



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ
ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

Εφαρμογή διαχείρισης ιατρικών δεδομένων σε έξυπνα ρολόγια WearOS

Διπλωματική Εργασία

Ευφροσύνη Χριστοδουλέα

Επιβλέπων: Παναγιώτης Τσανάκας

Καθηγητής ΕΜΠ

Αθήνα, Φεβρουάριος 2025



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ
ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

Εφαρμογή διαχείρισης ιατρικών δεδομένων σε έξυπνα ρολόγια WearOS

Διπλωματική Εργασία

Ευφροσύνη Χριστοδουλέα

Επιβλέπων: Παναγιώτης Τσανάκας

Καθηγητής ΕΜΠ

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 7^η Μαρτίου 2025.

.....

Π. Τσανάκας

Καθηγητής Ε.Μ.Π.

.....

Γ. Ματσόπουλος

Καθηγητής Ε.Μ.Π.

.....

Σ. Ξύδης

Καθηγητής Ε.Μ.Π.

Αθήνα, Φεβρουάριος 2025

Ευφροσύνη Χριστοδουλέα

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

Copyright © Ευφροσύνη Χριστοδουλέα, 2025

Με επιφύλαξη παντός δικαιώματος. All rights reserved.

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

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

Περίληψη

Η παρούσα διπλωματική εργασία αφορά την ανάπτυξη μιας εφαρμογής για έξυπνα ρολόγια που χρησιμοποιούν το λειτουργικό σύστημα WearOS, με σκοπό τη διαχείριση και ανάλυση ιατρικών δεδομένων. Η εφαρμογή σχεδιάστηκε για να λαμβάνει και να επεξεργάζεται μετρήσεις από τους ενσωματωμένους αισθητήρες του Samsung Galaxy Watch 4, εστιάζοντας στην ανάλυση ηλεκτροκαρδιογραφήματος (ECG) για την ανίχνευση αρρυθμιών και κολποκοιλιακού αποκλεισμού (AV Block). Επιπλέον, υποστηρίζει τη μέτρηση καρδιακού ρυθμού (HR), κορεσμού οξυγόνου στο αίμα (SpO₂), ανάλυσης βιοηλεκτρικής εμπέδησης (BIA) και φωτοπληθυσμογραφίας (PPG).

Η εφαρμογή λειτουργεί τοπικά στη συσκευή χωρίς την ανάγκη εξωτερικής βάσης δεδομένων, πραγματοποιώντας την ανάλυση των σημάτων σε πραγματικό χρόνο. Παράλληλα, παρέχει τη δυνατότητα αποστολής ειδοποιήσεων για υπενθύμιση μετρήσεων και στέλνει τα αποτελέσματα μέσω email στον χρήστη. Για την ανάπτυξη της εφαρμογής χρησιμοποιήθηκε η γλώσσα Java σε περιβάλλον Android Studio, ενώ οι αλγόριθμοι επεξεργασίας ECG και ανίχνευσης αρρυθμιών αναπτύχθηκαν σε Python μέσω της βιβλιοθήκης Chaquopy.

Η μελέτη επικεντρώνεται επίσης στη βελτίωση της ακρίβειας των μετρήσεων μέσω κατάλληλων φίλτρων και ανάλυσης μεταβλητότητας καρδιακού ρυθμού (HRV), ενώ προτείνεται μελλοντική ενσωμάτωση τεχνητής νοημοσύνης για πιο αξιόπιστη ανίχνευση καρδιακών διαταραχών. Συμπερασματικά, η εφαρμογή προσφέρει μια ολοκληρωμένη λύση για την προσωπική παρακολούθηση υγείας, συμβάλλοντας στην έγκαιρη διάγνωση πιθανών καρδιακών παθήσεων.

Λέξεις Κλειδιά

WearOS, ηλεκτροκαρδιογράφημα, ECG, αρρυθμίες, AV Block, καρδιακός ρυθμός, HRV, BIA, PPG, Samsung Galaxy Watch 4, Java, Python, υγεία

Abstract

This thesis focuses on the development of a medical data management application for smartwatches running WearOS. The application is designed to collect and process measurements from the built-in sensors of the Samsung Galaxy Watch 4, with a primary emphasis on analyzing electrocardiogram (ECG) signals for the detection of arrhythmias and atrioventricular (AV) block. Additionally, it supports heart rate (HR) monitoring, blood oxygen saturation (SpO₂), bioelectrical impedance analysis (BIA), and photoplethysmography (PPG).

The application operates locally on the device without requiring an external database, performing real-time signal processing. It also includes features for measurement reminders via notifications and allows users to send their results via email. The development was carried out in Java within the Android Studio environment, while ECG signal processing and arrhythmia detection algorithms were implemented in Python using the Chaquopy library.

The study also explores methods to improve measurement accuracy through signal filtering and heart rate variability (HRV) analysis, with future enhancements proposed in machine learning integration for more reliable cardiac abnormality detection. In conclusion, the application provides a comprehensive solution for personal health monitoring, contributing to the early diagnosis of potential heart conditions.

Key Words

WearOS, electrocardiogram, ECG, arrhythmias, AV Block, heart rate, HRV, BIA, PPG, Samsung Galaxy Watch 4, Java, Python, health

Ευχαριστίες

Η παρούσα διπλωματική εργασία σηματοδοτεί την ολοκλήρωση των σπουδών μου στη Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών του Εθνικού Μετσόβιου Πολυτεχνείου. Η εκπόνηση αυτής της εργασίας δεν θα ήταν εφικτή χωρίς την πολύτιμη καθοδήγηση και υποστήριξη συγκεκριμένων ανθρώπων, στους οποίους θα ήθελα να εκφράσω την ειλικρινή μου ευγνωμοσύνη. Αρχικά, οφείλω να ευχαριστήσω τον επιβλέποντα καθηγητή μου και κοσμήτορα της σχολής μου, κ. Παναγιώτη Τσανάκα, για τη συνεχή του καθοδήγηση και τη στήριξή του καθ' όλη τη διάρκεια της ερευνητικής μου προσπάθειας. Η επιστημονική του καθοδήγηση, η διάθεσή του για συζήτηση και η πολύτιμη εμπειρία του συνέβαλαν καθοριστικά στην ολοκλήρωση της εργασίας μου. Επίσης, θα ήθελα να εκφράσω τις ευχαριστίες μου στα μέλη της τριμελούς εξεταστικής επιτροπής, κ Γεώργιο Ματσόπουλο και κ. Σωτήριο Ξύδη, για τον χρόνο που διέθεσαν στην αξιολόγηση της εργασίας μου και τις χρήσιμες παρατηρήσεις τους. Ιδιαίτερες ευχαριστίες απευθύνω στον υποψήφιο διδάκτορα κ. Κωνσταντίνο Αθανασιάδη για τη συμβολή του στην πορεία της διπλωματικής μου, την πολύτιμη βοήθειά του και τις εύστοχες υποδείξεις του, που συνέβαλαν σημαντικά στην εμβάθυνση και τεχνική βελτίωση της εργασίας μου. Τέλος, ένα βαθύ και ειλικρινές ευχαριστώ στους γονείς μου, που με αμέριστη αγάπη, κατανόηση και υποστήριξη, υλική και ψυχολογική, στάθηκαν δίπλα μου όλα αυτά τα χρόνια, προσφέροντάς μου τη δύναμη και την αυτοπεποίθηση να φτάσω ως εδώ.

Στους γονείς μου

Περιεχόμενα

Περίληψη	5
Λέξεις κλειδιά	5
Abstract	7
Key words	7
Περιεχόμενα	11
Ευρετήριο Πινάκων	14
Ευρετήριο Εικόνων	14
Λίστα Ακρωνυμίων	16
1. Galaxy Watch και εργαλεία ανάπτυξης	18
1.1 Το έξυπνο ρολόι Galaxy Watch 4	19
1.1.1 Γενικά	19
1.1.2 Αισθητήρες	19
1.1.3 Τεχνικά χαρακτηριστικά	20
1.2 Java	20
1.3 Android Studio	20
1.4 Python	21
1.5 Samsung Health Sensor SDK	21
1.5.1 Γενικά	21
1.5.2 Δυνατότητες του Samsung Health Sensor SDK	22
1.5.3 Περιορισμοί	22
1.6 Wear	23
1.6.1 Γενικά	23
1.6.2 Λειτουργίες και δυνατότητες	23
1.6.3 Πλεονεκτήματα	24
1.6.4 Μειονεκτήματα	24
1.7 Activity	24
1.7.1 Γενικά	24
1.7.2 Λειτουργίες και δυνατότητες	25
1.7.3 Μειονεκτήματα	25
1.8 AppCompat	25
1.8.1 Γενικά	25
1.8.2 Λειτουργίες και δυνατότητες	26
1.8.3 Μειονεκτήματα	26
1.9 Material	26
1.9.1 Γενικά	26
1.9.2 Λειτουργίες και δυνατότητες	27
1.9.3 Μειονεκτήματα	27

1.10 ConstraintLayout	27
1.10.1 Γενικά	27
1.10.2 Λειτουργίες και δυνατότητες	28
1.10.3 Πλεονεκτήματα	28
1.10.4 Μειονεκτήματα	28
1.11 Google Play Services – Wearable API	29
1.11.1 Γενικά	29
1.11.2 Λειτουργίες και δυνατότητες	29
1.11.3 Πλεονεκτήματα	29
1.11.4 Μειονεκτήματα	30
1.12 Apache Commons Math3	30
1.12.1 Γενικά	30
1.12.2 Λειτουργίες και δυνατότητες	30
1.12.3 Πλεονεκτήματα	31
1.12.4 Μειονεκτήματα	31
1.13 JDSP	31
1.13.1 Γενικά	31
1.13.2 Λειτουργίες και δυνατότητες	31
1.13.3 Πλεονεκτήματα	32
1.13.4 Μειονεκτήματα	33
1.14 Chaquopy	33
1.14.1 Γενικά	33
1.14.2 Λειτουργίες - Μειονεκτήματα	33
1.15 RecyclerView	33
1.15.1 Γενικά	33
1.15.2 Λειτουργίες και δυνατότητες	34
1.15.3 Πλεονεκτήματα	34
1.15.4 Μειονεκτήματα	35
1.16 SendGrid Java	35
1.16.1 Γενικά	35
1.16.2 Λειτουργίες και δυνατότητες	35
1.16.3 Πλεονεκτήματα	36
1.16.4 Μειονεκτήματα	36
1.17 Apache HttpClient	36
1.17.1 Γενικά	36
1.17.2 Λειτουργίες και δυνατότητες	37
1.17.3 Πλεονεκτήματα	37
1.17.4 Μειονεκτήματα	37
2. Η σχέση της εφαρμογής με την υγεία	39
2.1 Εισαγωγή	40
2.2 Καρδιακοί παλμοί, Ηλεκτροκαρδιογράφημα (ECG) και Διακύμανση καρδιακής συχνότητας (HRV)	40
2.2.1 Γενικά	40
2.2.2 Παθήσεις της Καρδιάς	41

2.2.3 Αρρυθμίες	43
2.2.3.1 Γενικά	43
2.2.3.2 Τύποι Αρρυθμιών.....	43
2.2.3.3 Ανίχνευση.....	44
2.2.3.4 Συμπτώματα και Διαχείριση	44
2.2.4 AV Block.....	45
2.2.4.1 Γενικά	45
2.2.4.2 Τύποι AV Block	45
2.2.4.3 Ανίχνευση.....	46
2.2.4.4 Συμπτώματα και Διαχείριση	46
2.3 Υπολογισμός Αρρυθμιών από Single-Channel ECG	46
2.3.1 Γενικά	46
2.3.2 Διαδικασία Ανάλυσης Αρρυθμιών σε Single-Channel ECG.....	46
2.3.2.1 Εντοπισμός του R-Peak και Υπολογισμός του RR-Interval.....	46
2.3.2.2 Ανίχνευση Ανώμαλων Καρδιακών Ρυθμών	47
2.3.3 Περιορισμοί του Single-Channel ECG	48
2.4 Εντοπισμός AV Block από Single-Channel ECG	48
2.4.1 Γενικά	48
2.4.2 Βήματα εντοπισμού	48
2.5 Οξυγόνο στο αίμα	50
2.5.1 Γενικά	50
2.5.2 Υποδειγματικές τιμές.....	51
2.5.3 Έλλειψη οξυγόνου στο αίμα	51
2.6 Ανάλυση Βιοηλεκτρικής Εμπέδησης (BIA)	52
2.6.1 Γενικά	52
2.6.2 Βασικές αρχές λειτουργίας της BIA	52
2.6.3 Χρήσεις της BIA	52
2.6.4 Παράγοντες που επηρεάζουν την ακρίβεια της BIA	53
2.7 Φωτοπληθυσμογραφία (PPG).....	53
2.7.1 Γενικά	53
2.7.2 Βασικές αρχές λειτουργίας της PPG	53
2.7.3 Χρήσεις της PPG.....	53
2.7.4 Πλεονεκτήματα της PPG	54
2.7.5 Παράγοντες που επηρεάζουν την ακρίβεια της PPG	54
3. Δομή της εφαρμογής και μεθοδολογία υλοποίησής της	55
3.1 Εισαγωγή.....	56
3.2 Ευέλικτη μεθοδολογία (Agile)	56
3.3 Δομή της εφαρμογής	57
3.4 Επικοινωνία εφαρμογής με το Samsung Health.....	58
3.5 Κόστος ανάπτυξης της διπλωματικής εργασίας	58
4. Παρουσίαση και ερμηνεία κώδικα.....	60

