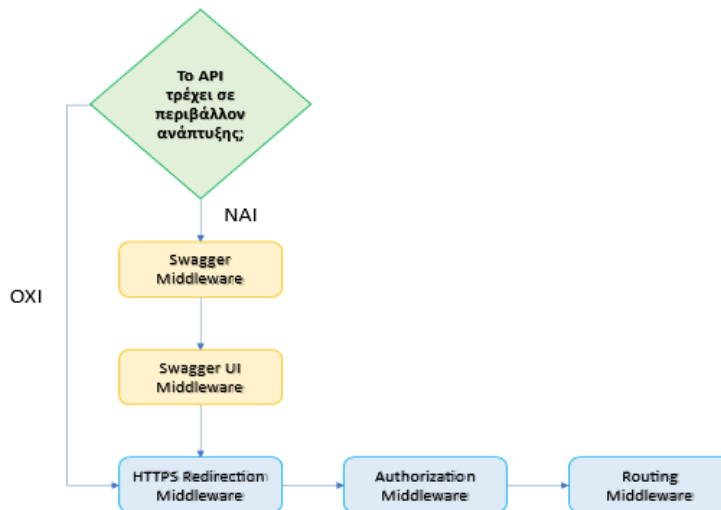


5.3.1 API

Η επικοινωνία μεταξύ του Front-End και του Back-End επιτυγχάνεται μέσω των API. Τα API έχουν υλοποιηθεί με την τεχνολογία του ASP .NET Core framework, το οποίο ανήκει στο οικοσύστημα του .NET Core και είναι ειδικά σχεδιασμένο για να δημιουργεί κώδικα που αξιοποιεί το HTTP πρωτόκολλο (HTTP services). Αυτά τα services μπορούν να χρησιμοποιηθούν για να εξυπηρετήσουν ένα ευρύ φάσμα clients συμπεριλαμβανομένων εφαρμογών ιστού, εφαρμογών κινητού αλλά και συστημάτων Back-End. Η δρομολόγηση των αιτημάτων (routing) γίνεται μέσω κλάσεων που ονομάζονται ελεγκτές (controllers) οι οποίοι περιλαμβάνουν συγκεκριμένες ομάδες endpoints που συνδέονται με διαφορετικές λειτουργίες του Back-End.

Πολύ σημαντικό για τη λειτουργία, την ευελιξία και την επεκτασιμότητα του συστήματος, είναι ότι τα ASP .NET Core API παρέχουν τη δυνατότητα στον μηχανικό λογισμικού να χρησιμοποιήσει το μοντέλο σχεδιασμού Dependency Injection (DI). Στην ουσία αυτό το σχεδιαστικό μοντέλο καθιστά διαθέσιμα προς εκτέλεση εντός ενός controller, τμήματα κώδικα που επιτελούν συγκεκριμένες λειτουργίες του συστήματος αλλά υλοποιούνται σε άλλες κλάσεις εκτός controller. Αυτό παρέχει τη δυνατότητα να αποφευχθεί η εκτέλεση μεγάλου όγκου κώδικα εντός μίας HTTP μεθόδου με αποτέλεσμα τον καλύτερο διαχωρισμό των διαδικασιών του συστήματος. Με αυτό τον τρόπο, αν χρειαστούν αλλαγές στον κώδικα του Back-End, αυτές θα γίνουν σε ένα συγκεκριμένο σημείο χωρίς να επηρεαστούν άλλα μέρη του συστήματος.

Η διαχείριση των αιτημάτων στο ASP .NET Core ονομάζεται request handling pipeline και είναι η εκτέλεσης μίας ακολουθίας διαφορετικών τμημάτων κώδικα που ονομάζονται middlewares και που βρίσκονται ανάμεσα στο λειτουργικό σύστημα και την εφαρμογή. Ένα middleware έχει τη δυνατότητα να διαχειρίζεται αιτήματα, να τα τροποποιεί, να τα διαβιβάζει στο επόμενο middleware ή να βραχυκυκλώνει το pipeline δημιουργώντας και επιστρέφοντας απευθείας κάποια απάντηση σε αυτά. Στο παρακάτω σχήμα φαίνεται το request handling pipeline για τα API του Back-End.



Σχήμα 5.3: Rest handling pipeline

Swagger Middleware – Swagger UI Middleware

Αρχικά γίνεται έλεγχος για το αν το API τρέχει σε περιβάλλον ανάπτυξης ή περιβάλλον παραγωγής. Αν τρέχει σε περιβάλλον ανάπτυξης τα πρώτα middlewares του pipeline είναι αυτά που ανοίγουν ένα εργαλείο δοκιμών που ονομάζεται Swagger. Μέσω του Swagger ο μηχανικός λογισμικού έχει τη δυνατότητα να δοκιμάζει άμεσα τα endpoints που έχει υλοποιήσει και να αντλεί πληροφορίες για τυχόν σφάλματα που μπορεί να προκύψουν κατά την εκτέλεση του κώδικα. Ο Swagger χωρίζει τα endpoints ανά controller, δείχνει τη μορφή και τη δομή που έχουν τα δεδομένα που εισάγονται για την εκτέλεση μίας μεθόδου ή τα δεδομένα που θα πάρουμε ως απάντηση από αυτή. Με αυτό τον τρόπο διευκολύνει τις δοκιμές και τη διασύνδεση του Back-End με το Front-End.

HTTPS Redirection Middleware

Επόμενο middleware στο pipeline είναι το HTTPS Redirection που όταν λαμβάνει ένα HTTP αίτημα το ανακατευθύνει σε HTTPS αίτημα έτσι ώστε η επικοινωνία μεταξύ client – server να είναι κρυπτογραφημένη και ασφαλής.

Authorization Middleware

Αυτό το middleware ελέγχει αν ο χρήστης είναι εξουσιοδοτημένος να αποκτήσει στο πρόσβαση στους πόρους που αιτείται. Αν είναι πιστοποιημένος χρήστης τότε επιτρέπει την πρόσβαση στο επόμενο middleware του pipeline, αν όχι τότε η ακολουθία τερματίζει και το middleware επιστρέφει στον χρήστη τον HTTP κωδικό κατάστασης 401. Ο κωδικός αυτός σημαίνει ότι ο client που υποβάλλει το αίτημα δεν είναι εξουσιοδοτημένος.

Routing Middleware

Το τελευταίο middleware του pipeline, ενσωματώνει τους controllers του API στο σύστημα δρομολόγησης του back – end. Αυτό πρακτικά σημαίνει ότι όταν έρχεται ένα HTTP αίτημα, το σύστημα δρομολόγησης θα αναζητήσει το endpoint στον αντίστοιχο controller.

Σε αυτό το Back-End υπάρχουν τρία API:

- Ένα API που αναλαμβάνει τα αιτήματα που υποβάλλει ο ασθενής μέσω της εφαρμογής κινητού,
- Ένα API που εξυπηρετεί τα αιτήματα που υποβάλλει ο ιατρός μέσω της εφαρμογής ιστού και
- ένα API που αναλαμβάνει να στέλνει αυτοματοποιημένα μηνύματα στους χρήστες των δύο εφαρμογών.

Πολύ σημαντική διαδικασία για τη λειτουργία του συστήματος, είναι η πιστοποίηση του χρήστη μέσω του API. Το HTTP είναι ένα πρωτόκολλο χωρίς κατάσταση, το οποίο σημαίνει ότι κάθε αίτημα που δέχεται το API είναι ανεξάρτητο από προηγούμενα αιτήματα, άρα το API δεν έχει καμία γνώση πέρα από το τρέχον αίτημα. Αυτό μπορεί να δημιουργήσει αρκετές δυσκολίες διότι σε μία εφαρμογή ο μεγαλύτερος όγκος δεδομένων που αιτείται να λάβει ο εκάστοτε χρήστης, συνδέονται με τον λογαριασμό του. Άρα, είναι απαραίτητο να πιστοποιηθεί ότι το αίτημα που έλαβε ο διακομιστής προήλθε από τον αντίστοιχο χρήστη, προτού του δώσει πρόσβαση σε ευαίσθητες πληροφορίες. Αυτόν τον ρόλο αναλαμβάνει η Json Web Token εξουσιοδότηση (JWT).

Στο μοντέλο JWT, ο διακομιστής επιστρέφει ένα κρυπτογραφημένο διακριτικό στον πελάτη. Αυτό το κρυπτογραφημένο διακριτικό είναι ουσιαστικά ένα αντικείμενο JSON που περιέχει πληροφορίες της ταυτότητας του, οι οποίες κρυπτογραφήθηκαν χρησιμοποιώντας ένα κρυφό κλειδί από τον διακομιστή. Σε επόμενα αιτήματα HTTP, ο πελάτης πρέπει απλώς να περάσει το JWT στην επικεφαλίδα και ο διακομιστής έχει τη δυνατότητα αποκρυπτογράφησης του διακριτικού και ελέγχου ταυτότητας του χρήστη. [14]

Ένα βασικό πλεονέκτημα του JWT είναι ότι εξαλείφει την ανάγκη για προσωρινή αποθήκευση στη μνήμη ή στη βάση δεδομένων κάποιας πληροφορίας για την αναγνώριση του χρήστη. Τα τελευταία χρόνια, ο έλεγχος ταυτότητας JSON Web Token έχει διαδοθεί στον κόσμο των εφαρμογών. Αυτό οφείλεται στην άνοδο της αρχιτεκτονικής REST στην ανάπτυξη εφαρμογών. Αυτό που κάνει τα JSON Web Tokens πολύ προσαρμοσμένα στις υπηρεσίες RESTful είναι ότι μπορούν να αποθηκεύουν δεδομένα σχετικά με τον χρήστη και τα προνόμιά του και να αποστέλλονται εύκολα και με ασφάλεια κάθε φορά που ο πελάτης καλεί ένα end point. Σε κάθε αίτημα, ο διακομιστής αποκωδικοποιεί το κρυπτογραφημένο JWT, αναγνωρίζει τον χρήστη, ελέγχει αν έχει το δικαίωμα πρόσβασης, εκπληρώνει το αίτημα που ζητά και απαντά ανάλογα. [15]

5.3.2 Κώδικας Διαχείρισης των Δεδομένων

Αυτό το μέρος του back – end περιλαμβάνει τον κώδικα που καλείται στους controllers των API και είναι υπεύθυνο για την διαχείριση, ανάλυση και επεξεργασία των δεδομένων του συστήματος. Αποτελείται από κλάσεις που ονομάζονται Services και περιέχουν μεθόδους οι οποίες εκτελούν συγκεκριμένες λειτουργίες που σχετίζονται με τη διαχείριση των δεδομένων. Χαρακτηρίζεται ως καλή πρακτική, κάθε controller να έχει και ένα αντίστοιχο service που θα καλεί για να υλοποιεί τη λογική διαχείρισης του αιτήματος που δέχεται. Όπως έχει ήδη αναφερθεί, κάθε controller δίνει πρόσβαση σε συγκεκριμένες ομάδες endpoints που διαχειρίζονται αιτήματα τα οποία αναφέρονται σε παρόμοια δεδομένα. Με αυτό τον τρόπο, κάθε controller περιλαμβάνει endpoints που καλούν το αντίστοιχο service το οποίο αναλαμβάνει την υλοποίηση της λογικής του Back-End για τα δεδομένα αυτά.

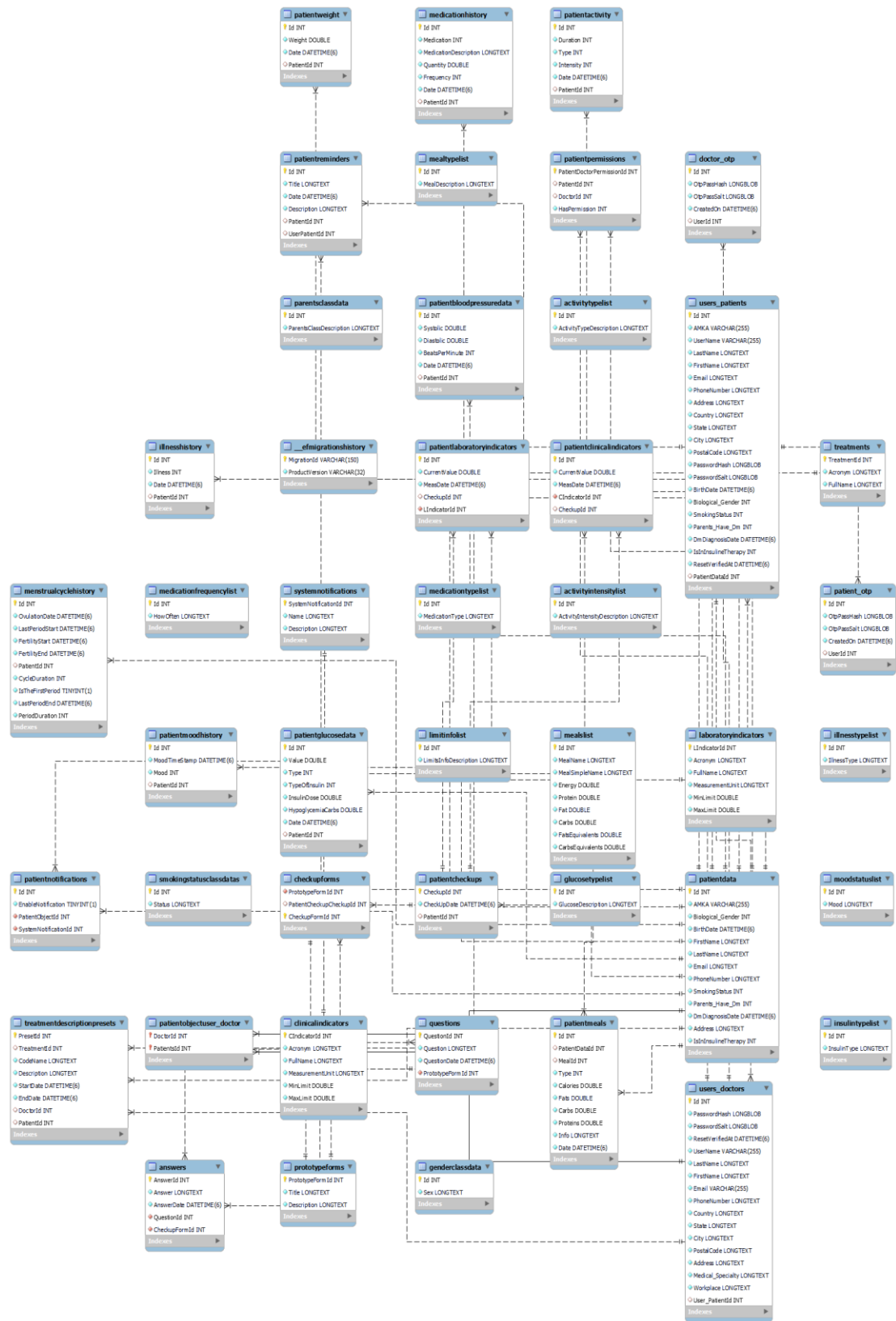
Συγκεκριμένα σε αυτή την εργασία, υπάρχουν τρεις κατηγορίες services, μία για κάθε API. Κάθε API έχει τόσα services όσοι είναι και οι αντίστοιχοι controllers του. Αυτό διευκολύνει στις δοκιμές και τη διαχείριση σφαλμάτων, διότι κατηγοριοποιώντας τον κώδικα ανάλογα με το είδος της «δουλειάς» που καλείται να εκτελέσει, είναι δυνατές διορθώσεις ή και αλλαγές βελτιστοποίησης, πολύ πιο σύντομα και στοχευμένα.

5.3.3 Βάση Δεδομένων

Η βάση που επιλέχθηκε για την οργάνωση και αποθήκευση των δεδομένων του συστήματος είναι η MySQL. Πρόκειται για ένα σύστημα διαχείρισης δεδομένων σχεσιακής βάσης (Relational Database Management System – RDBMS), το οποίο είναι ανοικτού κώδικα και χρησιμοποιείται ευρέως για την εξυπηρέτηση διάφορων εφαρμογών. Τέλος χρησιμοποιεί τη γλώσσα SQL για την πρόσβαση, τη διαχείριση και τον χειρισμό των δεδομένων της.

Ο σχεδιασμός της βάσης έγινε με την code – first μέθοδο. Με αυτή τη μέθοδο αρχικά σχεδιάζονται οι κλάσεις – μοντέλα του συστήματος και έπειτα το .NET μέσω του Entity Framework Core αναλαμβάνει τον σχεδιασμό της βάσης αντίστοιχα με τον τρόπο που ορίστηκαν τα μοντέλα. Το Entity Framework Core είναι ένα εργαλείο το οποίο εφαρμόζει μία τεχνική μετατροπής δεδομένων μεταξύ συστημάτων διαφορετικών τύπων που είναι μεταξύ τους ασύμβατοι, σε αντικειμενοστραφείς γλώσσες προγραμματισμού. Με αυτό τον τρόπο είναι δυνατή η διαχείριση των δεδομένων της βάσης χωρίς την ανάγκη υλοποίησης κώδικα με SQL αλλά γράφοντας αποκλειστικά C#.

Παρακάτω φαίνεται το φυσικό σχήμα της βάσης (σελ.):



Σχήμα 5.4: Φυσικό σχήμα της βάσης

Σημείωση: Στα επόμενα κεφάλαια θα παρουσιαστούν με μεγαλύτερη λεπτομέρεια οι πίνακες, τα πεδία και οι σχέσεις όπως χρησιμοποιούνται από το σύστημα

6. Υλοποίηση συστήματος υποστήριξης κλινικών αποφάσεων για τη διαχείριση του σακχαρώδους διαβήτη

Με βάση τα όσα αναφέρθηκαν προηγουμένως, αυτό το κεφάλαιο θα χωριστεί σε 4 ενότητες. Η πρώτη ενότητα περιγράφει την αρχιτεκτονική της βάσης δεδομένων, αναφέρει όλους τους πίνακες οντοτήτων που διαχειρίζεται το σύστημα και τους περιγράφει αναλυτικά. Οι επόμενες δύο περιγράφουν τη λογική που εκτελεί ο κώδικας του Back-End για την ικανοποίηση των αιτημάτων του ασθενούς και του ιατρού. Η τελευταία ενότητα περιγράφει τη λογική που εκτελεί ο κώδικας που σχετίζεται με το email.

6.1 Αρχιτεκτονική της Βάσης Δεδομένων και ανάλυση των πινάκων οντοτήτων

Στο σχήμα 5.4 του προηγούμενου κεφαλαίου, φαίνεται το φυσικό σχήμα της βάσης δεδομένων. Τα δεδομένα μας είναι χωρισμένα σε 4 μεγάλες κατηγορίες, στα δεδομένα που αφορούν την εφαρμογή του ιατρού, στα δεδομένα που αφορούν την εφαρμογή του ασθενούς, στα δεδομένα που μοιράζονται οι δύο χρήστες και τους επιτρέπουν να αλληλοεπιδρούν και τέλος στα σταθερά δεδομένα που σχετίζονται με τη λειτουργία του συστήματος.

6.1.1 Δεδομένα που αφορούν την εφαρμογή του ιατρού

Τα δεδομένα που σχετίζονται με την εφαρμογή του ιατρού μπορούμε να τα κατηγοριοποιήσουμε σε πέντε ομάδες.

1. Τα δεδομένα που σχετίζονται με την πιστοποίηση του χρήστη – ιατρού,
2. τα δεδομένα που σχετίζονται με το ιστορικό των ασθενών του,
3. τα δεδομένα που σχετίζονται με τις εξετάσεις του ασθενούς,
4. τα δεδομένα που σχετίζονται με φόρμες που συμπληρώνει ο ιατρός για τον ασθενή ανά εξέταση και
5. τα δεδομένα που σχετίζονται με τις θεραπείες που προτείνει ο ιατρός στον ασθενή.

Δεδομένα Πιστοποίησης χρήστη – ιατρού

Ο παρακάτω πίνακας περιλαμβάνει δεδομένα για την οντότητα του χρήστη – ιατρού (User_Doctor) και περιέχει πληροφορίες που σχετίζονται με την πρόσβαση στον λογαριασμό του (UserName, Password):

users_doctors	
Id (Primary Key)	Αριθμός εγγραφής στη βάση
PasswordHash	Συμβολοσειρά αναπαράστασης του κωδικού πρόσβασης έπειτα από κρυπτογράφηση με τον hashing αλγόριθμο.
PasswordSalt	Συμβολοσειρά που μπαίνει στην αρχή του κωδικού πρόσβασης και πριν τη διαδικασία του hashing.
ResetVerifiedAt	Ημερομηνία που πιστοποιήθηκε η αλλαγή κωδικού από τον χρήστη.
UserName	Όνομα χρήστη
LastName	Επώνυμο Χρήστη
FirstName	Μικρό Όνομα Χρήστη
Email	Διεύθυνση ηλεκτρονικού ταχυδρομίου
PhoneNumber	Αριθμός Τηλεφώνου
Country	Χώρα
State	Περιφέρεια
City	Πόλη
PostalCode	Ταχυδρομικός Κώδικας
Medical_Specialty	Ειδικότητα του ιατρού
Workplace	Το Νοσοκομείο που εργάζεται

Πίνακας 6.1.1.1: Χρήστης – ιατρός

Ο παρακάτω πίνακας περιλαμβάνει δεδομένα για την οντότητα του κωδικού μίας χρήσης του ιατρού (Doctor_OneTimePassword):

doctor_otp	
Id (Primary Key)	Αριθμός εγγραφής στη βάση
OtpPassHash	Συμβολοσειρά αναπαράστασης του κωδικού μίας χρήσης έπειτα από κρυπτογράφηση με τον hashing αλγόριθμο.
OtpPassSalt	Συμβολοσειρά που μπαίνει στην αρχή του κωδικού μίας χρήσης και πριν τη διαδικασία του hashing.
CreatedOn	Ημερομηνία δημιουργίας του κωδικού μίας χρήσης
UserId (Foreign Key)	Αριθμός συσχέτισης του κωδικού μίας πρόσβασης με τον χρήστη γιατρό.

Πίνακας 6.1.1.2: Κωδικός μίας χρήσης

Οι οντότητες User_Doctor και Doctor_OneTimePassword , σχετίζονται μεταξύ τους με τη σχέση One to Many που σημαίνει ότι ένας χρήστης μπορεί να έχει πολλούς κωδικούς μίας χρήσης.

Δεδομένα ιστορικού των ασθενών

Ο παρακάτω πίνακας περιλαμβάνει δεδομένα για την οντότητα του ιστορικού ενός ασθενή (PatientObject):

patientData	
Id (Primary Key)	Αριθμός εγγραφής στη βάση
AMKA	ΑΜΚΑ του ασθενούς
Biological_Gender	Αριθμός που δείχνει το φύλλο του ασθενούς. Παίρνει τις τιμές 1 για το θηλυκό και 2 για το αρσενικό.
BirthDate	Ημερομηνία Γέννησης του ασθενούς
FirstName	Μικρό Όνομα Ασθενή
LastName	Επώνυμο Ασθενή
Email	Διεύθυνση ηλεκτρονικού ταχυδρομίου

PhoneNumber	Αριθμός Τηλεφώνου
SmokingStatusClass	Αριθμός που δείχνει την κατάσταση του ασθενούς σε σχέση με το κάπνισμα. Η τιμή 1 αντιστοιχεί σε ασθενή που δεν ήταν ποτέ καπνιστής, η τιμή 2 σε πρώην καπνιστή και η τιμή 3 σε καπνιστή.
Parents_Have_Dm	Αριθμός που δείχνει το ιστορικό του ασθενούς όσον αφορά τον διαβήτη. Η τιμή 1 αντιστοιχεί στην περίπτωση που κανένας από τους δύο γονείς του είχαν διαβήτη, η τιμή 2 στην περίπτωση που η μητέρα του ασθενούς είχε διαβήτη, η τιμή 3 στην περίπτωση που ο πατέρας του ασθενούς είχε διαβήτη και η τιμή 4 στην περίπτωση που και οι δύο γονείς έχουν ιστορικό.
DmDiagnosisDate	Ημερομηνία Διάγνωσης
Address	Διεύθυνση Κατοικίας
IsInInsulineTherapy	Λογική τιμή που περιγράφει αν ο ασθενής είναι σε θεραπείας ινσουλίνης.

Πίνακας 6.1.1.3: Ιστορικό κάποιου ασθενή

Οι οντότητες `User_Doctor` και `PatientObject`, συνδέονται μεταξύ τους με σχέση `Many to Many`. Αυτό σημαίνει ότι ένας ιατρός μπορεί να παρακολουθεί το ιστορικό πολλών ασθενών και το κάθε ιστορικό ασθενή να παρακολουθείται από πολλούς ιατρούς.

Ο παρακάτω πίνακας είναι ένας πίνακας που χρησιμοποιεί η βάση ως ενδιάμεσο (`join table`) για να υλοποιήσει τη παραπάνω σχέση `Many to Many` και περιλαμβάνει τα κλειδιά των οντοτήτων `User_Doctor` και `PatientObject`. Το όνομά του είναι `patientobjectuser_doctor`.

patientObjectUser_doctor	
DoctorId (Foreign Key)	Id του χρήστη ιατρού της εφαρμογής
PatientsId (Foreign Key)	Id του ιστορικού του ασθενούς

Πίνακας 6.1.1.4: Πίνακας σχέσεων ιατρού – ιστορικού ασθενή

Όπως φαίνεται παραπάνω, ο κάθε ιατρός συνδέεται με σχέση One to Many με την οντότητα σχέση ιατρού – ιστορικού ασθενή και το κάθε ιστορικό ασθενή με μία σχέση One to Many με την οντότητα ιατρού – ιστορικού ασθενή.

Δεδομένα που σχετίζονται με τις εξετάσεις του ασθενούς

Ο παρακάτω πίνακας περιλαμβάνει δεδομένα για την οντότητα ενός κλινικού δείκτη (ClinicalIndicator):

clinicalIndicators	
CIndicatorId (Primary Key)	Αριθμός εγγραφής στη βάση
Acronym	Συντομογραφία κλινικού δείκτη
FullName	Πλήρες όνομα κλινικού δείκτη
MeasurementUnit	Μονάδα Μέτρησης
MinLimit	Κατώτατο Όριο
MaxLimit	Ανώτατο Όριο

Πίνακας 6.1.1.5: Πίνακας κλινικών δεικτών

Ο παρακάτω πίνακας περιλαμβάνει δεδομένα για την οντότητα ενός εργαστηριακού δείκτη (LaboratoryIndicator):

laboratoryIndicators	
LIndicatorId (Primary Key)	Αριθμός εγγραφής στη βάση
Acronym	Συντομογραφία εργαστηριακού δείκτη
FullName	Πλήρες όνομα εργαστηριακού δείκτη
MeasurementUnit	Μονάδα Μέτρησης
MinLimit	Κατώτατο Όριο

MaxLimit	Ανώτατο Όριο
-----------------	--------------

Πίνακας 6.1.1.6: Πίνακας κλινικών δεικτών

Ο παρακάτω πίνακας περιλαμβάνει δεδομένα για την οντότητα μίας εξέτασης για ένα συγκεκριμένο ασθενή (PatientCheckup). Η οντότητα PatientCheckup σχετίζεται με σχέση One to Many με τον πίνακα PatientObject, δηλαδή το ιστορικό ενός ασθενή μπορεί να σχετίζεται με πολλές εξετάσεις.

patientCheckups	
CheckupId (Primary Key)	Αριθμός εγγραφής στη βάση
CheckupDate	Ημερομηνία πραγματοποίησης της εξέτασης
PatientId (ForeignKey)	Id του ιστορικού του ασθενούς

Πίνακας 6.1.1.7: Πίνακας εξετάσεων ασθενή

Ο παρακάτω πίνακας περιλαμβάνει δεδομένα για την οντότητα της μέτρησης ενός κλινικού δείκτη συγκεκριμένου ασθενή (PatientClinicalIndicators). Η οντότητα PatientClinicalIndicators και η οντότητα ClinicalIndicator συσχετίζονται με One to Many σχέση, δηλαδή ένας κλινικός δείκτης μπορεί να έχει πολλές μετρήσεις.

patientClinicalIndicators	
Id (Primary Key)	Αριθμός εγγραφής στη βάση
CurrentValue	Τιμή του κλινικού δείκτη του ασθενούς
MeasDate	Ημερομηνία λήψης της μέτρησης
CIIndicatorId (Foreign Key)	Id του κλινικού δείκτη
CheckupId (Foreign Key)	Id της εξέτασης για την οποία έγινε η λήψη της μέτρησης αυτού του δείκτη

Πίνακας 6.1.1.8: Πίνακας μετρήσεων κλινικών δεικτών

Ο παρακάτω πίνακας περιλαμβάνει δεδομένα για την οντότητα της μέτρησης ενός εργαστηριακού δείκτη συγκεκριμένου ασθενή (PatientLaboratoryIndicator). Η οντότητα PatientLaboratoryIndicator και η οντότητα LaboratoryIndicator συσχετίζονται με One to Many σχέση, δηλαδή ένας εργαστηριακός δείκτης μπορεί να έχει πολλές μετρήσεις.

patientLaboratoryIndicators	
Id (Primary Key)	Αριθμός εγγραφής στη βάση
CurrentValue	Τιμή του εργαστηριακού δείκτη του ασθενούς
MeasDate	Ημερομηνία λήψης της μέτρησης

CIndicatorId (Foreign Key)	Id του εργαστηριακού δείκτη
CheckupId (Foreign Key)	Id της εξέτασης για την οποία έγινε η λήψη της μέτρησης αυτού του δείκτη

Πίνακας 6.1.1.9: Πίνακας μετρήσεων εργαστηριακών δεικτών

Φόρμες που συμπληρώνει ο ιατρός για τον ασθενή ανά εξέταση

Ο παρακάτω πίνακας περιλαμβάνει δεδομένα για την οντότητα που περιγράφει ένα πρότυπο μοντέλο φόρμας μίας εξέτασης ενός ασθενή (PrototypeForm).

prototypeForms	
PrototypeFormId (Primary Key)	Αριθμός εγγραφής στη βάση
Title	Τίτλος της φόρμας
Description	Περιγραφή της φόρμας

Πίνακας 6.1.1.10: Πίνακας πρότυπων μοντέλων φόρμας

Ο παρακάτω πίνακας περιλαμβάνει δεδομένα για την οντότητα μίας ερώτησης που μπορεί να βρίσκεται σε ένα πρότυπο μοντέλο φόρμας (QuestionForm). Η σχέση μεταξύ της οντότητας PrototypeForm και QuestionForm είναι One to Many, δηλαδή ένα πρότυπο μοντέλο φόρμας μπορεί να έχει πολλές ερωτήσεις.

questions	
QuestionId (Primary Key)	Αριθμός εγγραφής στη βάση
Question	Ερώτηση
QuestionDate	Ημερομηνία υποβολής της ερώτησης
PrototypeFormId (Foreign Key)	Id του πρότυπου μοντέλου φόρμας στο οποίο ανήκει η ερώτηση

Πίνακας 6.1.1.11: Πίνακας ερωτήσεων για πρότυπο μοντέλο φόρμας

Ο επόμενος πίνακας περιλαμβάνει δεδομένα για την οντότητα μίας φόρμας συγκεκριμένης εξέτασης (CheckupForm). Κάθε εξέταση περιλαμβάνει πολλές φόρμες, άρα η σχέση μεταξύ της οντότητας PatientCheckup και της οντότητας CheckupForm είναι One to Many. Κάθε φόρμα αντιστοιχεί σε ένα πρότυπο μοντέλο φόρμας, άρα η οντότητα PrototypeForm και η οντότητα CheckupForm έχουν σχέση One to Many (κάθε πρότυπο μοντέλο φόρμας μπορεί να έχει πολλές φόρμες).

checkupForms	
CheckupFormId (Primary Key)	Αριθμός εγγραφής στη βάση
PatientCheckupCheckupId (Foreign Key)	Id της εξέτασης στην οποία περιλαμβάνεται η φόρμα
PrototypeFormId (Foreign Key)	Id του πρότυπου μοντέλου φόρμας στο οποία αντιστοιχεί η συγκεκριμένη φόρμα

Πίνακας 6.1.1.12: Πίνακας με φόρμες ανά εξέταση

Τέλος, ο παρακάτω πίνακας περιλαμβάνει δεδομένα για την οντότητα που περιγράφει μία απάντηση σε μία ερώτηση μίας φόρμας που ανήκει σε συγκεκριμένη εξέταση (AnswerForm). Κάθε απάντηση αντιστοιχεί σε μία ερώτηση ενός πρότυπου μοντέλου φόρμας. Άρα κάθε ερώτηση θα έχει πολλές απαντήσεις οι οποίες θα διαφέρουν ανάλογα με την εξέταση. Συμπερασματικά η σχέση μεταξύ οντότητας QuestionForm και AnswerForm είναι One to Many. Επίσης ως γενόμενο του προηγούμενου λογισμού κάθε φόρμα μίας εξέτασης μπορεί να έχει πολλές απαντήσεις για κάθε ερώτηση που περιέχεται στο πρότυπο μοντέλο φόρμας στο οποίο αντιστοιχεί. Άρα η σχέση μεταξύ της οντότητας CheckupForm και της οντότητας AnswerForm είναι One to Many.

answers	
AnswerId (Primary Key)	Αριθμός εγγραφής στη βάση
Answer	Απάντηση
AnswerDate	Ημερομηνία υποβολής της απάντησης
QuestionId	Id της ερώτησης με την οποία γίνεται η σύνδεση
CheckupFormId	Id της φόρμας της εξέτασης στην οποία βρίσκεται η απάντηση

Πίνακας 6.1.1.13: Πίνακας απαντήσεων στις ερωτήσεις μιας φόρμας

Δεδομένα που σχετίζονται με τις θεραπείες που προτείνει ο ιατρός

Ο ιατρός έχει τη δυνατότητα να προτείνει και να αναθέτει θεραπείες σε ασθενείς στους οποίους έχει πρόσβαση στο ιστορικό τους. Η οντότητα Treatment περιγράφει ένα πρότυπο θεραπείας. Τα δεδομένα για τη συγκεκριμένη οντότητα, βρίσκονται στον παρακάτω πίνακα:

treatments	
TreatmentId (Primary Key)	Αριθμός εγγραφής στη βάση
Acronym	Συντομογραφία της ονομασίας της θεραπείας
FullName	Πλήρες όνομα της θεραπείας

Πίνακας 6.1.1.14: Πίνακας προτύπων θεραπείας

Για να αναθέσει ο ιατρός σε έναν ασθενή μία συγκεκριμένη θεραπεία θα χρειαστεί την οντότητα TreatmentDescriptionPreset η οποία συνδέει ένα πρότυπο θεραπείας, με ένα γιατρό και το ιστορικό ενός ασθενή. Ένας ιατρός, ένας ασθενής και ένα πρότυπο θεραπείας μπορούν να έχουν πολλά TreatmentDescriptionPreset οπότε η σχέση μεταξύ των τριών οντοτήτων και της TreatmentDescriptionPreset είναι One to Many. Στον παρακάτω πίνακα βρίσκονται τα δεδομένα που περιγράφουν τη συγκεκριμένη οντότητα:

treatmentDescriptionPresets	
PresetId (Primary Key)	Αριθμός εγγραφής στη βάση
TreatmentId (Foreign Key)	Id του προτύπου θεραπείας
CodeName	Συντομογραφία της θεραπείας
Description	Περιγραφή της θεραπείας
StartDate	Ημερομηνία έναρξης της θεραπείας
EndDate	Ημερομηνία λήξης της θεραπείας
DoctorId	Id του χρήστη ιατρού της εφαρμογής
PatientId	Id του ιστορικού του ασθενούς

Πίνακας 6.1.1.15: Πίνακας θεραπειών

6.1.2 Δεδομένα που αφορούν την εφαρμογή του ασθενούς

Τα δεδομένα που σχετίζονται με την εφαρμογή του ασθενούς μπορούμε να τα χωρίσουμε σε αυτά που χρειάζονται για την πιστοποίηση του χρήστη – ασθενή και σε

αυτά που αναφέρονται στην καθημερινότητα του όπως για παράδειγμα η διάθεση του ή οι διατροφικές του συνήθειες.