4.1 Εισαγωγή.....	61
4.2 Εφαρμογή στο smart watch	61
4.2.1 Άδεια για ειδοποιήσεις, καταγραφή μετρήσεων	61
4.2.2 Περιήγηση.....	63
4.2.3 Δημιουργία ειδοποιήσεων	64
4.2.4 Μέτρηση Ηλεκτροκαρδιογραφήματος (ECG) και Ανάλυση Καρδιακών Διαταραχών.....	68
4.2.5 Λήψη μετρήσεων Καρδιακού ρυθμού (HR) και Κορεσμού Οξυγόνου (SpO2)	80
4.2.6 Λήψη μετρήσεων BIA.....	86
4.2.7 Μέτρηση Φωτοπληθυσμογραφίας (PPG) – Πράσινο, Κόκκινο και Υπέρυθρο Φως	87
4.3 Συμπεράσματα και Ιδέες για αναβαθμίσεις της εφαρμογής	88
4.3.1 Τρέχουσα κατάσταση και δυνατότητες της εφαρμογής	88
4.3.2 Βελτίωση ακρίβειας και ανάλυσης δεδομένων	88
4.3.3 Επέκταση λειτουργιών υγείας και χρήση πρόσθετων αισθητήρων	89
4.3.4 Βελτίωση εμπειρίας χρήστη και διαδραστικότητας	89
4.3.5 Διασύνδεση με Cloud και Wearables	89
Βιβλιογραφία	90

Ευρετήριο πινάκων

Πίνακας 1. Τύποι AV Block και χαρακτηριστικά τους	49
---	----

Ευρετήριο εικόνων

Εικόνα 1. Οι αισθητήρες του Galaxy Watch 4	19
Εικόνα 2. Το περιβάλλον του Android Studio	21
Εικόνα 3. Παράδειγμα χρήσης του FFT για ανάλυση ενός σήματος	32
Εικόνα 4. Ανάλυση του συμπλέγματος QRS σε ηλεκτροκαρδιογράφημα	49
Εικόνα 5. Πρώτου βαθμού AV Block.....	49
Εικόνα 6. Δεύτερου βαθμού Mobitz I (Wenckebach) AV Block.....	50
Εικόνα 7. Δεύτερου βαθμού Mobitz II AV Block	50
Εικόνα 8. Δευτέρου βαθμού (High Grade) AV Block	50
Εικόνα 9. Τρίτου βαθμού AV Block	50
Εικόνα 10. Ευέλικτη μεθοδολογία	56
Εικόνα 11. Πλάνο κοστολόγησης SendGrid	59
Εικόνα 12. Κώδικας για ερώτηση άδειας Body Sensors	61
Εικόνα 13. Κώδικας για ερώτηση άδειας ειδοποιήσεων	61
Εικόνα 14. Κώδικας για διαχείριση ειδοποιήσεων	62
Εικόνα 15. Permissions στο AndroidManifest.xml.....	62
Εικόνα 16. Ερώτηση άδειας ειδοποιήσεων	62
Εικόνα 17. Κουμπιά αποδοχής απόρριψης άδειας	62

Εικόνα 18. Ερώτηση άδειας αισθητήρων	62
Εικόνα 19. Διάφορες οθόνες τις εφαρμογής, κομμάτι κώδικα από AndroidManifest.xml..	63
Εικόνα 20. Κεντρική οθόνη, μενού της εφαρμογής	63
Εικόνα 21. Κώδικας για μετάβαση σε νέα δραστηριότητα μέσω κουμπιού	64
Εικόνα 22. Ειδοποίηση.....	64
Εικόνα 23. Εσωτερικό ειδοποίησης με επιλογή ανοίγματος της εφαρμογής	64
Εικόνα 24. Δημιουργία καναλιού ειδοποιήσεων, έλεγχος permissions και προγραμματισμός ειδοποιήσεων	65
Εικόνα 25. Κώδικας για υπολογισμό κοντινότερης προγραμματισμένης ειδοποίησης	65
Εικόνα 26. Προγραμματίζει ειδοποίηση, ακόμα και σε κατάσταση εξοικονόμησης ενέργειας	66
Εικόνα 27. Δημιουργία ειδοποίησης και εμφάνιση με ήχο, δόνηση και κουμπί για άνοιγμα της εφαρμογής.....	67
Εικόνα 28. Δημιουργία καναλιού ειδοποιήσεων με δόνηση, ήχο και προβολή στην οθόνη κλειδώματος.....	67
Εικόνα 29. Σύνδεση με HealthTrackingService.....	68
Εικόνα 30. Προσθήκη Tracker και αρχικοποίηση EcgListener	68
Εικόνα 31. Αρχικοποίηση ECG Tracker	68
Εικόνα 32. Ανάκτηση και αποθήκευση δεδομένων ECG	69
Εικόνα 33. Οθόνη μετρήσεων ECG και live εμφάνιση τιμών	69
Εικόνα 34. Οθόνες αποτελεσμάτων και κουμπιών για αποστολή αποτελεσμάτων μέσω email και για επιστροφή στο αρχικό μενού	70
Εικόνα 35. Email με τα αποτελέσματα που στέλνεται μέσω της εφαρμογής στον χρήστη .	70
Εικόνα 36. Αποθήκευση και κλείσιμο αρχείου για δεδομένα ECG	71
Εικόνα 37. Αρχικοποίηση περιβάλλοντος Python στο Android	71
Εικόνα 38. Εκτέλεση Python Script για επεξεργασία ECG σε ξεχωριστό thread	71
Εικόνα 39. Φιλτράρισμα, επεξεργασία και οπτικοποίηση ECG σήματος	72
Εικόνα 40. Έλεγχος αρρυθμίας μέσω Python Script.....	73
Εικόνα 41. Επεξεργασία ECG και υπολογισμός καρδιακού ρυθμού	74
Εικόνα 42. Ανίχνευση αρρυθμιών από καρδιακό ρυθμό και RR-διαστήματα	75
Εικόνα 43. Οπτικοποίηση ECG σήματος και επισήμανση R-peaks	75
Εικόνα 44. Φιλτράρισμα και προεπεξεργασία σήματος ECG	76
Εικόνα 45. Ανίχνευση και ταξινόμηση AV Block	76
Εικόνα 46. Οπτικοποίηση σήματος ECG και PR διαστημάτων.....	77
Εικόνα 47. Αποστολή αποτελεσμάτων ECG μέσω email με συνημμένα γραφήματα	78
Εικόνα 48. Υλοποίηση αποστολής email με συνημμένα μέσω SendGrid σε Java.....	79
Εικόνα 49. Οθόνες μέτρησης HR και SpO2	80
Εικόνα 50. Αρχικοποίηση και παρακολούθηση SpO2 και καρδιακού ρυθμού	80
Εικόνα 51. Έναρξη και διακοπή μέτρησης SpO2.....	81
Εικόνα 52. Ενημέρωση UI με δεδομένα HR.....	81
Εικόνα 53. Ενημέρωση κατάστασης και UI για SpO2 μέτρηση	82
Εικόνα 54. Ενημέρωση δεδομένων SpO2 και ειδοποίηση trackers.....	83
Εικόνα 55. Ανάγνωση και επεξεργασία δεδομένων HR	83
Εικόνα 56. Ενημέρωση του UI με τα δεδομένα HR	84

Εικόνα 57. Ενημέρωση κατάστασης μετρητή HR	85
Εικόνα 58. Οθόνη μετρήσεων BIA.....	86
Εικόνα 59. Οθόνη μέτρησης PPG	87

Λίστα Ακρωνυμίων

ECG:	Electrocardiogram
HRV:	Heart Rate Variability
HR:	Heart Rate
PPG:	Photoplethysmography
SpO ₂ :	Oxygen Saturation
BIA:	Bioelectrical Impedance Analysis
SDK:	Software Development Kit
API:	Application Programming Interface
AV Block:	Atrioventricular Block)
JDSP:	Java Digital Signal Processing
JVM:	Java Virtual Machine
GPS:	Global Positioning System
Wi-Fi:	Wireless Fidelity
LTE:	Long Term Evolution
IP:	Internet Protocol
SMTP:	Simple Mail Transfer Protocol
BMR:	Basal Metabolic Rate
FFM:	Fat-Free Mass
ICD:	Implantable Cardioverter Defibrillator
NTLM:	Microsoft NT LAN Manager
SSL/TLS:	Secure Sockets Layer / Transport Layer Security
REST API:	Representational State Transfer API
UI:	User Interface
IDE:	Integrated Development Environment
XML:	Extensible Markup Language
JSON:	JavaScript Object Notation
HTTP:	HyperText Transfer Protocol
MVP:	Model-View-Presenter
AI:	Artificial Intelligence
AM:	Amplitude Modulation
AMOLED:	Active-Matrix Organic Light-Emitting Diode
MIME:	Multipurpose Internet Mail Extensions

Κεφάλαιο 1. Galaxy Watch και εργαλεία ανάπτυξης

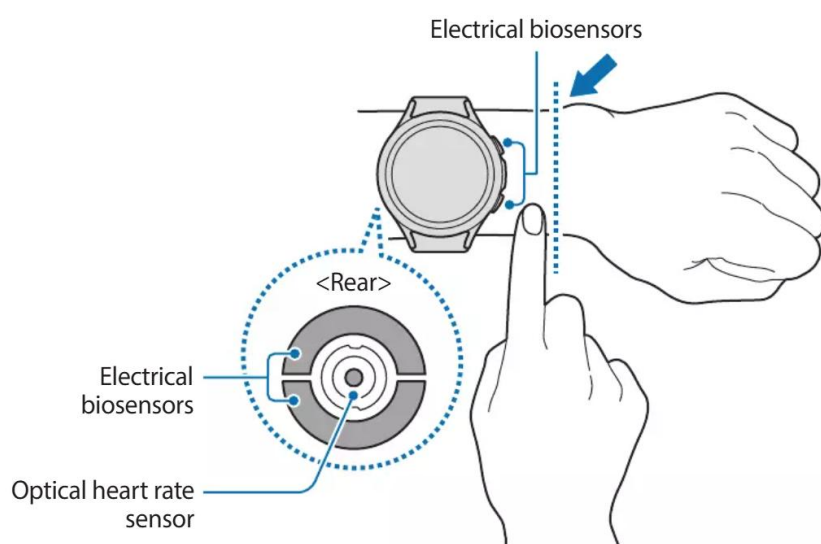
1.1 Το έξυπνο ρολόι Galaxy Watch 4

1.1.1 Γενικά

Το Samsung Galaxy Watch 4 είναι ένα προηγμένο έξυπνο ρολόι που αναπτύχθηκε από τη Samsung και αποτελεί ένα από τα πρώτα μοντέλα της εταιρείας που ενσωματώνει το λειτουργικό σύστημα Wear OS, σε συνεργασία με τη Google. Είναι το πρώτο ρολόι της στο οποίο η Samsung ενσωμάτωσε τον αισθητήρα οξυγόνου στο αίμα και τον αισθητήρα βιοηλεκτρικής εμπέδησης. Για αυτόν τον λόγο, η ανάπτυξη της εφαρμογής βασίστηκε σε αυτό το ρολόι, ώστε να μην αποκλειστεί η συμβατότητα κυρίως με τα μεταγενέστερα μοντέλα της σειράς Galaxy Watch. Το Galaxy Watch 4 επιλέχθηκε ύστερα από συγκριτική έρευνα με άλλα smartwatches, καθώς προσφέρει υψηλή ακρίβεια στις μετρήσεις του, ενώ διαθέτει προηγμένους αισθητήρες για ηλεκτροκαρδιογράφημα (ECG), οξυγόνο στο αίμα (SpO_2), βιοηλεκτρική εμπέδηση (BIA), παλμούς (PPG) και άλλες κρίσιμες παραμέτρους υγείας. Επιπλέον, με τη διαρκή υποστήριξη της Samsung μέσω ενημερώσεων λογισμικού, προστίθενται συνεχώς νέες λειτουργίες και βελτιώσεις, ενισχύοντας την ακρίβεια και τις δυνατότητες του ρολογιού στον τομέα της υγείας.

1.1.2 Αισθητήρες

Το Samsung Galaxy Watch 4 διαθέτει μια σειρά από προηγμένους αισθητήρες για την παρακολούθηση της υγείας και της φυσικής κατάστασης. Περιλαμβάνει τον BioActive Sensor, που συνδυάζει οπτικό και ηλεκτρικό αισθητήρα καρδιακών παλμών, καθώς και αισθητήρα βιοηλεκτρικής εμπέδησης για τη μέτρηση της σύστασης σώματος. Επιπλέον, είναι εξοπλισμένο με αξελερόμετρο, γυροσκόπιο, βαρόμετρο, αισθητήρα φωτός, GPS και μαγνητόμετρο, παρέχοντας ολοκληρωμένη καταγραφή δραστηριοτήτων και δεδομένων υγείας. [1]



Εικόνα 1. Οι αισθητήρες του Galaxy Watch 4

1.1.3 Τεχνικά χαρακτηριστικά

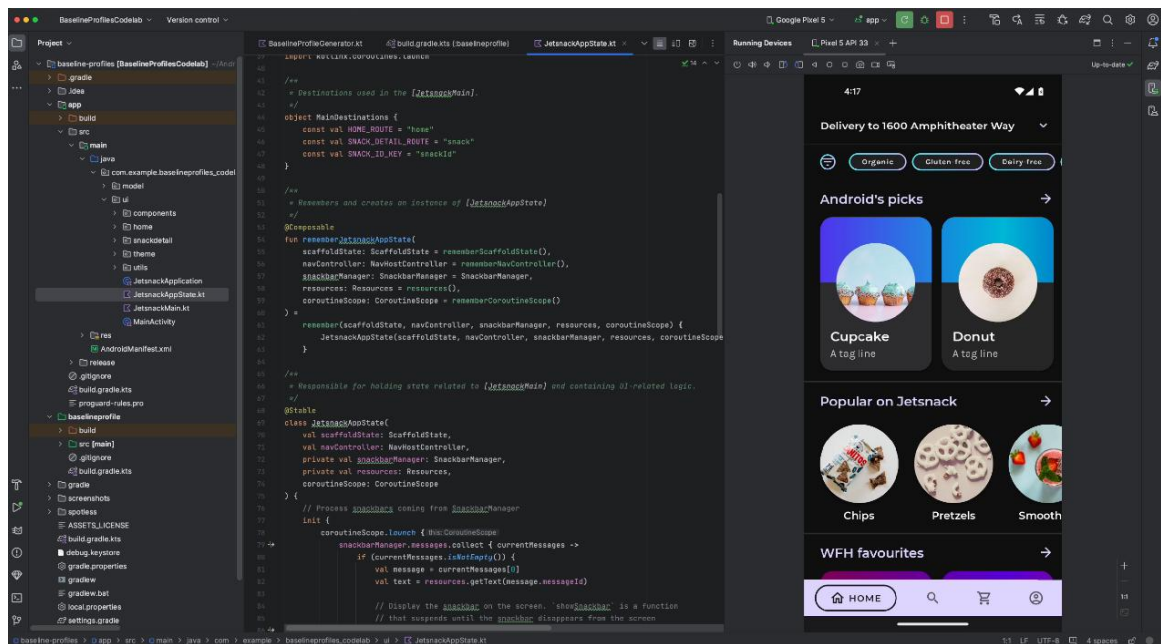
Το Samsung Galaxy Watch 4 διαθέτει τον διπύρνηνο επεξεργαστή Exynos W920 στα 1.18GHz, συνοδευόμενο από 1.5GB RAM και 16GB εσωτερικής μνήμης. Υποστηρίζει Bluetooth 5.0, Wi-Fi 802.11 a/b/g/n 2.4 & 5GHz, GPS, GLONASS, Galileo, Beidou, καθώς και LTE στη συμβατή έκδοση με eSIM. Διαθέτει οθόνη Super AMOLED ανάλυσης 396x396 (40mm) ή 450x450 (44mm) με δυνατότητα Always-On Display και προστασία Gorilla Glass DX+. Είναι ανθεκτικό σε νερό και σκόνη με πιστοποιήσεις 5ATM, IP68 και MIL-STD-810G, ενώ η μπαταρία του φτάνει τις 247mAh (40mm) ή 361mAh (44mm), προσφέροντας αυτονομία 1-2 ημερών.

1.2 Java

Η Java είναι μια αντικειμενοστρεφής γλώσσα προγραμματισμού γενικής χρήσης, ανεπτυγμένη από την Sun Microsystems και που αργότερα οικειοποιήθηκε από την Oracle Corporation. Κυκλοφόρησε για πρώτη φορά το 1995 και είναι σχεδιασμένη με γνώμονα τη φορητότητα, επιτρέποντας την εκτέλεση κώδικα σε διαφορετικές πλατφόρμες μέσω της Java Virtual Machine (JVM). Χρησιμοποιείται ευρέως στην ανάπτυξη εφαρμογών για επιχειρησιακά συστήματα, κινητές συσκευές (Android), web εφαρμογές και διακομιστές. Μια από τις βασικές αρχές της Java είναι το "Write Once, Run Anywhere" (WORA), που σημαίνει ότι ο κώδικας μπορεί να τρέξει σε οποιοδήποτε σύστημα διαθέτει συμβατή JVM, χωρίς να απαιτούνται τροποποιήσεις. Επιπλέον, η Java περιλαμβάνει μια πλούσια βιβλιοθήκη (Java Standard Library) και υποστηρίζει πολλαπλά frameworks, όπως το Spring, Hibernate και Java EE, διευκολύνοντας την ανάπτυξη εφαρμογών μεγάλης κλίμακας. Χρησιμοποιείται επίσης στην ανάπτυξη Big Data συστημάτων, τεχνητής νοημοσύνης και εφαρμογών cloud, ενώ συνεχίζει να εξελίσσεται βελτιώνοντας την απόδοση και προσθέτοντας συνεχώς σύγχρονα χαρακτηριστικά. [2]

1.3 Android Studio

Το Android Studio είναι το επίσημο Ολοκληρωμένο Περιβάλλον Ανάπτυξης (IDE) της Google για την ανάπτυξη εφαρμογών Android. Βασισμένο στο IntelliJ IDEA της JetBrains, προσφέρει μια σειρά από εργαλεία για τη δημιουργία, τη δοκιμή και τη διαχείριση εφαρμογών για συσκευές με λειτουργικό Android. Κυκλοφόρησε για πρώτη φορά το 2013, αντικαθιστώντας το Eclipse ως το προεπιλεγμένο IDE για Android ανάπτυξη. Το Android Studio παρέχει συμπληρωματική σύνταξη (code completion), debugging εργαλεία, έναν ισχυρό emulator, υποστήριξη για Kotlin, Java και C++, καθώς και ένα σύστημα διαχείρισης εξαρτήσεων με το Gradle. Η τελευταία σταθερή έκδοσή του περιλαμβάνει βελτιώσεις στην απόδοση και νέα χαρακτηριστικά όπως Live Edit και Jetpack Compose integration, διευκολύνοντας την ταχύτερη και πιο αποδοτική ανάπτυξη εφαρμογών. Το Android Studio είναι διαθέσιμο για Windows, macOS και Linux, επιτρέποντας στους προγραμματιστές να αναπτύξουν εφαρμογές για smartphones, tablets, wearables και Android TV. [3]



Εικόνα 2. Το περιβάλλον του Android Studio

1.4 Python

Η Python είναι μια από τις πιο δημοφιλείς και ευέλικτες γλώσσες προγραμματισμού, γνωστή για την καθαρή σύνταξή της και την απλότητα στη χρήση της. Χρησιμοποιείται σε ένα ευρύ φάσμα εφαρμογών, από ανάπτυξη λογισμικού και ανάλυση δεδομένων μέχρι τεχνητή νοημοσύνη, αυτοματισμούς συστημάτων και ανάπτυξη web εφαρμογών. Η Python είναι γλώσσα υψηλού επιπέδου και δυναμικής τυποποίησης, γεγονός που την καθιστά ιδιαίτερα εύκολη στην εκμάθηση και ιδανική για αρχάριους αλλά και έμπειρους προγραμματιστές. Επιπλέον, διαθέτει μια τεράστια κοινότητα χρηστών και μια πληθώρα βιβλιοθηκών (όπως NumPy, Pandas, TensorFlow, Flask, Django) που καλύπτουν σχεδόν κάθε ανάγκη στον χώρο της πληροφορικής. Η σύνταξή της είναι εμπνευσμένη από την απλότητα και την αναγνωσιμότητα, γεγονός που επιτρέπει τη γρήγορη ανάπτυξη και διατήρηση του κώδικα. Χρησιμοποιείται τόσο από μεγάλες εταιρείες τεχνολογίας (Google, Facebook, Netflix) όσο και από ερευνητές και αναλυτές δεδομένων. Στη συγκεκριμένη εργασία, η Python αξιοποιήθηκε για την ανάλυση και επεξεργασία σήματος και δεδομένων υγείας, την αυτοματοποίηση διαδικασιών και την επεξεργασία και αποθήκευση αρχείων, προσφέροντας ευελιξία και αξιοπιστία στον σχεδιασμό και την υλοποίηση της εφαρμογής. [4]

1.5 Samsung Health Sensor SDK

1.5.1 Γενικά

Το Samsung Health Sensor SDK είναι ένα σύνολο εργαλείων και προγραμματιστικών διεπαφών (APIs) που επιτρέπει την πρόσβαση και διαχείριση δεδομένων υγείας που

συλλέγονται από τις συσκευές της Samsung, όπως τα Galaxy Watch και τα smartphones της εταιρείας. Το SDK αυτό έχει σχεδιαστεί για να δώσει στους προγραμματιστές τη δυνατότητα να δημιουργούν εφαρμογές υγείας που αξιοποιούν τους ενσωματωμένους αισθητήρες, χωρίς να παραβιάζεται το απόρρητο του χρήστη. Μέσω του Samsung Health Sensor SDK, οι εφαρμογές μπορούν να λαμβάνουν δεδομένα από το ηλεκτροκαρδιογράφημα (ECG), τους παλμούς της καρδιάς (PPG), την οξυμετρία (SpO₂), τη βιοηλεκτρική εμπέδηση (BIA), καθώς και άλλους αισθητήρες που βρίσκονται στα Galaxy Watch. Η Samsung, με το SDK αυτό, προσφέρει έναν ασφαλή και αξιόπιστο τρόπο συλλογής δεδομένων σε πραγματικό χρόνο, διατηρώντας τον πλήρη έλεγχο στον χρήστη σχετικά με το ποια δεδομένα θέλει να διαμοιράζεται και σε ποιες εφαρμογές. [5]

1.5.2 Δυνατότητες του Samsung Health Sensor SDK

Το Samsung Health Sensor SDK προσφέρει πληθώρα δυνατοτήτων, επιτρέποντας την ανάπτυξη εφαρμογών που επεξεργάζονται δεδομένα υγείας απευθείας στη συσκευή του χρήστη. Οι σημαντικότερες λειτουργίες περιλαμβάνουν:

- Άμεση πρόσβαση στους αισθητήρες του Galaxy Watch: Το SDK επιτρέπει την καταγραφή και ανάλυση δεδομένων από αισθητήρες, όπως το ECG, το PPG, το SpO₂, την πίεση αίματος και την ανάλυση σύστασης σώματος (BIA), χωρίς την ανάγκη και τη δέσμευση που απαιτούν οι εξωτερικοί servers.
- Αποθήκευση και ανάλυση δεδομένων σε πραγματικό χρόνο: Σε αντίθεση με κάποιες πλατφόρμες που απαιτούν σύνδεση cloud, το Samsung Health Sensor SDK δίνει τη δυνατότητα στις εφαρμογές να επεξεργάζονται τα δεδομένα απευθείας στη συσκευή, επιτρέποντας ακόμα και την ανίχνευση αρρυθμιών ή αποκλεισμών του καρδιακού ρυθμού χωρίς να απαιτείται σύνδεση στο διαδίκτυο.
- Προστασία απορρήτου και έλεγχος χρήστη: Ο χρήστης μπορεί να επιλέξει ποιες εφαρμογές έχουν πρόσβαση στα δεδομένα του, ενώ όλες οι εφαρμογές που χρησιμοποιούν το SDK πρέπει να ζητούν ρητή άδεια για κάθε τύπο δεδομένων που επιθυμούν να χρησιμοποιήσουν.
- Υποστήριξη για ειδοποιήσεις υγείας: Οι εφαρμογές μπορούν να αξιοποιήσουν τις δυνατότητες των αισθητήρων για να στέλνουν ειδοποιήσεις στο χρήστη σε περίπτωση που εντοπιστούν ανωμαλίες, όπως ένας ακανόνιστος καρδιακός ρυθμός ή ένα πολύ χαμηλό επίπεδο οξυγόνου στο αίμα του.

Η δυνατότητα χρήσης του SDK σε Wear OS-based smartwatches της Samsung, σε συνδυασμό με τα πλούσια δεδομένα που μπορεί να συλλέξει, καθιστά το Samsung Health Sensor SDK ιδιαίτερα χρήσιμο για εφαρμογές που επικεντρώνονται στη συνεχή παρακολούθηση της υγείας του χρήστη.

1.5.3 Περιορισμοί

Παρόλο που το Samsung Health Sensor SDK παρέχει εκτεταμένες δυνατότητες για τη διαχείριση δεδομένων υγείας, υπάρχουν ορισμένοι περιορισμοί που πρέπει να ληφθούν υπόψη:

- Διαθεσιμότητα μόνο σε συσκευές Samsung: Παρότι το SDK βασίζεται στο Wear OS, η πλήρης υποστήριξη και η ακριβής μέτρηση των δεδομένων είναι διαθέσιμη μόνο σε συγκεκριμένα Galaxy Watch, γεγονός που περιορίζει τη χρήση του σε άλλες Wear OS συσκευές.
- Απουσία cloud συγχρονισμού από προεπιλογή: Το Samsung Health Sensor SDK δεν διαθέτει ενσωματωμένη αποθήκευση στο cloud, όπως άλλες πλατφόρμες (π.χ. HealthKit της Apple). Αν οι εφαρμογές χρειάζονται αποθήκευση δεδομένων για μεγαλύτερο χρονικό διάστημα, οι προγραμματιστές πρέπει να υλοποιήσουν δικές τους λύσεις αποθήκευσης.
- Αδυναμία πρόσβασης σε δεδομένα στο background: Όπως και άλλες πλατφόρμες, το SDK δεν επιτρέπει συνεχή πρόσβαση σε δεδομένα όταν η συσκευή είναι κλειδωμένη για λόγους ασφαλείας. Ο χρήστης πρέπει να ανοίξει την εφαρμογή για να ξεκινήσει η συλλογή δεδομένων.
- Περιορισμοί στην επεξεργασία ευαίσθητων δεδομένων: Τα δεδομένα όπως το ECG και το PPG υπόκεινται σε αυστηρούς κανονισμούς ασφαλείας, και ορισμένες λειτουργίες ενδέχεται να μην είναι διαθέσιμες σε συγκεκριμένες περιοχές, καθώς η χρήση ιατρικών δεδομένων υπόκειται σε διαφορετικές ρυθμίσεις ανάλογα με τη χώρα.

Παρά τους περιορισμούς αυτούς, το Samsung Health Sensor SDK αποτελεί μια ισχυρή λύση για εφαρμογές που βασίζονται στην ανάλυση βιομετρικών δεδομένων, προσφέροντας υψηλή ακρίβεια και δυνατότητες real-time επεξεργασίας απευθείας στο smartwatch, χωρίς την ανάγκη εξωτερικών υποδομών.