Δεδομένα Πιστοποίησης χρήστη – ιατρού

Ο παρακάτω πίνακας περιλαμβάνει δεδομένα για την οντότητα του χρήστη – ασθενή (User_Patient) και περιέχει πληροφορίες που σχετίζονται με την πρόσβαση στον λογαριασμό του (UserName, Password) και προσωπικά του δεδομένα. Κάθε χρήστης ασθενής είναι δυνατόν να συνδέεται με ένα και μόνο ιστορικό – ασθενή. Άρα έχουμε σχέση One to One μεταξύ των οντοτήτων User_Patient και PatientObject:

users_patients	
Id (Primary Key)	Αριθμός εγγραφής στη βάση
AMKA	Το ΑΜΚΑ του ασθενούς
PasswordHash	Συμβολοσειρά αναπαράστασης του κωδικού πρόσβασης έπειτα από κρυπτογράφηση με τον hashing αλγόριθμο.
PasswordSalt	Συμβολοσειρά που μπαίνει στην αρχή του κωδικού πρόσβασης και πριν τη διαδικασία του hashing.
ResetVerifiedAt	Ημερομηνία που πιστοποιήθηκε η αλλαγή κωδικού από τον χρήστη.
UserName	Όνομα χρήστη
LastName	Επώνυμο Χρήστη
FirstName	Μικρό Όνομα Χρήστη
Email	Διεύθυνση ηλεκτρονικού ταχυδρομίου
PhoneNumber	Αριθμός Τηλεφώνου
Country	Χώρα
State	Περιφέρεια
City	Πόλη

PostalCode	Ταχυδρομικός Κώδικας
Address	Διεύθυνση κατοικίας ασθενή
BirthDate	Ημερομηνία Γέννησης
Biological_Gender	Αριθμός που δείχνει το φύλλο του ασθενούς. Παίρνει τις τιμές 1 για το θηλυκό και 2 για το αρσενικό.
SmokingStatus	Αριθμός που δείχνει την κατάσταση του ασθενούς σε σχέση με το κάπνισμα. Η τιμή 1 αντιστοιχεί σε ασθενή που δεν ήταν ποτέ καπνιστής, η τιμή 2 σε πρώην καπνιστή και η τιμή 3 σε καπνιστή.
Parents_Have_Dm	Αριθμός που δείχνει το ιστορικό του ασθενούς όσον αφορά τον διαβήτη. Η τιμή 1 αντιστοιχεί στην περίπτωση που κανένας από τους δύο γονείς του είχαν διαβήτη, η τιμή 2 στην περίπτωση που η μητέρα του ασθενούς είχε διαβήτη, η τιμή 3 στην περίπτωση που ο πατέρας του ασθενούς είχε διαβήτη και η τιμή 4 στην περίπτωση που και οι δύο γονείς έχουν ιστορικό.
DmDiagnosisDate	Ημερομηνία Διάγνωσης
IsInInsulineTherapy	Λογική τιμή που περιγράφει αν ο ασθενής είναι σε θεραπείας ινσουλίνης.
PatientDataId (Foreign Key)	Id του ιστορικού ασθενή με το οποίο συνδέεται ο χρήστης - ασθενής

Πίνακας 6.1.2.1: Χρήστης - ασθενής

Ο παρακάτω πίνακας περιλαμβάνει δεδομένα για την οντότητα του κωδικού μίας χρήσης του ασθενούς (Patient_OneTimePassword):

patient_otp	
Id (Primary Key)	Αριθμός εγγραφής στη βάση
OtpPassHash	Συμβολοσειρά αναπαράστασης του κωδικού μίας χρήσης έπειτα από κρυπτογράφηση με τον hashing αλγόριθμο.

OtpPassSalt	Συμβολοσειρά που μπαίνει στην αρχή του κωδικού μίας χρήσης και πριν τη διαδικασία του hashing.
CreatedOn	Ημερομηνία δημιουργίας του κωδικού μίας χρήσης
UserId (Foreign Key)	Αριθμός συσχέτισης του κωδικού μίας πρόσβασης με τον χρήστη ασθενή.

Πίνακας 6.1.2.2: Κωδικός μίας χρήσης ασθενή

Οι οντότητες User_Patient και Doctor_OneTimePassword , σχετίζονται μεταξύ τους με τη σχέση One to Many. Στην περίπτωση μας ένας χρήστης μπορεί να έχει πολλούς κωδικούς μίας χρήσης.

Δεδομένα που περιγράφουν την καθημερινότητα του χρήστη

Όλοι οι οντότητες που περιγράφονται σε αυτή την ενότητα, σχετίζονται με την οντότητα PatientObject. Η μεταξύ τους σχέση είναι One to Many. Δηλαδή ένα ιστορικό ασθενή μπορεί να αντιστοιχεί με πολλές καταγραφές διάθεσης ή με πολλές καταγραφές γλυκόζης κοκ.

Ο παρακάτω πίνακας περιλαμβάνει δεδομένα για την οντότητα της διάθεσης του χρήστη – ασθενή (DailyMood).

patientmoodhistory	
Id (Primary Key)	Αριθμός εγγραφής στη βάση
MoodTimeStamp	Ημερομηνία υποβολής διάθεσης
Mood	Αριθμός που δείχνει τη διάθεση του χρήστη. Η τιμή 1 αντιστοιχεί στην περίπτωση ουδέτερης διάθεσης, η τιμή 2 στην περίπτωση κακής διάθεσης, η τιμή 3 στην περίπτωση μέτριας προς κακής διάθεσης, η τιμή 4 στην περίπτωση καλής διάθεσης και η τιμή 5 στην περίπτωση μέτριας προς κακής διάθεσης.
PatientId (Foreign Key)	Id του ιστορικού ασθενή

Πίνακας 6.1.2.3: Πίνακας διάθεσης του ασθενούς

Ο παρακάτω πίνακας περιλαμβάνει δεδομένα για την οντότητα της καθημερινής γλυκόζης του χρήστη – ασθενή (PatientGlucoseType).

patientGlucoseData	
Id (Primary Key)	Αριθμός εγγραφής στη βάση
Value	Αριθμητική τιμή της γλυκόζης σε mg / dl
Type	Αριθμός που δείχνει το είδος της μέτρησης της γλυκόζης, δηλαδή ποια χρονική στιγμή της μέρας έγινε η λήψη της μέτρησης. Η τιμή 1 αντιστοιχεί στην γλυκόζη πριν το γεύμα, η τιμή 2 στη γλυκόζη μετά το γεύμα, η τιμή 3 στην μέτρηση πριν τον ύπνο και η τιμή 4 στη γλυκόζη διόρθωσης.
TypeOfInsulin	Αριθμός που δείχνει το είδος της ινσουλίνης που λαμβάνει ο ασθενής. Η τιμή 1 αντιστοιχεί στην ινσουλίνη ταχείας δράσης, η τιμή 2 στην ινσουλίνη βραχείας δράσης, η τιμή 3 στην ινσουλίνη ενδιάμεσης δράσης, η τιμή 4 στην μικτή ινσουλίνη και η τιμή 5 στη μακράς δράσης ινσουλίνη.
InsulinDose	Αριθμητική τιμή της ποσότητας της δόσης της ινσουλίνης
HypoglycemiaCarbs	Αριθμός που δηλώνει την ποσότητα υδατανθράκων που καταναλώνονται για την άμεση αντιμετώπιση της υπογλυκαιμίας
Date	Ημερομηνία υποβολής γλυκόζης
PatientId (Foreign Key)	Id του ιστορικού ασθενή

Πίνακας 6.1.2.4: Πίνακας καθημερινής γλυκόζης του ασθενούς

Ο παρακάτω πίνακας περιλαμβάνει δεδομένα για την οντότητα της καθημερινής αρτηριακής πίεσης του χρήστη – ασθενή (PatientBloodPressure).

patientBloodPressureData	
Id (Primary Key)	Αριθμός εγγραφής στη βάση
Systolic	Αριθμητική τιμή της συστολικής πίεσης σε mmHg
Diastolic	Αριθμητική τιμή της διαστολικής πίεσης σε mmHg

BeatsPerMinute	Αριθμός παλμών ανά λεπτό του ασθενούς
Date	Ημερομηνία υποβολής δεδομένων πίεσης
PatientId (Foreign Key)	Id του ιστορικού ασθενή

Πίνακας 6.1.2.5: Πίνακας καθημερινής πίεσης του ασθενούς

Ο παρακάτω πίνακας περιλαμβάνει δεδομένα για την οντότητα ενός γεύματος (MealInfo).

mealsList	
Id (Primary Key)	Αριθμός εγγραφής στη βάση
MealName	Όνομα γεύματος
MealSimpleName	Συντομογραφία του ονόματος του γεύματος
Energy	Ποσότητα θερμίδων σε γραμμάρια ανά 100g γεύματος
Protein	Ποσότητα πρωτεϊνών σε γραμμάρια ανά 100g γεύματος
Fat	Ποσότητα λιπαρών σε γραμμάρια ανά 100g γεύματος
Carbs	Ποσότητα υδατανθράκων σε γραμμάρια ανά 100g γεύματος
FatsEquivalents	Ποσοστό λιπαρών επί του συνόλου του γεύματος
CarbsEquivalents	Ποσοστό υδατανθράκων επί του συνόλου του γεύματος

Πίνακας 6.1.2.6: Πίνακας πρότυπων γευμάτων

Ο παρακάτω πίνακας περιλαμβάνει δεδομένα για την οντότητα του γεύματος του χρήστη – ασθενή (PatientMeal). Το γεύμα που καταναλώνει ένας ασθενής αντιστοιχεί σε ένα από τα πρότυπα γεύματα του πίνακα 6.17. Η σχέση μεταξύ των οντοτήτων MealInfo και PatientMeal είναι One to Many.

patientMeals	
Id (Primary Key)	Αριθμός εγγραφής στη βάση
Type	Αριθμός που δηλώνει το είδος του γεύματος. Η τιμή 1 αντιστοιχεί σε πρωινό, η τιμή 2 σε δεκατιανό, η τιμή 3 σε μεσημεριανό, η τιμή 4 σε σνακ το απόγευμα και η τιμή 5 σε βραδινό.
Calories	Συνολικός αριθμός θερμίδων του γεύματος
Fats	Συνολικός αριθμός λιπαρών του γεύματος
Carbs	Συνολικός αριθμός υδατανθράκων του γεύματος
Proteins	Συνολικός αριθμός πρωτεϊνών του γεύματος
Info	Σημειώσεις του ασθενούς για το γεύμα
Date	Ημερομηνία καταγραφής του γεύματος
MealId (Foreign Key)	Κωδικός πρότυπου γεύματος
PatientDataId (Foreign Key)	Id του ιστορικού ασθενή

Πίνακας 6.1.2.7: Πίνακας γεύματος του ασθενούς

Ο παρακάτω πίνακας περιλαμβάνει δεδομένα για την οντότητα της καθημερινής άσκησης του χρήστη – ασθενή (PatientActivity).

patientActivity	
Id (Primary Key)	Αριθμός εγγραφής στη βάση
Duration	Διάρκεια της άσκησης σε λεπτά
Type	Αριθμός που δείχνει το είδος της άσκησης. Η τιμή 1 αντιστοιχεί σε περπάτημα, η τιμή 2 σε τρέξιμο, η τιμή 3 σε αερόβια άσκηση, η τιμή 4 σε ποδήλατο, η τιμή 5 σε κολύμβηση, η τιμή 6 σε yoga και η τιμή 7 σε κάτι άλλο.

Intensity	Αριθμός που δείχνει την ένταση της άσκησης. Η τιμή 1 αντιστοιχεί σε ήπια άσκηση, η τιμή 2 σε κανονική και η τιμή 3 σε έντονη.
Date	Ημερομηνία υποβολής άσκησης του ασθενούς
PatientId (Foreign Key)	Id του ιστορικού του ασθενούς

Πίνακας 6.1.2.8: Πίνακας καθημερινής άσκησης του ασθενούς

Ο παρακάτω πίνακας περιλαμβάνει δεδομένα για την οντότητα της καθημερινής μέτρησης του βάρους του χρήστη – ασθενή (PatientWeight).

patientWeight	
Id (Primary Key)	Αριθμός εγγραφής στη βάση
Weight	Αριθμητική τιμή του σωματικού βάρους του ασθενούς
Date	Ημερομηνία υποβολής σωματικού βάρους
PatientId (Foreign Key)	Id του ιστορικού του ασθενούς

Πίνακας 6.1.2.9: Πίνακας καθημερινής άσκησης του ασθενούς

Ο παρακάτω πίνακας περιλαμβάνει δεδομένα για την οντότητα της ασθένειας, διαφορετικής του διαβήτη, του χρήστη (PatientIllness).

illnessHistory	
Id (Primary Key)	Αριθμός εγγραφής στη βάση
Illness	Αριθμός που δείχνει το είδος της ασθένειας. Η τιμή 1 αντιστοιχεί σε Covid, η τιμή 2 σε γρίπη, η τιμή 3 σε πονοκέφαλο, η τιμή 4 σε πυρετό, η τιμή 5 σε εντερική διαταραχή και η τιμή 6 σε στομαχική διαταραχή.
Date	Ημερομηνία υποβολής ασθένειας του ασθενούς

PatientId (Foreign Key)	Id του ιστορικού του ασθενούς
--------------------------------	-------------------------------

Πίνακας 6.1.2.10: Πίνακας ασθένειας του ασθενούς

Ο παρακάτω πίνακας περιλαμβάνει δεδομένα για την οντότητα της καθημερινής φαρμακευτικής αγωγής του χρήστη – ασθενή (PatientMedication).

medicationHistory	
Id (Primary Key)	Αριθμός εγγραφής στη βάση
Medication	Αριθμός που δείχνει το είδος της φαρμακευτικής αγωγής. Η τιμή 1 αντιστοιχεί σε φάρμακο για τον διαβήτη, η τιμή 2 σε φάρμακο υπέρτασης, η τιμή 3 σε θεραπεία υπογλυκαιμίας, η τιμή 4 σε αντιθρομβωτικά, η τιμή 5 σε νευρολογικά χάπια, η τιμή 6 σε αναλγητικά και η τιμή 7 σε κάτι άλλο.
MedicationDescription	Περιγραφή της φαρμακευτικής αγωγής
Frequency	Αριθμός που δείχνει τη συχνότητα λήψης της συγκεκριμένης αγωγής. Η τιμή 1 αντιστοιχεί σε καθημερινή λήψη, η τιμή 2 σε εβδομαδιαία και η τιμή 3 σε έκτακτη.
Quantity	Ποσότητα φαρμάκου που λαμβάνεται
Date	Ημερομηνία υποβολής φαρμακευτικής αγωγής του ασθενούς
PatientId (Foreign Key)	Id του ιστορικού του ασθενούς

Πίνακας 6.1.2.11: Πίνακας καθημερινής φαρμακευτικής αγωγής του ασθενούς

Ο παρακάτω πίνακας περιλαμβάνει δεδομένα για την οντότητα του έμμηνου κύκλου του χρήστη – ασθενή (MenstrualCycle) στην περίπτωση που ο χρήστης είναι θηλυκό.

menstrualCycleHistory	
Id (Primary Key)	Αριθμός εγγραφής στη βάση

OvulationDate	Ημερομηνία ωορρηξίας
LastPeriodStart	Ημερομηνία έναρξης τελευταίας περιόδου
FertilityStart	Ημερομηνία έναρξης γόνιμων ημερών
FertilityEnd	Ημερομηνία λήξης γόνιμων ημερών
CycleDuration	Διάρκεια έμμηνου κύκλου σε μέρες
IsTheFirstPeriod	Λογική τιμή που ελέγχει αν η εγγραφή είναι η πρώτη του ασθενούς
LastPeriodEnd	Ημερομηνία λήξης περιόδου
PeriodDuration	Διάρκεια περιόδου σε μέρες
PatientId (Foreign Key)	Id του ιστορικού του ασθενούς

Πίνακας 6.1.2.12: Πίνακας έμμηνου κύκλου του ασθενούς

Ο παρακάτω πίνακας περιλαμβάνει δεδομένα για την οντότητα μίας ειδοποίησης – σημείωσης που μπορεί να προσθέσει ο χρήστης – ασθενής (PatientReminder).

patientreminders	
Id (Primary Key)	Αριθμός εγγραφής στη βάση
Title	Τίτλος σημείωσης - ειδοποίησης
Date	Ημερομηνία που θέλουμε να ενεργοποιηθεί η σημείωσης - ειδοποίηση
Description	Περιγραφή της σημείωσης - ειδοποίησης
PatientId (Foreign Key)	Id του ιστορικού του ασθενούς

Πίνακας 6.1.2.13: Πίνακας ειδοποιήσεων του χρήστη

6.1.3 Δεδομένα που αφορούν και τις δύο εφαρμογές

Όπως έχει περιγραφεί στις ενότητες 6.1.1 και 6.1.2 όλες οι οντότητες του συστήματος που αφορούν τον ασθενή, συσχετίζονται με την οντότητα PatientObject και αυτό διότι η οντότητα αυτή περιγράφει το ιστορικό του ασθενούς. Το ιστορικό του ασθενούς είναι το κοινό στοιχείο ενδιαφέροντος και των δύο εφαρμογών. Έτσι ο πίνακας 6.3, με όνομα patientData, είναι προσβάσιμος και από τις δύο εφαρμογές.

Ο χρήστης - ιατρός μπορεί να αιτηθεί την άδεια του χρήστη – ασθενή έτσι ώστε να αποκτήσει πρόσβαση στους πίνακες 6.17 patientGlucoseData, 6.18 patientBloodPressureData, 6.21 patientActivity και 6.22 patientWeight.

Ο παρακάτω πίνακας περιέχει τα δεδομένα για την οντότητα που περιγράφει την άδεια που ζητάει ο χρήστης ιατρός από τον χρήστη – ασθενή για να αποκτήσει πρόσβαση στους τέσσερις πίνακες που αναφέραμε προηγουμένως (PatientDoctorPermission). Η οντότητα PatientObject και η οντότητα User_Doctor έχουν σχέση One to Many με την οντότητα PatientDoctorPermission, δηλαδή ένας χρήστης – ιατρός μπορεί να έχει πολλές άδειες και ένα ιστορικό ασθενή μπορεί να έχει πολλές άδειες.

patientPermissions	
PatientDoctorPermissionId (Primary Key)	Αριθμός εγγραφής στη βάση
PatientId (Foreign Key)	Id του ιστορικού του ασθενούς
DoctorId (Foreign Key)	Id του χρήστη - ιατρού
HasPermission	Λογική τιμή που δηλώνει αν ο ιατρός έχει άδεια ή όχι

Πίνακας 6.1.3.1: Πίνακας για τις άδειες ιατρού – ασθενή

Το Back-End σύστημα μπορεί να στέλνει ειδοποιήσεις στους χρήστες των δύο εφαρμογών οι οποίες περιγράφονται από την οντότητα SystemNotification. Για να ειδοποιηθεί ο χρήστης – ασθενής, θα πρέπει μία SystemNotification να συνδεθεί με αυτόν. Αυτό εξυπηρετεί η οντότητα PatientNotification. Η οντότητα PatientNotification και η οντότητα SystemNotification συνδέονται με σχέση One to , δηλαδή ένα System Notification μπορεί να έχει πολλά PatientNotification

Τα δεδομένα της οντότητας SystemNotification, βρίσκονται στον παρακάτω πίνακα:

systemNotifications	
SystemNotificationId (Primary Key)	Αριθμός εγγραφής στη βάση
Name	Όνομα ειδοποίησης
Description	Περιγραφή Ειδοποίησης

Πίνακας 6.1.3.2: Πίνακας ειδοποιήσεων συστήματος

Τα δεδομένα της οντότητας PatientNotification, βρίσκονται στον παρακάτω πίνακα:

patientNotifications	
Id (Primary Key)	Αριθμός εγγραφής στη βάση
EnableNotification	Λογική τιμή που υποδεικνύει αν η ειδοποίηση θα φτάνει στον χρήστη – ασθενή.
PatientObjectId (Foreign Key)	Id του ιστορικού του ασθενούς
SystemNotificationId (Foreign Key)	Id του System Notification

Πίνακας 6.1.3.3: Πίνακας ειδοποιήσεων χρήστη – ασθενή

6.1.4 Σταθερά δεδομένα που αφορούν το σύστημα

Για τη σωστή λειτουργία του Back-End συστήματος και κατ' επέκταση των δύο εφαρμογών, σχεδιάστηκαν κάποιες οντότητες δεδομένων οι οποίες θα μπορούσαν μελλοντικά να είναι προσβάσιμες μόνο στον διαχειριστή του συστήματος. Κάποιες από αυτές τις οντότητες είναι ήδη προσβάσιμες στα πλαίσια της συγκεκριμένης υλοποίησης, για κάποιον επιβεβαιωμένο χρήστη – γιατρό. Σε επόμενο κεφάλαιο αναλυθούν η επεκτασιμότητα και μελλοντικές βελτιστοποιήσεις του συστήματος.

1. Κλινικοί και εργαστηριακοί Δείκτες

Στην ενότητα 6.1.1 που περιγράφονται τα δεδομένα τα οποία αφορούν την εφαρμογή του ιατρού, και έγινε αναφορά του πίνακες 6.1.1.5 clinicalIndicators και 6.1.1.6 laboratoryIndicators. Οι πίνακες αυτοί περιλαμβάνουν τους κλινικούς και εργαστηριακούς δείκτες που θέλει ο ιατρός να εξετάσει σε τακτά χρονικά διαστήματα. Οι δείκτες αυτοί είναι συγκεκριμένοι και δε δέχονται μεταβολή.

2. Θεραπείες

Στην ίδια ενότητα εξετάσαμε τη δυνατότητα του ιατρού να αναθέσει θεραπείες σε κάποιο ασθενή που είναι διασυνδεδεμένος με το ιστορικό του. Οι θεραπείες αυτές βασίζονται σε κάποια συγκεκριμένα πρότυπα θεραπειών (πίνακας 6.1.1.14) τα οποία ο ιατρός δεν μπορεί να παρακάμψει.

3. Δεδομένα που αποτελούν πιθανές περιπτώσεις για τον χαρακτηρισμό άλλων δεδομένων

Στο σύστημα, για την περιγραφή κάποιων δεδομένων χρησιμοποιούνται κάποια πεδία που ονομάζονται τύποι, όπως για παράδειγμα τύπος έντασης της άσκησης ή ο τύπος άσκησης. Αυτοί οι τύποι προς το παρόν είναι enum classes. Όμως έχουν σχεδιαστεί οντότητες που αποθηκεύονται στη βάση έτσι ώστε να παρέχεται μεγαλύτερη ευελιξία στο back – end σύστημα. Οι οντότητες αυτές είναι οι παρακάτω:

SmokingStatusClassData:

Πιθανές περιπτώσεις για τη σχέση του ασθενούς με το κάπνισμα.

smokingstatusclassdatas	
Id (Primary Key)	Αριθμός εγγραφής στη βάση
Status	Πιθανή σχέση του ασθενούς με το κάπνισμα

Πίνακας 6.1.4.1: Καταστάσεις σχέσης ασθενή με το κάπνισμα

GenderClassData:

Πιθανές περιπτώσεις για το φύλο του ασθενούς.

genderclassdata	
Id (Primary Key)	Αριθμός εγγραφής στη βάση
Sex	Φύλο του ασθενούς

Πίνακας 6.1.4.2: Περιπτώσεις φύλου του ασθενούς

ParentsClassData:

Πιθανές περιπτώσεις για το ιστορικό σχετικά με τον ΣΔ των γονέων του ασθενούς.

parentsclassdata	
Id (Primary Key)	Αριθμός εγγραφής στη βάση
ParentsClassDescription	Περίπτωση για το ιστορικό σχετικά με τον ΣΔ των γονέων του ασθενούς

Πίνακας 6.1.4.3: Περιπτώσεις για το ιστορικό σχετικά με τον ΣΔ των γονέων του ασθενούς

ActivityIntensityClassData:

Πιθανές περιπτώσεις για την ένταση της άσκησης του ασθενούς.

activityintensitylist	
Id (Primary Key)	Αριθμός εγγραφής στη βάση
ActivityIntensityDescription	Ένταση της άσκησης του ασθενούς

Πίνακας 6.1.4.4: Περιπτώσεις για την ένταση της άσκησης του ασθενούς

ActivityTypeClassData:

Πιθανές περιπτώσεις για το είδος της άσκησης του ασθενούς.

activitytypelist	
Id (Primary Key)	Αριθμός εγγραφής στη βάση
ActivityTypeDescription	Είδος της άσκησης του ασθενούς

Πίνακας 6.1.4.5: Περιπτώσεις για το είδος της άσκησης του ασθενούς

GlucoseTypeClassData:

Πιθανές περιπτώσεις για το είδος τη μέτρηση γλυκόζης που λαμβάνει ο ασθενής.

glucosetypelist	
Id (Primary Key)	Αριθμός εγγραφής στη βάση
GlucoseDescription	Είδος της μέτρησης γλυκόζης που λαμβάνει ο ασθενής

Πίνακας 6.1.4.6: Περιπτώσεις για το είδος τη μέτρηση γλυκόζης που λαμβάνει ο ασθενής

MoodStatusClassData:

Πιθανές περιπτώσεις για τη διάθεση του ασθενούς.

moodstatuslist	
Id (Primary Key)	Αριθμός εγγραφής στη βάση
Mood	Διάθεση του ασθενούς

Πίνακας 6.1.4.7: Περιπτώσεις για τη διάθεση του ασθενούς

MedicationTypeListData:

Πιθανές περιπτώσεις φαρμακευτικής αγωγής.

medicationtypelist	
Id (Primary Key)	Αριθμός εγγραφής στη βάση
MedicationType	Είδος φαρμακευτικής αγωγής

Πίνακας 6.1.4.8: Περιπτώσεις για τη φαρμακευτική αγωγή του ασθενούς

MedicationFrequencyData:

Πιθανές περιπτώσεις συχνότητας λήψης της φαρμακευτικής αγωγής.

medicationfrequencylist	
Id (Primary Key)	Αριθμός εγγραφής στη βάση
HowOften	Συχνότητα λήψης φαρμακευτικής αγωγής

Πίνακας 6.1.4.9: Περιπτώσεις συχνότητας λήψης της φαρμακευτικής αγωγής

InsulinTypeClassData:

Πιθανές είδη ινσουλίνης που λαμβάνει ο ασθενής.

insulintypelist	
Id (Primary Key)	Αριθμός εγγραφής στη βάση
InsulinType	Είδη ινσουλίνης που λαμβάνει ο ασθενής

Πίνακας 6.1.4.10: Είδη ινσουλίνης που λαμβάνει ο ασθενής

IllnessTypeClassData:

Περιπτώσεις ασθένειας που μπορεί να αντιμετωπίζει ο ασθενής για μια συγκεκριμένη χρονική περίοδο.

illnesstypelist	
Id (Primary Key)	Αριθμός εγγραφής στη βάση
IllnessType	Είδη ασθενειών

Πίνακας 6.1.4.11: Είδη ασθενειών

LimitsInfoData:

Τα είδη των ορίων για τους κλινικούς και εργαστηριακούς δείκτες.

limitinfolist	
Id (Primary Key)	Αριθμός εγγραφής στη βάση
LimitsInfoDescription	Είδος ορίου δείκτη

Πίνακας 6.1.4.12: Είδη ορίων εργαστηριακών και κλινικών δεικτών

4. Πρότυπα γεύματα

Όπως αναλύθηκε στην ενότητα 6.1.2, τα γεύματα που συμπληρώνει ο ασθενής στην εφαρμογή του, βασίζονται σε κάποια πρότυπα. Ο ασθενής μπορεί να συμπληρώσει τη λίστα του μόνο με προεπιλεγμένα γεύματα χωρίς να τα παρακάμψει.

5. Ειδοποιήσεις του συστήματος

Στην ενότητα 6.1.3 αναλύθηκε ο τρόπος με τον οποίο ο ασθενής ενημερώνεται ότι έχει αίτημα διασύνδεσης από τον γιατρό. Για τη σωστή και έγκαιρη ενημέρωση του ιατρού και του ασθενούς, πρέπει το σύστημα να χρησιμοποιεί κάποιες πρότυπες ειδοποιήσεις,

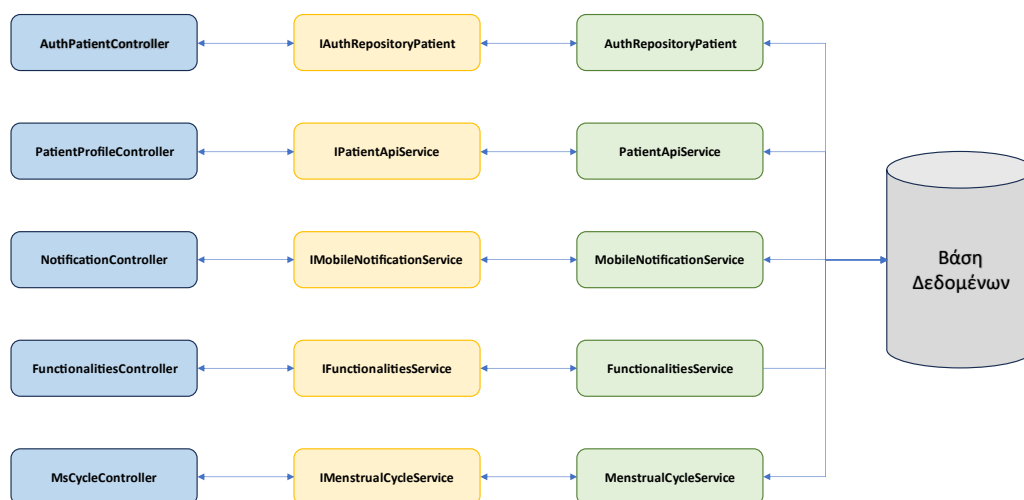
διαφορετικές για τον καθένα, που θα αποστέλλονται όμως και στους δύο. Αυτός είναι ο λόγος που δημιουργήθηκε ξεχωριστή οντότητα η οποία θα έχει δεδομένα στη βάση σταθερά και ανεξάρτητα από τους δύο χρήστες.

Τέλος, η πληροφορία που είτε εισέρχεται στο σύστημα με τη μορφή αιτημάτων των χρηστών (requests) είτε εξέρχεται από αυτό με τη μορφή απαντήσεων (responses) που επιστρέφουν τα API, μοιάζουν με τις οντότητες που αναπτύχθηκαν προηγουμένων και αναπαρίστανται με τους πίνακες της βάσης, αλλά δεν είναι ίδιες. Οι πληροφορίες που βρίσκονται αποθηκευμένες στη βάση ενός συστήματος, είναι ευαίσθητα δεδομένα και δεν πρέπει να είναι άμεσα προσβάσιμα. Για παράδειγμα οι κωδικοί πρόσβασης των χρηστών, δεν αποθηκεύονται στη βάση με τη μορφή που εισάγονται από τον χρήστη, αποθηκεύεται το hash και το salt του κωδικού έτσι ώστε να μην υπάρχει πρόσβαση σε αυτό. Επιπλέον από μία οντότητα που είναι αποθηκευμένη στη βάση, δεν είναι πάντα απαραίτητη η μεταφορά ολόκληρης της πληροφορίας που προσφέρει, αλλά ένα μέρος αυτής. Για αυτούς τους λόγους σχεδιάστηκαν οντότητες που ονομάζονται «Δεδομένα Μεταφοράς Αντικειμένων» ή Data Transfer Objects (DTO). Σε κάθε μία από τις παρακάτω ενότητες που αναλύονται τα API του συστήματος θα γίνει περιγραφή και των DTO που χρησιμοποιούνται σε κάθε περίπτωση.

6.2 API της εφαρμογής του ασθενούς

Το API του ασθενούς, αποτελείται από πέντε controllers όπου ο καθένας μέσω των endpoints του δίνει πρόσβαση σε, παρόμοιες μεταξύ τους, λειτουργίες της εφαρμογής. Οι controllers διαμορφώνουν τα url των endpoints που περιέχουν. Το όνομα του controller βρίσκεται μέσα στο url και είναι η συμβολοσειρά (string) που κλείνεται σε άγκιστρα. Η γενική μορφή του είναι η εξής: <http://localhost:5288/{controller}>.

Στο παρακάτω σχήμα φαίνεται η δομή του API και του κώδικα που καλεί με στόχο την επεξεργασία των δεδομένων που αιτείται ο ασθενής μέσω της εφαρμογής του.



Σχήμα 6.2.1: Αρχιτεκτονική API της εφαρμογής του ασθενούς

Όπως αναφέρθηκε στην ενότητα 6.1, η πληροφορία ρέει στο σύστημα με την χρήση DTO. Στους παρακάτω πίνακες φαίνονται τα DTO που χρησιμοποιούνται σε αυτό το API.

PatientRegisterDto	
AMKA	Το AMKA του ασθενούς
UserName	Όνομα χρήστη
LastName	Επώνυμο Χρήστη
FirstName	Μικρό Όνομα Χρήστη
BirthDate	Ημερομηνία Γέννησης
Biological_Gender	Αριθμός που δείχνει το φύλλο του ασθενούς. Παίρνει τις τιμές 1 για το θηλυκό και 2 για το αρσενικό.
SmokingStatus	Αριθμός που δείχνει την κατάσταση του ασθενούς σε σχέση με το κάπνισμα. Η τιμή 1 αντιστοιχεί σε ασθενή που δεν ήταν ποτέ καπνιστής, η τιμή 2 σε πρώην καπνιστή και η τιμή 3 σε καπνιστή.
Parents_Have_Dm	Αριθμός που δείχνει το ιστορικό του ασθενούς όσον αφορά τον διαβήτη. Η τιμή 1 αντιστοιχεί στην περίπτωση που κανένας από τους δύο γονείς του είχαν διαβήτη, η τιμή 2 στην περίπτωση που η μητέρα του ασθενούς είχε διαβήτη, η τιμή 3 στην περίπτωση που ο πατέρας του ασθενούς είχε διαβήτη και η τιμή 4 στην περίπτωση που και οι δύο γονείς έχουν ιστορικό.
DmDiagnosisDate	Ημερομηνία Διάγνωσης
IsInInsulineTherapy	Λογική τιμή που περιγράφει αν ο ασθενής είναι σε θεραπείας ινσουλίνης.
Email	Διεύθυνση ηλεκτρονικού ταχυδρομείου

Password	Κωδικός πρόσβασης του ασθενούς (πρέπει να αποτελείται από 6 χαρακτήρες)
ConfirmPassword	Επιβεβαίωση του κωδικού πρόσβασης. Αν δεν ταιριάζουν οι δύο κωδικοί, ο χρήστης ενημερώνεται με σχετικό μήνυμα.
PhoneNumber	Αριθμός τηλεφώνου ασθενή
Address	Διεύθυνση κατοικίας ασθενή
Country	Χώρα
State	Περιφέρεια
City	Πόλη
PostalCode	Ταχυδρομικός Κώδικας

Πίνακας 6.2.1: DTO για την εγγραφή του χρήστη - ασθενή

PatientLoginDto	
UserName	Όνομα χρήστη
Password	Κωδικός Χρήστη

Πίνακας 6.2.2: DTO για την είσοδο του χρήστη στην εφαρμογή

ResetPasswordDto	
Email	Email του χρήστη
Password	Κωδικός πρόσβασης του ασθενούς (πρέπει να αποτελείται από 6 χαρακτήρες)
ConfirmPassword	Επιβεβαίωση του κωδικού πρόσβασης. Αν δεν ταιριάζουν οι δύο κωδικοί, ο χρήστης ενημερώνεται με σχετικό μήνυμα.

Πίνακας 6.2.3: DTO για να αλλάξει ο ασθενής τον κωδικό πρόσβασης του στην εφαρμογή

GetPatientUserDto	
AMKA	Το ΑΜΚΑ του ασθενούς
UserName	Όνομα χρήστη
LastName	Επώνυμο Χρήστη
FirstName	Μικρό Όνομα Χρήστη
Email	Διεύθυνση ηλεκτρονικού ταχυδρομίου
PhoneNumber	Αριθμός τηλεφώνου ασθενή
Address	Διεύθυνση κατοικίας ασθενή
Country	Χώρα
State	Περιφέρεια
City	Πόλη
PostalCode	Ταχυδρομικός Κώδικας
Doctors	Οι ιατροί που παρακολουθούν το ιστορικό του ασθενούς
PatientData	Το ιστορικό του ασθενούς

Πίνακας 6.2.4: DTO για τα δεδομένα του λογαριασμού του χρήστη – ασθενή

GetPatientDoctorDto	
Id	Κωδικός εγγραφής του ιατρού στη βάση
LastName	Επώνυμο ιατρού

FirstName	Μικρό Όνομα ιατρού
Email	Διεύθυνση ηλεκτρονικού ταχυδρομείου ιατρού
PhoneNumber	Αριθμός τηλεφώνου ιατρού
MedicalSpecialty	Ειδικότητα του ιατρού
Workplace	Νοσοκομείο στο οποίο εργάζεται ο ιατρός

Πίνακας 6.2.5: DTO για τα δεδομένα του ιατρού που μπορεί να δει ο χρήστης - ασθενής

UpdateUserPatientDto	
AMKA	Το AMKA του ασθενούς
UserName	Όνομα χρήστη
LastName	Επώνυμο Χρήστη
FirstName	Μικρό Όνομα Χρήστη
Email	Διεύθυνση ηλεκτρονικού ταχυδρομείου
PhoneNumber	Αριθμός τηλεφώνου ασθενή
Address	Διεύθυνση κατοικίας ασθενή
Country	Χώρα
State	Περιφέρεια
City	Πόλη
PostalCode	Ταχυδρομικός Κώδικας

Πίνακας 6.2.6: DTO για τα δεδομένα του χρήστη ασθενή τα οποία μπορεί να τροποποιήσει.

UpdateDoctorPermissionDto	
AMKA	AMKA του χρήστη - ασθενή
DoctorId	Κωδικός εγγραφής του ιατρού στη βάση για τον οποίο θέλει ο χρήστης – ασθενής να επεξεργαστεί το αίτημα άδειας πρόσβασης στα δεδομένα που εισάγει ο ασθενής στο σύστημα μέσω της εφαρμογής του κινητού.
HasPermission	Λογική τιμή που δηλώνει αν ο ιατρός έχει άδεια πρόσβασης στα δεδομένα που εισάγει ο ασθενής στο σύστημα μέσω της εφαρμογής του κινητού.

Πίνακας 6.2.7: DTO για την τροποποίηση της άδειας που δίνει ο ασθενής στον γιατρό για την απόκτηση πρόσβασης στα δεδομένα του.

GetPatientDoctorPermissionDto	
PatientDoctorPermissionId	Κωδικός εγγραφής της άδειας πρόσβασης που σχετίζεται με συγκεκριμένο ζεύγος ασθενή – ιατρού.
DoctorId	Κωδικός εγγραφής του ιατρού στη βάση για τον οποίο θέλει ο χρήστης – ασθενής να επεξεργαστεί το αίτημα άδειας πρόσβασης στα δεδομένα που εισάγει ο ασθενής στο σύστημα μέσω της εφαρμογής του κινητού.
PatientId	Κωδικός εγγραφής του ιστορικού του ασθενούς στη βάση.
HasPermission	Λογική τιμή που δηλώνει αν ο ιατρός έχει άδεια πρόσβασης στα δεδομένα που εισάγει ο ασθενής στο σύστημα μέσω της εφαρμογής του κινητού.

Πίνακας 6.2.8: DTO που περιγράφει την άδεια που δίνει ο ασθενής στον γιατρό για την απόκτηση πρόσβασης στα δεδομένα του.

GetPatientMoodDto	
MoodTimeStamp	Χρονική στιγμή που έγινε η εγγραφή «Διάθεσης» στη βάση.
Mood	Το είδος της διάθεσης.
Patient	Ιστορικό του ασθενούς με το οποίο συνδέεται ο χρήστης.

Πίνακας 6.2.9: DTO που περιγράφει συγκεκριμένη διάθεση του χρήστη – ασθενή.

AddPatientMoodDto	
Mood	Το είδος της διάθεσης.

Πίνακας 6.2.10: DTO για εγγραφή νέα διάθεσης του ασθενούς στη βάση.

GetDailyGlucoseDto	
Id	Κωδικός εγγραφής δεδομένων γλυκόζης στη βάση.
Value	Αριθμητική τιμή της μέτρησης γλυκόζης.
Type	Τύπος γλυκόζης
Date	Ημερομηνία εγγραφής στη βάση

Πίνακας 6.2.11: DTO για τα δεδομένα γλυκόζης συγκεκριμένης εγγραφής στη βάση.

AddDailyGlucoseDto	
Value	Αριθμητική τιμή της μέτρησης γλυκόζης.
Type	Τύπος γλυκόζης
Date	Ημερομηνία εγγραφής στη βάση

Πίνακας 6.2.12: DTO για εισαγωγή νέας εγγραφής δεδομένων γλυκόζης στη βάση.

GetGlucosePercentsDto	
PreMealPercent	Ποσοστό προγευματικής γλυκόζης
AfterMealPercent	Ποσοστό μεταγευματικής γλυκόζης
PreSleepPercent	Ποσοστό γλυκόζης πριν τον ύπνο

Πίνακας 6.2.13: DTO για τα ποσοστά γλυκόζης ανά τύπο του ασθενούς

GetBloodPressureDto	
Id	Κωδικός εγγραφής δεδομένων πίεσης στη βάση.
Systolic	Αριθμητική τιμή της συστολικής πίεσης.
Diastolic	Αριθμητική τιμή της διαστολικής πίεσης.
BeatsPerMinute	Παλμοί ανά λεπτό
Date	Ημερομηνία εγγραφής στη βάση

Πίνακας 6.2.14: DTO για τα δεδομένα πίεσης συγκεκριμένης εγγραφής στη βάση.

AddBloodPressureDto	
Systolic	Αριθμητική τιμή της συστολικής πίεσης.
Diastolic	Αριθμητική τιμή της διαστολικής πίεσης.
BeatsPerMinute	Παλμοί ανά λεπτό
Date	Ημερομηνία εγγραφής στη βάση

Πίνακας 6.2.15: DTO για εισαγωγή νέας εγγραφής δεδομένων πίεσης στη βάση.

GetPatientMealDto	
Id	Κωδικός εγγραφής δεδομένων γεύματος στη βάση.

Meal	Πληροφορίες γεύματος
Type	Είδος γεύματος
Calories	Θερμίδες
Fats	Λιπαρά
Carbs	Λιπαρά
Proteins	Πρωτεΐνες
Info	Σημειώσεις
Date	Ημερομηνία εγγραφής γεύματος στη βάση

Πίνακας 6.2.16: DTO για τα δεδομένα γεύματος συγκεκριμένης εγγραφής στη βάση.

AddPatientMealDto	
MealInfoId	Κωδικός εγγραφής δεδομένων γεύματος στη βάση.
Type	Είδος γεύματος
Calories	Θερμίδες
Fats	Λιπαρά
Carbs	Λιπαρά
Proteins	Πρωτεΐνες
Info	Σημειώσεις
Date	Ημερομηνία εγγραφής γεύματος στη βάση

Πίνακας 6.2.17: DTO για εισαγωγή νέας εγγραφής δεδομένων γεύματος στη βάση.

GetPatientActivityDto	
Id	Κωδικός εγγραφής δεδομένων γεύματος στη βάση.
Duration	Διάρκεια άσκησης
Type	Είδος άσκησης
Intensity	Ένταση της άσκησης
Date	Ημερομηνία εγγραφής άσκησης στη βάση

Πίνακας 6.2.18: DTO για τα δεδομένα άσκησης συγκεκριμένης εγγραφής στη βάση.

AddPatientActivityDto	
Duration	Διάρκεια άσκησης
Type	Είδος άσκησης
Intensity	Ένταση της άσκησης
Date	Ημερομηνία εγγραφής άσκησης στη βάση

Πίνακας 6.2.19: DTO για εισαγωγή νέας εγγραφής δεδομένων άσκησης στη βάση.

GetPatientWeightDto	
Id	Κωδικός εγγραφής δεδομένων σωματικού βάρους στη βάση.
Weight	Αριθμητική τιμή σωματικού βάρους
Date	Ημερομηνία εγγραφής σωματικού βάρους στη βάση

Πίνακας 6.2.20: DTO για τα δεδομένα σωματικού βάρους συγκεκριμένης εγγραφής στη βάση.

AddPatientWeightDto	
Weight	Αριθμητική τιμή σωματικού βάρους
Date	Ημερομηνία εγγραφής σωματικού βάρους στη βάση

Πίνακας 6.2.21: DTO για εισαγωγή νέας εγγραφής δεδομένων σωματικού βάρους στη βάση.

GetPatientIllnessDto	
Id	Κωδικός εγγραφής δεδομένων ασθένειας στη βάση.
Illness	Είδος ασθένειας
Date	Ημερομηνία εγγραφής ασθένειας στη βάση

Πίνακας 6.2.22: DTO για τα δεδομένα ασθένειας συγκεκριμένης εγγραφής στη βάση.

AddPatientMedicationDto	
Illness	Είδος ασθένειας
Date	Ημερομηνία εγγραφής ασθένειας στη βάση

Πίνακας 6.2.23: DTO για εισαγωγή νέας εγγραφής δεδομένων ασθένειας στη βάση.

GetPatientMedicationDto	
Id	Κωδικός εγγραφής δεδομένων φαρμακευτικής αγωγής στη βάση.
Medication	Είδος φαρμακευτικής αγωγής
MedicationDescription	Περιγραφή φαρμακευτικής αγωγής
Date	Ημερομηνία εγγραφής φαρμακευτικής αγωγής στη βάση

Quantity	Ποσότητα δόσης φαρμακευτικής αγωγής
Frequency	Συχνότητα λήψης φαρμακευτικής αγωγής

Πίνακας 6.2.24: DTO για τα δεδομένα φαρμακευτικής αγωγής συγκεκριμένης εγγραφής στη βάση.

AddPatientMedicationDto	
Medication	Είδος φαρμακευτικής αγωγής
MedicationDescription	Περιγραφή φαρμακευτικής αγωγής
Date	Ημερομηνία εγγραφής φαρμακευτικής αγωγής στη βάση
Quantity	Ποσότητα δόσης φαρμακευτικής αγωγής
Frequency	Συχνότητα λήψης φαρμακευτικής αγωγής

Πίνακας 6.2.25: DTO για εισαγωγή νέας εγγραφής δεδομένων φαρμακευτικής αγωγής στη βάση.

GetMenstrualCycleDto	
Id	Αριθμός εγγραφής στη βάση
OvulationDate	Ημερομηνία ωορρηξίας
LastPeriodStart	Ημερομηνία έναρξης τελευταίας περιόδου
FertilityStart	Ημερομηνία έναρξης γόνιμων ημερών
FertilityEnd	Ημερομηνία λήξης γόνιμων ημερών
CycleDuration	Διάρκεια έμμηνου κύκλου σε μέρες
LastPeriodEnd	Ημερομηνία λήξης περιόδου
PatientAmka	ΑΜΚΑ του ασθενούς

Πίνακας 6.2.26: DTO για τα δεδομένα έμμηνου κύκλου συγκεκριμένης εγγραφής στη βάση.

AddMenstrualCycleDto	
LastPeriodStart	Ημερομηνία έναρξης τελευταίας περιόδου
LastPeriodEnd	Ημερομηνία λήξης τελευταίας περιόδου

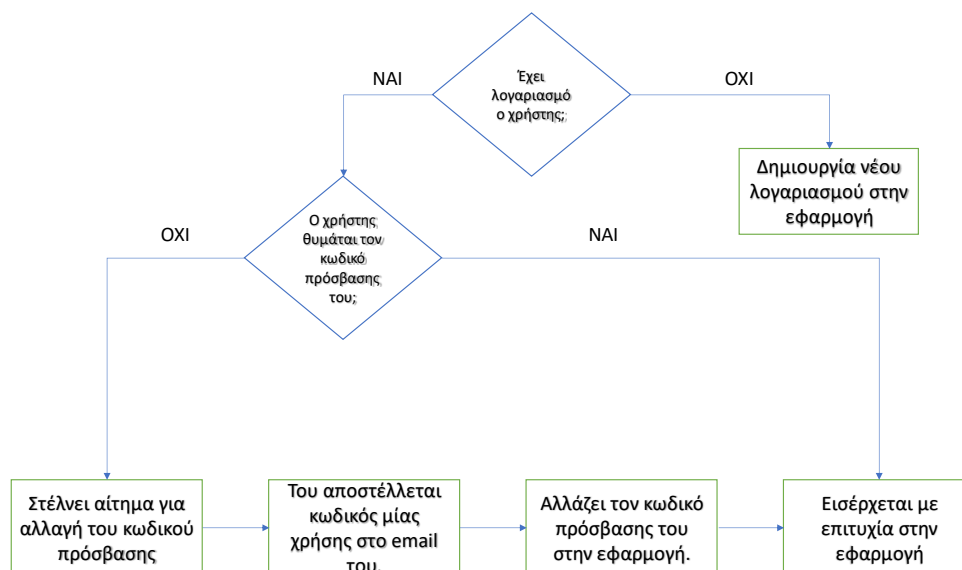
Πίνακας 6.2.27: DTO για εισαγωγή νέας εγγραφής δεδομένων έμμηνου κύκλου στη βάση.

GetPatientNotificationDto	
Id	Κωδικός εγγραφής δεδομένων ασθένειας στη βάση.
Notification	Είδος ειδοποίησης συστήματος
EnableNotification	Λογική τιμή που δείχνει αν ο χρήστης επιθυμεί να λαμβάνει την ειδοποίηση

Πίνακας 6.2.26: DTO για τα δεδομένα ειδοποίησης του χρήστη – ασθενή.

6.2.1 AuthPatientController

Ο controller αυτός υλοποιήθηκε για να διαχειριστεί το κομμάτι της εγγραφής και πιστοποίησης του ασθενούς στο σύστημα. Στο παρακάτω σχήμα φαίνεται η λογική που υλοποιεί το Back-End για την εγγραφή και την είσοδο του ασθενούς στην εφαρμογή.



Σχήμα 3.1 2: Διαδικασία εγγραφής και εισόδου του ασθενούς στην εφαρμογή

Στον παρακάτω πίνακα φαίνονται τα endpoints που περιέχει, τα δεδομένα που πρέπει να δέχεται το καθένα και τα δεδομένα που επιστρέφει.

AuthPatientController (http://localhost:5288/AuthPatient)					
A/A	Endpoint	HTTP Μέθοδος	Περιγραφή	Δεδομένα Εισόδου στο endpoint	Δεδομένα εισόδου στο σώμα της μεθόδου
1	/Register	POST	Κάνει εγγραφή νέου χρήστη - ασθενή	-	PatientRegisterDto Πίνακας 6.2.1
2	/Login	POST	Κάνει είσοδο του χρήστη στην εφαρμογή	-	PatientLoginDto Πίνακας 6.2.2
3	/forgot-password	POST	Κάνει αίτημα για να αλλάξει τον κωδικό πρόσβασης του στην εφαρμογή.	Email και AMKA του ασθενούς.	-
4	/verify-otp-password	POST	Ελέγχει την εγκυρότητα του κωδικού μίας χρήσης που τοποθετεί ο ασθενής για την αλλαγή του Password του.	Email και AMKA ασθενή και κωδικός μίας χρήσης που έλαβε με email	-
5	/reset-password	POST	Αλλάζει τον κωδικό πρόσβασης του χρήστη στην εφαρμογή	AMKA του ασθενούς	ResetPasswordDto Πίνακας 6.2.3

Πίνακας 6.2.27: Endpoints του AuthPatient Controller

Στον παρακάτω πίνακα φαίνονται τα μηνύματα ή τα δεδομένα που επιστρέφει το κάθε ένα από τα παραπάνω.

AuthPatientController: Endpoint 1 Responses

A/A	Περίπτωση	Status Code	Λεδομένα που επιστρέφει	Μήνυμα
1	Ο χρήστης συμπληρώνει σωστά τη φόρμα εγγραφής του στην εφαρμογή	200: Success	Τον κωδικό εγγραφής του χρήστη στη βάση.	Ειδοποιεί τον χρήστη ότι πρέπει να επικοινωνήσει με τον γιατρό του για να διασυνδεθούν.
2	Ο χρήστης συμπληρώνει User Name που υπάρχει εγγεγραμμένο στη βάση	400: Bad Request	Επιστρέφει τον αριθμό 0.	Ειδοποιεί τον χρήστη ότι υπάρχει ήδη κάποιος που χρησιμοποιεί το ίδιο User Name.
3	Ο χρήστης συμπληρώνει AMKA που υπάρχει εγγεγραμμένο στη βάση	400: Bad Request	Επιστρέφει τον αριθμό 0.	Ειδοποιεί τον χρήστη ότι υπάρχει ήδη κάποιος που χρησιμοποιεί το ίδιο AMKA.
4	Ο χρήστης εισάγει Password με λιγότερους από 6 χαρακτήρες	400: Bad Request	-	Ειδοποιεί τον χρήστη ότι το Password πρέπει να έχει τουλάχιστον 6 χαρακτήρες.
5	Ο χρήστης εισάγει ConfirmPassword διαφορετικό από το Password	400: Bad Request		Ειδοποιεί τον χρήστη ότι το Password και το ConfirmPassword δεν ταιριάζουν.
6	Η εφαρμογή δε φτάνει ποτέ στο endpoint	500: Internal Server Error	-	Δεν επιστρέφεται κάποιο μήνυμα αλλά σημαίνει ότι το API δεν είναι διαθέσιμο.

Πίνακας 6.2.28: Πίνακας με τα πιθανά responses του Endpoint 1 του πίνακα 6.2.4

AuthPatientController: Endpoint 2 Responses

A/A	Περίπτωση	Status Code	Δεδομένα που επιστρέφει	Μήνυμα
1	Ο χρήστης συμπληρώνει σωστά τον συνδυασμό User Name – Password.	200: Success	Ένα string που αναπαριστά το JWT.	“Ok”
2	Ο χρήστης συμπληρώνει λανθασμένα το User Name του.	400: Bad Request	-	Ειδοποιεί τον χρήστη ότι δε βρέθηκε ο λογαριασμός του.
3	Ο χρήστης συμπληρώνει λανθασμένα το Password του.	400: Bad Request	-	Ειδοποιεί τον χρήστη ότι εισάγει λανθασμένο Password.
4	Η εφαρμογή δε φτάνει ποτέ στο endpoint	500: Internal Server Error	-	Δεν επιστρέφεται κάποιο μήνυμα αλλά σημαίνει ότι το API δεν είναι διαθέσιμο.

Πίνακας 6.2.29: Πίνακας με τα πιθανά responses του Endpoint 2 του πίνακα 6.2.4

AuthPatientController: Endpoint 3 Responses

A/A	Περίπτωση	Status Code	Δεδομένα που επιστρέφει	Μήνυμα
1	Ο χρήστης συμπληρώνει σωστά το email και το ΑΜΚΑ του.	200: Success	Ένα string “Ok”.	Ειδοποιεί τον χρήστη ότι του έστειλε email με κωδικό μίας χρήσης.
2	Ο χρήστης συμπληρώνει	200: Success	-	Ειδοποιεί τον χρήστη ότι δε βρέθηκε ο

	λανθασμένα το email του			λογαριασμός του.
3	Ο χρήστης συμπληρώνει λανθασμένα το ΑΜΚΑ του.	200: Success	-	Ειδοποιεί τον χρήστη ότι δε βρέθηκε ο λογαριασμός του.
4	Η εφαρμογή δε φτάνει ποτέ στο endpoint	500: Internal Server Error	-	Δεν επιστρέφεται κάποιο μήνυμα αλλά σημαίνει ότι το API δεν είναι διαθέσιμο.

Πίνακας 6.2.30: Πίνακας με τα πιθανά responses του Endpoint 3 του πίνακα 6.2.4

AuthPatientController: Endpoint 4 Responses				
A/A	Περίπτωση	Status Code	Δεδομένα που επιστρέφει	Μήνυμα
1	Ο χρήστης συμπληρώνει σωστά το email, το ΑΜΚΑ του και τον κωδικό μίας χρήσης που του ήρθε με email.	200: Success	-	Ειδοποιεί τον χρήστη ότι μπορεί να αλλάξει το Password του.
2	Ο χρήστης συμπληρώνει σωστά τα πεδία email, ΑΜΚΑ και κωδικό μίας χρήσης αλλά ο κωδικός μίας χρήσης έχει λήξει	200: Success	-	Ειδοποιεί τον χρήστη ότι ο κωδικός μίας χρήσης έχει λήξει.
3	Ο χρήστης συμπληρώνει λανθασμένα το email του.	200: Success	-	Ειδοποιεί τον χρήστη ότι δε βρέθηκε ο λογαριασμός του.
4	Ο χρήστης συμπληρώνει	200: Success	-	Ειδοποιεί τον χρήστη ότι δε βρέθηκε ο

	λανθασμένα το ΑΜΚΑ του.			λογαριασμός του.
5	Ο χρήστης συμπληρώνει λανθασμένα τον κωδικό μίας χρήσης του.	200: Success	-	Ειδοποιεί τον χρήστη ότι εισάγει λάθος κωδικό μίας χρήσης.
6	Η εφαρμογή δε φτάνει ποτέ στο endpoint	500: Internal Server Error	-	Δεν επιστρέφεται κάποιο μήνυμα αλλά σημαίνει ότι το API δεν είναι διαθέσιμο.

Πίνακας 6.2.31: Πίνακας με τα πιθανά responses του Endpoint 4 του πίνακα 6.2.4

AuthPatientController: Endpoint 5 Responses				
A/A	Περίπτωση	Status Code	Δεδομένα που επιστρέφει	Μήνυμα
1	Ο χρήστης συμπληρώνει σωστά τα ζητούμενα πεδία	200: Success	-	Ειδοποιεί τον χρήστη ότι άλλαξε το Password του.
2	Ο χρήστης συμπληρώνει σωστά τα ζητούμενα πεδία αλλά ο κωδικός μίας χρήσης έχει λήξει.	200: Success	-	Ειδοποιεί τον χρήστη ότι ο κωδικός μίας χρήσης έχει λήξει.
3	Ο χρήστης συμπληρώνει λανθασμένα το email του.	200: Success	-	Ειδοποιεί τον χρήστη ότι δε βρέθηκε ο λογαριασμός του.
4	Ο χρήστης συμπληρώνει λανθασμένα το ΑΜΚΑ του.	200: Success	-	Ειδοποιεί τον χρήστη ότι δε βρέθηκε ο λογαριασμός του.

5	Ο χρήστης εισάγει Password με λιγότερους από 6 χαρακτήρες	400: Bad Request	-	Ειδοποιεί τον χρήστη ότι το Password πρέπει να έχει τουλάχιστον 6 χαρακτήρες.
6	Ο χρήστης εισάγει ConfirmPassword διαφορετικό από το Password	400: Bad Request		Ειδοποιεί τον χρήστη ότι το Password και το ConfirmPassword δεν ταιριάζουν.
7	Η εφαρμογή δε φτάνει ποτέ στο endpoint	500: Internal Server Error	-	Δεν επιστρέφεται κάποιο μήνυμα αλλά σημαίνει ότι το API δεν είναι διαθέσιμο.

Πίνακας 6.2.32: Πίνακας με τα πιθανά responses του Endpoint 5 του πίνακα 6.2.4

6.2.2 PatientProfileController

Αυτός ο controller διαχειρίζεται όλα τα αιτήματα που σχετίζονται με τον λογαριασμό του χρήστη - ασθενή. Μέσω των endpoints αυτού του controller, ο χρήστης μπορεί να επεξεργαστεί πληροφορίες του λογαριασμού του, να δει τη λίστα με τους ιατρούς που τον παρακολουθούν και να διαχειριστεί αιτήματα των γιατρών του. Τα endpoints που περιλαμβάνει φαίνονται στον παρακάτω πίνακα:

PatientProfileController (http://localhost:5288/PatientProfile)					
A/A	Endpoint	HTTP Μέθοδος	Περιγραφή	Λεδομένα Εισόδου στο endpoint	Λεδομένα εισόδου στο σώμα της μεθόδου
1	/patientProfile	GET	Επιστρέφει τον λογαριασμό του χρήστη - ασθενή	-	-
2	/doctors	GET	Επιστρέφει τους ιατρούς που έχουν διασυνδεθεί με το	-	-

			ιστορικό του χρήστη - ασθενή		
3	/updateProfile	PUT	Κάνει αίτημα για να αλλάξει κάποια πεδία που σχετίζονται με τον λογαριασμό του χρήστη - ασθενή	-	UpdateUserPatientDto Πίνακας 6.2.6
4	/permissions	GET	Επιστρέφει όλα τα δεδομένα που σχετίζονται με τα αιτήματα πρόσβασης των γιατρών στα δεδομένα που εισάγει ο ασθενής από την εφαρμογή του κινητού του.	-	-
5	/addPermission	POST	Προσθέτει άδεια πρόσβασης στα δεδομένα του κατόπιν αιτήματος από κάποιο γιατρό που είναι διασυνδεδεμένος με το ιστορικό του.	-	UpdateDoctorPermissionDto Πίνακας 6.2.7
6	/updatePermission	PUT	Ο χρήστης μπορεί να αλλάξει την επιλογή του σχετικά με το αν ο γιατρός μπορεί να έχει άδεια πρόσβασης στα δεδομένα του.	-	UpdateDoctorPermissionDto Πίνακας 6.2.7

Πίνακας 6.2.33: Endpoints του PatientProfile Controller

PatientProfileController: Endpoint 1 Responses

A/A	Περίπτωση	Status Code	Δεδομένα που επιστρέφει	Μήνυμα
1	Ο χρήστης έχει πιστοποιηθεί από το API με JWT	200: Success	GetPatientUserDto Πίνακας 6.2.4	“OK”
2	Ο χρήστης δεν έχει πιστοποιηθεί από το API με JWT	401: Unauthorized	-	Δεν επιστρέφεται κάποιο μήνυμα αλλά σημαίνει ότι ο χρήστης δεν έχει πρόσβαση στο endpoint διότι δεν έχει πιστοποιηθεί από το API.
3	Η εφαρμογή δε φτάνει ποτέ στο endpoint	500: Internal Server Error	-	Δεν επιστρέφεται κάποιο μήνυμα αλλά σημαίνει ότι το API δεν είναι διαθέσιμο.