1.6 Wear

1.6.1 Γενικά

Η βιβλιοθήκη Wear της Google αποτελεί ένα σύνολο εργαλείων και APIs που επιτρέπει την ανάπτυξη εφαρμογών για Wear OS, το λειτουργικό σύστημα της Google για έξυπνα ρολόγια. Υποστηρίζει ανάπτυξη σε Java και Kotlin μέσα από το Android Studio, παρέχοντας στους προγραμματιστές ένα εύχρηστο περιβάλλον για τη δημιουργία διαδραστικών και αποδοτικών εφαρμογών. Η βιβλιοθήκη επιτρέπει την επικοινωνία μεταξύ smartwatch και κινητού τηλεφώνου, τη χρήση αισθητήρων υγείας, την καταγραφή φυσικής δραστηριότητας και τη δημιουργία προσαρμοσμένων διεπαφών χρήστη. [6]

1.6.2 Λειτουργίες και δυνατότητες

Η βιβλιοθήκη Wear προσφέρει μια σειρά από λειτουργίες που επιτρέπουν στους προγραμματιστές να δημιουργήσουν πλούσιες εμπειρίες χρήσης. Μέσω του Data Layer API, επιτυγχάνεται η αποστολή δεδομένων μεταξύ του smartwatch και του κινητού τηλεφώνου, καθιστώντας δυνατή τη συγχρονισμένη ανταλλαγή πληροφοριών, όπως καρδιακοί παλμοί και βήματα. Η Watch Face API επιτρέπει τη δημιουργία προσαρμοσμένων καντράν, ενώ η Complications API προσθέτει διαδραστικά στοιχεία που παρέχουν χρήσιμες πληροφορίες απευθείας στο ρολόι. Παράλληλα, η Messaging API επιτρέπει την ανταλλαγή μηνυμάτων

μεταξύ των συσκευών. Ιδιαίτερα σημαντική είναι η ενσωμάτωση του Health Services API, το οποίο επιτρέπει τη συλλογή και ανάλυση δεδομένων υγείας με ενεργειακά αποδοτικό τρόπο, βελτιώνοντας τη διάρκεια ζωής της μπαταρίας. Επιπλέον, οι εφαρμογές Wear OS μπορούν να αξιοποιήσουν gesture detection, παρέχοντας σύγχρονους εναλλακτικούς τρόπους αλληλεπίδρασης μέσω κινήσεων του χεριού.

1.6.3 Πλεονεκτήματα

Ένα από τα βασικά πλεονεκτήματα της βιβλιοθήκης Wear είναι η άμεση ενσωμάτωσή της στο Android οικοσύστημα, επιτρέποντας στις εφαρμογές smartwatch να συνεργάζονται με υπάρχουσες εφαρμογές στο κινητό. Η χρήση του Health Services API συμβάλλει στη βελτιστοποίηση της κατανάλωσης ενέργειας, καθώς μειώνει τον φόρτο επεξεργασίας των δεδομένων. Παράλληλα, η βιβλιοθήκη υποστηρίζει σύγχρονες αρχιτεκτονικές ανάπτυξης, όπως MVVM και Jetpack Compose, διευκολύνοντας την ανάπτυξη καλά δομημένων και εύκολα διαχειρίσιμων εφαρμογών. Η ύπαρξη του Wear OS Emulator στο Android Studio επιτρέπει στους προγραμματιστές να δοκιμάζουν τις εφαρμογές τους χωρίς την ανάγκη φυσικής συσκευής, επιταχύνοντας τη διαδικασία ανάπτυξης και εντοπισμού σφαλμάτων.

1.6.4 Μειονεκτήματα

Παρά τα σημαντικά πλεονεκτήματα, η ανάπτυξη εφαρμογών με τη βιβλιοθήκη Wear αντιμετωπίζει ορισμένες προκλήσεις. Το κυριότερο μειονέκτημα είναι η περιορισμένη διάρκεια ζωής της μπαταρίας των περισσότερων smartwatches, γεγονός που απαιτεί ιδιαίτερη προσοχή στη βελτιστοποίηση των εφαρμογών. Επίσης, οι μικρές οθόνες των wearables περιορίζουν την πολυπλοκότητα του UI, απαιτώντας την ανάπτυξη μινιμαλιστικών και λειτουργικών διεπαφών. Επιπλέον, το οικοσύστημα Wear OS είναι λιγότερο διαδεδομένο σε σχέση με το Android για smartphones, γεγονός που μπορεί να περιορίσει τη διείσδυση των εφαρμογών στην αγορά. Δε μπορούμε να αγνοήσουμε πως παρά την παρουσία emulator δεν γίνονται δοκιμές σε χαρακτηριστικά των εφαρμογών που χρησιμοποιούν τους αισθητήρες, απαιτώντας την κατοχή συσκευής. Τέλος, ορισμένοι κατασκευαστές χρησιμοποιούν τροποποιημένες εκδόσεις του Wear OS, προκαλώντας ζητήματα συμβατότητας και απαιτώντας επιπλέον προσαρμογές στις εφαρμογές.

1.7 Activity

1.7.1 Γενικά

Το Activity είναι μία από τις βασικότερες κλάσεις στο Android Framework, καθώς αντιπροσωπεύει μία οθόνη διεπαφής χρήστη (UI) σε μια εφαρμογή. Κάθε Android εφαρμογή αποτελείται συνήθως από πολλαπλά Activities, τα οποία επιτρέπουν στους χρήστες να αλληλεπιδρούν με την εφαρμογή. Το Activity διαχειρίζεται τον κύκλο ζωής του μέσω μιας σειράς callback μεθόδων (onCreate(), onStart(), onResume(), κ.λπ.), οι οποίες βοηθούν στη διαχείριση της κατάστασης της εφαρμογής και στη βελτιστοποίηση της απόδοσής της. [7]

1.7.2 Λειτουργίες και δυνατότητες

Το Activity παρέχει τις βασικές δυνατότητες για την εμφάνιση UI και τη διαχείριση της αλληλεπίδρασης του χρήστη. Ένα Activity μπορεί να περιέχει διαφορετικές διατάξεις (layouts) και στοιχεία UI, επιτρέποντας στους χρήστες να εκτελούν ενέργειες, να εισάγουν δεδομένα και να πλοηγούνται σε άλλες οθόνες. Η Intent System του Android επιτρέπει την επικοινωνία μεταξύ διαφορετικών Activities, τόσο μέσα στην ίδια εφαρμογή όσο και μεταξύ εφαρμογών προσφέροντας διαλειτουργικότητα και βελτιωμένη εμπειρία χρήστη. Μια από τις σημαντικότερες λειτουργίες του Activity είναι η διαχείριση του κύκλου ζωής. Πιο συγκεκριμένα, το σύστημα Android μπορεί να διακόψει ή να καταστρέψει ένα Activity όταν ο χρήστης μεταβαίνει σε άλλο Activity ή όταν το σύστημα χρειάζεται να ελευθερώσει πόρους. Μέσω των callback μεθόδων (`onPause()`, `onStop()`, `onDestroy()`), οι προγραμματιστές μπορούν να αποθηκεύουν δεδομένα, να σταματούν λειτουργίες ή να απελευθερώνουν πόρους ώστε να αποφεύγονται διαρροές μνήμης. Επιπλέον, το Android υποστηρίζει fragment-based σχεδιασμό, όπου ένα Activity μπορεί να φιλοξενήσει Fragments, επιτρέποντας πιο δυναμικές και επαναχρησιμοποιούμενες διεπαφές χρήστη. Το `AppCompatActivity` είναι μια εκτεταμένη έκδοχή του Activity που παρέχει συμβατότητα με παλαιότερες εκδόσεις του Android και υποστηρίζει σύγχρονα UI components, όπως το Toolbar και το Material Design.

1.7.3 Μειονεκτήματα

Η χρήση πολλών Activities μπορεί να οδηγήσει σε πολύπλοκο σχεδιασμό εφαρμογής και σε αυξημένο overhead στη διαχείριση της μνήμης. Κάθε νέο Activity που εκκινείται αποθηκεύεται στη στοίβα δραστηριοτήτων (back stack), και αν δεν διαχειρίζεται σωστά, μπορεί να οδηγήσει σε υπερβολική κατανάλωση πόρων. Επιπλέον, το παραδοσιακό Activity-based UI έχει αρχίσει να αντικαθίσταται από Fragment-based αρχιτεκτονικές, καθώς τα Fragments προσφέρουν μεγαλύτερη ευελιξία και μπορούν να επαναχρησιμοποιηθούν σε διαφορετικά σημεία της εφαρμογής. Η διαχείριση του κύκλου ζωής του Activity είναι πολύπλοκη και απαιτεί ιδιαίτερη προσοχή στη διατήρηση της κατάστασης κατά τη διάρκεια αλλαγών, όπως η περιστροφή της οθόνης ή η είσοδος σε λειτουργία εξοικονόμησης ενέργειας.

1.8 AppCompatActivity

1.8.1 Γενικά

Η βιβλιοθήκη AppCompatActivity είναι μέρος του Android Jetpack και χρησιμοποιείται για τη διατήρηση της συμβατότητας των Android εφαρμογών με παλαιότερες εκδόσεις του λειτουργικού συστήματος. Ο κύριος ρόλος της είναι να επιτρέπει τη χρήση νεότερων χαρακτηριστικών UI και APIs σε συσκευές με παλαιότερες εκδόσεις Android, προσφέροντας μια ομοιογενή εμπειρία χρήστη. Χρησιμοποιείται ευρέως από προγραμματιστές που θέλουν να εξασφαλίσουν ότι οι εφαρμογές τους λειτουργούν σωστά σε ένα ευρύ φάσμα συσκευών, ανεξαρτήτως της έκδοσης του Android που χρησιμοποιούν. [8]

1.8.2 Λειτουργίες και δυνατότητες

Η βιβλιοθήκη AppCompat προσφέρει μια σειρά από υποκατηγορίες UI components και themes που βοηθούν τους προγραμματιστές να διατηρούν μια συνεπή και ομοιόμορφη εμφάνιση στις εφαρμογές τους. Περιλαμβάνει το AppCompatActivity, το οποίο επεκτείνει τη βασική Activity και προσφέρει υποστήριξη για το Action Bar, ακόμα και σε παλαιότερες εκδόσεις του Android. Επίσης, επιτρέπει τη χρήση των Material Design Components, όπως Toolbar, DrawerLayout, και NavigationView, προσφέροντας ένα σύγχρονο περιβάλλον χρήσης ακόμα και σε παλιότερες συσκευές. Μία από τις πιο χρήσιμες λειτουργίες της είναι η υποστήριξη για το Dark Mode, το οποίο μπορεί να εφαρμοστεί μέσω του DayNight theme, προσαρμόζοντας αυτόματα την εμφάνιση της εφαρμογής ανάλογα με τις ρυθμίσεις του χρήστη ή την ώρα της ημέρας. Επιπλέον, η βιβλιοθήκη παρέχει backward-compatible widgets όπως AppCompatActivity, AppCompatActivity, και AppCompatActivity, που εξασφαλίζουν σταθερότητα στη συμπεριφορά και στην εμφάνιση των UI στοιχείων ανεξαρτήτως έκδοσης Android. Τέλος, η βιβλιοθήκη υποστηρίζεται από τη Google, γεγονός που σημαίνει ότι λαμβάνει συχνά ενημερώσεις και νέες δυνατότητες.

1.8.3 Μειονεκτήματα

Ένα βασικό πρόβλημα είναι το αυξημένο μέγεθος της εφαρμογής, καθώς η χρήση της προσθέτει επιπλέον κώδικα και εξαρτήσεις που μπορεί να επηρεάσουν την απόδοση, ιδιαίτερα σε low-end συσκευές. Επιπλέον, η ανάγκη για backward compatibility μπορεί να περιορίσει τη χρήση ορισμένων νεότερων APIs, αναγκάζοντας τους προγραμματιστές να βασίζονται σε παλιότερες μεθόδους που ίσως να μην είναι τόσο αποδοτικές. Ένα ακόμα μειονέκτημα είναι ότι, καθώς το Android εξελίσσεται, η βιβλιοθήκη AppCompat γίνεται λιγότερο απαραίτητη για τις σύγχρονες εφαρμογές, δεδομένου ότι οι περισσότερες συσκευές υποστηρίζουν ήδη νεότερες εκδόσεις του λειτουργικού συστήματος. Πολλοί προγραμματιστές πλέον μεταβαίνουν σε AndroidX και άλλα Jetpack Components, τα οποία προσφέρουν καλύτερη απόδοση και λιγότερη πολυπλοκότητα.

1.9 Material

1.9.1 Γενικά

Η βιβλιοθήκη Material Components for Android (Material) είναι μέρος του Android Jetpack και χρησιμοποιείται για την υλοποίηση του Material Design, της γλώσσας σχεδιασμού που έχει αναπτύξει η Google. Παρέχει ένα σύγχρονο, συνεπές και προσαρμόσιμο UI για Android εφαρμογές, βελτιώνοντας την εμπειρία χρήσης. Χρησιμοποιείται σε εφαρμογές που ακολουθούν τις σχεδιαστικές αρχές του Material Design, ενσωματώνοντας στοιχεία όπως κουμπιά, κάρτες, διαλόγους και μεταβάσεις με βελτιωμένη εμφάνιση και λειτουργικότητα. [9]

1.9.2 Λειτουργίες και δυνατότητες

Η βιβλιοθήκη Material παρέχει μια σειρά από προκαθορισμένα UI components που βοηθούν τους προγραμματιστές να δημιουργήσουν εφαρμογές με μοντέρνα αισθητική και

ομοιογενή συμπεριφορά. Περιλαμβάνει στοιχεία όπως το `MaterialButton`, το `TextInputLayout` για φόρμες εισαγωγής δεδομένων, το `BottomNavigationView` για εύκολη πλοήγηση και το `FloatingActionButton` για γρήγορη εκτέλεση ενεργειών. Ένα από τα βασικά χαρακτηριστικά της βιβλιοθήκης είναι η υποστήριξη του `Theming`, επιτρέποντας την εύκολη προσαρμογή της εμφάνισης της εφαρμογής μέσω του `MaterialTheme`, καθώς και η υποστήριξη για `Dark Mode`, το οποίο μπορεί να εφαρμοστεί δυναμικά ανάλογα με τις προτιμήσεις του χρήστη. Επιπλέον, η βιβλιοθήκη περιλαμβάνει `Motion Components` που επιτρέπουν την προσθήκη ρευστών κινήσεων και μεταβάσεων, βελτιώνοντας την εμπειρία χρήστη και περαιτέρω ενσωματώνει `advanced typography` μέσω του `MaterialTextView`, καθώς και συστήματα `grid` και `layout` που βοηθούν στη δημιουργία προσαρμόσιμων διεπαφών. Υποστηρίζει τη χρήση του `ShapeableImageView` και του `CardView`, επιτρέποντας τη διαχείριση σχημάτων και σκιών στα UI στοιχεία, δίνοντας μια πιο δυναμική και ευχάριστη οπτική εμπειρία.

1.9.3 Μειονεκτήματα

Ένα από τα βασικά προβλήματα της βιβλιοθήκης αυτής είναι ότι προσθέτει επιπλέον εξαρτήσεις και αυξάνει το μέγεθος της εφαρμογής, καθώς ενσωματώνει πολλά UI components και animations. Σε συσκευές με περιορισμένους πόρους, η χρήση πολύπλοκων Material elements μπορεί να επηρεάσει την απόδοση της εφαρμογής. Επιπλέον, η υλοποίηση του Material Design μπορεί να απαιτεί περισσότερη πολυπλοκότητα στην ανάπτυξη, ειδικά αν η εφαρμογή δεν είναι πλήρως σχεδιασμένη με βάση τις αρχές του Material Design. Η σωστή ενσωμάτωση Motion Components και Theming μπορεί να απαιτεί επιπλέον χρόνο και δοκιμές. Τέλος, παρόλο που η βιβλιοθήκη ενημερώνεται συχνά από τη Google, ορισμένα νέα στοιχεία μπορεί να είναι ασταθή ή να αλλάζουν μεταξύ εκδόσεων, απαιτώντας επιπλέον συντήρηση της εφαρμογής.

1.10 ConstraintLayout

1.10.1 Γενικά

Η `ConstraintLayout` είναι μια προηγμένη διάταξη (layout) στο Android που επιτρέπει τη δημιουργία ευέλικτων και αποδοτικών UI με δυναμικό και προσαρμόσιμο σχεδιασμό. Παρουσιάστηκε ως μέρος του Android Jetpack για να αντικαταστήσει πιο παλιές διατάξεις όπως το `RelativeLayout` και το `LinearLayout`, προσφέροντας μεγαλύτερη ευελιξία, βελτιωμένη απόδοση και ευκολία σχεδίασης. Το κύριο χαρακτηριστικό της είναι ότι επιτρέπει την τοποθέτηση στοιχείων βασισμένη σε περιορισμούς (constraints), δηλαδή σχέσεις μεταξύ των στοιχείων στο UI, χωρίς να απαιτείται η χρήση εμφωλευμένων διατάξεων (nested layouts). [10]

1.10.2 Λειτουργίες και δυνατότητες

Το `ConstraintLayout` επιτρέπει τη δημιουργία πολύπλοκων διεπαφών χρήστη χωρίς τη χρήση πολλαπλών επιπέδων διατάξεων, γεγονός που μειώνει την πολυπλοκότητα και βελτιώνει την απόδοση της εφαρμογής. Χρησιμοποιεί constraints (περιορισμούς) για να καθορίσει τη θέση και το μέγεθος των στοιχείων, επιτρέποντας τον προσαρμοστικό

σχεδιασμό σε διαφορετικά μεγέθη οθόνης. Μια από τις βασικές λειτουργίες της βιβλιοθήκης είναι η Chain & Grouping, η οποία επιτρέπει τη δημιουργία ομαδοποιημένων στοιχείων με ευέλικτη στοίχιση και κατανομή χώρου. Επιπλέον, το Guideline & Barrier System προσφέρει έναν αόρατο οδηγό που βοηθά στην οργάνωση του UI, ιδιαίτερα σε δυναμικά layouts. Ένα ακόμα βασικό χαρακτηριστικό είναι το MotionLayout, το οποίο βασίζεται στο ConstraintLayout και επιτρέπει τη δημιουργία πολύπλοκων animations και μεταβάσεων με απλό και εύχρηστο τρόπο. Το MotionLayout υποστηρίζει keyframes, διαβαθμισμένες κινήσεις και προσαρμοσμένα animations, καθιστώντας το ιδανικό για μοντέρνα και διαδραστικά UI. Το ConstraintLayout υποστηρίζει επίσης percent-based layouts, επιτρέποντας στα στοιχεία να καταλαμβάνουν ποσοστιαία τμήματα της οθόνης, κάτι ιδιαίτερα χρήσιμο για responsive σχεδιασμό, κάτι που επιθυμούν οι περισσότερες σύγχρονες εφαρμογές.

1.10.3 Πλεονεκτήματα

Το κύριο πλεονέκτημα του ConstraintLayout είναι ότι βελτιώνει την απόδοση της εφαρμογής, μειώνοντας το overhead από εμφωλευμένες διατάξεις και βελτιώνοντας τους χρόνους rendering. Σε αντίθεση με το LinearLayout και το RelativeLayout, όπου η εμφώλευση πολλών layouts μπορεί να οδηγήσει σε χαμηλότερη απόδοση, το ConstraintLayout διατηρεί το UI απλό και αποδοτικό. Ένα ακόμα σημαντικό πλεονέκτημα είναι ότι διευκολύνει τη δημιουργία responsive UI που προσαρμόζονται εύκολα σε διαφορετικές διαστάσεις οθόνης. Χρησιμοποιώντας constraints, guidelines και barriers, οι προγραμματιστές μπορούν να σχεδιάσουν προσαρμόσιμες διεπαφές χωρίς να χρειάζονται ξεχωριστά layouts για κάθε διάταξη. Επιπλέον, το MotionLayout προσφέρει ενσωματωμένη υποστήριξη για animations, διευκολύνοντας τη δημιουργία μοντέρνων και ρευστών UI, χωρίς να απαιτούνται πολύπλοκες custom animations μέσω κώδικα. Τέλος, το ConstraintLayout υποστηρίζεται πλήρως από το Android Studio Layout Editor, επιτρέποντας στους προγραμματιστές να σχεδιάζουν εύκολα διαδραστικά layouts μέσω drag-and-drop, χωρίς να χρειάζεται να γράψουν κώδικα XML με το χέρι.

1.10.4 Μειονεκτήματα

Παρά τα πλεονεκτήματά του, το ConstraintLayout έχει μεγαλύτερη καμπύλη εκμάθησης σε σχέση με άλλες διατάξεις, καθώς απαιτεί κατανόηση των constraints και των μηχανισμών τοποθέτησης των στοιχείων. Οι νέοι προγραμματιστές ενδέχεται να χρειαστούν χρόνο για να εξοικειωθούν με την καθοδήγηση μέσω constraints, καθώς και με τη χρήση Guidelines, Barriers και Chains. Επιπλέον, ενώ το ConstraintLayout βελτιώνει την απόδοση, σε απλά UI με λίγα στοιχεία μπορεί να θεωρηθεί περιττό, καθώς ένας απλός LinearLayout ή FrameLayout μπορεί να είναι πιο αποδοτικός και εύκολος στη χρήση. Ένα άλλο μειονέκτημα είναι ότι η δημιουργία πολύπλοκων constraints μπορεί να οδηγήσει σε δυσκολία συντήρησης του κώδικα, ειδικά αν υπάρχουν πάρα πολλές αλληλεξαρτήσεις μεταξύ των στοιχείων. Αυτό μπορεί να καταστήσει το UI δυσνόητο και δύσκολο στην τροποποίηση αργότερα. Τέλος, η χρήση του MotionLayout, παρόλο που είναι συχνή, δεν αντικαθιστά πλήρως τα παραδοσιακά animation frameworks, καθώς υπάρχουν περιπτώσεις όπου custom animations μέσω code (ObjectAnimator, Lottie, OpenGL) είναι πιο ευέλικτες και αποδοτικές.

1.11 Google Play Services – Wearable API

1.11.1 Γενικά

Το Google Play Services - Wearable API είναι μια βιβλιοθήκη που επιτρέπει την επικοινωνία και τον συγχρονισμό δεδομένων μεταξύ ενός Wear OS smartwatch και ενός Android smartphone. Είναι μέρος του Google Play Services και παρέχει ένα σύνολο APIs που επιτρέπουν τη μεταφορά μηνυμάτων, αρχείων, ειδοποιήσεων και άλλων δεδομένων μεταξύ των συσκευών. Αυτή η βιβλιοθήκη είναι ιδανική για εφαρμογές που απαιτούν συνεχή επικοινωνία μεταξύ κινητού και smartwatch, επιτρέποντας συγχρονισμένες εμπειρίες χρήστη, όπως η καταγραφή δεδομένων υγείας, οι ειδοποιήσεις και η απομακρυσμένη εκτέλεση εντολών από το ρολόι στο κινητό και αντίστροφα. [11]

1.11.2 Λειτουργίες και δυνατότητες

Το Google Play Services - Wearable API παρέχει διάφορους μηχανισμούς επικοινωνίας μέσω των ακόλουθων APIs:

Data Layer API: Επιτρέπει τη συγχρονισμένη αποθήκευση και μεταφορά δεδομένων μεταξύ του smartwatch και του κινητού. Είναι ιδανικό για δεδομένα όπως καρδιακοί παλμοί, βήματα, θερμίδες, ειδοποιήσεις και ρυθμίσεις χρήστη.

MessageClient API: Υποστηρίζει real-time αποστολή μηνυμάτων μεταξύ των συσκευών. Είναι κατάλληλο για περιπτώσεις όπου μια συσκευή θέλει να στείλει μια άμεση εντολή στην άλλη, όπως ο χειρισμός της μουσικής από το smartwatch ή η αποστολή μιας ειδοποίησης από το κινητό.

ChannelClient API: Χρησιμοποιείται για τη μεταφορά αρχείων μεταξύ των συσκευών, όπως ήχοι, εικόνες και άλλα μεγάλα δεδομένα. Είναι ιδιαίτερα χρήσιμο για εφαρμογές που συγχρονίζουν αρχεία μεταξύ smartphone και smartwatch.

CapabilityClient API: Δίνει τη δυνατότητα στις συσκευές να εντοπίζουν ποιες λειτουργίες είναι διαθέσιμες σε κάθε Wear OS συσκευή. Με αυτόν τον τρόπο, η εφαρμογή μπορεί να γνωρίζει αν η Wear OS συσκευή υποστηρίζει συγκεκριμένες δυνατότητες πριν προσπαθήσει να τις χρησιμοποιήσει.

Επιπλέον, η βιβλιοθήκη υποστηρίζει τη δημιουργία Complications, δηλαδή μικρών widgets που εμφανίζονται απευθείας στο καντράν του ρολογιού, παρέχοντας πληροφορίες όπως ο καιρός, οι καρδιακοί παλμοί, τα βήματα ή το επίπεδο μπαταρίας.

1.11.3 Πλεονεκτήματα

Το Google Play Services - Wearable API παρέχει αξιόπιστη και ενεργειακά αποδοτική επικοινωνία μεταξύ κινητού και smartwatch. Εξασφαλίζει συγχρονισμό δεδομένων χωρίς να απαιτείται συνεχής σύνδεση, καθώς το σύστημα μπορεί να αποθηκεύσει και να συγχρονίσει δεδομένα όταν οι συσκευές είναι διαθέσιμες. Επιπλέον, η υποστήριξη real-time messaging μέσω του MessageClient API επιτρέπει την ταχύτατη ανταλλαγή δεδομένων, όπως ειδοποιήσεις, εντολές και άμεσες αλληλεπιδράσεις, βελτιώνοντας την εμπειρία χρήστη. Η βιβλιοθήκη ενσωματώνεται πλήρως στο Google Play Services, γεγονός

που εξασφαλίζει σταθερότητα και συνεχείς ενημερώσεις, ενώ η χρήση του CapabilityClient API βοηθά στον εντοπισμό συμβατών Wear OS συσκευών, αποτρέποντας προβλήματα συμβατότητας. Τέλος, είναι ιδανικό για εφαρμογές που απαιτούν συνεχή επικοινωνία, όπως fitness tracking apps, απομακρυσμένος έλεγχος multimedia, ειδοποιήσεις υγείας και ενσωμάτωση με IoT συσκευές.

1.11.4 Μειονεκτήματα

Η εξάρτησή από το Google Play Services οδηγεί στη μη λειτουργία του σε συσκευές που δεν υποστηρίζουν τις υπηρεσίες της Google, όπως ορισμένα Wear OS smartwatches από Κινέζους κατασκευαστές ή συσκευές χωρίς Play Store. Επιπλέον, η σωστή ρύθμιση του Data Layer API λόγω της πολυπλοκότητάς του, απαιτεί καλή κατανόηση των callbacks και των listeners, γεγονός που μπορεί να δυσκολέψει τους αρχάριους προγραμματιστές. Ακόμα, παρόλο που υποστηρίζει ChannelClient API για μεταφορά αρχείων, η αποστολή μεγάλων αρχείων μέσω Bluetooth μπορεί να είναι αργή στη μεταφορά μεγάλου όγκου δεδομένων και σε πολλές περιπτώσεις είναι προτιμότερη η χρήση cloud storage ή Wi-Fi Direct. Τέλος, αν η σύνδεση μεταξύ smartwatch και smartphone είναι αδύναμη (π.χ. Bluetooth αποσυνδέσεις), ενδέχεται να υπάρχουν καθυστερήσεις στην αποστολή δεδομένων ή απώλειες μηνυμάτων.