Πίνακας 6.2.34: Πίνακας με τα πιθανά responses του Endpoint 1 του πίνακα 6.2.33

PatientProfileController: Endpoint 2 Responses

A/A	Περίπτωση	Status Code	Δεδομένα που επιστρέφει	Μήνυμα
1	Ο χρήστης έχει πιστοποιηθεί από το API με JWT	200: Success	GetPatientDoctorDto Πίνακας 6.2.5	“OK”
2	Ο χρήστης δεν έχει πιστοποιηθεί από το API με JWT	401: Unauthorized	-	Δεν επιστρέφεται κάποιο μήνυμα αλλά σημαίνει ότι ο χρήστης δεν έχει

				πρόσβαση στο endpoint διότι δεν έχει πιστοποιηθεί από το API.
3	Η εφαρμογή δε φτάνει ποτέ στο endpoint	500: Internal Server Error	-	Δεν επιστρέφεται κάποιο μήνυμα αλλά σημαίνει ότι το API δεν είναι διαθέσιμο.

Πίνακας 6.2.35: Πίνακας με τα πιθανά responses του Endpoint 2 του πίνακα 6.2.33

PatientProfileController: Endpoint 3 Responses				
A/A	Περίπτωση	Status Code	Δεδομένα που επιστρέφει	Μήνυμα
1	Ο χρήστης έχει πιστοποιηθεί από το API με JWT	200: Success	Ανανεωμένο το GetPatientUserDto Πίνακας 6.2.4	“OK”
2	Ο χρήστης δεν έχει πιστοποιηθεί από το API με JWT	401: Unauthorized	-	Δεν επιστρέφεται κάποιο μήνυμα αλλά σημαίνει ότι ο χρήστης δεν έχει πρόσβαση στο endpoint διότι δεν έχει πιστοποιηθεί από το API.
3	Η εφαρμογή δε φτάνει ποτέ στο endpoint	500: Internal Server Error	-	Δεν επιστρέφεται κάποιο μήνυμα αλλά σημαίνει ότι το API δεν είναι διαθέσιμο.

Πίνακας 6.2.36: Πίνακας με τα πιθανά responses του Endpoint 3 του πίνακα 6.2.33

PatientProfileController: Endpoint 4 Responses

A/A	Περίπτωση	Status Code	Δεδομένα που επιστρέφει	Μήνυμα
1	Ο χρήστης έχει πιστοποιηθεί από το API με JWT και έχει αιτήματα πρόσβασης.	200: Success	Λίστα με GetPatientDoctorPermissionDto	“OK”
2	Ο χρήστης έχει πιστοποιηθεί από το API με JWT και δεν έχει αιτήματα πρόσβασης από κάποιο γιατρό.	200: Success	-	Ειδοποιεί τον χρήστη ότι δεν έχει αιτήματα από κάποιο γιατρό.
3	Ο χρήστης δεν έχει πιστοποιηθεί από το API με JWT	401: Unauthorized	-	Δεν επιστρέφεται κάποιο μήνυμα αλλά σημαίνει ότι ο χρήστης δεν έχει πρόσβαση στο endpoint διότι δεν έχει πιστοποιηθεί από το API.
4	Η εφαρμογή δε φτάνει ποτέ στο endpoint	500: Internal Server Error	-	Δεν επιστρέφεται κάποιο μήνυμα αλλά σημαίνει ότι το API δεν

				είναι διαθέσιμο.
--	--	--	--	------------------

Πίνακας 6.2.37: Πίνακας με τα πιθανά responses του Endpoint 4 του πίνακα 6.2.33

PatientProfileController: Endpoint 5 Responses				
A/A	Περίπτωση	Status Code	Δεδομένα που επιστρέφει	Μήνυμα
1	Ο χρήστης έχει πιστοποιηθεί από το API με JWT.	200: Success	GetPatientDoctorPermissionDto	“OK”
2	Ο χρήστης έχει πιστοποιηθεί από το API με JWT και εισάγει λανθασμένο ΑΜΚΑ	200: Success	-	Ειδοποιεί τον χρήστη ότι εισάγει λάθος ΑΜΚΑ.
3	Ο χρήστης δεν έχει πιστοποιηθεί από το API με JWT	401: Unauthorized	-	Δεν επιστρέφεται κάποιο μήνυμα αλλά σημαίνει ότι ο χρήστης δεν έχει πρόσβαση στο endpoint διότι δεν έχει πιστοποιηθεί από το API.
4	Η εφαρμογή δε φτάνει ποτέ στο endpoint	500: Internal Server Error	-	Δεν επιστρέφεται κάποιο μήνυμα αλλά σημαίνει ότι το API δεν

				είναι διαθέσιμο.
--	--	--	--	------------------

Πίνακας 6.2.38: Πίνακας με τα πιθανά responses του Endpoint 5 του πίνακα 6.2.33

PatientProfileController: Endpoint 6 Responses				
A/A	Περίπτωση	Status Code	Δεδομένα που επιστρέφει	Μήνυμα
1	Ο χρήστης έχει πιστοποιηθεί από το API με JWT.	200: Success	GetPatientUserDto Πίνακας 6.2.4	“OK”
2	Ο χρήστης έχει πιστοποιηθεί από το API με JWT και εισάγει λανθασμένο ΑΜΚΑ	200: Success	-	Ειδοποιεί τον χρήστη ότι εισάγει λάθος ΑΜΚΑ.
3	Ο χρήστης έχει πιστοποιηθεί από το API με JWT και εισάγει λανθασμένο γιατρό	200: Success	-	Ειδοποιεί τον χρήστη ότι δε βρέθηκε ο ιατρός.
3	Ο χρήστης δεν έχει πιστοποιηθεί από το API με JWT	401: Unauthorized	-	Δεν επιστρέφεται κάποιο μήνυμα αλλά σημαίνει ότι ο χρήστης δεν έχει πρόσβαση στο endpoint διότι δεν έχει πιστοποιηθεί από το API.
4	Η εφαρμογή δε φτάνει ποτέ στο endpoint	500: Internal Server Error	-	Δεν επιστρέφεται κάποιο μήνυμα αλλά

				σημαίνει ότι το API δεν είναι διαθέσιμο.
--	--	--	--	--

Πίνακας 6.2.39: Πίνακας με τα πιθανά responses του Endpoint 5 του πίνακα 6.2.33

6.2.3 FunctionalitiesController

Ο παρακάτω controller διαχειρίζεται όλα τα requests που σχετίζονται με τα δεδομένα της καθημερινότητας του ασθενούς που εκείνος εισάγει στο σύστημα μέσω της εφαρμογής του κινητού του. Αυτά τα endpoints προσφέρουν τη δυνατότητα στον χρήστη να προσθέσει δεδομένα για τη γλυκόζη και για την αρτηριακή του πίεση. Επίσης προσθέτει τα καθημερινά γεύματα, την άσκησή του, το βάρος του, τον αν νοσεί από κάποια άλλη ασθένεια και τη φαρμακευτική του αγωγή. Έχει τη δυνατότητα να προσθέσει την καθημερινή του διάθεση αλλά και να λάβει την τελευταία που αποθήκευσε στο σύστημα.

Με βάση τα δεδομένα γλυκόζης που εισάγει στο σύστημα σε καθημερινή βάση μπορεί να λάβει αναλύσεις για τα δεδομένα γλυκόζης της τελευταίας εβδομάδας, των τελευταίων 15 ημερών και του τελευταίου μήνα. Επίσης λαμβάνει ποσοστό επί του συνόλου των μετρήσεων για κάθε είδος μέτρησης γλυκόζης και για χρονικό διάστημα μίας εβδομάδας, 15 ημερών και ενός μήνα. Τέλος λαμβάνει ένα ποσοστό για κάθε μέτρηση επί του συνόλου των μετρήσεων που έχουν εισαχθεί από τον ασθενή στο σύστημα.

Στον παρακάτω πίνακα φαίνονται όλα τα endpoints που περιλαμβάνει:

FunctionalitiesController (http://localhost:5288/PatientProfile)					
A/A	Endpoint	HTTP Μέθοδος	Περιγραφή	Δεδομένα Εισόδου στο endpoint	Δεδομένα εισόδου στο σώμα της μεθόδου
1	/addGlucose	POST	Προσθέτει δεδομένα γλυκόζης στη βάση.	-	AddDailyGlucoseDto Πίνακας 6.2.12
2	/weekglucose	GET	Επιστρέφει λίστα με τα δεδομένα γλυκόζης της τελευταίας εβδομάδας.	-	-

3	/semimonthglucose	GET	Επιστρέφει λίστα με τα δεδομένα γλυκόζης των τελευταίων 15 ημερών.	-	-
4	/monthglucose	GET	Επιστρέφει λίστα με τα δεδομένα γλυκόζης του τελευταίου μήνα.	-	-
5	/weekglucosepercent	GET	Επιστρέφει το ποσοστό επί του συνόλου των εβδομαδιαίων μετρήσεων, για κάθε είδος μέτρησης γλυκόζης.	-	-
6	/semimonthglucosepercent	GET	Επιστρέφει το ποσοστό επί του συνόλου των μετρήσεων 15 ημερών, για κάθε είδος μέτρησης γλυκόζης.	-	-
7	/monthglucosepercent	GET	Επιστρέφει το ποσοστό επί του συνόλου των μηνιαίων μετρήσεων, για κάθε είδος μέτρησης γλυκόζης.	-	-
8	/totalglucosepercent	GET	Επιστρέφει το ποσοστό επί του συνόλου των μετρήσεων, για κάθε είδος μέτρησης γλυκόζης.	-	-

9	/addBloodPressure	POST	Προσθέτει δεδομένα πίεσης στη βάση	-	AddBloodPressureDto Πίνακας 6.2.15
10	/addMeal	POST	Προσθέτει δεδομένα γεύματος στη βάση	-	AddPatientMealDto Πίνακας 6.2.17
11	/addActivity	POST	Προσθέτει δεδομένα άσκησης στη βάση	-	AddPatientActivityDto Πίνακας 6.2.19
12	/addWeight	POST	Προσθέτει δεδομένα σωματικού βάρους στη βάση	-	AddPatientWeightDto Πίνακας 6.2.21
13	/addIllness	POST	Προσθέτει δεδομένα ασθένειας στη βάση	-	AddPatientIllnessDto Πίνακας 6.2.23
14	/addMedication	POST	Προσθέτει δεδομένα φαρμακευτικής αγωγής στη βάση	-	AddPatientMedicationDto Πίνακας 6.2.25
15	/addMood	POST	Προσθέτει δεδομένα διάθεσης στη βάση	-	AddPatientMoodDto Πίνακας 6.2.10
16	/currentMood	GET	Φέρνει την τελευταία εγγραφή διάθεσης	-	-

			του χρήστη από τη βάση		
--	--	--	------------------------	--	--

Πίνακας 6.2.40: Endpoints του Functionalities Controller

FunctionalitiesController: Endpoint 1 Responses				
A/A	Περίπτωση	Status Code	Δεδομένα που επιστρέφει	Μήνυμα
1	Ο χρήστης έχει πιστοποιηθεί από το API με JWT.	200: Success	GetDailyGlucoseDto Πίνακας 6.2.11	“OK”
2	Ο χρήστης δεν έχει πιστοποιηθεί από το API με JWT	401: Unauthorized	-	Δεν επιστρέφεται κάποιο μήνυμα αλλά σημαίνει ότι ο χρήστης δεν έχει πρόσβαση στο endpoint διότι δεν έχει πιστοποιηθεί από το API.
3	Η εφαρμογή δε φτάνει ποτέ στο endpoint	500: Internal Server Error	-	Δεν επιστρέφεται κάποιο μήνυμα αλλά σημαίνει ότι το API δεν είναι διαθέσιμο.

Πίνακας 6.2.41: Πίνακας με τα πιθανά responses του Endpoint 1 του πίνακα 6.2.40

FunctionalitiesController: Endpoint 2 Responses				
A/A	Περίπτωση	Status Code	Δεδομένα που επιστρέφει	Μήνυμα
1	Ο χρήστης έχει πιστοποιηθεί	200: Success	Λίστα με GetDailyGlucoseDto	“OK”

	από το API με JWT.		Πίνακας 6.2.11 για την τελευταία εβδομάδα	
2	Ο χρήστης έχει πιστοποιηθεί από το API με JWT αλλά δεν έχει δεδομένα γλυκόζης	200: Success	-	Ειδοποιείται ο χρήστης ότι δεν έχει δεδομένα γλυκόζης.
3	Ο χρήστης δεν έχει πιστοποιηθεί από το API με JWT	401: Unauthorized	-	Δεν επιστρέφεται κάποιο μήνυμα αλλά σημαίνει ότι ο χρήστης δεν έχει πρόσβαση στο endpoint διότι δεν έχει πιστοποιηθεί από το API.
4	Η εφαρμογή δε φτάνει ποτέ στο endpoint	500: Internal Server Error	-	Δεν επιστρέφεται κάποιο μήνυμα αλλά σημαίνει ότι το API δεν είναι διαθέσιμο.

Πίνακας 6.2.42: Πίνακας με τα πιθανά responses του Endpoint 2 του πίνακα 6.2.40

FunctionalitiesController: Endpoint 3 Responses

A/A	Περίπτωση	Status Code	Δεδομένα που επιστρέφει	Μήνυμα
1	Ο χρήστης έχει πιστοποιηθεί από το API με JWT.	200: Success	Λίστα με GetDailyGlucoseDto Πίνακας 6.2.11 για τις τελευταίες 15 μέρες	“OK”
2	Ο χρήστης έχει πιστοποιηθεί από το API με JWT αλλά δεν	200: Success	-	Ειδοποιείται ο χρήστης ότι δεν έχει

	έχει δεδομένα γλυκόζης			δεδομένα γλυκόζης.
3	Ο χρήστης δεν έχει πιστοποιηθεί από το API με JWT	401: Unauthorized	-	Δεν επιστρέφεται κάποιο μήνυμα αλλά σημαίνει ότι ο χρήστης δεν έχει πρόσβαση στο endpoint διότι δεν έχει πιστοποιηθεί από το API.
4	Η εφαρμογή δε φτάνει ποτέ στο endpoint	500: Internal Server Error	-	Δεν επιστρέφεται κάποιο μήνυμα αλλά σημαίνει ότι το API δεν είναι διαθέσιμο.

Πίνακας 6.2.43: Πίνακας με τα πιθανά responses του Endpoint 3 του πίνακα 6.2.40

FunctionalitiesController: Endpoint 4 Responses				
A/A	Περίπτωση	Status Code	Δεδομένα που επιστρέφει	Μήνυμα
1	Ο χρήστης έχει πιστοποιηθεί από το API με JWT.	200: Success	Λίστα με GetDailyGlucoseDto Πίνακας 6.2.11 για τον τελευταίο μήνα	“OK”
2	Ο χρήστης έχει πιστοποιηθεί από το API με JWT αλλά δεν έχει δεδομένα γλυκόζης	200: Success	-	Ειδοποιείται ο χρήστης ότι δεν έχει δεδομένα γλυκόζης.
3	Ο χρήστης δεν έχει πιστοποιηθεί	401: Unauthorized	-	Δεν επιστρέφεται κάποιο μήνυμα αλλά

	από το API με JWT			σημαίνει ότι ο χρήστης δεν έχει πρόσβαση στο endpoint διότι δεν έχει πιστοποιηθεί από το API.
4	Η εφαρμογή δε φτάνει ποτέ στο endpoint	500: Internal Server Error	-	Δεν επιστρέφεται κάποιο μήνυμα αλλά σημαίνει ότι το API δεν είναι διαθέσιμο.

Πίνακας 6.2.44: Πίνακας με τα πιθανά responses του Endpoint 4 του πίνακα 6.2.40

FunctionalitiesController: Endpoint 5 Responses				
A/A	Περίπτωση	Status Code	Δεδομένα που επιστρέφει	Μήνυμα
1	Ο χρήστης έχει πιστοποιηθεί από το API με JWT.	200: Success	GetGlucosePercentsDto για τον τελευταία εβδομάδα Πίνακας 6.2.13	“OK”
2	Ο χρήστης έχει πιστοποιηθεί από το API με JWT αλλά δεν έχει δεδομένα γλυκόζης	200: Success	-	Ειδοποιείται ο χρήστης ότι δεν έχει δεδομένα γλυκόζης.
3	Ο χρήστης δεν έχει πιστοποιηθεί από το API με JWT	401: Unauthorized	-	Δεν επιστρέφεται κάποιο μήνυμα αλλά σημαίνει ότι ο χρήστης δεν έχει πρόσβαση στο endpoint διότι δεν έχει

				πιστοποιηθεί από το API.
3	Η εφαρμογή δε φτάνει ποτέ στο endpoint	500: Internal Server Error	-	Δεν επιστρέφεται κάποιο μήνυμα αλλά σημαίνει ότι το API δεν είναι διαθέσιμο.

Πίνακας 6.2.45: Πίνακας με τα πιθανά responses του Endpoint 5 του πίνακα 6.2.40

FunctionalitiesController: Endpoint 6 Responses				
A/A	Περίπτωση	Status Code	Δεδομένα που επιστρέφει	Μήνυμα
1	Ο χρήστης έχει πιστοποιηθεί από το API με JWT.	200: Success	GetGlucosePercentsDto για τις τελευταίες 15 μέρες Πίνακας 6.2.13	“OK”
2	Ο χρήστης έχει πιστοποιηθεί από το API με JWT αλλά δεν έχει δεδομένα γλυκόζης	200: Success	-	Ειδοποιείται ο χρήστης ότι δεν έχει δεδομένα γλυκόζης.
3	Ο χρήστης δεν έχει πιστοποιηθεί από το API με JWT	401: Unauthorized	-	Δεν επιστρέφεται κάποιο μήνυμα αλλά σημαίνει ότι ο χρήστης δεν έχει πρόσβαση στο endpoint διότι δεν έχει πιστοποιηθεί από το API.
3	Η εφαρμογή δε φτάνει ποτέ στο endpoint	500: Internal Server Error	-	Δεν επιστρέφεται κάποιο μήνυμα αλλά

				σημαίνει ότι το API δεν είναι διαθέσιμο.
--	--	--	--	--

Πίνακας 6.2.46: Πίνακας με τα πιθανά responses του Endpoint 6 του πίνακα 6.2.40

FunctionalitiesController: Endpoint 7 Responses				
A/A	Περίπτωση	Status Code	Δεδομένα που επιστρέφει	Μήνυμα
1	Ο χρήστης έχει πιστοποιηθεί από το API με JWT.	200: Success	GetGlucosePercentsDto για τον τελευταίο μήνα Πίνακας 6.2.13	“OK”
2	Ο χρήστης έχει πιστοποιηθεί από το API με JWT αλλά δεν έχει δεδομένα γλυκόζης	200: Success	-	Ειδοποιείται ο χρήστης ότι δεν έχει δεδομένα γλυκόζης.
3	Ο χρήστης δεν έχει πιστοποιηθεί από το API με JWT	401: Unauthorized	-	Δεν επιστρέφεται κάποιο μήνυμα αλλά σημαίνει ότι ο χρήστης δεν έχει πρόσβαση στο endpoint διότι δεν έχει πιστοποιηθεί από το API.
3	Η εφαρμογή δε φτάνει ποτέ στο endpoint	500: Internal Server Error	-	Δεν επιστρέφεται κάποιο μήνυμα αλλά σημαίνει ότι το API δεν είναι διαθέσιμο.

Πίνακας 6.2.47: Πίνακας με τα πιθανά responses του Endpoint 7 του πίνακα 6.2.40

FunctionalitiesController: Endpoint 8 Responses

A/A	Περίπτωση	Status Code	Δεδομένα που επιστρέφει	Μήνυμα
1	Ο χρήστης έχει πιστοποιηθεί από το API με JWT.	200: Success	GetGlucosePercentsDto για όλα τα δεδομένα γλυκόζης Πίνακας 6.2.13	“OK”
2	Ο χρήστης έχει πιστοποιηθεί από το API με JWT αλλά δεν έχει δεδομένα γλυκόζης	200: Success	-	Ειδοποιείται ο χρήστης ότι δεν έχει δεδομένα γλυκόζης.
3	Ο χρήστης δεν έχει πιστοποιηθεί από το API με JWT	401: Unauthorized	-	Δεν επιστρέφεται κάποιο μήνυμα αλλά σημαίνει ότι ο χρήστης δεν έχει πρόσβαση στο endpoint διότι δεν έχει πιστοποιηθεί από το API.
3	Η εφαρμογή δε φτάνει ποτέ στο endpoint	500: Internal Server Error	-	Δεν επιστρέφεται κάποιο μήνυμα αλλά σημαίνει ότι το API δεν είναι διαθέσιμο.

Πίνακας 6.2.48: Πίνακας με τα πιθανά responses του Endpoint 8 του πίνακα 6.2.40

FunctionalitiesController: Endpoint 9 Responses

A/A	Περίπτωση	Status Code	Δεδομένα που επιστρέφει	Μήνυμα
1	Ο χρήστης έχει πιστοποιηθεί	200: Success	GetBloodPressureDto	“OK”

	από το API με JWT.		Πίνακας 6.2.14	
2	Ο χρήστης δεν έχει πιστοποιηθεί από το API με JWT	401: Unauthorized	-	Δεν επιστρέφεται κάποιο μήνυμα αλλά σημαίνει ότι ο χρήστης δεν έχει πρόσβαση στο endpoint διότι δεν έχει πιστοποιηθεί από το API.
3	Η εφαρμογή δε φτάνει ποτέ στο endpoint	500: Internal Server Error	-	Δεν επιστρέφεται κάποιο μήνυμα αλλά σημαίνει ότι το API δεν είναι διαθέσιμο.

Πίνακας 6.2.48: Πίνακας με τα πιθανά responses του Endpoint 9 του πίνακα 6.2.40

FunctionalitiesController: Endpoint 10 Responses				
A/A	Περίπτωση	Status Code	Δεδομένα που επιστρέφει	Μήνυμα
1	Ο χρήστης έχει πιστοποιηθεί από το API με JWT.	200: Success	GetPatientMealDto Πίνακας 6.2.16	“OK”
2	Ο χρήστης δεν έχει πιστοποιηθεί από το API με JWT	401: Unauthorized	-	Δεν επιστρέφεται κάποιο μήνυμα αλλά σημαίνει ότι ο χρήστης δεν έχει πρόσβαση στο endpoint διότι δεν έχει

				πιστοποιηθεί από το API.
3	Η εφαρμογή δε φτάνει ποτέ στο endpoint	500: Internal Server Error	-	Δεν επιστρέφεται κάποιο μήνυμα αλλά σημαίνει ότι το API δεν είναι διαθέσιμο.

Πίνακας 6.2.49: Πίνακας με τα πιθανά responses του Endpoint 10 του πίνακα 6.2.40

FunctionalitiesController: Endpoint 11 Responses				
A/A	Περίπτωση	Status Code	Δεδομένα που επιστρέφει	Μήνυμα
1	Ο χρήστης έχει πιστοποιηθεί από το API με JWT.	200: Success	GetPatientActivityDto Πίνακας 6.2.18	“OK”
2	Ο χρήστης δεν έχει πιστοποιηθεί από το API με JWT	401: Unauthorized	-	Δεν επιστρέφεται κάποιο μήνυμα αλλά σημαίνει ότι ο χρήστης δεν έχει πρόσβαση στο endpoint διότι δεν έχει πιστοποιηθεί από το API.
3	Η εφαρμογή δε φτάνει ποτέ στο endpoint	500: Internal Server Error	-	Δεν επιστρέφεται κάποιο μήνυμα αλλά σημαίνει ότι το API δεν είναι διαθέσιμο.

Πίνακας 6.2.49: Πίνακας με τα πιθανά responses του Endpoint 11 του πίνακα 6.2.40

FunctionalitiesController: Endpoint 12 Responses

A/A	Περίπτωση	Status Code	Δεδομένα που επιστρέφει	Μήνυμα
1	Ο χρήστης έχει πιστοποιηθεί από το API με JWT.	200: Success	AddPatientWeightDto Πίνακας 6.2.21	“OK”
2	Ο χρήστης δεν έχει πιστοποιηθεί από το API με JWT	401: Unauthorized	-	Δεν επιστρέφεται κάποιο μήνυμα αλλά σημαίνει ότι ο χρήστης δεν έχει πρόσβαση στο endpoint διότι δεν έχει πιστοποιηθεί από το API.
3	Η εφαρμογή δε φτάνει ποτέ στο endpoint	500: Internal Server Error	-	Δεν επιστρέφεται κάποιο μήνυμα αλλά σημαίνει ότι το API δεν είναι διαθέσιμο.

Πίνακας 6.2.50: Πίνακας με τα πιθανά responses του Endpoint 12 του πίνακα 6.2.40

FunctionalitiesController: Endpoint 13 Responses

A/A	Περίπτωση	Status Code	Δεδομένα που επιστρέφει	Μήνυμα
1	Ο χρήστης έχει πιστοποιηθεί από το API με JWT.	200: Success	GetPatientIllnessDto Πίνακας 6.2.22	“OK”
2	Ο χρήστης δεν έχει πιστοποιηθεί	401: Unauthorized	-	Δεν επιστρέφεται κάποιο μήνυμα αλλά

	από το API με JWT			σημαίνει ότι ο χρήστης δεν έχει πρόσβαση στο endpoint διότι δεν έχει πιστοποιηθεί από το API.
3	Η εφαρμογή δε φτάνει ποτέ στο endpoint	500: Internal Server Error	-	Δεν επιστρέφεται κάποιο μήνυμα αλλά σημαίνει ότι το API δεν είναι διαθέσιμο.

Πίνακας 6.2.51: Πίνακας με τα πιθανά responses του Endpoint 13 του πίνακα 6.2.40

FunctionalitiesController: Endpoint 14 Responses				
A/A	Περίπτωση	Status Code	Δεδομένα που επιστρέφει	Μήνυμα
1	Ο χρήστης έχει πιστοποιηθεί από το API με JWT.	200: Success	GetPatientMedicationDto Πίνακας 6.2.24	“OK”
2	Ο χρήστης δεν έχει πιστοποιηθεί από το API με JWT	401: Unauthorized	-	Δεν επιστρέφεται κάποιο μήνυμα αλλά σημαίνει ότι ο χρήστης δεν έχει πρόσβαση στο endpoint διότι δεν έχει πιστοποιηθεί από το API.
3	Η εφαρμογή δε φτάνει ποτέ στο endpoint	500: Internal Server Error	-	Δεν επιστρέφεται κάποιο μήνυμα αλλά σημαίνει ότι

				το API δεν είναι διαθέσιμο.
--	--	--	--	-----------------------------

Πίνακας 6.2.52: Πίνακας με τα πιθανά responses του Endpoint 14 του πίνακα 6.2.40

FunctionalitiesController: Endpoint 15 Responses				
A/A	Περίπτωση	Status Code	Λεδομένα που επιστρέφει	Μήνυμα
1	Ο χρήστης έχει πιστοποιηθεί από το API με JWT.	200: Success	GetPatientMoodDto Πίνακας 6.2.9	“OK”
2	Ο χρήστης δεν έχει πιστοποιηθεί από το API με JWT	401: Unauthorized	-	Δεν επιστρέφεται κάποιο μήνυμα αλλά σημαίνει ότι ο χρήστης δεν έχει πρόσβαση στο endpoint διότι δεν έχει πιστοποιηθεί από το API.
3	Η εφαρμογή δε φτάνει ποτέ στο endpoint	500: Internal Server Error	-	Δεν επιστρέφεται κάποιο μήνυμα αλλά σημαίνει ότι το API δεν είναι διαθέσιμο.

Πίνακας 6.2.53: Πίνακας με τα πιθανά responses του Endpoint 15 του πίνακα 6.2.40

FunctionalitiesController: Endpoint 16 Responses				
A/A	Περίπτωση	Status Code	Λεδομένα που επιστρέφει	Μήνυμα
1	Ο χρήστης έχει πιστοποιηθεί	200: Success	GetPatientMoodDto	“OK”

	από το API με JWT.		Πίνακας 6.2.9	
2	Ο χρήστης δεν έχει πιστοποιηθεί από το API με JWT	401: Unauthorized	-	Δεν επιστρέφεται κάποιο μήνυμα αλλά σημαίνει ότι ο χρήστης δεν έχει πρόσβαση στο endpoint διότι δεν έχει πιστοποιηθεί από το API.
3	Η εφαρμογή δε φτάνει ποτέ στο endpoint	500: Internal Server Error	-	Δεν επιστρέφεται κάποιο μήνυμα αλλά σημαίνει ότι το API δεν είναι διαθέσιμο.

Πίνακας 6.2.54: Πίνακας με τα πιθανά responses του Endpoint 16 του πίνακα 6.2.40

6.2.4 MsCycleController

Αυτός ο controller περιλαμβάνει όλα τα endpoints που αφορούν τον έμμηνο κύκλο. Αν ο χρήστης είναι γένους θηλυκού, τότε έχει πρόσβαση στα δεδομένα αυτής της ενότητας. Με αυτά τα endpoints η ασθενής μπορεί να προσθέσει τον μηνιαίο κύκλο της, να λάβει ετήσια εκτίμηση έμμηνου κύκλου αλλά και να λάβει ιστορικό των κύκλων που έχει αποθηκεύσει στο σύστημα.

Στον παρακάτω πίνακα φαίνονται τα endpoints που περιλαμβάνει:

MsCycleController (http://localhost:5288/MsCycle)					
A/A	Endpoint	HTTP Μέθοδος	Περιγραφή	Δεδομένα Εισόδου στο endpoint	Δεδομένα εισόδου στο σώμα της μεθόδου
1	/ addcycle	POST	Προσθέτει δεδομένα ενός έμμηνου κύκλου	-	AddMenstrual CycleDto Πίνακας 6.2.27
2	/ cycles/history"	GET	Επιστρέφει λίστα με όλα τα δεδομένα έμμηνου κύκλου που σχετίζονται με αυτό τον χρήστη	-	-
3	/ annualCycles	GET	Επιστρέφει λίστα με την πρόβλεψη των έμμηνων κύκλων για διάστημα ενός έτους.	-	-

Πίνακας 6.2.55: Endpoints του MsCycle Controller

Στους παρακάτω πίνακες φαίνονται τα responses των endpoints του πίνακα 6.2.55:

MsCycleController: Endpoint 1 Responses				
A/A	Περίπτωση	Status Code	Δεδομένα που επιστρέφει	Μήνυμα
1	Ο χρήστης έχει πιστοποιηθεί από το API με JWT.	200: Success	GetMenstrualCycleDto Πίνακας 6.2.26	“OK”
2	Ο χρήστης έχει πιστοποιηθεί από το API με JWT αλλά έχει δηλώσει ως	200: Success	-	Ειδοποιεί τον χρήστη ότι λόγω φύλου δεν έχει πρόσβαση σε

	φύλο το αρσενικό			αυτή τη λειτουργία.
3	Ο χρήστης δεν έχει πιστοποιηθεί από το API με JWT	401: Unauthorized	-	Δεν επιστρέφεται κάποιο μήνυμα αλλά σημαίνει ότι ο χρήστης δεν έχει πρόσβαση στο endpoint διότι δεν έχει πιστοποιηθεί από το API.
4	Η εφαρμογή δε φτάνει ποτέ στο endpoint	500: Internal Server Error	-	Δεν επιστρέφεται κάποιο μήνυμα αλλά σημαίνει ότι το API δεν είναι διαθέσιμο.