1.12 Apache Commons Math3

1.12.1 Γενικά

Η Apache Commons Math3 είναι μια ισχυρή και ευρέως χρησιμοποιούμενη βιβλιοθήκη για μαθηματικούς και αριθμητικούς υπολογισμούς στην Java. Παρέχει ένα πλούσιο σύνολο εργαλείων για γραμμική άλγεβρα, στατιστική ανάλυση, θεωρία αριθμών, αριθμητική ανάλυση και βελτιστοποίηση, καθιστώντας την κατάλληλη για επιστημονικές και χρηματοοικονομικές εφαρμογές. Είναι ιδιαίτερα χρήσιμη για προγραμματιστές που χρειάζονται ακριβείς, αποδοτικούς και έτοιμους προς χρήση μαθηματικούς αλγορίθμους, χωρίς να χρειάζεται να τους υλοποιήσουν από το μηδέν. [12]

1.12.2 Λειτουργίες και δυνατότητες

Η βιβλιοθήκη Apache Commons Math3 παρέχει ένα σύνολο προηγμένων αριθμητικών λειτουργιών που καλύπτουν ένα ευρύ φάσμα εφαρμογών. Περιλαμβάνει εργαλεία για τη διαχείριση διανυσμάτων και πινάκων μέσω της RealMatrix και της RealVector, επιτρέποντας τη διεξαγωγή πράξεων όπως πολλαπλασιασμός, αντίστροφη μήτρα και ιδιοτιμές. Παρέχει επίσης ένα ισχυρό σύστημα για στατιστικές αναλύσεις, συμπεριλαμβανομένων βασικών στατιστικών μετρικών (μέσος όρος, διασπορά, τυπική απόκλιση), καθώς και υποστήριξη για πιθανοτικές κατανομές (Gaussian, Poisson, Exponential, κ.λπ.). Επιπλέον, περιλαμβάνει αλγορίθμους βελτιστοποίησης, όπως η μέθοδος Simplex, καθώς και μεθόδους εύρεσης ριζών εξισώσεων (Newton-Raphson, Brent's Method). Υποστηρίζει επίσης αριθμητική ολοκλήρωση (μέθοδοι Trapezoidal, Simpson), καθώς και διαχείριση πολυωνύμων μέσω της PolynomialFunction. Για προβλήματα που αφορούν τη θεωρία αριθμών, παρέχει λειτουργίες όπως εύρεση μέγιστου

κοινού διαιρέτη (GCD), ανάλυση σε πρώτους αριθμούς και αριθμητικές πράξεις με BigInteger.

1.12.3 Πλεονεκτήματα

Ένα σημαντικό πλεονέκτημα αποτελεί ότι η βιβλιοθήκη είναι ανοιχτού κώδικα και υποστηρίζεται από την Apache Software Foundation, εξασφαλίζοντας μακροχρόνια υποστήριξη και συνεχείς βελτιώσεις. Οι δομές δεδομένων και οι μαθηματικοί αλγόριθμοι είναι σχεδιασμένοι με ανθεκτικότητα και ακρίβεια, γεγονός που την καθιστά ιδιαίτερα χρήσιμη σε τομείς όπως η επιστημονική έρευνα, η μηχανική μάθηση, τα χρηματοοικονομικά και η αριθμητική ανάλυση.

1.12.4 Μειονεκτήματα

Η Apache Commons Math3 δεν υποστηρίζει GPU επιτάχυνση, κάτι που σημαίνει ότι για πολύ μεγάλους αριθμητικούς υπολογισμούς, άλλες βιβλιοθήκες όπως TensorFlow (για Java) ή NumPy (σε Python) μπορεί να είναι πιο αποδοτικές. Επίσης, δεν είναι πάντα η πιο γρήγορη λύση για ορισμένους εξειδικευμένους αλγόριθμους, όπου βιβλιοθήκες όπως JAMA ή Colt μπορεί να προσφέρουν καλύτερες επιδόσεις σε συγκεκριμένες εφαρμογές. Ένα ακόμα μειονέκτημα είναι το μέγεθος των εξαρτήσεων της, καθώς η πλήρης βιβλιοθήκη περιλαμβάνει πολλές κλάσεις που μπορεί να είναι περιττές για μια ελαφριά εφαρμογή. Τέλος, η βιβλιοθήκη δεν υποστηρίζεται ενεργά από την Apache από το 2016, και αν και εξακολουθεί να χρησιμοποιείται ευρέως, δεν λαμβάνει νέες σημαντικές ενημερώσεις ή προσθήκες.

1.13 JDSP

1.13.1 Γενικά

Η JDSP είναι μια βιβλιοθήκη επεξεργασίας ψηφιακών σημάτων (Digital Signal Processing - DSP) για Java, η οποία επιτρέπει την υλοποίηση διαφόρων τεχνικών επεξεργασίας σημάτων με αποδοτικό και εύκολο τρόπο. Χρησιμοποιείται κυρίως σε εφαρμογές που απαιτούν φιλτράρισμα, ανάλυση Fourier, επεξεργασία ήχου, επεξεργασία σήματος αισθητήρων και ανάλυση συχνοτήτων. Είναι ιδιαίτερα χρήσιμη για επιστημονικές και μηχανικές εφαρμογές, καθώς και για αναλύσεις δεδομένων που σχετίζονται με χρονικές σειρές, βιοϊατρικά σήματα και εφαρμογές IoT. Η βιβλιοθήκη JDSP έχει σχεδιαστεί για να προσφέρει υψηλή απόδοση, ευκολία στη χρήση και ευέλικτες υλοποιήσεις βασικών αλγορίθμων DSP, επιτρέποντας στους προγραμματιστές να ενσωματώνουν λειτουργίες επεξεργασίας σήματος σε εφαρμογές Java χωρίς την ανάγκη χρήσης εξωτερικών γλωσσών όπως MATLAB ή Python. [13]

1.13.2 Λειτουργίες και δυνατότητες

Η JDSP παρέχει ένα πλήρες σύνολο εργαλείων για την επεξεργασία και ανάλυση σημάτων, καλύπτοντας βασικές και προηγμένες τεχνικές DSP. Οι κυριότερες λειτουργίες της περιλαμβάνουν:

- Γρήγορος Μετασχηματισμός Fourier (Fast Fourier Transform - FFT): Παρέχει γρήγορη ανάλυση συχνοτήτων χρησιμοποιώντας FFT και IFFT (Inverse FFT), επιτρέποντας τη μετατροπή σημάτων από το πεδίο του χρόνου στο πεδίο των συχνοτήτων.
- Φιλτράρισμα σημάτων: Υποστηρίζει διάφορα ψηφιακά φίλτρα FIR και IIR (π.χ. Butterworth, Chebyshev, Bessel) για απομάκρυνση θορύβου και εξαγωγή σημαντικών χαρακτηριστικών από τα δεδομένα.
- Συσχέτιση και σύγκλιση σημάτων: Επιτρέπει την εφαρμογή convolution και cross-correlation, χρήσιμα για την ανάλυση και επεξεργασία σημάτων σε πραγματικό χρόνο.
- Μετασχηματισμός κυματιδίων (Wavelet Transform): Παρέχει δυνατότητες Wavelet decomposition, χρήσιμο για την επεξεργασία βιοϊατρικών σημάτων όπως ΗΚΓ (ECG) και EEG.
- Επεξεργασία σήματος αισθητήρων: Περιλαμβάνει εργαλεία για την ανάλυση δεδομένων από επιταχυνσιόμετρα, γυροσκόπια και άλλους αισθητήρες IoT, κάνοντας την ιδανική για εφαρμογές wearable τεχνολογίας και ανάλυσης υγειονομικών δεδομένων.
- Παραγωγή και διαμόρφωση σημάτων: Υποστηρίζει δημιουργία συνθετικών σημάτων (π.χ. ημιτονοειδή, τετραγωνικά κύματα) και διαμόρφωση πλάτους, συχνότητας και φάσης (AM, FM, PM).

```
import com.github.psambit9791.jdsp.transform.FastFourier;

public class FFTExample {
    public static void main(String[] args) {
        double[] signal = {0.0, 1.0, 0.0, -1.0}; // Παράδειγμα σήματος
        FastFourier fft = new FastFourier(signal);
        double[][] fftResult = fft.fft(); // Υπολογισμός FFT
        System.out.println("FFT Real Part: " + Arrays.toString(fftResult[0]));
        System.out.println("FFT Imaginary Part: " + Arrays.toString(fftResult[1]));
    }
}
```

Εικόνα 3. Παράδειγμα χρήσης του FFT για ανάλυση ενός σήματος

1.13.3 Πλεονεκτήματα

Η JDSP παρέχει ένα πλήρες και αποδοτικό σύνολο εργαλείων για επεξεργασία ψηφιακών σημάτων, καθιστώντας την ιδανική για αναλύσεις δεδομένων αισθητήρων, ανάλυση ήχου, εφαρμογές IoT και επιστημονικές μελέτες. Ένα από τα σημαντικότερα πλεονεκτήματά της είναι ότι παρέχει υψηλή απόδοση, χρησιμοποιώντας αποδοτικούς αλγορίθμους για πράξεις όπως FFT, φίλτρα FIR/IIR και convolution, χωρίς να απαιτείται GPU επιτάχυνση. Ένα ακόμα πλεονέκτημα είναι η ευκολία ενσωμάτωσης και χρήσης, καθώς παρέχει καλά τεκμηριωμένο API, που διευκολύνει την ενσωμάτωση της επεξεργασίας σημάτων σε εφαρμογές Java. Σε αντίθεση με άλλες λύσεις που βασίζονται σε Python (όπως SciPy ή NumPy), η JDSP τρέχει απευθείας σε περιβάλλον Java, καθιστώντας την ιδανική για Android εφαρμογές, Big Data ανάλυση και IoT. Επιπλέον, η JDSP υποστηρίζει πολλές βασικές και προχωρημένες τεχνικές DSP σε μία ενιαία βιβλιοθήκη, μειώνοντας την ανάγκη χρήσης εξωτερικών εργαλείων, όπως

MATLAB, SciPy ή OpenCV για επεξεργασία σημάτων. Παράλληλα, επιτρέπει τη χρήση Wavelet Transforms, κάτι που δεν είναι κοινό σε πολλές βιβλιοθήκες DSP.

1.13.4 Μειονεκτήματα

Ένα από τα μειονεκτήματα της JDSP είναι ότι δεν υποστηρίζει GPU acceleration, γεγονός που μπορεί να περιορίσει την απόδοση σε υπολογιστικά εντατικές εφαρμογές που απαιτούν real-time DSP με υψηλή ανάλυση. Για εφαρμογές που απαιτούν υψηλές επιδόσεις, ίσως είναι καλύτερη επιλογή βιβλιοθήκες όπως TensorFlow (για Java) ή CUDA-based DSP libraries. Επιπλέον, η JDSP είναι λιγότερο διαδεδομένη σε σύγκριση με άλλες DSP βιβλιοθήκες, όπως αυτές που υπάρχουν σε Python (SciPy, NumPy) ή C++ (FFTW, OpenCV). Αυτό σημαίνει ότι η κοινότητα ανάπτυξης της είναι μικρότερη, και ως εκ τούτου, η υποστήριξη και η τεκμηρίωση μπορεί να είναι περιορισμένες σε σχέση με άλλες πιο γνωστές βιβλιοθήκες. Τέλος, η ακρίβεια και η σταθερότητα ορισμένων αλγορίθμων δεν έχει ελεγχθεί εκτενώς σε βιομηχανικές εφαρμογές, επομένως για κρίσιμες εφαρμογές (π.χ. ιατρικά σήματα) μπορεί να χρειαστεί επιπλέον δοκιμή και επικύρωση.

1.14 Chaquopy

1.14.1 Γενικά

Το Chaquopy είναι ένα SDK που επιτρέπει την εκτέλεση κώδικα Python μέσα σε εφαρμογές Android. Ενσωματώνεται στο Android Studio και διευκολύνει την ανάπτυξη εφαρμογών που απαιτούν λειτουργικότητες της Python, χωρίς να απαιτείται ολική μετατροπή της εφαρμογής, όπως ανάλυση δεδομένων, μηχανική μάθηση και επιστημονικούς υπολογισμούς, ενώ ταυτόχρονα υποστηρίζει την αλληλεπίδραση με Java/Kotlin, γεγονός που το καθιστά ευέλικτο για σύνθετες εφαρμογές. [14]

1.14.2 Δυνατότητες - Μειονεκτήματα

Υποστηρίζει τη χρήση βιβλιοθηκών Python, όπως NumPy, Pandas, OpenCV και TensorFlow, κάνοντάς το ιδανικό για εφαρμογές που απαιτούν ισχυρούς αλγορίθμους ανάλυσης. Επιπλέον, διαθέτει εργαλείο διαχείρισης πακέτων, μέσω του οποίου μπορούν να εγκατασταθούν εξωτερικές βιβλιοθήκες απευθείας από το pip. Η αυξημένη πολυπλοκότητα της ανάπτυξης, λόγω της ταυτόχρονης χρήσης δύο διαφορετικών γλωσσών προγραμματισμού, αποτελεί μειονέκτημά του. Επιπλέον, μπορεί να έχει θέματα επιδόσεων σε εφαρμογές που απαιτούν πολύ υψηλή ταχύτητα εκτέλεσης, καθώς η Python δεν είναι τόσο αποδοτική όσο η Java/Kotlin σε native Android περιβάλλοντα.