Πίνακας 6.2.56: Πίνακας με τα πιθανά responses του Endpoint 1 του πίνακα 6.2.55

MsCycleController: Endpoint 2 Responses

A/A	Περίπτωση	Status Code	Δεδομένα που επιστρέφει	Μήνυμα
1	Ο χρήστης έχει πιστοποιηθεί από το API με JWT.	200: Success	Λίστα του τύπου GetMenstrualCycleDto Πίνακας 6.2.26	“OK”
2	Ο χρήστης έχει πιστοποιηθεί από το API με JWT αλλά έχει δηλώσει ως φύλο το αρσενικό	200: Success	-	Ειδοποιεί τον χρήστη ότι λόγω φύλου δεν έχει πρόσβαση σε αυτή τη λειτουργία.
3	Ο χρήστης δεν έχει πιστοποιηθεί	401: Unauthorized	-	Δεν επιστρέφεται κάποιο

	από το API με JWT			μήνυμα αλλά σημαίνει ότι ο χρήστης δεν έχει πρόσβαση στο endpoint διότι δεν έχει πιστοποιηθεί από το API.
4	Η εφαρμογή δε φτάνει ποτέ στο endpoint	500: Internal Server Error	-	Δεν επιστρέφεται κάποιο μήνυμα αλλά σημαίνει ότι το API δεν είναι διαθέσιμο.

Πίνακας 6.2.56: Πίνακας με τα πιθανά responses του Endpoint 2 του πίνακα 6.2.55

MsCycleController: Endpoint 3 Responses

A/A	Περίπτωση	Status Code	Δεδομένα που επιστρέφει	Μήνυμα
1	Ο χρήστης έχει πιστοποιηθεί από το API με JWT.	200: Success	Λίστα του τύπου GetMenstrualCycleDto Πίνακας 6.2.26	“OK”
2	Ο χρήστης έχει πιστοποιηθεί από το API με JWT αλλά έχει δηλώσει ως φύλο το αρσενικό	200: Success	-	Ειδοποιεί τον χρήστη ότι λόγω φύλου δεν έχει πρόσβαση σε αυτή τη λειτουργία.
3	Ο χρήστης δεν έχει πιστοποιηθεί από το API με JWT	401: Unauthorized	-	Δεν επιστρέφεται κάποιο μήνυμα αλλά σημαίνει ότι ο χρήστης δεν έχει πρόσβαση στο endpoint

				διότι δεν έχει πιστοποιηθεί από το API.
4	Η εφαρμογή δε φτάνει ποτέ στο endpoint	500: Internal Server Error	-	Δεν επιστρέφεται κάποιο μήνυμα αλλά σημαίνει ότι το API δεν είναι διαθέσιμο.

Πίνακας 6.2.57: Πίνακας με τα πιθανά responses του Endpoint 3 του πίνακα 6.2.55

6.2.5 NotificationController

Σε αυτό τον controller περιλαμβάνονται τα endpoints που σχετίζονται με τις ειδοποιήσεις που λαμβάνει ο χρήστης στην εφαρμογή του. Αυτός ο controller έχει ένα μόνο endpoint το οποίο φαίνεται στον παρακάτω πίνακα:

NotificationController (http://localhost:5288/NotificationController)					
A/A	Endpoint	HTTP Μέθοδος	Περιγραφή	Δεδομένα Εισόδου στο endpoint	Δεδομένα εισόδου στο σώμα της μεθόδου
1	/list	GET	Φέρνει τη λίστα με τις ειδοποιήσεις που σχετίζονται με αιτήματα που έχει στείλει ο ιατρός στον χρήστη έτσι ώστε να πάρει άδεια πρόσβασης στα δεδομένα που εισάγει στην εφαρμογή του κινητού του.	-	-

Πίνακας 6.2.58: Endpoints του Notification Controller

Στον παρακάτω πίνακα φαίνεται τι επιστρέφει αυτό το endpoint:

NotificationController: Endpoint 1 Responses				
A/A	Περίπτωση	Status Code	Δεδομένα που επιστρέφει	Μήνυμα
1	Ο χρήστης έχει πιστοποιηθεί από το API με JWT.	200: Success	Λίστα του τύπου GetPatientNotificationDto Πίνακας 6.2.26	“OK”
2	Ο χρήστης έχει πιστοποιηθεί από το API με JWT αλλά δεν έχει αιτήματα από ιατρούς.	200: Success	-	Ειδοποιεί τον χρήστη ότι δεν έχει αιτήματα.
3	Ο χρήστης δεν έχει πιστοποιηθεί από το API με JWT	401: Unauthorized	-	Δεν επιστρέφεται κάποιο μήνυμα αλλά σημαίνει ότι ο χρήστης δεν έχει πρόσβαση στο endpoint διότι δεν έχει πιστοποιηθεί από το API.
4	Η εφαρμογή δε φτάνει ποτέ στο endpoint	500: Internal Server Error	-	Δεν επιστρέφεται κάποιο μήνυμα αλλά σημαίνει ότι το API δεν είναι διαθέσιμο.

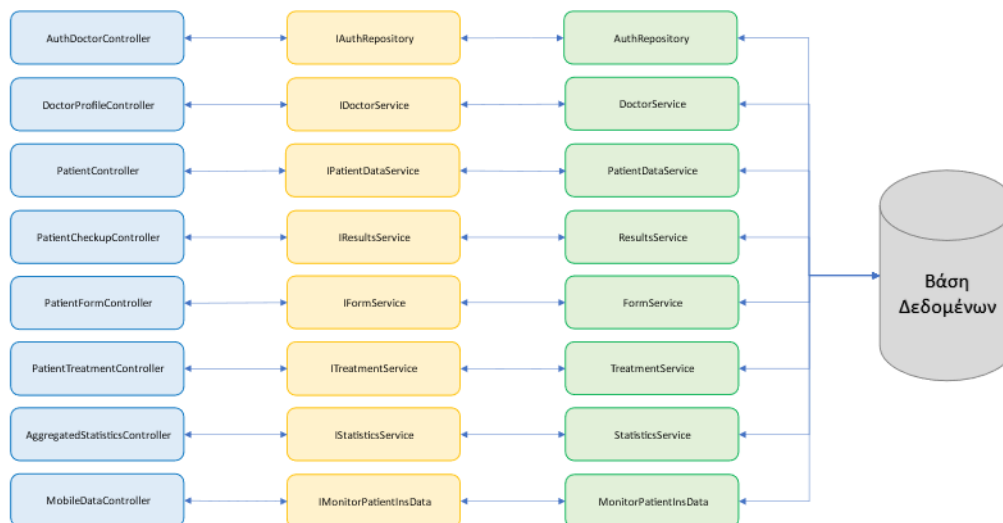
Πίνακας 6.2.59: Πίνακας με τα πιθανά responses του Endpoint 1 του πίνακα 6.2.58

6.3 API της εφαρμογής του ιατρού

Το API που εξυπηρετεί την εφαρμογή του ιατρού αποτελείται από οκτώ controllers όπου ο καθένας μέσω των endpoints του δίνει πρόσβαση σε διάφορες λειτουργίες της εφαρμογής. Όπως αναφέρθηκε στην ενότητα 6.2 για το API της εφαρμογής του ασθενούς, το όνομα του controller βρίσκεται μέσα στο url και είναι η συμβολοσειρά

(string) μέσα στα άγκιστρα. Η γενική μορφή του είναι η εξής: <http://localhost:5230/{controller}>.

Στο παρακάτω σχήμα φαίνεται η δομή του API και του κώδικα που καλεί με στόχο



Σχήμα 6.3.1: Αρχιτεκτονική του API της εφαρμογής του ιατρού

την επεξεργασία των δεδομένων που αιτείται ο χρήστης - ιατρός μέσω της εφαρμογής του.

Τα DTO που σχεδιάστηκαν για αυτό το API είναι τα παρακάτω:

DoctorRegisterDto	
Password	Κωδικός πρόσβασης στην εφαρμογή
ConfirmPassword	Επιβεβαίωση του κωδικού πρόσβασης για την εφαρμογή
UserName	Όνομα χρήστη
LastName	Επώνυμο Χρήστη
FirstName	Μικρό Όνομα Χρήστη
Email	Διεύθυνση ηλεκτρονικού ταχυδρομείου
PhoneNumber	Αριθμός Τηλεφώνου

Country	Χώρα
State	Περιφέρεια
City	Πόλη
PostalCode	Ταχυδρομικός Κώδικας
Medical_Specialty	Ειδικότητα του ιατρού
Workplace	Το Νοσοκομείο που εργάζεται
Address	Διεύθυνση του χρήστη - ιατρού

Πίνακας 6.3.1: DTO για την εγγραφή του ιατρού στο σύστημα

DoctorLoginDto	
UserName	Όνομα χρήστη
Password	Κωδικός Χρήστη

Πίνακας 6.3.2: DTO για την είσοδο του ιατρού στο σύστημα

GetDoctorDto	
Id	Κωδικός εγγραφής του ιατρού στη βάση
UserName	Όνομα χρήστη
LastName	Επώνυμο Χρήστη
FirstName	Μικρό Όνομα Χρήστη
Email	Διεύθυνση ηλεκτρονικού ταχυδρομείου
PhoneNumber	Αριθμός Τηλεφώνου
Country	Χώρα

State	Περιφέρεια
City	Πόλη
PostalCode	Ταχυδρομικός Κώδικας
Medical_Specialty	Ειδικότητα του ιατρού
Workplace	Το Νοσοκομείο που εργάζεται
Address	Διεύθυνση του χρήστη - ιατρού
Patients	Ιστορικά ασθενών που έχουν διασυνδεθεί με τον γιατρό

Πίνακας 6.3.3: DTO για τον λογαριασμό του ιατρού στο σύστημα

UpdateDoctorDto	
UserName	Όνομα χρήστη
LastName	Επώνυμο Χρήστη
FirstName	Μικρό Όνομα Χρήστη
Email	Διεύθυνση ηλεκτρονικού ταχυδρομείου
PhoneNumber	Αριθμός Τηλεφώνου
Country	Χώρα
State	Περιφέρεια
City	Πόλη
PostalCode	Ταχυδρομικός Κώδικας
Medical_Specialty	Ειδικότητα του ιατρού
Workplace	Το Νοσοκομείο που εργάζεται

Address	Διεύθυνση του χρήστη - ιατρού
----------------	-------------------------------

Πίνακας 6.3.4: DTO για την επεξεργασία του λογαριασμού του ιατρού στο σύστημα

GetPatientDto	
Id	Αριθμός εγγραφής στη βάση του ιστορικού του ασθενούς
AMKA	ΑΜΚΑ του ασθενούς
Biological_Gender	Αριθμός που δείχνει το φύλλο του ασθενούς. Παίρνει τις τιμές 1 για το θηλυκό και 2 για το αρσενικό.
BirthDate	Ημερομηνία Γέννησης του ασθενούς
FirstName	Μικρό Όνομα Ασθενή
LastName	Επώνυμο Ασθενή
Email	Διεύθυνση ηλεκτρονικού ταχυδρομείου
PhoneNumber	Αριθμός Τηλεφώνου
SmokingStatus	Αριθμός που δείχνει την κατάσταση του ασθενούς σε σχέση με το κάπνισμα. Η τιμή 1 αντιστοιχεί σε ασθενή που δεν ήταν ποτέ καπνιστής, η τιμή 2 σε πρώην καπνιστή και η τιμή 3 σε καπνιστή.
Parents_Have_Dm	Αριθμός που δείχνει το ιστορικό του ασθενούς όσον αφορά τον διαβήτη. Η τιμή 1 αντιστοιχεί στην περίπτωση που κανένας από τους δύο γονείς του είχαν διαβήτη, η τιμή 2 στην περίπτωση που η μητέρα του ασθενούς είχε διαβήτη, η τιμή 3 στην περίπτωση που ο πατέρας του ασθενούς είχε διαβήτη και η τιμή 4 στην περίπτωση που και οι δύο γονείς έχουν ιστορικό.
DmDiagnosisDate	Ημερομηνία Διάγνωσης
Address	Διεύθυνση Κατοικίας
IsInInsulineTherapy	Λογική τιμή που περιγράφει αν ο ασθενής είναι σε θεραπείας ινσουλίνης.

DoctorPermissions	Άδειες που έχει δώσει ο ασθενής σε ιατρούς για να έχουν πρόσβαση στα δεδομένα που εισάγει στο σύστημα.
--------------------------	--

Πίνακας 6.3.5: DTO για το ιστορικό του ασθενούς

PatientByAmkaDto	
PatientAmka	ΑΜΚΑ του ασθενούς

Πίνακας 6.3.6: DTO για την αναζήτηση του ιστορικού του ασθενούς με βάση το ΑΜΚΑ

AddDoctorPatientDto	
PatientAmka	ΑΜΚΑ του ασθενούς

Πίνακας 6.3.7: DTO για τη διασύνδεση του ιστορικού του ασθενούς με τον γιατρό

AddPatientDto	
AMKA	ΑΜΚΑ του ασθενούς
Biological_Gender	Αριθμός που δείχνει το φύλλο του ασθενούς. Παίρνει τις τιμές 1 για το θηλυκό και 2 για το αρσενικό.
BirthDate	Ημερομηνία Γέννησης του ασθενούς
FirstName	Μικρό Όνομα Ασθενή
LastName	Επώνυμο Ασθενή
Email	Διεύθυνση ηλεκτρονικού ταχυδρομείου
PhoneNumber	Αριθμός Τηλεφώνου
SmokingStatus	Αριθμός που δείχνει την κατάσταση του ασθενούς σε σχέση με το κάπνισμα. Η τιμή 1 αντιστοιχεί σε ασθενή που δεν ήταν ποτέ καπνιστής, η τιμή 2 σε πρώην καπνιστή και η τιμή 3 σε καπνιστή.
Parents_Have_Dm	Αριθμός που δείχνει το ιστορικό του ασθενούς όσον αφορά τον διαβήτη. Η τιμή 1 αντιστοιχεί στην περίπτωση που κανένας από τους δύο γονείς του είχαν διαβήτη, η τιμή 2 στην περίπτωση που η μητέρα του ασθενούς είχε διαβήτη, η

	τιμή 3 στην περίπτωση που ο πατέρας του ασθενούς είχε διαβήτη και η τιμή 4 στην περίπτωση που και οι δύο γονείς έχουν ιστορικό.
DmDiagnosisDate	Ημερομηνία Διάγνωσης
Address	Διεύθυνση Κατοικίας
IsInInsulineTherapy	Λογική τιμή που περιγράφει αν ο ασθενής είναι σε θεραπείας ινσουλίνης.

Πίνακας 6.3.8: DTO για την εγγραφή ιστορικού νέου ασθενή στο σύστημα

GetPatientCheckUpDto	
CheckupId	Id εγγραφής της εξέτασης στη βάση
CheckUpDate	Ημερομηνία εγγραφής της εξέτασης
PatientClinical	Λίστα με τα αποτελέσματα των κλινικών δεικτών της εξέτασης
PatientLaboratory	Λίστα με τα αποτελέσματα των εργαστηριακών δεικτών της εξέτασης
CheckupForms	Λίστα με τις φόρμες του ιατρού αυτής της εξέτασης

Πίνακας 6.3.9: DTO για την εξέταση ενός ασθενή

AddPatientCheckupDto	
CheckUpDate	Ημερομηνία εγγραφής της εξέτασης
Fasting_Glucose	Αριθμητική τιμή γλυκόζης νηστείας
LDL_Cholesterol	Αριθμητική τιμή χοληστερόλης LDL
Triglycerides	Αριθμητική τιμή τριγλυκεριδίων
HDL_Apolipoprotein_B	Αριθμητική τιμή HDL απολιποπρωτεΐνη B

HDL_Cholesterol	Αριθμητική τιμή χοληστερόλης HDL
Urea	Αριθμητική τιμή ουρίας
Microalbuminuria	Αριθμητική τιμή μικρολευκωματινουρία
Glycosylated_Hemoglobin	Αριθμητική τιμή γλυκοζυλιωμένης αιμοσφαιρίνης
Waist_Circuference	Αριθμητική τιμή περιφέρειας μέσης
Weight	Αριθμητική τιμή σωματικού βάρους
Height	Αριθμητική τιμή του ύψους
Hip_Circuference	Αριθμητική τιμή περιφέρειας
Systolic_Blood_Pressure	Αριθμητική τιμή συστολικής πίεσης
Diastolic_Blood_Pressure	Αριθμητική τιμή διαστολικής πίεσης
Hypertention	Υπέρταση

Πίνακας 6.3.10: DTO για την προσθήκη νέας εξέτασης ενός ασθενή

DateRequestDto	
RequestedDate	Ημερομηνία εγγραφής της επιθυμητής εξέτασης

Πίνακας 6.3.11: DTO για την αναζήτηση μίας εξέτασης με βάση την ημερομηνία

UpdatePatientCheckupDto	
CheckUpDate	Ημερομηνία εγγραφής της εξέτασης
Fasting_Glucose	Αριθμητική τιμή γλυκόζης νηστείας
LDL_Cholesterol	Αριθμητική τιμή χοληστερόλης LDL
Triglycerides	Αριθμητική τιμή τριγλυκεριδίων

HDL_Apolipoprotein_B	Αριθμητική τιμή HDL απολιποπρωτεΐνη Β
HDL_Cholesterol	Αριθμητική τιμή χοληστερόλης HDL
Urea	Αριθμητική τιμή ουρίας
Microalbuminuria	Αριθμητική τιμή μικρολευκωματινουρία
Glycosylated_Hemoglobin	Αριθμητική τιμή γλυκοζυλιωμένης αιμοσφαιρίνης
Waist_Circuference	Αριθμητική τιμή περιφέρειας μέσης
Weight	Αριθμητική τιμή σωματικού βάρους
Height	Αριθμητική τιμή του ύψους
Hip_Circuference	Αριθμητική τιμή περιφέρειας
Systolic_Blood_Pressure	Αριθμητική τιμή συστολικής πίεσης
Diastolic_Blood_Pressure	Αριθμητική τιμή διαστολικής πίεσης
Hypertention	Υπέρταση

Πίνακας 6.3.12: DTO για την επεξεργασία και τροποποίηση μίας εξέτασης

GetPatientLaboratoryDto	
Id	Id εγγραφής του εργαστηριακού δείκτη της εξέτασης στη βάση
CurrentValue	Αριθμητική τιμή του δείκτη
MeasDate	Ημερομηνία λήψης της μέτρησης
LIndicatorId	Id εγγραφής του πρότυπου εργαστηριακού δείκτη στη βάση
LabIndicator	Πρότυπος εργαστηριακός δείκτης

Limits	Χαρακτηρισμός της τιμής σε σχέση με τα όρια του πρότυπου δείκτη
---------------	---

Πίνακας 6.3.13: DTO για κάποιο εργαστηριακό δείκτη μίας συγκεκριμένης εξέτασης

GetPrototypeFormDto	
PrototypeFormId	Αριθμός εγγραφής στη βάση
Title	Τίτλος της φόρμας
Description	Περιγραφή της φόρμας
FormDate	Ημερομηνία εγγραφής της φόρμας στη βάση
Questions	Ερωτήσεις της πρότυπης φόρμας

Πίνακας 6.3.14: DTO πρότυπης φόρμας

AddPrototypeFormDto	
Title	Τίτλος της φόρμας
Description	Περιγραφή της φόρμας

Πίνακας 6.3.15: DTO προσθήκης νέας πρότυπης φόρμας

EditPrototypeFormDto	
FormId	Αριθμός εγγραφής στη βάση
Title	Τίτλος της φόρμας
Description	Περιγραφή της φόρμας

Πίνακας 6.3.16: DTO επεξεργασίας πρότυπης φόρμας

AddQuestionFormDto	
PrototypeFormId	Αριθμός εγγραφής πρότυπης φόρμας στη βάση
Question	Ερώτηση
QuestionDate	Ημερομηνία εγγραφής νέας ερώτησης

Πίνακας 6.3.17: DTO προσθήκης νέας ερώτησης σε πρότυπη φόρμα

EditQuestionDto και GetQuestionDto	
PrototypeFormId	Αριθμός εγγραφής πρότυπης φόρμας στη βάση
QuestionId	Id εγγραφής ερώτησης στη βάση
Question	Ερώτηση
QuestionDate	Ημερομηνία εγγραφής νέας ερώτησης

Πίνακας 6.3.18: DTO επεξεργασίας – τροποποίησης ερώτησης σε πρότυπη φόρμα

DeleteQuestionFormDto	
PrototypeFormId	Αριθμός εγγραφής πρότυπης φόρμας στη βάση
QuestionId	Id εγγραφής ερώτησης στη βάση

Πίνακας 6.3.19: DTO διαγραφής ερώτησης από πρότυπη φόρμα

GetCheckupFormDto	
CheckupFormId	Αριθμός εγγραφής στη βάση της φόρμας της εξέτασης
PrototypeFormId	Αριθμός εγγραφής πρότυπης φόρμας στη βάση

PrototypeForm	Πρότυπη φόρμα
Title	Τίτλος της φόρμας
Description	Περιγραφή της φόρμας
CheckupId	Ημερομηνία εγγραφής της εξέτασης στη βάση
FormAnswers	Απαντήσεις στις ερωτήσεις της πρότυπης φόρμας στην οποία βασίζεται η φόρμα της εξέτασης

Πίνακας 6.3.20: DTO φόρμας συγκεκριμένης εξέτασης

AddCheckupFormDto	
PrototypeFormId	Αριθμός εγγραφής πρότυπης φόρμας στη βάση
PatientCheckupCheckupId	Id εγγραφής της εξέτασης στη βάση

Πίνακας 6.3.21: DTO για την προσθήκη φόρμας σε συγκεκριμένη εξέταση

GetAnswerDto	
AnswerId	Αριθμός εγγραφής στη βάση της απάντησης
Answer	Απάντηση
AnswerDate	Ημερομηνία εγγραφής της απάντησης
CheckupFormId	Αριθμός εγγραφής στη βάση της φόρμας της εξέτασης
QuestionId	Αριθμός εγγραφής στη βάση της ερώτησης στην οποία απαντά
Question	Ερώτηση στην οποία απαντά

Πίνακας 6.3.22: DTO για την απάντηση σε ερώτηση φόρμας σε συγκεκριμένη εξέταση

AddAnswerQuestionDto	
CheckupId	Αριθμός εγγραφής στη βάση της εξέτασης
Answer	Απάντηση
AnswerDate	Ημερομηνία εγγραφής της απάντησης
CheckupFormId	Αριθμός εγγραφής στη βάση της φόρμας της εξέτασης
QuestionId	Αριθμός εγγραφής στη βάση της ερώτησης στην οποία απαντά
Question	Ερώτηση στην οποία απαντά

Πίνακας 6.3.23: DTO για την προσθήκη απάντησης σε ερώτηση πρότυπης φόρμας για συγκεκριμένη εξέταση

TreatmentDto	
TreatmentId	Αριθμός εγγραφής πρότυπης θεραπείας στη βάση
Acronym	Συντομογραφία θεραπείας
FullName	Πλήρες όνομα θεραπείας

Πίνακας 6.3.23: DTO για το πρότυπο θεραπείας

TreatmentDescriptionPresetDto	
PresetId	Αριθμός εγγραφής στη βάση της θεραπείας συγκεκριμένου ασθενή
Treatment	Πρότυπο θεραπείας (Πίνακας 6.3.23)
CodeName	Σύντομη ονομασία
Description	Περιγραφή

StartDate	Ημερομηνία έναρξης θεραπείας
EndDate	Ημερομηνία λήξης θεραπείας

Πίνακας 6.3.24: DTO για θεραπεία συγκεκριμένου ασθενή

DateRequestDto	
RequestedDate	Ημερομηνία έναρξης θεραπείας

Πίνακας 6.3.25: DTO για την αναζήτηση συγκεκριμένης θεραπείας με βάση την ημερομηνία έναρξης

UpdateTreatmentDescriptionPresetDto	
PresetId	Αριθμός εγγραφής στη βάση της θεραπείας συγκεκριμένου ασθενή
CodeName	Σύντομη ονομασία
Description	Περιγραφή
StartDate	Ημερομηνία έναρξης θεραπείας
EndDate	Ημερομηνία λήξης θεραπείας

Πίνακας 6.3.26: DTO για την επεξεργασία - τροποποίηση θεραπείας σε συγκεκριμένο ασθενή

AddTreatmentDescriptionPresetDto	
Acronym	Συντομογραφία πρότυπης θεραπείας
CodeName	Σύντομη ονομασία
Description	Περιγραφή
StartDate	Ημερομηνία έναρξης θεραπείας
EndDate	Ημερομηνία λήξης θεραπείας

Πίνακας 6.3.27: DTO για την επεξεργασία - τροποποίηση θεραπείας σε συγκεκριμένο ασθενή

AgeStatistics	
Minor	Ποσοστό των ασθενών κάτω των 18 ετών
Under25	Ποσοστό των ασθενών μεταξύ 18 και 25 ετών
Between26_44	Ποσοστό των ασθενών μεταξύ 26 και 44 ετών
Between45_65	Ποσοστό των ασθενών μεταξύ 45 και 65 ετών
Over65	Ποσοστό των ασθενών άνω των 65 ετών

Πίνακας 6.3.28: DTO για την παρουσίαση ποσοστών για την εμφάνιση της ασθένειας με βάση την ηλικία

GetGlucosePercentsDto	
PreMealPercent	Ποσοστό μετρήσεων προγευματικής γλυκόζης
AfterMealPercent	Ποσοστό μετρήσεων μεταγευματικής γλυκόζης
PreSleepPercent	Ποσοστό μετρήσεων γλυκόζης προ ύπνου

Πίνακας 6.3.29: DTO για την παρουσίαση ποσοστών της μέτρησης γλυκόζης συγκεκριμένου ασθενή από τα δεδομένα που εισάγει ο ίδιος στο σύστημα μέσω της εφαρμογής του κινητού του

6.3.1 AuthDoctorController

Ο controller αυτός εξυπηρετεί την εγγραφή και την πιστοποίηση του χρήστη – ιατρού με σκοπό την είσοδο του στην εφαρμογή ιστού.

Στον παρακάτω πίνακα φαίνονται τα endpoints που περιέχει:

AuthDoctorController (http://localhost: 5230/AuthDoctor)					
A/A	Endpoint	HTTP Μέθοδος	Περιγραφή	Δεδομένα Εισόδου endpoint	Δεδομένα στο εισόδο στο σώμα της μεθόδου
1	/Register	POST	Κάνει την εγγραφή νέου χρήστη – ιατρού στο σύστημα.	-	DoctorRegisterDto Πίνακας 6.3.1
2	/Login	POST	Πιστοποιεί τον χρήστη και του δίνει πρόσβαση στις λειτουργίες της εφαρμογής του ιστού.	-	DoctorLoginDto Πίνακας 6.3.2

Πίνακας 6.3.30: Endpoints του controller AuthDoctorController

Στους παρακάτω πίνακες φαίνονται αναλυτικά τα responses των endpoints του πίνακα 6.3.30:

AuthDoctorController: Endpoint 1 Responses				
A/A	Περίπτωση	Status Code	Δεδομένα που επιστρέφει	Μήνυμα
1	Ο χρήστης συμπληρώνει τα πεδία που του ζητήθηκαν με επιτυχία.	200: Success	Επιστρέφει τον κωδικό εγγραφής του χρήστη – ιατρού στη βάση	“OK”
2	Ο χρήστης συμπληρώνει τα πεδία που του ζητήθηκαν με επιτυχία αλλά τοποθετεί User Name που χρησιμοποιεί άλλος εγγεγραμμένος χρήστης.	400: Bad Request	-	Ειδοποιεί τον χρήστη ότι υπάρχει εγγραφή στην εφαρμογή με αυτό το User Name.

3	Ο χρήστης εισάγει Password με λιγότερους από 6 χαρακτήρες.	400: Bad Request	-	Ειδοποιεί τον χρήστη ότι πρέπει να δημιουργήσει Password με τουλάχιστον 6 χαρακτήρες.
4	Ο χρήστης εισάγει Confirm Password διαφορετικό από το Password.	Ο χρήστης εισάγει Password με λιγότερους από 6 χαρακτήρες.	-	Ειδοποιεί τον χρήστη ότι το Password και το Confirm Password δεν ταιριάζουν.
5	Η εφαρμογή δε φτάνει ποτέ στο endpoint	500: Internal Server Error	-	Δεν επιστρέφεται κάποιο μήνυμα αλλά σημαίνει ότι το API δεν είναι διαθέσιμο.

Πίνακας 6.3.31: Responses του endpoint 1 του controller AuthDoctorController για τον πίνακα 6.3.30

AuthDoctorController: Endpoint 2 Responses

A/A	Περίπτωση	Status Code	Δεδομένα που επιστρέφει	Μήνυμα
1	Ο χρήστης συμπληρώνει σωστά τα πεδία User Name και Password.	200: Success	Επιστρέφει ένα string που περιγράφει το JWT	“OK”
2	Ο χρήστης συμπληρώνει λάθος User Name	400: Bad Request	-	Ειδοποιεί τον χρήστη ότι δεν υπάρχει εγγραφή στην εφαρμογή με αυτό το User Name.

3	Ο χρήστης συμπληρώνει λάθος Password	400: Bad Request	-	Ειδοποιεί τον χρήστη ότι έχει συμπληρώσει λάθος Password.
4	Η εφαρμογή δε φτάνει ποτέ στο endpoint	500: Internal Server Error	-	Δεν επιστρέφεται κάποιο μήνυμα αλλά σημαίνει ότι το API δεν είναι διαθέσιμο.

Πίνακας 6.3.32: Responses του endpoint 1 του controller AuthDoctorController για τον πίνακα 6.3.30

6.3.2 DoctorProfileController

Ο controller αυτός δίνει τη δυνατότητα στον γιατρό να δει και να διαχειριστεί τον λογαριασμό του.

Στον παρακάτω πίνακα φαίνονται τα endpoints που περιέχει:

DoctorProfileController (http://localhost: 5230/DoctorProfile)						
A/A	Endpoint	HTTP Μέθοδος	Περιγραφή	Δεδομένα Εισόδου endpoint	στο	Δεδομένα εισόδου στο σώμα της μεθόδου
1	/doctorProfile	GET	Επιστρέφει τα στοιχεία του λογαριασμού του ιατρού καθώς και μία λύση με τους ασθενείς του.	-		-
2	/	PUT	Δίνει τη δυνατότητα στον ασθενή να τροποποιήσει τα προσωπικά του στοιχεία.	-		UpdateDoctorDto Πίνακας 6.3.4

Πίνακας 6.3.33: Endpoints του controller DoctorProfileController

Στους παρακάτω πίνακες φαίνονται αναλυτικά τα responses των endpoints του πίνακα 6.3.33:

DoctorProfileController: Endpoint 1 Responses				
A/A	Περίπτωση	Status Code	Δεδομένα που επιστρέφει	Μήνυμα
1	Ο χρήστης έχει πιστοποιηθεί από το API με JWT.	200: Success	Λίστα του τύπου GetDoctorDto Πίνακας 6.3.3	-
2	Ο χρήστης δεν έχει πιστοποιηθεί από το API με JWT	401: Unauthorized	-	Δεν επιστρέφεται κάποιο μήνυμα αλλά σημαίνει ότι ο χρήστης δεν έχει πρόσβαση στο endpoint διότι δεν έχει πιστοποιηθεί από το API.
3	Η εφαρμογή δε φτάνει ποτέ στο endpoint	500: Internal Server Error	-	Δεν επιστρέφεται κάποιο μήνυμα αλλά σημαίνει ότι το API δεν είναι διαθέσιμο.

Πίνακας 6.3.34: Responses του endpoint 1 του controller DoctorProfileController για τον πίνακα 6.3.33

DoctorProfileController: Endpoint 2 Responses				
A/A	Περίπτωση	Status Code	Δεδομένα που επιστρέφει	Μήνυμα
1	Ο χρήστης συμπληρώνει σωστά τα πεδία που του	200: Success	GetDoctorDto Πίνακας 6.3.3	“OK”

	ζητούνται από την εφαρμογή.			
2	Ο χρήστης συμπληρώνει λάθος το email που είναι διασυνδεδεμένο με τον λογαριασμό του.	200: Success	-	Ειδοποιεί τον χρήστη ότι δεν υπάρχει εγγραφή στην εφαρμογή με αυτό το email.
3	Ο χρήστης δεν έχει πιστοποιηθεί από το API με JWT	401: Unauthorized	-	Δεν επιστρέφεται κάποιο μήνυμα αλλά σημαίνει ότι ο χρήστης δεν έχει πρόσβαση στο endpoint διότι δεν έχει πιστοποιηθεί από το API.
4	Η εφαρμογή δε φτάνει ποτέ στο endpoint	500: Internal Server Error	-	Δεν επιστρέφεται κάποιο μήνυμα αλλά σημαίνει ότι το API δεν είναι διαθέσιμο.