1.15 RecyclerView

1.15.1 Γενικά

Το RecyclerView είναι ένα από τα πιο ισχυρά και αποδοτικά στοιχεία UI που παρέχονται από το Android Jetpack, σχεδιασμένο για τη διαχείριση και την εμφάνιση μεγάλων λιστών

και πλεγμάτων δεδομένων. Αντικαθιστά το παλιότερο ListView και GridView, προσφέροντας βελτιωμένη απόδοση, μεγαλύτερη ευελιξία και υποστήριξη για προσαρμόσιμες διατάξεις. Το RecyclerView χρησιμοποιεί μια αποδοτική μέθοδο ανακύκλωσης (view recycling), όπου οι προβολές (views) που δεν είναι πλέον ορατές επαναχρησιμοποιούνται αντί να δημιουργούνται από την αρχή, μειώνοντας έτσι την κατανάλωση μνήμης και βελτιώνοντας την απόδοση της εφαρμογής. [15]

1.15.2 Λειτουργίες και δυνατότητες

Το RecyclerView παρέχει προηγμένες δυνατότητες προβολής δεδομένων, επιτρέποντας τη δημιουργία λίστων, πλεγμάτων και custom layouts με ευέλικτο τρόπο. Μερικές από τις βασικές του λειτουργίες περιλαμβάνουν:

- **ViewHolder Pattern:** Το RecyclerView χρησιμοποιεί το ViewHolder, ένα σχεδιαστικό πρότυπο που επιτρέπει τη βελτιστοποίηση της απόδοσης, διατηρώντας αναφορές στα στοιχεία UI για την αποφυγή περιττών κλήσεων findViewById().
- **LayoutManager:** Επιτρέπει τη διάταξη των στοιχείων με διάφορους τρόπους, όπως κάθετη ή οριζόντια λίστα (LinearLayoutManager), πλέγμα (GridLayoutManager) ή ελεύθερη διάταξη (StaggeredGridLayoutManager).
- **ItemDecoration:** Παρέχει υποστήριξη για προσαρμοσμένα διαχωριστικά και διακοσμητικά στοιχεία μεταξύ των αντικειμένων της λίστας, βελτιώνοντας την εμφάνιση του RecyclerView.
- **ItemAnimator:** Επιτρέπει την εύκολη προσθήκη animations κατά την εισαγωγή, διαγραφή ή αναδιάταξη αντικειμένων στη λίστα.
- **DiffUtil:** Χρησιμοποιείται για αποδοτικές ενημερώσεις δεδομένων, συγκρίνοντας διαφορές μεταξύ παλιών και νέων δεδομένων και ανανεώνοντας μόνο τα απαραίτητα στοιχεία.
- **Drag & Drop και Swipe Gestures:** Επιτρέπει στους χρήστες να μετακινούν αντικείμενα ή να διαγράφουν στοιχεία με swipe, χρησιμοποιώντας το ItemTouchHelper

1.15.3 Πλεονεκτήματα

Ένα από τα σημαντικότερα πλεονεκτήματα του RecyclerView είναι η αποδοτική διαχείριση μνήμης μέσω του View Recycling, που επιτρέπει τη χρήση ενός περιορισμένου αριθμού προβολών (views), ακόμα και σε πολύ μεγάλες λίστες δεδομένων. Αυτό μειώνει την κατανάλωση μνήμης και βελτιώνει την απόδοση, ιδιαίτερα σε συσκευές με χαμηλή επεξεργαστική ισχύ. Επιπλέον, το RecyclerView παρέχει μεγαλύτερη ευελιξία από το παλιό ListView, καθώς υποστηρίζει προσαρμοσμένες διατάξεις, animations και drag & drop λειτουργίες, δίνοντας στους προγραμματιστές τον πλήρη έλεγχο της εμφάνισης και λειτουργικότητας. Άλλο ένα σημαντικό πλεονέκτημα είναι η εύκολη διαχείριση δεδομένων, χάρη σε εργαλεία όπως το DiffUtil, που επιτρέπουν γρήγορη ενημέρωση της λίστας χωρίς περιττές ανανεώσεις. Η χρήση ItemAnimator και ItemDecoration επιτρέπει την προσθήκη

απαλών μεταβάσεων και προσαρμοσμένων διαχωριστικών, βελτιώνοντας την εμπειρία χρήστη.

1.15.4 Μειονεκτήματα

Παρόλο που το RecyclerView είναι ισχυρό και αποδοτικό, έχει μεγαλύτερη πολυπλοκότητα σε σύγκριση με το ListView, καθώς απαιτεί την υλοποίηση πολλών διαφορετικών στοιχείων (Adapter, ViewHolder, LayoutManager). Για αρχάριους προγραμματιστές, η αρχική εκμάθηση μπορεί να είναι πιο δύσκολη. Επίσης, η υλοποίηση ορισμένων βασικών λειτουργιών, όπως τα click listeners, δεν είναι ενσωματωμένη όπως στο ListView, και απαιτεί την προσθήκη custom interfaces ή χρήσης setOnClickListener μέσα στον ViewHolder. Περαιτέρω, αν δεν χρησιμοποιηθεί σωστά το ViewHolder και το DiffUtil, η λίστα μπορεί να επαναφορτώνει όλα τα στοιχεία της, οδηγώντας σε κακή απόδοση και υψηλή κατανάλωση μνήμης. Για το λόγο αυτό, είναι σημαντικό να γίνεται σωστή χρήση των εργαλείων που προσφέρει το RecyclerView.

1.16 SendGrid Java

1.16.1 Γενικά

Η SendGrid Java είναι η επίσημη βιβλιοθήκη της SendGrid για την αποστολή email μέσω της πλατφόρμας Twilio SendGrid χρησιμοποιώντας Java. Επιτρέπει στους προγραμματιστές να στέλνουν email μέσω του SendGrid API, χωρίς να χρειάζεται να διαχειρίζονται SMTP διακομιστές ή να υλοποιούν πολύπλοκες διαδικασίες αποστολής email. Η βιβλιοθήκη απλοποιεί τη δημιουργία, αποστολή και διαχείριση email, υποστηρίζοντας λειτουργίες όπως HTML emails, συνημμένα αρχεία, email templates, authentication και email tracking. Είναι κατάλληλη για διαχειριστικά συστήματα, εφαρμογές e-commerce, email marketing πλατφόρμες και ειδοποιήσεις χρηστών. [16]

1.16.2 Λειτουργίες και δυνατότητες

Η SendGrid Java παρέχει ένα ολοκληρωμένο σύνολο δυνατοτήτων για τη δημιουργία, αποστολή και διαχείριση email μέσω της πλατφόρμας SendGrid. Ένα από τα βασικά χαρακτηριστικά της είναι η υποστήριξη για απλά και HTML emails, επιτρέποντας στους προγραμματιστές να στέλνουν μηνύματα είτε ως απλό κείμενο είτε με πλούσια μορφοποίηση HTML. Αυτό είναι χρήσιμο για την αποστολή επαγγελματικών email, newsletters και ειδοποιήσεων. Επιπλέον, η βιβλιοθήκη υποστηρίζει την αποστολή email σε πολλαπλούς παραλήπτες, προσφέροντας δυνατότητα για CC (Carbon Copy) και BCC (Blind Carbon Copy). Αυτό καθιστά εύκολη την αποστολή ενός email σε διάφορους χρήστες ταυτόχρονα, είτε πρόκειται για ενημερωτικά email είτε για μηνύματα συναλλαγών. Μια ακόμα σημαντική λειτουργία είναι η δυνατότητα διαχείρισης συνημμένων αρχείων, επιτρέποντας την προσθήκη εγγράφων, εικόνων και άλλων αρχείων στα email μέσω Base64 κωδικοποίησης. Επιπλέον, η SendGrid Java API επιτρέπει τη χρήση έτοιμων email templates, τα οποία μπορούν να διαμορφωθούν μέσω του SendGrid UI, διευκολύνοντας έτσι την αποστολή προσαρμοσμένων email χωρίς να χρειάζεται κώδικας HTML μέσα στην εφαρμογή. Η βιβλιοθήκη προσφέρει επίσης εργαλεία email tracking και analytics,

επιτρέποντας την παρακολούθηση του αν ένα email παραδόθηκε, ανοιχτήκε ή αν ο χρήστης έκανε κλικ σε κάποιον σύνδεσμο. Αυτή η λειτουργία είναι εξαιρετικά χρήσιμη για εταιρείες που θέλουν να αξιολογούν την απόδοση των email τους, ειδικά σε email marketing καμπάνιες. Για την ασφάλεια και την αποφυγή spam, το SendGrid Java API υποστηρίζει SPF, DKIM και DMARC authentication, εξασφαλίζοντας ότι τα emails αποστέλλονται από αξιόπιστες πηγές και δεν φιλτράρονται ως spam. Επιπλέον, υποστηρίζει webhooks, τα οποία επιτρέπουν στις εφαρμογές να λαμβάνουν ειδοποιήσεις σε πραγματικό χρόνο για γεγονότα όπως bounces, spam complaints και απορρίψεις email, βελτιώνοντας τη διαχείριση αποστολής email.

1.16.3 Πλεονεκτήματα

Ένα από τα κύρια πλεονεκτήματα της βιβλιοθήκης SendGrid Java είναι ότι εξαλείφει την ανάγκη διαχείρισης SMTP διακομιστών, καθώς λειτουργεί μέσω REST API, μειώνοντας την πολυπλοκότητα της αποστολής email. Η υποστήριξη δυναμικών προτύπων και email tracking καθιστά εύκολη την παρακολούθηση της απόδοσης των email, ενώ η δυνατότητα αποστολής email με HTML μορφοποίηση και συνημμένα αρχεία προσφέρει ευελιξία σε διαφορετικά σενάρια χρήσης. Επιπλέον, η εύκολη ενσωμάτωση με εφαρμογές Java επιτρέπει στους προγραμματιστές να στέλνουν email με απλό κώδικα, ενώ η υποστήριξη SPF, DKIM και DMARC authentication εξασφαλίζει υψηλή παραδοσιμότητα των email και αποτρέπει τη σήμανση τους ως spam. Ένα ακόμα πλεονέκτημα είναι η υποστήριξη webhooks, που επιτρέπει σε εφαρμογές να λαμβάνουν ανατροφοδότηση για την κατάσταση των email σε πραγματικό χρόνο.

1.16.4 Μειονεκτήματα

Βασικό πρόβλημα της βιβλιοθήκης αυτής είναι ότι η λειτουργία της εξαρτάται αποκλειστικά από την πλατφόρμα Twilio SendGrid, πράγμα που σημαίνει ότι αν το SendGrid αντιμετωπίσει downtime ή αν ένας λογαριασμός φτάσει το όριο αποστολών, τα email δεν θα σταλούν. Επιπλέον, το δωρεάν πακέτο του SendGrid περιορίζει την αποστολή σε 100 email την ημέρα, καθιστώντας το ακατάλληλο για εφαρμογές με μεγάλο όγκο αποστολής email χωρίς αναβάθμιση σε πληρωμένο πακέτο. Άλλο ένα μειονέκτημα είναι ότι η αποστολή email μέσω API μπορεί να έχει καθυστερήσεις, ειδικά σε περιπτώσεις υψηλού όγκου email, καθώς απαιτείται επικοινωνία με εξωτερικό server αντί για άμεση παράδοση μέσω SMTP. Τέλος, η βιβλιοθήκη δεν περιλαμβάνει ενσωματωμένη υποστήριξη για μαζική αποστολή email (bulk sending), κάτι που απαιτεί πρόσθετη υλοποίηση μέσω custom κώδικα ή χρήσης ξεχωριστών υπηρεσιών της SendGrid.

1.17 Apache HttpClient

1.17.1 Γενικά

Η Apache HttpClient είναι μια ευέλικτη και ισχυρή βιβλιοθήκη HTTP για Java, η οποία χρησιμοποιείται για την εκτέλεση HTTP αιτήσεων και λήψεων δεδομένων από απομακρυσμένους διακομιστές. Είναι μέρος του Apache HttpComponents, ενός ευρύτερου συνόλου εργαλείων για τη διαχείριση δικτυακής επικοινωνίας σε Java. Η βιβλιοθήκη

διευκολύνει την αποστολή GET, POST, PUT, DELETE και άλλων HTTP requests, παρέχοντας πλήρη έλεγχο των αιτήσεων, παραμετροποιήσιμα headers, authentication support, cookie management και connection pooling. Χρησιμοποιείται ευρέως σε RESTful APIs, web scrapers, IoT εφαρμογές και διαχειριστικά συστήματα, όπου η επικοινωνία με web services είναι κρίσιμη. [17]

1.17.2 Λειτουργίες και δυνατότητες

Το Apache HttpClient παρέχει ένα σύνολο εργαλείων για τη δημιουργία και εκτέλεση HTTP αιτήσεων, προσφέροντας πλήρη υποστήριξη για σύγχρονες και ασφαλείς επικοινωνίες μέσω του πρωτοκόλλου HTTP. Μια από τις βασικές του λειτουργίες είναι η υποστήριξη διαφορετικών μεθόδων HTTP, επιτρέποντας την εύκολη εκτέλεση GET, POST, PUT, DELETE, PATCH και HEAD requests, οι οποίες χρησιμοποιούνται συχνά σε RESTful APIs. Ένα άλλο σημαντικό χαρακτηριστικό είναι η υποστήριξη authentication. Το HttpClient υποστηρίζει διάφορους τύπους πιστοποίησης χρηστών, όπως Basic, Digest, NTLM και Kerberos authentication, επιτρέποντας την ασφαλή πρόσβαση σε προστατευμένα web services. Επιπλέον, διαθέτει ενσωματωμένη διαχείριση cookies, κάτι που είναι χρήσιμο για εφαρμογές που απαιτούν session persistence. Η βιβλιοθήκη υποστηρίζει επίσης connection pooling, επιτρέποντας τη διαχείριση πολλαπλών ταυτόχρονων συνδέσεων προς διαφορετικούς servers, βελτιώνοντας έτσι την απόδοση και μειώνοντας την καθυστέρηση στις κλήσεις HTTP. Παράλληλα, παρέχει λεπτομερή διαχείριση HTTP headers και request parameters, επιτρέποντας στους προγραμματιστές να προσθέσουν custom headers, user-agents, timeouts και retry policies στις αιτήσεις τους.

1.17.3 Πλεονεκτήματα

Το Apache HttpClient είναι μια ώριμη και σταθερή βιβλιοθήκη που χρησιμοποιείται σε πολλές εφαρμογές λόγω της ευελιξίας και της αξιοπιστίας της. Ένα από τα σημαντικότερα πλεονεκτήματά της είναι ότι παρέχει πλήρη έλεγχο της επικοινωνίας HTTP, επιτρέποντας στους προγραμματιστές να προσαρμόσουν τα requests και τις αποκρίσεις με βάση τις ανάγκες τους. Η υποστήριξη authentication και secure connections (HTTPS, SSL/TLS), αποτελεί ένα ακόμα πλεονέκτημα, καθιστώντας την κατάλληλη για εφαρμογές που χρειάζονται ασφαλή επικοινωνία με web services. Επιπλέον, η δυνατότητα για connection pooling και timeout management βοηθά στη βελτίωση της απόδοσης σε εφαρμογές που εκτελούν μεγάλο όγκο δικτυακών κλήσεων. Η βιβλιοθήκη παρέχει επίσης λεπτομερή logging και debugging δυνατότητες, καθιστώντας εύκολη τη διάγνωση προβλημάτων στις HTTP αιτήσεις. Τέλος, είναι εύκολη στην ενσωμάτωση με frameworks όπως Spring Boot και Apache Camel, επιτρέποντας την αποτελεσματική διαχείριση RESTful υπηρεσιών σε enterprise εφαρμογές.