Πίνακας 6.3.35: Responses του endpoint 1 του controller DoctorProfileController για τον πίνακα 6.3.33

6.3.3 PatientController

Ο controller αυτός δίνει τη δυνατότητα στον γιατρό να δει και να διαχειριστεί το ιστορικό των ασθενών του με τους οποίους είναι διασυνδεδεμένος. Συγκεκριμένα, μετά την επιτυχημένη είσοδο του στην εφαρμογή μπορεί να λάβει μία λίστα με το ιστορικό των ασθενών με τους οποίους έχει διασυνδεθεί. Μπορεί επίσης να προσθέσει ένα νέο ιστορικό ασθενή και η διασύνδεση του με αυτόν γίνεται αυτόματα. Επιπλέον μπορεί να αναζητήσει μία εγγραφή ιστορικού με βάση το ΑΜΚΑ και αν δεν υπάρχει στη λίστα του να πραγματοποιήσει διασύνδεση. Τέλος του δίνεται η δυνατότητα να διαγράψει το ιστορικό του ασθενούς.

Στον παρακάτω πίνακα φαίνονται τα endpoints που περιέχει:

PatientController (http://localhost: 5230/Patient)

A/A	Endpoint	HTTP Μέθοδος	Περιγραφή	Δεδομένα Εισόδου στο endpoint	Δεδομένα εισόδου στο σώμα της μεθόδου
1	/GetAll	GET	Επιστρέφει μία λίστα με το ιστορικό όλων των ασθενών με τους οποίους είναι διασυνδεδεμένος.	-	-
2	/patientExists	POST	Μέσω αυτού του endpoint ο χρήστης εξετάζει αν υπάρχει στο σύστημα ιστορικό για συγκεκριμένο ασθενή.	-	PatientByAmkaDto Πίνακας 6.3.6
3	/relationships	POST	Μέσω αυτού του endpoint ο χρήστης διασυνδέεται με το ιστορικό συγκεκριμένου ασθενή.	-	AddDoctorPatientDto Πίνακας 6.3.7
4	/addPatient	POST	Μέσω αυτού του endpoint ο χρήστης προσθέτει νέο ιστορικό για συγκεκριμένο ασθενή.	-	AddPatientDto Πίνακας 6.3.8
5	/delete	DELETE	Μέσω αυτού του endpoint ο χρήστης διαγράφει το ιστορικό συγκεκριμένου ασθενή.	-	PatientByAmkaDto Πίνακας 6.3.6

Πίνακας 6.3.36: Endpoints του controller DoctorProfileController

Στους παρακάτω πίνακες φαίνονται αναλυτικά τα responses των endpoints του πίνακα 6.3.36:

PatientController: Endpoint 1 Responses				
A/A	Περίπτωση	Status Code	Δεδομένα που επιστρέφει	Μήνυμα
1	Ο χρήστης έχει πιστοποιηθεί από το API με JWT.	200: Success	Λίστα του τύπου GetPatientDto Πίνακας 6.3.5	-
2	Ο χρήστης δεν έχει πιστοποιηθεί από το API με JWT	401: Unauthorized	-	Δεν επιστρέφεται κάποιο μήνυμα αλλά σημαίνει ότι ο χρήστης δεν έχει πρόσβαση στο endpoint διότι δεν έχει πιστοποιηθεί από το API.
3	Η εφαρμογή δε φτάνει ποτέ στο endpoint	500: Internal Server Error	-	Δεν επιστρέφεται κάποιο μήνυμα αλλά σημαίνει ότι το API δεν είναι διαθέσιμο.

Πίνακας 6.3.37: Responses του endpoint 1 του controller PatientController για τον πίνακα 6.3.36

PatientController: Endpoint 2 Responses				
A/A	Περίπτωση	Status Code	Δεδομένα που επιστρέφει	Μήνυμα
1	Ο χρήστης συμπληρώνει ΑΜΚΑ ιστορικού	200: Success	GetPatientDto Πίνακας 6.3.5	Ενημερώνει τον χρήστη ότι είναι διασυνδεδεμένος με αυτό το

	ασθενή που είναι διασυνδεδεμένο με τον λογαριασμό του.			ιστορικό ασθενή.
2	Ο χρήστης συμπληρώνει ΑΜΚΑ ιστορικού ασθενή που δεν υπάρχει εγγεγραμμένο στο σύστημα.	200: Success	-	Ειδοποιεί τον χρήστη ότι δεν υπάρχει ιστορικό ασθενή στο σύστημα με αυτό το ΑΜΚΑ και θα πρέπει να το προσθέσει εκείνος.
3	Ο χρήστης συμπληρώνει ΑΜΚΑ ιστορικού ασθενή που υπάρχει εγγεγραμμένο στο σύστημα αλλά δεν είναι διασυνδεδεμένο με τον λογαριασμό του.	200: Success	-	Ειδοποιεί τον χρήστη ότι υπάρχει ιστορικό ασθενή στο σύστημα με αυτό το ΑΜΚΑ αλλά θα πρέπει να διασυνδεθεί με αυτό.
4	Ο χρήστης δεν έχει πιστοποιηθεί από το API με JWT	401: Unauthorized	-	Δεν επιστρέφεται κάποιο μήνυμα αλλά σημαίνει ότι ο χρήστης δεν έχει πρόσβαση στο endpoint διότι δεν έχει πιστοποιηθεί από το API.
5	Η εφαρμογή δε φτάνει ποτέ στο endpoint	500: Internal Server Error	-	Δεν επιστρέφεται κάποιο μήνυμα αλλά σημαίνει

				ότι το API δεν είναι διαθέσιμο.
--	--	--	--	---------------------------------

Πίνακας 6.3.38: Responses του endpoint 1 του controller PatientController για τον πίνακα 6.3.36

PatientController: Endpoint 3 Responses				
A/A	Περίπτωση	Status Code	Δεδομένα που επιστρέφει	Μήνυμα
1	Ο χρήστης συμπληρώνει ΑΜΚΑ ιστορικού ασθενή που υπάρχει εγγεγραμμένο στο σύστημα αλλά δεν είναι διασυνδεδεμένο με τον λογαριασμό του.	200: Success	GetPatientDto Πίνακας 6.3.5	Ενημερώνει τον χρήστη ότι είναι διασυνδεδεμένος με αυτό το ιστορικό ασθενή.
2	Ο χρήστης συμπληρώνει ΑΜΚΑ ιστορικού ασθενή που δεν υπάρχει εγγεγραμμένο στο σύστημα.	200: Success	-	Ειδοποιεί τον χρήστη ότι δεν υπάρχει ιστορικό ασθενή στο σύστημα με αυτό το ΑΜΚΑ.
3	Ο χρήστης συμπληρώνει ΑΜΚΑ ιστορικού ασθενή που υπάρχει εγγεγραμμένο στο σύστημα και είναι διασυνδεδεμένο με τον λογαριασμό του.	200: Success	-	Ειδοποιεί τον χρήστη ότι είναι ήδη διασυνδεδεμένος με αυτό το ιστορικό.

4	Ο χρήστης δεν έχει πιστοποιηθεί από το API με JWT	401: Unauthorized	-	Δεν επιστρέφεται κάποιο μήνυμα αλλά σημαίνει ότι ο χρήστης δεν έχει πρόσβαση στο endpoint διότι δεν έχει πιστοποιηθεί από το API.
5	Η εφαρμογή δε φτάνει ποτέ στο endpoint	500: Internal Server Error	-	Δεν επιστρέφεται κάποιο μήνυμα αλλά σημαίνει ότι το API δεν είναι διαθέσιμο.

Πίνακας 6.3.39: Responses του endpoint 1 του controller PatientController για τον πίνακα 6.3.36

PatientController: Endpoint 4 Responses

A/A	Περίπτωση	Status Code	Δεδομένα που επιστρέφει	Μήνυμα
1	Ο χρήστης έχει πιστοποιηθεί από το API με JWT.	200: Success	Λίστα του τύπου GetPatientDto Πίνακας 6.3.5	-
3	Ο χρήστης δεν έχει πιστοποιηθεί από το API με JWT	401: Unauthorized	-	Δεν επιστρέφεται κάποιο μήνυμα αλλά σημαίνει ότι ο χρήστης δεν έχει πρόσβαση στο endpoint διότι δεν έχει πιστοποιηθεί από το API.
4	Η εφαρμογή δε φτάνει ποτέ στο endpoint	500: Internal Server Error	-	Δεν επιστρέφεται κάποιο μήνυμα αλλά σημαίνει

				ότι το API δεν είναι διαθέσιμο.
--	--	--	--	---------------------------------

Πίνακας 6.3.40: Responses του endpoint 1 του controller PatientController για τον πίνακα 6.3.36

PatientController: Endpoint 5 Responses				
A/A	Περίπτωση	Status Code	Δεδομένα επιστρέφει	Μήνυμα
1	Ο χρήστης συμπληρώνει ΑΜΚΑ ιστορικού ασθενή που υπάρχει εγγεγραμμένο στο σύστημα και είναι διασυνδεδεμένο με τον λογαριασμό του.	200: Success	-	Ενημερώνει τον χρήστη ότι ετο ιστορικό αφαιρέθηκε με επιτυχία από το σύστημα.
2	Ο χρήστης συμπληρώνει ΑΜΚΑ ιστορικού ασθενή που δεν υπάρχει εγγεγραμμένο στο σύστημα.	200: Success	-	Ειδοποιεί τον χρήστη ότι δεν υπάρχει ιστορικό ασθενή στο σύστημα με αυτό το ΑΜΚΑ.
3	Ο χρήστης συμπληρώνει ΑΜΚΑ ιστορικού ασθενή που υπάρχει εγγεγραμμένο στο σύστημα και δεν είναι διασυνδεδεμένο με τον λογαριασμό του.	200: Success	-	Ειδοποιεί τον χρήστη ότι δεν είναι ήδη διασυνδεδεμένος με αυτό το ιστορικό και δεν έχει το διακίωμα της διαγραφής του.

4	Ο χρήστης δεν έχει πιστοποιηθεί από το API με JWT	401: Unauthorized	-	Δεν επιστρέφεται κάποιο μήνυμα αλλά σημαίνει ότι ο χρήστης δεν έχει πρόσβαση στο endpoint διότι δεν έχει πιστοποιηθεί από το API.
5	Η εφαρμογή δε φτάνει ποτέ στο endpoint	500: Internal Server Error	-	Δεν επιστρέφεται κάποιο μήνυμα αλλά σημαίνει ότι το API δεν είναι διαθέσιμο.

Πίνακας 6.3.41: Responses του endpoint 1 του controller PatientController για τον πίνακα 6.3.36

6.3.4 PatientCheckupController

Ο controller αυτός δίνει τη δυνατότητα στον ιατρό να υποβάλλει αιτήματα σχετικά με τις εξετάσεις συγκεκριμένου ασθενούς. Μέσω αυτών των endpoints λαμβάνει ιστορικό για τις εξετάσεις συγκεκριμένου ασθενούς κάνοντας αναζήτηση με βάση το ΑΜΚΑ του. Με τον ίδιο τρόπο αναζήτησης λαμβάνει την τελευταία εξέταση για τον ασθενή του που έχει αποθηκεύσει στο σύστημα. Επίσης μπορεί να λάβει συγκεκριμένη εξέταση, υποβάλλοντας την ημερομηνία διεξαγωγής της και το ΑΜΚΑ του ασθενούς. Τέλος μπορεί να προσθέσει μία νέα εξέταση και να επεξεργαστεί κάποια υπάρχουσα.

Με βάση τα δεδομένα που αποθηκεύει ο ιατρός στο σύστημα μέσω των εξετάσεων για κάθε ασθενή, μπορεί να λάβει μία λίστα με τις μετρήσεις της LDL χοληστερόλης, της HDL χοληστερόλης, της γλυκοζυλιωμένης αιμοσφαιρίνης και της γλυκόζης νηστείας

Στον παρακάτω πίνακα φαίνονται τα endpoints που περιέχει:

PatientCheckupController (http://localhost:5230/PatientCheckup)					
A/A	Endpoint	HTTP Μέθοδος	Περιγραφή	Δεδομένα Εισόδου στο endpoint	Δεδομένα εισόδου στο σώμα της μεθόδου
1	/	POST	Επιστρέφει μία λίστα με όλες τις εξετάσεις ενός	-	PatientByAmkaDto

			συγκεκριμένου ασθενή κάνοντας αναζήτηση με το ΑΜΚΑ του.		Πίνακας 6.3.6
2	/GetLastPatientsCheck	POST	Επιστρέφει την τελευταία εξέταση ενός συγκεκριμένου ασθενή κάνοντας αναζήτηση με το ΑΜΚΑ του.	-	PatientByAmkaDto Πίνακας 6.3.6
3	/checkupdate/{amka}	POST	Μέσω αυτού του endpoint ο χρήστης εξετάζει αν υπάρχει στο σύστημα ιστορικό για συγκεκριμένο ασθενή.	Το ΑΜΚΑ του ασθενούς	DateRequestDto Πίνακας 6.3.11
4	/updateCheckup/{amka}	PUT	Μέσω αυτού του endpoint ο χρήστης μπορεί να επεξεργαστεί μία συγκεκριμένη εξέταση	Το ΑΜΚΑ του ασθενούς	UpdatePatientCheckupDto Πίνακας 6.3.12
5	/newCheckup/{amka}	POST	Μέσω αυτού του endpoint ο χρήστης προσθέτει νέα εξέταση για συγκεκριμένο ασθενή.	-	AddPatientCheckupDto Πίνακας 6.3.16
6	/ldlCholesterol	POST	Μέσω αυτού του endpoint ο χρήστης παίρνει μία λίστα με όλες τις τιμές για την LDL χοληστερόλη, που έχουν καταγραφεί σε όλες τις εξετάσεις του ασθενούς.	-	PatientByAmkaDto Πίνακας 6.3.6

7	/hdlCholesterol	POST	Μέσω αυτού του endpoint ο χρήστης παίρνει μία λίστα με όλες τις τιμές για την HDL χοληστερόλη, που έχουν καταγραφεί σε όλες τις εξετάσεις του ασθενούς.	-	PatientByAmkaDto Πίνακας 6.3.6
8	/Glycosylated	POST	Μέσω αυτού του endpoint ο χρήστης παίρνει μία λίστα με όλες τις τιμές για την γλυκοζυλιωμένη αιμοσφαιρίνη, που έχουν καταγραφεί σε όλες τις εξετάσεις του ασθενούς.	-	PatientByAmkaDto Πίνακας 6.3.6
9	/fastingGlucose	POST	Μέσω αυτού του endpoint ο χρήστης παίρνει μία λίστα με όλες τις τιμές για την γλυκόζη νηστείας, που έχουν καταγραφεί σε όλες τις εξετάσεις του ασθενούς.	-	PatientByAmkaDto Πίνακας 6.3.6

Πίνακας 6.3.42: Endpoints του controller PatientCheckupController

Στους παρακάτω πίνακες φαίνονται αναλυτικά τα responses των endpoints του πίνακα 6.3.42:

PatientCheckupController: Endpoint 1 Responses

A/A	Περίπτωση	Status Code	Δεδομένα που επιστρέφει	Μήνυμα
1	Ο χρήστης συμπληρώνει ΑΜΚΑ με το οποίο είναι διασυνδεδεμένος και ο ασθενής έχει checkup.	200: Success	Λίστα του τύπου GetPatientCheckUpDto Πίνακας 6.3.9	-
2	Ο χρήστης συμπληρώνει ΑΜΚΑ το οποίο δεν υπάρχει στο σύστημα	200: Success	-	Ο χρήστης ειδοποιείται ότι δεν υπάρχει το ιστορικό του ασθενούς για το οποίο αναζητά να λάβει εξετάσεις.
3	Ο χρήστης συμπληρώνει ΑΜΚΑ με το οποίο δεν έχει διασυνδεθεί.	200: Success	-	Ο χρήστης ειδοποιείται ότι δεν είναι διασυνδεδεμένος με το ιστορικό του ασθενούς για το οποίο αναζητά να λάβει εξετάσεις.
4	Ο ασθενής δεν έχει εξετάσεις.	200: Success	-	Ο χρήστης ειδοποιείται ότι δεν υπάρχουν εξετάσεις για αυτό τον ασθενή.
5	Ο χρήστης δεν έχει πιστοποιηθεί από το API με JWT	401: Unauthorized	-	Δεν επιστρέφεται κάποιο μήνυμα αλλά σημαίνει ότι ο χρήστης δεν έχει πρόσβαση στο endpoint διότι

				δεν έχει πιστοποιηθεί από το API.
6	Η εφαρμογή δε φτάνει ποτέ στο endpoint	500: Internal Server Error	-	Δεν επιστρέφεται κάποιο μήνυμα αλλά σημαίνει ότι το API δεν είναι διαθέσιμο.

Πίνακας 6.3.43: Responses του endpoint 1 του controller PatientCheckupController για τον πίνακα 6.3.42

PatientCheckupController: Endpoint 2 Responses				
A/A	Περίπτωση	Status Code	Δεδομένα που επιστρέφει	Μήνυμα
1	Ο χρήστης συμπληρώνει ΑΜΚΑ με το οποίο είναι διασυνδεδεμένος και ο ασθενής έχει checkup.	200: Success	Τελευταία εξέταση του τύπου GetPatientCheckUpDto Πίνακας 6.3.9	-
2	Ο χρήστης συμπληρώνει ΑΜΚΑ το οποίο δεν υπάρχει στο σύστημα	200: Success	-	Ο χρήστης ειδοποιείται ότι δεν υπάρχει το ιστορικό του ασθενούς για το οποίο αναζητά να λάβει εξετάσεις.
3	Ο χρήστης συμπληρώνει ΑΜΚΑ με το οποίο δεν έχει διασυνδεθεί.	200: Success	-	Ο χρήστης ειδοποιείται ότι δεν είναι διασυνδεδεμένος με το ιστορικό του ασθενούς για το οποίο αναζητά να λάβει εξετάσεις.

4	Ο ασθενής δεν έχει εξετάσεις.	200: Success	-	Ο χρήστης ειδοποιείται ότι δεν υπάρχουν εξετάσεις για αυτό τον ασθενή.
5	Ο χρήστης δεν έχει πιστοποιηθεί από το API με JWT	401: Unauthorized	-	Δεν επιστρέφεται κάποιο μήνυμα αλλά σημαίνει ότι ο χρήστης δεν έχει πρόσβαση στο endpoint διότι δεν έχει πιστοποιηθεί από το API.
6	Η εφαρμογή δε φτάνει ποτέ στο endpoint	500: Internal Server Error	-	Δεν επιστρέφεται κάποιο μήνυμα αλλά σημαίνει ότι το API δεν είναι διαθέσιμο.

Πίνακας 6.3.44: Responses του endpoint 1 του controller PatientCheckupController για τον πίνακα 6.3.42

PatientCheckupController: Endpoint 3 Responses				
A/A	Περίπτωση	Status Code	Λεδομένα που επιστρέφει	Μήνυμα
1	Ο χρήστης συμπληρώνει ΑΜΚΑ με το οποίο είναι διασυνδεδεμένος, ο ασθενής έχει checkup και υπάρχει checkup για αυτή την ημερομηνία.	200: Success	Εξέταση του τύπου GetPatientCheckUpDto Πίνακας 6.3.9	-
2	Ο χρήστης συμπληρώνει ΑΜΚΑ το οποίο	200: Success	-	Ο χρήστης ειδοποιείται ότι δεν υπάρχει το

	δεν υπάρχει στο σύστημα			ιστορικό του ασθενούς για το οποίο αναζητά να λάβει εξετάσεις.
3	Ο χρήστης συμπληρώνει ΑΜΚΑ με το οποίο δεν έχει διασυνδεθεί.	200: Success	-	Ο χρήστης ειδοποιείται ότι δεν είναι διασυνδεδεμένος με το ιστορικό του ασθενούς για το οποίο αναζητά να λάβει εξετάσεις.
4	Ο ασθενής δεν έχει εξετάσεις.	200: Success	-	Ο χρήστης ειδοποιείται ότι δεν υπάρχουν εξετάσεις για αυτό τον ασθενή.
5	Δεν υπάρχει εξέταση για αυτή την ημερομηνία	200: Success	-	Ο χρήστης ειδοποιείται ότι δεν υπάρχουν εξετάσεις για αυτή την ημερομηνία.
6	Ο χρήστης δεν έχει πιστοποιηθεί από το API με JWT	401: Unauthorized	-	Δεν επιστρέφεται κάποιο μήνυμα αλλά σημαίνει ότι ο χρήστης δεν έχει πρόσβαση στο endpoint διότι δεν έχει πιστοποιηθεί από το API.
7	Η εφαρμογή δε φτάνει ποτέ στο endpoint	500: Internal Server Error	-	Δεν επιστρέφεται κάποιο μήνυμα αλλά σημαίνει

				ότι το API δεν είναι διαθέσιμο.
--	--	--	--	---------------------------------

Πίνακας 6.3.45: Responses του endpoint 1 του controller PatientCheckupController για τον πίνακα 6.3.42

PatientCheckupController: Endpoint 4 Responses				
A/A	Περίπτωση	Status Code	Δεδομένα που επιστρέφει	Μήνυμα
1	Ο χρήστης συμπληρώνει σωστά το ΑΜΚΑ και την ημερομηνία της εξέτασης που θέλει να τροποποιήσει.	200: Success	Εξέταση του τύπου GetPatientCheckUpDto (Πίνακας 6.3.9) τροποποιημένη με βάση τα στοιχεία που άλλαξε ο χρήστης.	-
2	Ο χρήστης συμπληρώνει ΑΜΚΑ το οποίο δεν υπάρχει στο σύστημα	200: Success	-	Ο χρήστης ειδοποιείται ότι δεν υπάρχει το ιστορικό του ασθενούς για το οποίο αναζητά να λάβει εξετάσεις.
3	Ο χρήστης συμπληρώνει ΑΜΚΑ με το οποίο δεν έχει διασυνδεθεί.	200: Success	-	Ο χρήστης ειδοποιείται ότι δεν είναι διασυνδεδεμένος με το ιστορικό του ασθενούς για το οποίο αναζητά να λάβει εξετάσεις.
4	Ο ασθενής δεν έχει εξετάσεις.	200: Success	-	Ο χρήστης ειδοποιείται ότι δεν υπάρχουν εξετάσεις για αυτό τον ασθενή.
5	Δεν υπάρχει εξέταση για	200: Success	-	Ο χρήστης ειδοποιείται ότι

	αυτή την ημερομηνία			δεν υπάρχουν εξετάσεις για αυτή την ημερομηνία.
6	Ο χρήστης δεν έχει πιστοποιηθεί από το API με JWT	401: Unauthorized	-	Δεν επιστρέφεται κάποιο μήνυμα αλλά σημαίνει ότι ο χρήστης δεν έχει πρόσβαση στο endpoint διότι δεν έχει πιστοποιηθεί από το API.
7	Η εφαρμογή δε φτάνει ποτέ στο endpoint	500: Internal Server Error	-	Δεν επιστρέφεται κάποιο μήνυμα αλλά σημαίνει ότι το API δεν είναι διαθέσιμο.

Πίνακας 6.3.46: Responses του endpoint 4 του controller PatientCheckupController για τον πίνακα 6.3.42

PatientCheckupController: Endpoint 5 Responses				
A/A	Περίπτωση	Status Code	Δεδομένα που επιστρέφει	Μήνυμα
1	Ο χρήστης συμπληρώνει σωστά το ΑΜΚΑ του ασθενούς για τον οποίο θέλει να προσθέσει νέα εξέταση.	200: Success	Εξέταση του τύπου GetPatientCheckUpDto (Πίνακας 6.3.9).	-
2	Ο χρήστης συμπληρώνει ΑΜΚΑ το οποίο δεν	200: Success	-	Ο χρήστης ειδοποιείται ότι δεν υπάρχει το ιστορικό του ασθενούς για το οποίο αναζητά

	υπάρχει στο σύστημα			να λάβει εξετάσεις.
3	Ο χρήστης συμπληρώνει ΑΜΚΑ με το οποίο δεν έχει διασυνδεθεί.	200: Success	-	Ο χρήστης ειδοποιείται ότι δεν είναι διασυνδεδεμένος με το ιστορικό του ασθενούς για το οποίο αναζητά να λάβει εξετάσεις.
4	Ο χρήστης δεν έχει πιστοποιηθεί από το API με JWT	401: Unauthorized	-	Δεν επιστρέφεται κάποιο μήνυμα αλλά σημαίνει ότι ο χρήστης δεν έχει πρόσβαση στο endpoint διότι δεν έχει πιστοποιηθεί από το API.
5	Η εφαρμογή δε φτάνει ποτέ στο endpoint	500: Internal Server Error	-	Δεν επιστρέφεται κάποιο μήνυμα αλλά σημαίνει ότι το API δεν είναι διαθέσιμο.

Πίνακας 6.3.47: Responses του endpoint 5 του controller PatientCheckupControllery για τον πίνακα 6.3.42

PatientCheckupController: Endpoint 6 Responses

A/A	Περίπτωση	Status Code	Δεδομένα που επιστρέφει	Μήνυμα
1	Ο χρήστης συμπληρώνει σωστά το ΑΜΚΑ του ασθενούς για τον οποίο θέλει να	200: Success	Λίστα του τύπου GetPatientLaboratoryDto (Πίνακας 6.3.13).	-

	προσθέσει νέα εξέταση.			
2	Ο χρήστης συμπληρώνει ΑΜΚΑ το οποίο δεν υπάρχει στο σύστημα	200: Success	-	Ο χρήστης ειδοποιείται ότι δεν υπάρχει το ιστορικό του ασθενούς για το οποίο αναζητά να λάβει εξετάσεις.
3	Ο χρήστης συμπληρώνει ΑΜΚΑ με το οποίο δεν έχει διασυνδεθεί.	200: Success	-	Ο χρήστης ειδοποιείται ότι δεν είναι διασυνδεδεμένος με το ιστορικό του ασθενούς για το οποίο αναζητά να λάβει εξετάσεις.
4	Ο χρήστης δεν έχει πιστοποιηθεί από το API με JWT	401: Unauthorized	-	Δεν επιστρέφεται κάποιο μήνυμα αλλά σημαίνει ότι ο χρήστης δεν έχει πρόσβαση στο endpoint διότι δεν έχει πιστοποιηθεί από το API.
5	Η εφαρμογή δε φτάνει ποτέ στο endpoint	500: Internal Server Error	-	Δεν επιστρέφεται κάποιο μήνυμα αλλά σημαίνει ότι το API δεν είναι διαθέσιμο.

Πίνακας 6.3.48: Responses του endpoint 6 του controller PatientCheckupControlleryγια τον πίνακα 6.3.42

PatientCheckupController: Endpoint 7 Responses

A/A	Περίπτωση	Status Code	Δεδομένα που επιστρέφει	Μήνυμα
1	Ο χρήστης συμπληρώνει σωστά το ΑΜΚΑ του ασθενούς για τον οποίο θέλει να προσθέσει νέα εξέταση.	200: Success	Λίστα του τύπου GetPatientLaboratoryDto (Πίνακας 6.3.13).	-
2	Ο χρήστης συμπληρώνει ΑΜΚΑ το οποίο δεν υπάρχει στο σύστημα	200: Success	-	Ο χρήστης ειδοποιείται ότι δεν υπάρχει το ιστορικό του ασθενούς για το οποίο αναζητά να λάβει εξετάσεις.
3	Ο χρήστης συμπληρώνει ΑΜΚΑ με το οποίο δεν έχει διασυνδεθεί.	200: Success	-	Ο χρήστης ειδοποιείται ότι δεν είναι διασυνδεδεμένος με το ιστορικό του ασθενούς για το οποίο αναζητά να λάβει εξετάσεις.
4	Ο χρήστης δεν έχει πιστοποιηθεί από το API με JWT	401: Unauthorized	-	Δεν επιστρέφεται κάποιο μήνυμα αλλά σημαίνει ότι ο χρήστης δεν έχει πρόσβαση στο endpoint διότι δεν έχει πιστοποιηθεί από το API.

5	Η εφαρμογή δε φτάνει ποτέ στο endpoint	500: Internal Server Error	-	Δεν επιστρέφεται κάποιο μήνυμα αλλά σημαίνει ότι το API δεν είναι διαθέσιμο.
---	--	----------------------------	---	--

Πίνακας 6.3.49: Responses του endpoint 7 του controller PatientCheckupControllery για τον πίνακα 6.3.42

PatientCheckupController: Endpoint 8 Responses				
A/A	Περίπτωση	Status Code	Δεδομένα που επιστρέφει	Μήνυμα
1	Ο χρήστης συμπληρώνει σωστά το ΑΜΚΑ του ασθενούς για τον οποίο θέλει να προσθέσει νέα εξέταση.	200: Success	Λίστα του τύπου GetPatientLaboratoryDto (Πίνακας 6.3.13).	-
2	Ο χρήστης συμπληρώνει ΑΜΚΑ το οποίο δεν υπάρχει στο σύστημα	200: Success	-	Ο χρήστης ειδοποιείται ότι δεν υπάρχει το ιστορικό του ασθενούς για το οποίο αναζητά να λάβει εξετάσεις.
3	Ο χρήστης συμπληρώνει ΑΜΚΑ με το οποίο δεν έχει διασυνδεθεί.	200: Success	-	Ο χρήστης ειδοποιείται ότι δεν είναι διασυνδεδεμένος με το ιστορικό του ασθενούς για το οποίο αναζητά να λάβει εξετάσεις.
4	Ο χρήστης δεν έχει πιστοποιηθεί	401: Unauthorized	-	Δεν επιστρέφεται κάποιο μήνυμα αλλά σημαίνει

	από το API με JWT			ότι ο χρήστης δεν έχει πρόσβαση στο endpoint διότι δεν έχει πιστοποιηθεί από το API.
5	Η εφαρμογή δε φτάνει ποτέ στο endpoint	500: Internal Server Error	-	Δεν επιστρέφεται κάποιο μήνυμα αλλά σημαίνει ότι το API δεν είναι διαθέσιμο.

Πίνακας 6.3.50: Responses του endpoint 8 του controller PatientCheckupController για τον πίνακα 6.3.42

PatientCheckupController: Endpoint 9 Responses				
A/A	Περίπτωση	Status Code	Δεδομένα που επιστρέφει	Μήνυμα
1	Ο χρήστης συμπληρώνει σωστά το ΑΜΚΑ του ασθενούς για τον οποίο θέλει να προσθέσει νέα εξέταση.	200: Success	Λίστα του τύπου GetPatientLaboratoryDto (Πίνακας 6.3.13).	-
2	Ο χρήστης συμπληρώνει ΑΜΚΑ το οποίο δεν υπάρχει στο σύστημα	200: Success	-	Ο χρήστης ειδοποιείται ότι δεν υπάρχει το ιστορικό του ασθενούς για το οποίο αναζητά να λάβει εξετάσεις.
3	Ο χρήστης συμπληρώνει ΑΜΚΑ με το οποίο δεν έχει διασυνδεθεί.	200: Success	-	Ο χρήστης ειδοποιείται ότι δεν είναι διασυνδεδεμένος με το ιστορικό του ασθενούς για

				το οποίο αναζητά να λάβει εξετάσεις.
4	Ο χρήστης δεν έχει πιστοποιηθεί από το API με JWT	401: Unauthorized	-	Δεν επιστρέφεται κάποιο μήνυμα αλλά σημαίνει ότι ο χρήστης δεν έχει πρόσβαση στο endpoint διότι δεν έχει πιστοποιηθεί από το API.
5	Η εφαρμογή δε φτάνει ποτέ στο endpoint	500: Internal Server Error	-	Δεν επιστρέφεται κάποιο μήνυμα αλλά σημαίνει ότι το API δεν είναι διαθέσιμο.

Πίνακας 6.3.51: Responses του endpoint 9 του controller PatientCheckupControlleryμα τον πίνακα 6.3.42

6.3.5 PatientFormController

Ο controller αυτός δίνει τη δυνατότητα στον ιατρό να προσθέσει και να διαχειριστεί φόρμες τις οποίες συνδέει με εξετάσεις ασθενών με τους οποίους είναι διασυνδεδεμένος με το ιστορικό τους. Οι φόρμες αυτές βασίζονται σε πρότυπα φόρμας που είναι αποθηκευμένα στο σύστημα. Ο χρήστης να επεξεργαστεί, να δημιουργήσει και να αποθηκεύσει νέα πρότυπα. Επίσης μπορεί να δημιουργήσει ερωτήσεις στα πρότυπα αυτά, να επεξεργαστεί ή και να διαγράψει υπάρχουσες. Τέλος, λαμβάνοντας φόρμα που είναι διασυνδεδεμένη με συγκεκριμένη εξέταση, ο χρήστης μπορεί να προσθέσει απαντήσεις στις ερωτήσεις που περιέχει.

Στον παρακάτω πίνακα φαίνονται τα endpoints που περιέχει:

PatientFormController (http://localhost: 5230/PatientForm)					
A/A	Endpoint	HTTP Μέθοδος	Περιγραφή	Δεδομένα Εισόδου στο endpoint	Δεδομένα εισόδου στο σώμα της μεθόδου
1	/forms	GET	Επιστρέφει μία λίστα με όλες τις	-	-

			πρότυπες φόρμες που είναι αποθηκευμένες στο σύστημα.		
2	/form/{formid}	GET	Επιστρέφει τη ζητούμενη πρότυπη φόρμα.	Αριθμός που δηλώνει τον κωδικό με τον οποίο αποθηκεύτηκε η πρότυπη φόρμα στη βάση.	-
3	/newPrototypeForm	POST	Μέσω αυτού του endpoint ο χρήστης προσθέτει νέα πρότυπη φόρμα στο σύστημα.	-	AddPrototypeFormDto Πίνακας 6.3.15
4	/editForm	PUT	Μέσω αυτού του endpoint ο χρήστης μπορεί να επεξεργαστεί μία συγκεκριμένη πρότυπη φόρμα.	-	EditPrototypeFormDto Πίνακας 6.3.16
5	/deleteForm/{formid}	DELETE	Μέσω αυτού του endpoint ο χρήστης διαγράφει μία πρότυπη φόρμα.	Αριθμός που δηλώνει τον κωδικό με τον οποίο αποθηκεύτηκε η πρότυπη φόρμα στη βάση.	-
6	/newQuestion	POST	Μέσω αυτού του endpoint ο χρήστης δημιουργεί νέα ερώτηση για μία συγκεκριμένη πρότυπη φόρμα.	-	AddQuestionFormDto Πίνακας 6.3.17
7	/deleteQuestion	DELETE	Μέσω αυτού του endpoint ο χρήστης διαγράφει την	-	DeleteQuestionFormDto

			ερώτηση από το σύστημα.		Πίνακας 6.3.19
8	/editquestion	PUT	Μέσω αυτού του endpoint ο χρήστης επεξεργάζεται μία συγκεκριμένη ερώτηση.	-	EditQuestionDto Πίνακας 6.3.18
9	/addcheckupform	POST	Μέσω αυτού του endpoint ο χρήστης προσθέτει πρότυπη φόρμα σε κάποια εξέταση.	-	AddCheckupFormDto Πίνακας 6.3.21
10	/newAnswer	POST	Μέσω αυτού του endpoint ο χρήστης προσθέτει απάντηση στην ερώτηση συγκεκριμένης φόρμας κάποιας εξέτασης	-	AddAnswerQuestionDto Πίνακας 6.3.23

Πίνακας 6.3.52: Endpoints του controller PatientCheckupController

Στους παρακάτω πίνακες φαίνονται αναλυτικά τα responses των endpoints του πίνακα 6.3.52:

PatientFormController: Endpoint 1 Responses					
A/A	Περίπτωση	Status Code	Δεδομένα που επιστρέφει		Μήνυμα
1	Ο χρήστης έχει κάνει είσοδο στην εφαρμογή	200: Success	Λίστα του τύπου GetPrototypeFormDto Πίνακας 6.3.14	-	
2	Ο χρήστης δεν έχει πιστοποιηθεί	401: Unauthorized	-		Δεν επιστρέφεται κάποιο μήνυμα αλλά σημαίνει

	από το API με JWT			ότι ο χρήστης δεν έχει πρόσβαση στο endpoint διότι δεν έχει πιστοποιηθεί από το API.
3	Η εφαρμογή δε φτάνει ποτέ στο endpoint	500: Internal Server Error	-	Δεν επιστρέφεται κάποιο μήνυμα αλλά σημαίνει ότι το API δεν είναι διαθέσιμο.

Πίνακας 6.3.53: Responses του endpoint 1 του controller PatientFormController για τον πίνακα 6.3.52

PatientFormController: Endpoint 2 Responses				
A/A	Περίπτωση	Status Code	Δεδομένα που επιστρέφει	Μήνυμα
1	Υπάρχει η πρότυπη φόρμα που ζητά ο χρήστης.	200: Success	GetPrototypeFormDto Πίνακας 6.3.14	-
2	Δεν υπάρχει η πρότυπη φόρμα που ζητά ο χρήστης.	200: Success	-	Ο χρήστης ειδοποιείται ότι δεν υπάρχει η πρότυπη φόρμα που ζητά ο χρήστης.
3	Ο χρήστης δεν έχει πιστοποιηθεί από το API με JWT	401: Unauthorized	-	Δεν επιστρέφεται κάποιο μήνυμα αλλά σημαίνει ότι ο χρήστης δεν έχει πρόσβαση στο endpoint διότι δεν έχει

				πιστοποιηθεί από το API.
4	Η εφαρμογή δε φτάνει ποτέ στο endpoint	500: Internal Server Error	-	Δεν επιστρέφεται κάποιο μήνυμα αλλά σημαίνει ότι το API δεν είναι διαθέσιμο.

Πίνακας 6.3.54: Responses του endpoint 1 του controller PatientFormController για τον πίνακα 6.3.52

PatientFormController: Endpoint 3 Responses				
A/A	Περίπτωση	Status Code	Δεδομένα που επιστρέφει	Μήνυμα
1	Ο χρήστης έχει κάνει είσοδο στην εφαρμογή.	200: Success	GetPrototypeFormDto Πίνακας 6.3.14	-
2	Ο χρήστης δεν έχει πιστοποιηθεί από το API με JWT	401: Unauthorized	-	Δεν επιστρέφεται κάποιο μήνυμα αλλά σημαίνει ότι ο χρήστης δεν έχει πρόσβαση στο endpoint διότι δεν έχει πιστοποιηθεί από το API.
3	Η εφαρμογή δε φτάνει ποτέ στο endpoint	500: Internal Server Error	-	Δεν επιστρέφεται κάποιο μήνυμα αλλά σημαίνει ότι το API δεν είναι διαθέσιμο.

Πίνακας 6.3.55: Responses του endpoint 1 του controller PatientFormController για τον πίνακα 6.3.52

PatientFormController: Endpoint 4 Responses

A/A	Περίπτωση	Status Code	Δεδομένα που επιστρέφει	Μήνυμα
1	Υπάρχει η φόρμα που θέλει να τροποποιήσει ο χρήστης	200: Success	Τροποποιημένη τη φόρμα GetPrototypeFormDto Πίνακας 6.3.14	-
2	Δεν υπάρχει η φόρμα που θέλει να τροποποιήσει ο χρήστης	200: Success	-	Ο χρήστης ειδοποιείται ότι δεν υπάρχει η φόρμα που θέλει να τροποποιήσει.
3	Ο χρήστης δεν έχει πιστοποιηθεί από το API με JWT	401: Unauthorized	-	Δεν επιστρέφεται κάποιο μήνυμα αλλά σημαίνει ότι ο χρήστης δεν έχει πρόσβαση στο endpoint διότι δεν έχει πιστοποιηθεί από το API.
4	Η εφαρμογή δε φτάνει ποτέ στο endpoint	500: Internal Server Error	-	Δεν επιστρέφεται κάποιο μήνυμα αλλά σημαίνει ότι το API δεν είναι διαθέσιμο.

Πίνακας 6.3.56: Responses του endpoint 1 του controller PatientFormController για τον πίνακα 6.3.52

PatientFormController: Endpoint 5 Responses

A/A	Περίπτωση	Status Code	Δεδομένα επιστρέφει	Μήνυμα
1	Υπάρχει η φόρμα που θέλει να διαγράψει ο χρήστης	200: Success	-	Ο χρήστης ειδοποιείται ότι η πρότυπη φόρμα αφαιρέθηκε από το σύστημα.
2	Δεν υπάρχει η φόρμα που θέλει να διαγράψει ο χρήστης.	200: Success	-	Ο χρήστης ειδοποιείται ότι δεν υπάρχει η πρότυπη φόρμα.
3	Ο χρήστης δεν έχει πιστοποιηθεί από το API με JWT	401: Unauthorized	-	Δεν επιστρέφεται κάποιο μήνυμα αλλά σημαίνει ότι ο χρήστης δεν έχει πρόσβαση στο endpoint διότι δεν έχει πιστοποιηθεί από το API.
4	Η εφαρμογή δε φτάνει ποτέ στο endpoint	500: Internal Server Error	-	Δεν επιστρέφεται κάποιο μήνυμα αλλά σημαίνει ότι το API δεν είναι διαθέσιμο.

Πίνακας 6.3.57: Responses του endpoint 1 του controller PatientFormController για τον πίνακα 6.3.52

PatientFormController: Endpoint 6 Responses

A/A	Περίπτωση	Status Code	Δεδομένα επιστρέφει	Μήνυμα
1	Ο χρήστης προσθέτει ερώτηση για πρότυπη φόρμα που υπάρχει στο σύστημα.	200: Success	Την πρότυπη φόρμα του τύπου GetPrototypeFormDto (Πίνακας 6.3.14).	-
2	Ο χρήστης προσθέτει ερώτηση για πρότυπη φόρμα που δεν υπάρχει στο σύστημα.	200: Success	-	Ο χρήστης ειδοποιείται ότι δεν υπάρχει η πρότυπη φόρμα.
4	Ο χρήστης δεν έχει πιστοποιηθεί από το API με JWT	401: Unauthorized	-	Δεν επιστρέφεται κάποιο μήνυμα αλλά σημαίνει ότι ο χρήστης δεν έχει πρόσβαση στο endpoint διότι δεν έχει πιστοποιηθεί από το API.
5	Η εφαρμογή δε φτάνει ποτέ στο endpoint	500: Internal Server Error	-	Δεν επιστρέφεται κάποιο μήνυμα αλλά σημαίνει ότι το API δεν είναι διαθέσιμο.

Πίνακας 6.3.58: Responses του endpoint 1 του controller PatientFormController για τον πίνακα 6.3.52

PatientFormController: Endpoint 7 Responses

A/A	Περίπτωση	Status Code	Δεδομένα επιστρέφει	Μήνυμα
1	Υπάρχει η ερώτηση που θέλει να διαγράψει ο χρήστης στην πρότυπη φόρμα.	200: Success	Την πρότυπη φόρμα του τύπου GetPrototypeFormDto (Πίνακας 6.3.14).	-
2	Ο χρήστης αιτείται τη διαγραφή ερώτησης για πρότυπη φόρμα που δεν υπάρχει στο σύστημα.	200: Success	-	Ο χρήστης ειδοποιείται ότι δεν υπάρχει η πρότυπη φόρμα.
3	Ο χρήστης αιτείται τη διαγραφή ερώτησης που δεν υπάρχει στο σύστημα.	200: Success	-	Ο χρήστης ειδοποιείται ότι δεν υπάρχει η ερώτηση στο σύστημα.
4	Ο χρήστης δεν έχει πιστοποιηθεί από το API με JWT	401: Unauthorized	-	Δεν επιστρέφεται κάποιο μήνυμα αλλά σημαίνει ότι ο χρήστης δεν έχει πρόσβαση στο endpoint διότι δεν έχει πιστοποιηθεί από το API.
5	Η εφαρμογή δε φτάνει ποτέ στο endpoint	500: Internal Server Error	-	Δεν επιστρέφεται κάποιο μήνυμα αλλά σημαίνει ότι το API δεν

				είναι διαθέσιμο.
--	--	--	--	------------------

Πίνακας 6.3.59: Responses του endpoint 1 του controller PatientFormController για τον πίνακα 6.3.52

PatientFormController: Endpoint 8 Responses				
A/A	Περίπτωση	Status Code	Δεδομένα που επιστρέφει	Μήνυμα
1	Υπάρχει η ερώτηση που θέλει να τροποποιήσει ο χρήστης στην πρότυπη φόρμα.	200: Success	Την ερώτηση του τύπου GetQuestionDto (Πίνακας 6.3.18).	-
2	Ο χρήστης αιτείται την τροποποίηση ερώτησης για πρότυπη φόρμα που δεν υπάρχει στο σύστημα.	200: Success	-	Ο χρήστης ειδοποιείται ότι δεν υπάρχει η πρότυπη φόρμα.
3	Ο χρήστης αιτείται την τροποποίηση ερώτησης που δεν υπάρχει στο σύστημα.	200: Success	-	Ο χρήστης ειδοποιείται ότι δεν υπάρχει η ερώτηση στο σύστημα.
4	Ο χρήστης δεν έχει πιστοποιηθεί από το API με JWT	401: Unauthorized	-	Δεν επιστρέφεται κάποιο μήνυμα αλλά σημαίνει ότι ο χρήστης δεν έχει πρόσβαση στο endpoint διότι δεν έχει πιστοποιηθεί από το API.

5	Η εφαρμογή δε φτάνει ποτέ στο endpoint	500: Internal Server Error	-	Δεν επιστρέφεται κάποιο μήνυμα αλλά σημαίνει ότι το API δεν είναι διαθέσιμο.
---	--	----------------------------	---	--

Πίνακας 6.3.60: Responses του endpoint 1 του controller PatientFormController για τον πίνακα 6.3.52

PatientFormController: Endpoint 9 Responses				
A/A	Περίπτωση	Status Code	Δεδομένα που επιστρέφει	Μήνυμα
1	Υπάρχει η πρότυπη φόρμα που θέλει να προσθέσει ο χρήστης στην συγκεκριμένη εξέταση.	200: Success	Την ερώτηση του τύπου GetCheckupFormDto (Πίνακας 6.3.20).	-
2	Ο χρήστης αιτείται την προσθήκη πρότυπης φόρμας που δεν υπάρχει στο σύστημα.	200: Success	-	Ο χρήστης ειδοποιείται ότι δεν υπάρχει η πρότυπη φόρμα.
3	Ο χρήστης αιτείται την προσθήκη πρότυπης φόρμας σε εξέταση που δεν υπάρχει στο σύστημα.	200: Success	-	Ο χρήστης ειδοποιείται ότι δεν υπάρχει η εξέταση στο σύστημα.
4	Ο χρήστης δεν έχει πιστοποιηθεί από το API με JWT	401: Unauthorized	-	Δεν επιστρέφεται κάποιο μήνυμα αλλά σημαίνει ότι ο χρήστης δεν έχει πρόσβαση στο

				endpoint διότι δεν έχει πιστοποιηθεί από το API.
5	Η εφαρμογή δε φτάνει ποτέ στο endpoint	500: Internal Server Error	-	Δεν επιστρέφεται κάποιο μήνυμα αλλά σημαίνει ότι το API δεν είναι διαθέσιμο.

Πίνακας 6.3.61: Responses του endpoint 1 του controller PatientFormController για τον πίνακα 6.3.52

PatientFormController: Endpoint 10 Responses				
A/A	Περίπτωση	Status Code	Λεδομένα που επιστρέφει	Μήνυμα
1	Υπάρχει η ερώτηση που θέλει να προσθέσει ο χρήστης για ερώτηση σε φόρμα εξέτασης.	200: Success	Την απάντηση του τύπου GetAnswerDto (Πίνακας 6.3.22).	-
2	Ο χρήστης αιτείται την προσθήκη απάντησης αλλά δεν υπάρχει η εξέταση.	200: Success	-	Ο χρήστης ειδοποιείται ότι δεν υπάρχει η εξέταση.
3	Ο χρήστης αιτείται την προσθήκη απάντησης αλλά η εξέταση δεν έχει φόρμα.	200: Success	-	Ο χρήστης ειδοποιείται ότι δεν υπάρχει αυτή η φόρμα για τη συγκεκριμένη εξέταση.
4	Ο χρήστης αιτείται την	200: Success	-	Ο χρήστης ειδοποιείται ότι

	προσθήκη απάντησης αλλά δεν υπάρχει η ερώτηση που ζητά στη φόρμα της εξέτασης.			δεν η ερώτηση που αιτείται.
5	Ο χρήστης δεν έχει πιστοποιηθεί από το API με JWT	401: Unauthorized	-	Δεν επιστρέφεται κάποιο μήνυμα αλλά σημαίνει ότι ο χρήστης δεν έχει πρόσβαση στο endpoint διότι δεν έχει πιστοποιηθεί από το API.
6	Η εφαρμογή δε φτάνει ποτέ στο endpoint	500: Internal Server Error	-	Δεν επιστρέφεται κάποιο μήνυμα αλλά σημαίνει ότι το API δεν είναι διαθέσιμο.

Πίνακας 6.3.62: Responses του endpoint 1 του controller PatientFormController για τον πίνακα 6.3.52

6.3.6 PatientTreatmentController

Ο controller αυτός δίνει τη δυνατότητα στον ιατρό να προσθέσει και να διαχειριστεί θεραπείες για συγκεκριμένους ασθενείς με τους οποίους είναι διασυνδεδεμένος με το ιστορικό τους. Υπάρχουν αποθηκευμένες στο σύστημα κάποιες πρότυπες θεραπείες τις οποίες ο χρήστης μπορεί να λάβει σε λίστα με ένα αίτημα. Χρησιμοποιώντας το ΑΜΚΑ του ασθενούς του και την ημερομηνία ανάθεσης, ο ιατρός μπορεί να λάβει συγκεκριμένη θεραπεία που έχει χορηγήσει. Επίσης χρησιμοποιώντας το ΑΜΚΑ μπορεί να λάβει μία λίστα με όλες τις ενεργές θεραπείες που έχουν ανατεθεί σε συγκεκριμένο ασθενή. Τέλος έχει τη δυνατότητα να προσθέσει και να επεξεργαστεί κάποια θεραπεία που είναι αποθηκευμένη στη βάση.

Στον παρακάτω πίνακα φαίνονται τα endpoints που περιέχει:

PatientTreatmentController (http://localhost: 5230/PatientTreatment)					
A/A	Endpoint	HTTP Μέθοδος	Περιγραφή	Δεδομένα Εισόδου στο endpoint	Δεδομένα εισόδου στο σώμα της μεθόδου
1	/default	GET	Επιστρέφει μία λίστα με όλες τις πρότυπες θεραπείες που είναι αποθηκευμένες στο σύστημα.	-	-
2	/amka	POST	Επιστρέφει θεραπεία για συγκεκριμένο ασθενή με βάση την ημερομηνία.	Το ΑΜΚΑ του ασθενούς	DateRequestDto Πίνακας 6.3.11
3	/active	POST	Επιστρέφει μία λίστα με όλες τις ενεργές θεραπείες ενός ασθενή.	-	PatientByAmkaDto Πίνακας 6.3.6
4	/addTreatment/amka	POST	Μέσω αυτού του endpoint ο χρήστης δημιουργεί μία νέα θεραπεία για συγκεκριμένο ασθενή.	Το ΑΜΚΑ του ασθενούς	AddTreatmentDescriptionPresetDto Πίνακας 6.3.27
5	/updateTreatment/amka	PUT	Μέσω αυτού του endpoint ο χρήστης τροποποιεί μία θεραπεία συγκεκριμένου ασθενή.	Το ΑΜΚΑ του ασθενούς	UpdateTreatmentDescriptionPresetDto Πίνακας 6.3.26

Πίνακας 6.3.63: Endpoints του controller PatientTreatmentController

Στους παρακάτω πίνακες φαίνονται αναλυτικά τα responses των endpoints του πίνακα 6.3.63:

PatientTreatmentController: Endpoint 1 Responses

A/A	Περίπτωση	Status Code	Δεδομένα που επιστρέφει	Μήνυμα
1	Ο χρήστης έχει κάνει είσοδο στην εφαρμογή	200: Success	Λίστα του τύπου TreatmentDto Πίνακας 6.3.23	«Οκ»
2	Ο χρήστης δεν έχει πιστοποιηθεί από το API με JWT	401: Unauthorized	-	Δεν επιστρέφεται κάποιο μήνυμα αλλά σημαίνει ότι ο χρήστης δεν έχει πρόσβαση στο endpoint διότι δεν έχει πιστοποιηθεί από το API.
3	Η εφαρμογή δε φτάνει ποτέ στο endpoint	500: Internal Server Error	-	Δεν επιστρέφεται κάποιο μήνυμα αλλά σημαίνει ότι το API δεν είναι διαθέσιμο.

Πίνακας 6.3.64: Responses του endpoint 1 του controller PatientTreatmentController για τον πίνακα 6.3.63

PatientTreatmentController: Endpoint 2 Responses

A/A	Περίπτωση	Status Code	Δεδομένα που επιστρέφει	Μήνυμα
1	Υπάρχει θεραπεία για τον ασθενή με τη ζητούμενη ημερομηνία.	200: Success	TreatmentDescriptionPresetDt ο Πίνακας 6.3.24	«Οκ»
2	Δεν υπάρχει ο ασθενής.	200: Success	-	Ο χρήστης ειδοποιείται ότι δεν

				υπάρχει ο ασθενής.
3	Δεν υπάρχει θεραπεία του ασθενούς για τη ζητούμενη ημερομηνία.	200: Success	-	Ο χρήστης ειδοποιείται ότι δεν υπάρχει η ζητούμενη θεραπεία.
4	Ο χρήστης δεν έχει πιστοποιηθεί από το API με JWT	401: Unauthorized	-	Δεν επιστρέφεται κάποιο μήνυμα αλλά σημαίνει ότι ο χρήστης δεν έχει πρόσβαση στο endpoint διότι δεν έχει πιστοποιηθεί από το API.
5	Η εφαρμογή δε φτάνει ποτέ στο endpoint	500: Internal Server Error	-	Δεν επιστρέφεται κάποιο μήνυμα αλλά σημαίνει ότι το API δεν είναι διαθέσιμο.

Πίνακας 6.3.65: Responses του endpoint 2 του controller PatientTreatmentController για τον πίνακα 6.3.63

PatientTreatmentController: Endpoint 3 Responses				
A/A	Περίπτωση	Status Code	Δεδομένα που επιστρέφει	Μήνυμα
1	Ο χρήστης ζητάει θεραπείες για ασθενή που είναι διασυνδεδεμέν	200: Success	Λίστα του τύπου TreatmentDescriptionPreset Dto Πίνακας 6.3.24	«ΟΚ»

	ος με το ιστορικό του.			
2	Ο χρήστης ζητάει θεραπείες για ασθενή που δεν είναι διασυνδεδεμένος με το ιστορικό του.	200: Success	-	Ο χρήστης ειδοποιείται ότι δεν είναι διασυνδεδεμένος με το ιστορικό του ασθενούς.
3	Δεν υπάρχουν θεραπείες για τον ασθενή.	200: Success	-	Ο χρήστης ειδοποιείται ότι δεν υπάρχουν θεραπείες για τον ασθενή.
4	Ο χρήστης δεν έχει πιστοποιηθεί από το API με JWT	401: Unauthorized	-	Δεν επιστρέφεται κάποιο μήνυμα αλλά σημαίνει ότι ο χρήστης δεν έχει πρόσβαση στο endpoint διότι δεν έχει πιστοποιηθεί από το API.
5	Η εφαρμογή δε φτάνει ποτέ στο endpoint	500: Internal Server Error	-	Δεν επιστρέφεται κάποιο μήνυμα αλλά σημαίνει ότι το API δεν είναι διαθέσιμο.

Πίνακας 6.3.66: Responses του endpoint 2 του controller PatientTreatmentController για τον πίνακα 6.3.63

PatientTreatmentController: Endpoint 4 Responses

A/A	Περίπτωση	Status Code	Λεδομένα που επιστρέφει	Μήνυμα
1	Ο χρήστης είναι διασυνδεδεμένος με το ιστορικό του ασθενούς και υπάρχει η πρότυπη θεραπεία που αιτείται.	200: Success	TreatmentDescriptionPreset Dto Πίνακας 6.3.24	«ΟΚ»
2	Ο χρήστης δεν είναι διασυνδεδεμένος με το ιστορικό του ασθενούς.	200: Success	-	Ο χρήστης ειδοποιείται ότι δεν είναι διασυνδεδεμένος με το ιστορικό του ασθενούς.
3	Δεν υπάρχει η πρότυπη θεραπεία.	200: Success		Ο χρήστης ειδοποιείται ότι δεν υπάρχει η πρότυπη θεραπεία.
4	Ο χρήστης δεν έχει πιστοποιηθεί από το API με JWT	401: Unauthorized	-	Δεν επιστρέφεται κάποιο μήνυμα αλλά σημαίνει ότι ο χρήστης δεν έχει πρόσβαση στο endpoint διότι δεν έχει πιστοποιηθεί από το API.
5	Η εφαρμογή δε φτάνει ποτέ στο endpoint	500: Internal Server Error	-	Δεν επιστρέφεται κάποιο μήνυμα αλλά σημαίνει ότι το API δεν

				είναι διαθέσιμο.
--	--	--	--	------------------

Πίνακας 6.3.67: Responses του endpoint 4 του controller PatientTreatmentController για τον πίνακα 6.3.63

PatientTreatmentController: Endpoint 5 Responses				
A/A	Περίπτωση	Status Code	Δεδομένα που επιστρέφει	Μήνυμα
1	Ο χρήστης έχει συμπληρώσει ΑΜΚΑ ασθενή που έχει πρόσβαση στο ιστορικό του και υπάρχει η θεραπεία που αιτείται να επεξεργαστεί.	200: Success	Τροποποιημένη τη θεραπεία του τύπου TreatmentDescriptionPreset Dto (Πίνακας 6.3.24)	«Οκ»
2	Δεν είναι διασυνδεδεμένος με τον ασθενή του οποίου έχει συμπληρώσει το ΑΜΚΑ.	200: Success	-	Ο χρήστης ειδοποιείται ότι δεν είναι διασυνδεδεμένος με τον ασθενή.
3	Δεν υπάρχει η θεραπεία που θέλει να τροποποιήσει ο χρήστης.	200: Success	-	Ο χρήστης ειδοποιείται ότι δεν υπάρχει η θεραπεία που αναζητά ο χρήστης.
4	Ο χρήστης δεν έχει πιστοποιηθεί από το API με JWT	401: Unauthorized	-	Δεν επιστρέφεται κάποιο μήνυμα αλλά σημαίνει ότι ο χρήστης δεν έχει πρόσβαση στο endpoint διότι δεν έχει

				πιστοποιηθεί από το API.
5	Η εφαρμογή δε φτάνει ποτέ στο endpoint	500: Internal Server Error	-	Δεν επιστρέφεται κάποιο μήνυμα αλλά σημαίνει ότι το API δεν είναι διαθέσιμο.

Πίνακας 6.3.68: Responses του endpoint 5 του controller PatientTreatmentController για τον πίνακα 6.3.63

6.3.7 AggregatedStatisticsController

Ο controller αυτός δίνει τη δυνατότητα στον γιατρό να δει κάποια στατιστικά επί του συνόλου των ασθενών για τους οποίους έχει πρόσβαση στο ιστορικό τους, με βάση την ηλικία το φύλλο, το ιστορικό των γονέων σχετικά με το διαβήτη και άλλα χαρακτηριστικά. Στο Front-End καλείται ένα endpoint αυτού του controller, μέσω του οποίου ο ιατρός, λαμβάνει το ποσοστό εμφάνισης του διαβήτη, σε κάθε ηλικιακή κατηγορία, επί του συνόλου των ασθενών με τους οποίους είναι διασυνδεδεμένος.

AggregatedStatisticsController (http://localhost: 5230/AggregatedStatistics)					
A/A	Endpoint	HTTP Μέθοδος	Περιγραφή	Δεδομένα Εισόδου στο endpoint	Δεδομένα εισόδου στο σώμα της μεθόδου
1	/statistics/age	GET	Επιστρέφει στατιστικά εμφάνισης του διαβήτη με βάση την ηλικία, επί του συνόλου των ασθενών που παρακολουθεί ο χρήστης το ιστορικό τους.	-	-

Πίνακας 6.3.69: Endpoints του controller AggregatedStatisticsController

Στον παρακάτω πίνακα φαίνεται τα πιθανά responses του endpoint του πίνακα 6.3.69:

AggregatedStatisticsController: Endpoint 1 Responses

A/A	Περίπτωση	Status Code	Δεδομένα επιστρέφει	Μήνυμα
1	Ο χρήστης έχει κάνει είσοδο στην εφαρμογή.	200: Success	AgeStatistics Πίνακας 6.3.28	«Οκ»
2	Ο χρήστης δεν έχει διασυνδεθεί με το ιστορικό κάποιων ασθενών.	200: Success		Ο χρήστης ειδοποιείται ότι δεν έχει διασυνδεθεί με το ιστορικό ασθενών.
3	Ο χρήστης δεν έχει πιστοποιηθεί από το API με JWT	401: Unauthorized	-	Δεν επιστρέφεται κάποιο μήνυμα αλλά σημαίνει ότι ο χρήστης δεν έχει πρόσβαση στο endpoint διότι δεν έχει πιστοποιηθεί από το API.
4	Η εφαρμογή δε φτάνει ποτέ στο endpoint	500: Internal Server Error	-	Δεν επιστρέφεται κάποιο μήνυμα αλλά σημαίνει ότι το API δεν είναι διαθέσιμο.

Πίνακας 6.3.70: Responses του endpoint 1 του controller AggregatedStatisticsController για τον πίνακα 6.3.69

6.3.8 MobileDataController

Ο controller αυτός δίνει τη δυνατότητα στον χρήστη να δει κάποια δεδομένα από αυτά που εισάγει ο ασθενής στο σύστημα μέσω της εφαρμογής του. Για να μπορεί ο χρήστης – ιατρός να αποκτήσει πρόσβαση στα δεδομένα κάποιου ασθενή θα πρέπει ο χρήστης – ασθενής της εφαρμογής του κινητού να του δώσει άδεια για αυτό.

Όταν δημιουργείται ένας νέος χρήστης- ασθενής μέσω της εφαρμογής του κινητού αυτόματα δημιουργείται και μία νέα εγγραφή για τον πίνακα patientData (Πίνακας 6.1.1.3) της βάσης. Ο ασθενής πρέπει να επικοινωνήσει με τον γιατρό του έτσι ώστε

εκείνος να προσθέσει το ιστορικό του στη λίστα αυτών που παρακολουθεί (Endpoint 3 του PatientController, Πίνακας 6.3.39). Όταν ο ιατρός προσθέτει το ιστορικό ενός ασθενή στη λίστα αυτών που παρακολουθεί, ο ασθενής λαμβάνει μία ειδοποίηση η οποία τον καλεί να δώσει άδεια στον γιατρό για να παρακολουθεί τα δεδομένα που εκείνος εισάγει μέσω της εφαρμογής του κινητού του.

Τέλος όταν ένας χρήστης – ιατρός προσθέτει ένα νέο ιστορικό ασθενή στο σύστημα τότε αν ο ασθενής αυτός συνδέεται με κάποιο ενεργό χρήστη του έρχεται ειδοποίηση άδειας πρόσβασης στα δεδομένα με τον τρόπο που αναπτύχθηκε προηγουμένως.

Το Front – End καλεί μόνο ένα endpoint αυτού του controller το οποίο φαίνεται στον παρακάτω πίνακα:

MobileDataController (http://localhost: 5230/MobileData)					
A/A	Endpoint	HTTP Μέθοδος	Περιγραφή	Δεδομένα Εισόδου στο endpoint	Δεδομένα εισόδου στο σώμα της μεθόδου
1	/totalglucosepercent	POST	Επιστρέφει ποσοστά επί του συνόλου των μετρήσεων για την προγευματική, τη μεταγευματική και την προ ύπνου γλυκόζη λαμβάνοντας υπόψη όλες τις μετρήσεις γλυκόζης που έχει εισάγει ο χρήστης – ασθενής στο σύστημα.	-	PatientByAmkaDto Πίνακας 6.3.6

Πίνακας 6.3.71: Endpoints του controller MobileDataController

Στον παρακάτω πίνακα φαίνονται τα πιθανά responses του endpoint του πίνακα 6.3.71:

MobileDataController: Endpoint 1 Responses

A/A	Περίπτωση	Status Code	Δεδομένα επιστρέφει	Μήνυμα
1	Ο χρήστης έχει κάνει είσοδο στην εφαρμογή.	200: Success	GetGlucosePercentsDto Πίνακας 6.2.13	«Οκ»
2	Ο χρήστης δεν έχει διασυνδεθεί με το ιστορικό του ασθενούς.	200: Success		Ο χρήστης ειδοποιείται ότι δεν έχει διασυνδεθεί με το ιστορικό του ασθενούς.
3	Ο ασθενής δεν έχει δώσει άδεια πρόσβασης στον χρήστη	200: Success		Ο χρήστης ειδοποιείται ότι δεν έχει άδεια πρόσβασης στα δεδομένα που εισάγει ο ασθενής μέσω της εφαρμογής του.
4	Ο χρήστης δεν έχει πιστοποιηθεί από το API με JWT	401: Unauthorized	-	Δεν επιστρέφεται κάποιο μήνυμα αλλά σημαίνει ότι ο χρήστης δεν έχει πρόσβαση στο endpoint διότι δεν έχει πιστοποιηθεί από το API.
5	Η εφαρμογή δε φτάνει ποτέ στο endpoint	500: Internal Server Error	-	Δεν επιστρέφεται κάποιο μήνυμα αλλά σημαίνει ότι το API δεν είναι διαθέσιμο.

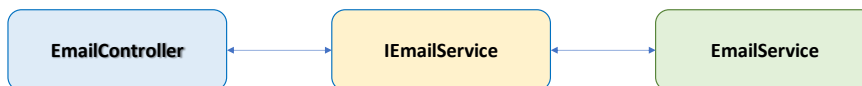
6.4 API για την αποστολή email στους χρήστες των δύο εφαρμογών

Στην ενότητα 5.3 του προηγούμενου κεφαλαίου φαίνεται στο σχήμα που περιγράφει την αρχιτεκτονική του Back-End συστήματος, ένα τρίτο API το οποίο καλείται μόνο από το Service που είναι υπεύθυνο για την ταυτοποίηση του χρήστη – ασθενή. Σε όλες τις σύγχρονες εφαρμογές, κομμάτι της ασφάλειας αποτελεί η λειτουργία της επαναφοράς του κωδικού πρόσβασης.

Στο παρόν Back-End υπάρχει υλοποίηση αυτής της λειτουργίας η οποία χρησιμοποιείται μόνο στην εφαρμογή κινητού για τον ασθενή. Τα endpoints 3, 4 και 5 του πίνακα 6.2.27 που περιγράφει τον controller που είναι υπεύθυνος για την πιστοποίηση του χρήστη – ασθενή περιγράφουν αναλυτικά αυτή τη διαδικασία. Η σειρά των ενεργειών του χρήστη είναι η εξής:

1. Ο ασθενής δηλώνει ότι ξέχασε τον κωδικό πρόσβασης του στο σύστημα και εισάγει το email και το ΑΜΚΑ του. Το endpoint 3 (Πίνακας 6.2.30) καλεί το υπεύθυνο Service και αυτό με τη σειρά του το API που είναι υπεύθυνο για την αποστολή του email.
2. Το Email API στέλνει έναν κωδικό μίας χρήσης στον χρήστη – ασθενή ο οποίος διαρκεί λήγει 5 λεπτά μετά τη δημιουργία του.
3. Ο χρήστης καλεί το endpoint 4 (Πίνακας 6.2.31) και πιστοποιεί τον κωδικό που έλαβε και τέλος
4. Με το endpoint 5 (Πίνακας 6.2.32) αλλάζει τον κωδικό πρόσβασης του στο σύστημα.

Η αρχιτεκτονική του Email API φαίνεται στο παρακάτω σχήμα:



Σχήμα 6.4.1: Αρχιτεκτονική του Email API

Για αυτό το API χρησιμοποιήθηκε το παρακάτω DTO:

EmailDto	
To	Email του παραλήπτη
Subject	Θέμα του email

Body	Κείμενο του email
-------------	-------------------

Πίνακας 6.4.1 DTO για την αποστολή email στον χρήστη της εφαρμογής του κινητού

Το endpoint του Email API το οποίο καλείται από το Service που αναφέρθηκε προηγουμένως, περιγράφεται στον παρακάτω πίνακα:

EmailController (http://localhost: 5294/api/Email)					
A/A	Endpoint	HTTP Μέθοδος	Περιγραφή	Δεδομένα Εισόδου στο endpoint	Δεδομένα εισόδου στο σώμα της μεθόδου
1	/	POST	Στέλνει email στον χρήστη – ασθενή με τον κωδικό μίας χρήσης που θα εισάγει στην εφαρμογή του για την επαναφορά του κωδικού πρόσβασης του.	-	EmailDto Πίνακας 6.4.1

Πίνακας 6.4.2 Endpoints του EmailController

EmailController: Endpoint 1 Responses				
A/A	Περίπτωση	Status Code	Δεδομένα που επιστρέφει	Μήνυμα
1	Το endpoint καλείται σωστά από το Service	200: Success	-	-
2	Το Service δε φτάνει ποτέ στο endpoint	500: Internal Server Error	-	Δεν επιστρέφεται κάποιο μήνυμα αλλά σημαίνει ότι το API δεν είναι διαθέσιμο.

Πίνακας 6.4.3 Responses για το Endpoint 1 του EmailController για τον πίνακα 6.4.2

Το email που φτάνει στον τελικό χρήστη φαίνεται στην παρακάτω εικόνα:

Reset Password Request



dmystem@outlook.com

Προς: Εσείς



Σάβ 22/6/2024 12:32 μ.μ.

Your reset password is: 5EFF5908EF62B40A2BF3

← Απάντηση

→ Προώθηση

Εικόνα 6.4.1: Email με τον κωδικό μίας χρήσης που στέλνει το σύστημα στον χρήστη – ασθενή.

7. Συμπεράσματα και Μελλοντικές Επεκτάσεις

7.1 Συμπεράσματα και χρησιμότητα της εφαρμογής

Όπως αναλύθηκε στα προηγούμενα κεφάλαια, η διαχείριση του Σακχαρώδους Διαβήτη είναι το πιο κρίσιμο συστατικό για την πορεία της υγείας του ασθενούς. Χρειάζεται να γίνει μέρος της καθημερινής ρουτίνας των ανθρώπων που νοσούν με τρόπο φιλικό έτσι ώστε να μη διακόψουν τη δέσμευση τους στη διαχείριση της νόσου. Το σύστημα που αναπτύχθηκε, αποτελεί μία ολοκληρωμένη πρόταση για τη διαχείριση αυτή διότι εξασφαλίζει με άμεσο και εύκολο τρόπο τα τρία κριτήρια που θα πρέπει να πληρούνται για την αντιμετώπιση της κατάστασης της νόσου:

1. Τη διαχείριση της νόσου από τους ασθενείς
2. Την παρακολούθηση και διαχείριση της κατάστασης των ασθενών από τους ιατρούς, και
3. Την αδιάλειπτη επικοινωνία μεταξύ ασθενή και ιατρού.

Το πιο σημαντικό κριτήριο για τον σχεδιασμό μίας εφαρμογής είναι η δημιουργία σχέσης δέσμευσης μεταξύ χρήστη και εφαρμογής. Η ραγδαία ενσωμάτωση των έξυπνων συσκευών στο ευρύ κοινωνικό σύνολο, οδήγησε στην ανάπτυξη εφαρμογών που δίνουν τη δυνατότητα στους χρήστες να καταγράφουν και να παρακολουθούν τις καθημερινές τους συνήθειες. Έτσι υπάρχουν εφαρμογές που για παράδειγμα καταγράφουν τη σωματική δραστηριότητα [27] και δίνουν ολοκληρωμένα προγράμματα άσκησης με στόχο τη βελτίωση της φυσικής κατάστασης, εφαρμογές που καταγράφουν τις διατροφικές συνήθειες και προσφέρουν ολοκληρωμένες προτάσεις για τη μείωση του σωματικού βάρους, εφαρμογές που καταγράφουν και εκτιμούν τον έμμηνο κύκλο των γυναικών κ.α. Δεδομένης της δυσκολίας που αντιμετωπίζει κάποιος να εισάγει δεδομένα σε πολλές διαφορετικές εφαρμογές καθημερινά, συγκεντρώθηκαν όλα τα ιατρικά δεδομένα που θα πρέπει ο ασθενής να εισάγει για την παρακολούθηση της πορείας του αλλά και εκείνα, που περιγράφουν συνήθειες της καθημερινότητας του και επηρεάζουν σε σημαντικό βαθμό την υγεία του. Έτσι σχεδιάστηκε ένα Back-End το οποίο θα εξυπηρετεί της ανάγκες μίας εφαρμογής κινητού, η οποία θα αντικαθιστά όλες τις υπόλοιπες εφαρμογές που σχετίζονται με την καθημερινότητα του ασθενούς αλλά επιπλέον θα μπορεί να προσθέτει τα ιατρικά του δεδομένα. Οπότε ο ασθενής πλέον μπορεί να εισάγει δεδομένα για τη γλυκόζη του για κάθε προγραμματισμένη μέτρηση για κάποια ώρα της ημέρας, μπορεί να εισάγει τις δόσεις ινσουλίνης αλλά και τα καθημερινά του γεύματα, το σωματικό του βάρος, την καθημερινή του φυσική άσκηση, τη διάθεση του και αν είναι θηλυκού γένους τον έμμηνο κύκλο του. Τέλος μπορεί να δει αποτελέσματα για την εξέλιξη της πορείας του με βάση τα δεδομένα της γλυκόζης που έχει αποθηκεύσει στο σύστημα αλλά και να λάβει εκτίμηση, ένα πλήρες ετήσιο ημερολόγιο του έμμηνου κύκλου του. Ο σχεδιασμός μίας εφαρμογής που συγκεντρώνει όλα αυτά τα χαρακτηριστικά, έχει προοπτικές να εξασφαλίσει την απαραίτητη δέσμευση του ασθενούς με αυτήν και κατ' επέκταση με τη διαχείριση του Σακχαρώδους Διαβήτη Τύπου 2.

Το δεύτερο σκέλος, αφορά τον σχεδιασμό που αναπτύχθηκε έτσι ώστε το Back-End να εξυπηρετήσει την εφαρμογή μέσω της οποίας ο ιατρός θα μπορεί να παρακολουθεί και να επεμβαίνει στην κατάσταση των ασθενών του. Μπορεί να βλέπει λίστες με το

ιστορικό των ασθενών του, να βλέπει, να προσθέτει και να διαχειρίζεται εξετάσεις που σχετίζονται με τους ασθενείς του αλλά και να αναθέτει θεραπείες που κρίνει ότι είναι απαραίτητο να τους χορηγήσει. Τέλος έχει τη δυνατότητα, να στείλει αίτημα στον ασθενή του, για να του ζητήσει την άδεια να μπορεί να λαμβάνει δεδομένα που ο ίδιος αποθηκεύει στο σύστημα μέσω της δικής του εφαρμογής. Αυτό δίνει τη δυνατότητα στον ιατρό να παρακολουθεί σε πραγματικό χρόνο την πορεία της υγείας του ασθενούς και να επέμβει άμεσα όταν χρειάζεται.

Όλα τα δεδομένα του συστήματος αποθηκεύονται σε μία κοινή βάση με αποτέλεσμα οι δύο εφαρμογές να χρησιμοποιούν με ευκολία όλες τις πληροφορίες που πρέπει να μοιράζονται από κοινού.

Συνολικά το σύστημα σχεδιάστηκε και αναπτύχθηκε με στόχο τη διευκόλυνση της διαχείρισης του σακχαρώδους διαβήτη τύπου 2 τόσο από τους ασθενείς όσο και από τους ιατρούς και συνολικά τη βελτίωση της ποιότητας ζωής των ανθρώπων που νοσούν.

7.2 Μελλοντικές Επεκτάσεις

1. Διαχειριστής συστήματος

Όπως αναφέρθηκε στην ενότητα 6.1 του παρόντος, έχουν σχεδιαστεί οντότητες οι οποίες αναπαριστούν σταθερά δεδομένα του συστήματος και είτε αφορούν δεδομένα των επαγγελματιών υγείας είτε αφορούν δεδομένα που εξυπηρετούν την καλή λειτουργία των δύο εφαρμογών. Τα δεδομένα αυτά δε θα πρέπει να μπορούν να μεταβάλλονται από τους χρήστες των δύο εφαρμογών είτε γιατί στην πρώτη περίπτωση υπάρχει κίνδυνος τα δεδομένα αυτά να χαρακτηριστούν ως μη έγκυρα είτε στη δεύτερη περίπτωση να τεθεί σε κίνδυνο η ασφάλεια του συστήματος.

Όσον αφορά τα σταθερά δεδομένα υγείας έχει υλοποιηθεί ένα API το οποίο δεν εξυπηρετεί τη διαχείριση τους. Μια σημαντική μελλοντική επέκταση του συστήματος είναι η προσθήκη σε αυτό τεχνικής πιστοποίησης χρήστη η οποία να βασίζεται σε ρόλους έτσι ώστε να είναι προσβάσιμο μόνο σε κάποιον που χαρακτηρίζεται ως διαχειριστής. Στον παρακάτω πίνακα παρουσιάζονται συνοπτικά τα endpoints που περιέχει ο controller αυτού του API:

SystemInfoController (http://localhost: 5216/SystemInfo)					
A/A	Endpoint	HTTP Μέθοδος	Περιγραφή	Δεδομένα Εισόδου στο endpoint	Δεδομένα εισόδου στο σώμα της μεθόδου
1	/smoking Status	GET	Επιστρέφει μία λίστα με τις περιπτώσεις	-	-

			σχέσης ασθενή - καπνίσματος		
2	/genderlist	GET	Επιστρέφει μία λίστα με τα φύλα των ασθενών	-	-
3	/parentsinfo	GET	Επιστρέφει μία λίστα με τις περιπτώσεις του ιστορικού των γονέων σε σχέση με τον διαβήτη.	-	-
4	/activityintensitylist	GET	Επιστρέφει μία λίστα με τις περιπτώσεις έντασης της άσκησης του ασθενούς.	-	-
5	/activitytypes	GET	Επιστρέφει μία λίστα με τα είδη άσκησης του ασθενούς.	-	-
6	/glucosetypes	GET	Επιστρέφει μία λίστα με τα είδη μέτρησης γλυκόζης.	-	-
7	/illnesstypes	GET	Επιστρέφει μία λίστα με τα είδη ασθένειας.	-	-
8	/insulintypes	GET	Επιστρέφει μία λίστα με τα είδη ινσουλίνης.	-	-
9	/mealtypes	GET	Επιστρέφει μία λίστα με τα είδη γευμάτων.	-	-

10	/medicationfrequencylist	GET	Επιστρέφει μία λίστα με τα είδη συχνότητας λήψης φαρμακευτικής αγωγής.	-	-
11	/medicationtypelist	GET	Επιστρέφει μία λίστα με τα είδη φαρμακευτικής αγωγής.	-	-
12	/moodstatuslist	GET	Επιστρέφει μία λίστα με τα είδη διάθεσης του ασθενούς.	-	-
13	/mealstlist	GET	Επιστρέφει μία λίστα με τα γεύματα.	-	-
14	/addmealstlist	POST	Προσθέτει πρότυπο γεύμα στη λίστα.-	-	-

Πίνακας 7.1.1: Endpoints του controller για τη διαχείριση σταθερών δεδομένων του συστήματος.

Με αυτό το API πρέπει να γίνει και όλη η διαχείριση των πρότυπων φόρμας και θεραπειάς που γίνεται έως τώρα από τον χρήστη – γιατρό αλλά και η διαχείριση των ειδοποιήσεων συστήματος οι οποίες αφορούν και τους δύο ασθενείς.

2. Υλοποίηση της διαδικασίας αλλαγής κωδικού πρόσβασης και στην εφαρμογή του ιατρού

Η ίδια υποστήριξη που παρέχει το Back-End για τη διαδικασία αλλαγής κωδικού πρόσβασης για την εφαρμογή του κινητού υπάρχει και για την εφαρμογή του ιστού και περιγράφεται στο σχήμα 3.1 2.

3. Δεδομένα που εισάγει ο ασθενής στο σύστημα μέσω της εφαρμογής του κινητού

Με την κατάλληλη άδεια πρόσβασης που παρέχει ο ασθενής στον γιατρό μέσω της εφαρμογής του, το Back-End παρέχει την υποστήριξη στο Front-End, οι δύο χρήστες να μπορούν να βλέπουν στατιστικά για τις μετρήσεις γλυκόζης, πίεσης και βάρους αλλά και για την σωματική άσκηση τα οποία υπολογίζονται

με βάση τα δεδομένα που εισάγει ο ασθενής στην εφαρμογή του κινητού. Τα στατιστικά αυτά αφορούν μετρήσεις σε διάστημα μίας εβδομάδας, 15 ημερών και ενός μήνα.

4. Email API

Για την αποστολή email στους χρήστες των δύο εφαρμογών, χρησιμοποιήθηκε ένας λογαριασμός που δημιουργήθηκε στον Email Server της Microsoft. Πρέπει να δημιουργηθεί ένας λογαριασμός στον Email Server του Πολυτεχνείου ή του Νοσοκομείου στο οποίο θα εγκατασταθεί το σύστημα έτσι ώστε να γίνεται επίσημα η αποστολή αυτών των μηνυμάτων.

5. Σχεδιασμός και ανάπτυξη υποδομής για ολοκληρωμένες προτάσεις προγραμμάτων

Όπως αναλύθηκε στην ενότητα 7.1, οι διαθέσιμες στην αγορά ολοκληρωμένες προτάσεις για την καταγραφή δεδομένων της καθημερινότητας, προσφέρουν προτάσεις προγραμμάτων για τους χρήστες, βασισμένα στα δεδομένα που συλλέγουν από αυτούς. Έτσι μπορεί να σχεδιαστεί υποσύστημα το οποίο θα συλλέγει το ιστορικό της φυσικής αγωγής του ασθενούς, το ιστορικό του βάρους και της αρτηριακής πίεσης και να προτείνει καθημερινό πρόγραμμα άσκησης με βάση τα χαρακτηριστικά του ασθενούς.

Με αντίστοιχο τρόπο και σε συνεργασία με τους θεράποντες ιατρούς, μπορούν να προτείνονται προγράμματα διατροφής που στόχο θα έχουν τον έλεγχο των επιπέδων σακχάρου στο αίμα και ο ιατρός να μπορεί να επεμβαίνει αν κρίνει ότι πρέπει να γίνει κάποια έκτακτη αλλαγή. Σε αυτό μπορεί να προστεθεί υποδομή για παιχνιδιοποίηση (*Zarkogianni, 2021*) της εφαρμογής του ασθενούς δημιουργώντας του προκλήσεις για κατάκτηση στόχων οι οποίες θα ενισχύουν την ψυχολογία του και θα του δίνουν κίνητρο να συνεχίσει την προσπάθεια.

6. Εγκατάσταση του Back-End

Τέλος πρέπει το Back-End σύστημα να εγκατασταθεί σε ένα υπολογιστικό σύστημα με πιστοποιημένη IP έτσι ώστε να είναι μονίμως διαθέσιμο και ασφαλές.

Βιβλιογραφία

[1] World Health Organization. (2023). *Diabetes*. <https://www.who.int/news-room/fact-sheets/detail/diabetes>

[2] National Institute of Diabetes and Digestive and Kidney Diseases. (2016). *Symptoms & Causes of Diabetes*. <https://www.niddk.nih.gov/health-information/diabetes/overview/symptoms-causes>

[3] American Diabetes Association. (2023). *Diabetes & DKA (Ketoacidosis)*. <https://diabetes.org/about-diabetes/complications/ketoacidosis-dka/dka-ketoacidosis-ketones>

[4] Steck, A. K., & Rewers, M. J. (2011). Genetics of Type 1 Diabetes. *Clinical Chemistry*, 57(2), 176. <https://doi.org/10.1373/CLINCHEM.2010.148221>

[5] Redondo, M. J., Steck, A. K., & Pugliese, A. (2018). Genetics of type 1 diabetes. *Pediatric Diabetes*, 19(3), 346. <https://doi.org/10.1111/PEDI.12597>

[6] Scherm, M. G., Serr, I., Zahm, A. M., Schug, J., Bellusci, S., Manfredini, R., Salb, V. K., Gerlach, K., Weigmann, B., Ziegler, A. G., Kaestner, K. H., & Daniel, C. (2019). miRNA142-3p targets Tet2 and impairs Treg differentiation and stability in models of type 1 diabetes. *Nature Communications*, 10(1). <https://doi.org/10.1038/s41467-019-13587-3>

[7] Pihoker, C., Gilliam, L. K., Hampe, C. S., & Lernmark, Å. (2005). Autoantibodies in Diabetes. *Diabetes*, 54(suppl_2), S52–S61. https://doi.org/10.2337/DIABETES.54.SUPPL_2.S52

[8] Eizirik, D. L., Szymczak, F., & Mallone, R. (2023). Why does the immune system destroy pancreatic β -cells but not α -cells in type 1 diabetes? *Nature Reviews Endocrinology* 2023 19:7, 19(7), 425–434. <https://doi.org/10.1038/s41574-023-00826-3>

[9] Reed, J., Bain, S., & Kanamarlapudi, V. (2021). A Review of Current Trends with Type 2 Diabetes Epidemiology, Aetiology, Pathogenesis, Treatments and Future Perspectives. *Diabetes, Metabolic Syndrome and Obesity*, 14, 3567–3602. <https://doi.org/10.2147/DMSO.S319895>

[10] Galicia-Garcia, U., Benito-Vicente, A., Jebari, S., Larrea-Sebal, A., Siddiqi, H., Uribe, K. B., Ostolaza, H., & Martín, C. (2020). Pathophysiology of Type 2 Diabetes Mellitus. *International Journal of Molecular Sciences* 2020, Vol. 21, Page 6275, 21(17), 6275. <https://doi.org/10.3390/IJMS21176275>

[11] Westman, E. C. (2021). Type 2 Diabetes Mellitus: A Pathophysiologic Perspective. *Frontiers in Nutrition*, 8, 707371. <https://doi.org/10.3389/FNUT.2021.707371/BIBTEX>

[12] Bellou, V., Belbasis, L., Tzoulaki, I., & Evangelou, E. (2018). Risk factors for type 2 diabetes mellitus: An exposure-wide umbrella review of meta-analyses. *PLOS ONE*, 13(3), e0194127. <https://doi.org/10.1371/JOURNAL.PONE.0194127>

[13] Hossen, A.-A., Rahman, A., Islam, S., Mia, M., & Ferdous, K. (2023). A Review: Assessment of HbA1c as A Diagnostic Tool in Diabetes Mellitus. In *Molecular Mechanism Research* (Vol. 1, Issue 1). <https://www.researchgate.net/publication/376523487>

[14] Ahmed, S., & Mahmood, Q. (2019). An authentication based scheme for applications using JSON web token. *Proceedings - 22nd International Multitopic Conference, INMIC 2019*. <https://doi.org/10.1109/INMIC48123.2019.9022766>

[15] Jánoky, L. V., Levendovszky, J., & Ekler, P. (2018). An analysis on the revoking mechanisms for JSON Web Tokens. *International Journal of Distributed Sensor Networks*, 14(9). <https://doi.org/https://journals.sagepub.com/doi/10.1177/1550147718801535>

- [16] S.G. Mougiakakou, K.S. Nikita, M.D. Protonotarios, G.K. Matsopoulos, "Neural Network based Decision Support System for Outpatients Diabetes Management," *Diabetes, Nutrition and Metabolism*, vol. 11(1), p. 74, 1998 .
- [17] S. Mougiakakou and K.S. Nikita, "A Neural Network Approach for Insulin Regime and Dose Adjustment in IDDM," *Diabetes Technology & Therapeutics*, vol. 2, no. 3, pp. 381-389, November 2000
- [18] D. Iliopoulou, K. Giokas, S. Mougiakakou, J. Stoitsis, A. Prentza, K. Nikita, "A Telematic System for Diabetes Management, Reporting and Patient Advice", *The Journal on Information Technology in Healthcare* 2005, 3(5), pp. 307-313, 2005.
- [19] S. Mougiakakou, A. Prountzou, K. Zarkogianni, and K.S. Nikita, "Prediction of glucose profile in children with type 1 diabetes mellitus using continuous glucose monitors and insulin pumps ", *Hormone Research* 70: 22-23 Suppl. 1, 2008.
- [20] S. G. Mougiakakou, C. Bartsocas, E. Bozas, N. Chaniotaki, D. Iliopoulou, I. Kouris, S. Pavlopoulos, A. Prountzou, M. Skevofylakas, A. Tsoukalis, K. Varotsis, A. Vazeou, K. Zarkogianni, K. S. Nikita, "SMARTDIAB: A Communication and Information Technology Approach for the Intelligent Monitoring, Management and Follow-up of Type 1 Diabetes Patients", *IEEE Transactions on Information Technology in Biomedicine, Special Issue: New and Emerging Trends in Bioinformatics and Bioengineering*, 14 (3), pp. 622-633, 2010.
- [21] M. Skevofilakas, K. Zarkogianni, B. G. Karamanos, K. S. Nikita, "A hybrid Decision Support System for the Risk Assessment of retinopathy development as a long term complication of Type 1 Diabetes Mellitus", *32nd International Conference of the IEEE Engineering in Medicine and Biology Society, August 31 - September 4, Buenos Aires, Argentina, 2010*.
- [22] Valavanis, I. K., Mougiakakou, S. G., Grimaldi, K. A., & Nikita, K. S. (2010). A multifactorial analysis of obesity as CVD risk factor: use of neural network based methods in a nutrigenetics context. *BMC bioinformatics*, 11(1), 1-10.
- [23] K. Zarkogianni, A. Vazeou, S.G. Mougiakakou, A. Prountzou, K.S. Nikita, "An insulin infusion advisory system based on autotuning nonlinear model-predictive control", *IEEE Transactions on Biomedical Engineering*, vol. 58, no. 9, pp. 2467-77, 2011
- [24] K. Dalakleidi, K. Zarkogianni, V. Karamanos, K.S. Nikita, "A hybrid genetic algorithm for the selection of critical features for cardiovascular complications prognosis in type 2 diabetes patients," *13th IEEE International Conference on Bioinformatics and BioEngineering (BIBE 2013), November 11-13, 2013, Chania, Greece*.
- [25] K. Zarkogianni, E. Litsa, A. Vazaiou, K.S. Nikita, "Personalized glucose-insulin metabolism model based on self-organizing maps for patients with type 1 diabetes

mellitus," *13th IEEE International Conference on BioInformatics and BioEngineering (BIBE 2013)*, November 11-13, 2013, Chania, Greece.

[26] K. Zarkogianni, K. Mitsis, M-T.Arredondo, G. Fico, A. Fioravanti, K.S. Nikita, "Neuro-fuzzy based glucose prediction model for patients with Type 1 diabetes mellitus", *IEEE-EMBS International Conference on Biomedical and Health Informatics (BHI)*, June 1-4, 2014, Valencia, Spain.

[27] K. Zarkogianni, K. Mitsis, E. Litsa, M.T. Arredondo, G. Fico, A. Fioravanti, and K.S. Nikita, "Comparative assessment of glucose prediction models for Patients with Type 1 Diabetes Mellitus applying sensors for glucose and physical activity monitoring", *Journal of Medical & Biological Engineering & Computing*, vol. 53, no. 12, pp. 1333-1343, 2015.

[28] K. Zarkogianni, E. Litsa, K. Mitsis, P. Wu, C. Kaddi, C. Cheng, M. Wang, K.S. Nikita, "A Review of Emerging Technologies for the Management of Diabetes Mellitus," *IEEE Transactions on Biomedical Engineering*, vol. 62, no. 12, pp. 2735-2749, 2015.

[29] K. Dalakleidi, K. Zarkogianni, A. Thanopoulou, K.S. Nikita, "Comparative assessment of statistical and machine learning techniques towards estimating the risk of developing type 2 diabetes and cardiovascular complications", *Expert Systems*, 2017.

[30] K. Zarkogianni, M. Athanasiou, A.C. Thanopoulou, K.S. Nikita, "Comparison of machine learning approaches towards assessing the risk of developing Cardiovascular disease as a long-term diabetes complication", *IEEE Journal of Biomedical and Health Informatics*, September 2018.

[31] G. Fico, L. Hernandez, J. Cancela, A. Dagliati, L. Sacchi, A. Martinez-Millana, J. Posada, L. Manero, J. Verdú, A. Facchinetti, M. Ottaviano, K. Zarkogianni, K. S. Nikita, L. Groop, R. Gabriel-Sanchez, L. Chiovato, V. Traver, J. Francisco Merino-Torres, C. Cobelli, R. Bellazzi, M. T. Arredondo, "What do healthcare professionals need to turn risk models for type 2 diabetes into usable computerized clinical decision support systems? Lessons learned from the MOSAIC project", *BMC medical informatics and decision making*, vol. 19(1), December 2019.

[32] K. Zarkogianni, K. S. Nikita, "Personal health systems for diabetes management, early diagnosis and prevention", *Consumer-Driven Technologies in Healthcare: Breakthroughs in Research and Practice*, pp. 191-218, IGI Global, 2019.

[33] M. Athanasiou, K. Sfrintzeri, A. C. Thanopoulou, K. S. Nikita, "An Explainable XGBoost-Based Approach Towards Assessing the Risk of Cardiovascular Disease in Patients with Type 2 Diabetes Mellitus," , " *20th IEEE International Conference on BioInformatics and BioEngineering (BIBE 2020)*, 2020.

[34] M. Athanasiou, K. Zarkogianni, K. Karytsas and K. S. Nikita, "An LSTM-based Approach Towards Automated Meal Detection from Continuous Glucose Monitoring

in Type 1 Diabetes Mellitus," *2021 IEEE 21st International Conference on Bioinformatics and Bioengineering (BIBE)*, pp. 1-5, 2021.

[35] Athanasiou, M., Zarkogianni, K., Monou, S., Kosteria, I., Vasilakis, I., Kanaka-Gantenbein, C., & Nikita, K. (2021, June). AN INTERPRETABLE LSTM-BASED PREDICTION MODEL FOR ASSESSING THE RISK OF HOSPITALIZATION AND RE-HOSPITALIZATION IN YOUTH WITH TYPE 1 DIABETES MELLITUS. *In DIABETES TECHNOLOGY & THERAPEUTICS (Vol. 23, pp. A27-A28)*. 140 HUGUENOT STREET, 3RD FL, NEW ROCHELLE, NY 10801 USA: MARY ANN LIEBERT, INC.

[36] Zarkogianni, K., Athanasiou, M., Mitsis, K., Chatzidaki, E., Polychronaki, N., Perakis, K., ... & Nikita, K. (2021, June). A COMPREHENSIVE APPROACH TO EMPOWER SELF-MANAGEMENT OF HEALTH IN CHILDHOOD OBESITY BASED ON GAMIFICATION MECHANISMS AND BIOFEEDBACK. *In DIABETES TECHNOLOGY & THERAPEUTICS (Vol. 23, pp. A57-A58)*. 140 HUGUENOT STREET, 3RD FL, NEW ROCHELLE, NY 10801 USA: MARY ANN LIEBERT, INC

[37] K. Zarkogianni, M. Athanasiou, K. Mitsis, I. Vasilakis, I. Kosteria, K. Perakis, D. Vergeti, D. Antonopoulou, E. Papachristou, V. Chioti, A. Voutetakis, E. Kalafatis, P. Pervanidou, C. Kanaka-Gantenbein, and K. S. Nikita (2021). The Endorse pilot trial. *In Diabetes Technology & Therapeutics*, 23, A68-A68.

[38] I. A. Vasilakis, I. Kosteria, K. Mitsis, K. Zarkogianni, L. Kalafatis, M. Athanasiou, K. Perakis, G. Papavasiliou, A. Taliou, N. Nicolaidis, V. Chioti, I. Tokou, D. Vergeti, D. Antonopoulou, E. Papachristou, V. Meklis, K. S. Nikita, and C. Kanaka-Gantenbein (2022). The ENDORSE feasibility pilot trial: Assessing the implementation of serious games strategy and artificial intelligence- based telemedicine in glycemic control improvement. *In Diabetes Technology & Therapeutics*, 24(SUPPL 1): A218-A219.

[39] M. Athanasiou, "Development of interpretable machine learning models to support diabetes management", *Doctoral Thesis, National Technical University of Athens*, 2023.

[40] E. Chatzidaki, V. Chioti, L. Mourtou, G. Papavasileiou, R.-A. Kitani, E. Kalafatis, K. Mitsis, M. Athanasiou, K. Zarkogianni, K. S. Nikita, et al., "Parenting Styles and Psychosocial Factors of Mother–Child Dyads Participating in the ENDORSE Digital Weight Management Program for Children and Adolescents during the COVID-19 Pandemic", *Children* 2024, 11, 107.

[41] K. Zarkogianni, E. Chatzidaki, N. Polychronaki, E. Kalafatis, N. C. Nicolaidis, A. Voutetakis, V. Chioti, R.-A Kitani, K. Mitsis, K. Perakis, M. Athanasiou, D. Antonopoulou, P. Pervanidou, C. Kanaka-Gantenbein, and Konstantina S. Nikita (2023). The ENDORSE Feasibility Study: Exploring the Use of M-Health, Artificial Intelligence and Serious Games for the Management of Childhood Obesity. *Nutrients* 15, no. 6: 1451.

[42] M. Athanasiou, N. Myropoulou, K. Zarkogianni, K. S. Nikita, “Development of Personalized Interpretable Multilevel Prediction Models for the Risk Assessment of Hypoglycemia in Type 1 Diabetes”, *EMBC*, 2024.