1.17.4 Μειονεκτήματα

Παρά τα πλεονεκτήματά του, το Apache HttpClient είναι πιο περίπλοκο σε σχέση με τα Java built-in HTTP classes (URLConnection), απαιτώντας περισσότερη ρύθμιση και διαμόρφωση για βασικές λειτουργίες. Επιπλέον, η παλαιότερη αρχιτεκτονική του (έκδοση 4.x) περιλαμβάνει πολλά interfaces και κλάσεις, γεγονός που μπορεί να το κάνει δύσκολο στη χρήση για αρχάριους προγραμματιστές. Για τον λόγο αυτό, η νεότερη έκδοση 5.x του

HttpClient έχει σχεδιαστεί για να είναι πιο απλή και πιο αποδοτική, αντικαθιστώντας σταδιακά τις παλαιότερες εκδόσεις. Ένα άλλο μειονέκτημα είναι ότι, αν δεν γίνει σωστή διαχείριση των συνδέσεων, η εφαρμογή μπορεί να αντιμετωπίσει memory leaks ή ανεπιθύμητες καθυστερήσεις στις κλήσεις HTTP. Επομένως, είναι σημαντικό να τερματίζουν πάντα οι HttpClient και HttpResponse μετά τη χρήση. Τέλος, αν και υποστηρίζει HTTPS, η ρύθμιση για custom SSL certificates και proxies μπορεί να είναι περίπλοκη, απαιτώντας επιπλέον διαμόρφωση.

Κεφάλαιο 2. Η σχέση της εφαρμογής με την υγεία

2.1 Εισαγωγή

Η ραγδαία εξέλιξη των φορητών τεχνολογιών και η ενσωμάτωση καινοτόμων εφαρμογών στην καθημερινότητα έχουν επιφέρει σημαντικές αλλαγές στη διαχείριση της υγείας και της ευεξίας των ατόμων. Οι φορητές συσκευές, όπως τα έξυπνα ρολόγια και οι συσκευές παρακολούθησης φυσικής κατάστασης, έχουν εξελιχθεί από απλά αξεσουάρ σε προηγμένα εργαλεία καταγραφής βιομετρικών δεδομένων. Μέσω αυτών, οι χρήστες μπορούν να παρακολουθούν κρίσιμους δείκτες υγείας, όπως οι καρδιακοί παλμοί, τα επίπεδα οξυγόνου στο αίμα, η ποιότητα του ύπνου και η καθημερινή φυσική δραστηριότητα. Αυτή η διαρκής συλλογή δεδομένων καθιστά εφικτή την έγκαιρη ανίχνευση πιθανών προβλημάτων υγείας, ενισχύοντας την προληπτική ιατρική και τη δυνατότητα άμεσης αντίδρασης σε επείγουσες καταστάσεις. [18]

Παράλληλα, η πρόοδος στην επεξεργασία δεδομένων και στις τεχνολογίες cloud computing έχει καταστήσει εφικτή τη συλλογή, ανάλυση και αξιοποίηση αυτών των πληροφοριών σε πραγματικό χρόνο. Αυτό έχει ως αποτέλεσμα τη βελτίωση της εξατομικευμένης ιατρικής, καθώς τα δεδομένα που συλλέγονται από τις φορητές συσκευές μπορούν να χρησιμοποιηθούν από τους επαγγελματίες υγείας για την καλύτερη προσαρμογή θεραπευτικών αγωγών στις ανάγκες κάθε ατόμου. Επιπλέον, η διαθεσιμότητα αυτών των δεδομένων μέσω εφαρμογών σε κινητές συσκευές διευκολύνει την αλληλεπίδραση μεταξύ ασθενών και γιατρών, βελτιώνοντας τη συνολική ποιότητα της παρεχόμενης φροντίδας.

Στο πλαίσιο αυτής της τεχνολογικής εξέλιξης, η παρούσα διπλωματική εργασία επικεντρώνεται στην ανάπτυξη μιας εφαρμογής για έξυπνα ρολόγια, η οποία αξιοποιεί τα δεδομένα υγείας που συλλέγει η συσκευή με σκοπό την ανάλυση και τη βελτίωση της προσωπικής ευεξίας του χρήστη. Μέσω της εφαρμογής, παρέχεται η δυνατότητα συνεχούς παρακολούθησης βιομετρικών παραμέτρων, καθώς και η αξιοποίησή τους για την εξαγωγή χρήσιμων συμπερασμάτων σχετικά με την υγεία και τη φυσική κατάσταση των χρηστών. Στις επόμενες ενότητες, θα παρουσιαστούν οι βασικές λειτουργίες της εφαρμογής, οι κατηγορίες δεδομένων που συλλέγει, καθώς και οι τρόποι με τους οποίους αυτά μπορούν να αξιοποιηθούν για τη βελτίωση της καθημερινότητας των χρηστών.

2.2 Καρδιακοί παλμοί, Ηλεκτροκαρδιογράφημα (ECG) και Διακύμανση καρδιακής συχνότητας (HRV)

2.2.1 Γενικά

Ηλεκτροκαρδιογράφημα (ECG)

Το ηλεκτροκαρδιογράφημα (ECG) αποτελεί μια βασική, μη επεμβατική εξέταση που επιτρέπει την καταγραφή της ηλεκτρικής δραστηριότητας της καρδιάς. Μέσω ηλεκτροδίων που τοποθετούνται στο σώμα, ανιχνεύονται τα ηλεκτρικά σήματα που παράγονται από την καρδιά κατά τη σύσπαση και χαλάρωσή της. Το αποτέλεσμα είναι μια χαρακτηριστική κυματομορφή που περιλαμβάνει επάρματα και διαστήματα, τα οποία ερμηνεύονται από ειδικούς για την αξιολόγηση της καρδιακής λειτουργίας. Τα σύγχρονα έξυπνα ρολόγια διαθέτουν τεχνολογία που επιτρέπει τη λήψη ενός βασικού μονοκαναλικού ECG, δίνοντας

τη δυνατότητα στον χρήστη να παρακολουθεί τη δραστηριότητα της καρδιάς του εύκολα και γρήγορα. Αν και δεν αντικαθιστούν τα πολυκάναλα ιατρικά ECG, μπορούν να αποτελέσουν ένα χρήσιμο εργαλείο για την καθημερινή καταγραφή δεδομένων και την ανάλυση πιθανών τάσεων στη λειτουργία της καρδιάς. [19], [20]

Καρδιακός Παλμός

Ο καρδιακός παλμός αναφέρεται στη ρυθμική διαστολή και συστολή των αρτηριών, αποτέλεσμα της ροής του αίματος που εκτοξεύεται από την καρδιά με κάθε σύσπαση. Σε κάθε χτύπο, η αριστερή κοιλία της καρδιάς στέλνει αίμα στην αορτή, η οποία, λόγω της ελαστικότητάς της, διαστέλλεται και στη συνέχεια επανέρχεται στη φυσιολογική της κατάσταση. Αυτή η διαδικασία δημιουργεί ένα κύμα πίεσης, γνωστό ως σφυγμικό κύμα, το οποίο μεταδίδεται στις περιφερειακές αρτηρίες. Η μέτρηση του καρδιακού ρυθμού (σε παλμούς ανά λεπτό) παρέχει μια ένδειξη της καρδιακής λειτουργίας και μπορεί να επηρεάζεται από παράγοντες όπως η φυσική δραστηριότητα, το στρες και η κατάσταση της υγείας. Ανωμαλίες στη συχνότητα των παλμών, όπως ταχυκαρδία ή βραδυκαρδία, μπορεί να σχετίζονται με υποκείμενες παθολογικές καταστάσεις, όπως καρδιακή ανεπάρκεια, αναιμία ή προβλήματα στο αυτόνομο νευρικό σύστημα. Τα έξυπνα ρολόγια χρησιμοποιούν οπτικούς αισθητήρες για την καταγραφή του καρδιακού παλμού, προσφέροντας στους χρήστες άμεση ενημέρωση για τις μεταβολές στη συχνότητα των παλμών τους. Αυτή η δυνατότητα επιτρέπει την καλύτερη κατανόηση της φυσιολογικής λειτουργίας της καρδιάς και μπορεί να αξιοποιηθεί για τη βελτίωση της φυσικής κατάστασης και της διαχείρισης του στρες. [21], [22]

Μεταβλητότητα Καρδιακών Παλμών (HRV)

Η μεταβλητότητα της καρδιακής συχνότητας (HRV) αναφέρεται στη διακύμανση του χρονικού διαστήματος μεταξύ διαδοχικών καρδιακών παλμών. Αποτελεί δείκτη της ικανότητας του οργανισμού να προσαρμόζεται σε διάφορες συνθήκες, όπως η σωματική άσκηση, η χαλάρωση ή οι καθημερινές δραστηριότητες. Μια υψηλή HRV συνδέεται με καλή καρδιαγγειακή υγεία και μεγαλύτερη ικανότητα προσαρμογής σε στρεσογόνες καταστάσεις, ενώ μια χαμηλή HRV μπορεί να αποτελεί ένδειξη αυξημένου κινδύνου για καρδιαγγειακές παθήσεις, χρόνιο στρες ή νευρολογικές διαταραχές. Η HRV χρησιμοποιείται στην έρευνα και στην κλινική πράξη για την εκτίμηση του καρδιακού τόνου, την παρακολούθηση της επίδρασης του στρες στον οργανισμό, καθώς και για τη διάγνωση διαταραχών, όπως η νευροκαρδιογενής συγκοπή και οι αρρυθμίες. Τα έξυπνα ρολόγια και άλλες φορητές συσκευές μπορούν να μετρήσουν την HRV, χρησιμοποιώντας δεδομένα από τον καρδιακό παλμό ή το ηλεκτροκαρδιογράφημα. Η παρακολούθηση αυτής της παραμέτρου μπορεί να βοηθήσει στην κατανόηση του επιπέδου της φυσιολογικής ισορροπίας του σώματος, επιτρέποντας στους χρήστες να διαχειρίζονται καλύτερα την ένταση της άσκησης, την ποιότητα του ύπνου και τη γενική ευεξία τους. [23]

2.2.2 Παθήσεις της Καρδιάς

Η καρδιά είναι ένα από τα πιο ζωτικά όργανα του ανθρώπινου σώματος, και οποιαδήποτε διαταραχή στη λειτουργία της μπορεί να επηρεάσει σημαντικά την υγεία. Οι παθήσεις της καρδιάς περιλαμβάνουν ένα ευρύ φάσμα διαταραχών που επηρεάζουν τη ροή του αίματος, τον καρδιακό ρυθμό και τη συνολική απόδοσή της. [24], [25]

1. Στεφανιαία Νόσος (Στεφανιαία Αθηροσκλήρωση)

Η στεφανιαία νόσος εμφανίζεται όταν οι αρτηρίες που τροφοδοτούν την καρδιά με αίμα στενεύουν λόγω της συσσώρευσης λιπαρών ουσιών, σχηματίζοντας αθηρωματικές πλάκες. Αυτή η κατάσταση μειώνει τη ροή του αίματος στην καρδιά και μπορεί να οδηγήσει σε πόνο στο στήθος (στηθάγχη) ή, σε πιο σοβαρές περιπτώσεις, σε έμφραγμα του μυοκαρδίου.

2. Καρδιακή Ανεπάρκεια

Η καρδιακή ανεπάρκεια συμβαίνει όταν η καρδιά αδυνατεί να αντλήσει αρκετό αίμα για να καλύψει τις ανάγκες του σώματος. Μπορεί να προκληθεί από προηγούμενες καρδιακές παθήσεις, υψηλή αρτηριακή πίεση ή άλλες καταστάσεις που επιβαρύνουν την καρδιακή λειτουργία. Τα συμπτώματα περιλαμβάνουν κόπωση, δύσπνοια και κατακράτηση υγρών.

3. Αρρυθμίες

Οι αρρυθμίες είναι διαταραχές στον καρδιακό ρυθμό, όπου η καρδιά μπορεί να χτυπά πολύ γρήγορα (ταχυκαρδία), πολύ αργά (βραδυκαρδία) ή με ακανόνιστο τρόπο. Οι πιο συχνές αρρυθμίες περιλαμβάνουν την κολπική μαρμαρυγή, τις κοιλιακές ταχυκαρδίες και τα έκτακτα κοιλιακά ή κοιλικά συστολικά επεισόδια.

4. Βαλβιδοπάθειες

Οι παθήσεις των καρδιακών βαλβίδων προκύπτουν όταν οι βαλβίδες της καρδιάς δεν ανοίγουν ή δεν κλείνουν σωστά, επηρεάζοντας την κυκλοφορία του αίματος. Αυτές οι διαταραχές μπορεί να είναι συγγενείς ή να αναπτυχθούν με την πάροδο του χρόνου λόγω λοιμώξεων, εκφυλιστικών μεταβολών ή άλλων καταστάσεων.

5. Υπέρταση (Υψηλή Αρτηριακή Πίεση)

Η υπέρταση είναι μια χρόνια κατάσταση όπου η πίεση του αίματος στις αρτηρίες είναι υψηλότερη από το φυσιολογικό, γεγονός που αυξάνει τον κίνδυνο εμφάνισης καρδιακών παθήσεων. Εάν δεν ελεγχθεί, μπορεί να οδηγήσει σε επιπλοκές όπως καρδιακή ανεπάρκεια, έμφραγμα ή εγκεφαλικό επεισόδιο.

6. Μυοκαρδιοπάθειες

Οι μυοκαρδιοπάθειες είναι παθήσεις που επηρεάζουν τον καρδιακό μυ, προκαλώντας τη διόγκωση, τη σκλήρυνση ή την αδυναμία της καρδιάς να συσπάται σωστά. Μπορεί να είναι γενετικές ή να προκληθούν από άλλες παθολογικές καταστάσεις, όπως λοιμώξεις ή υπερβολική κατανάλωση αλκοόλ.

7. Περικαρδίτιδα

Η περικαρδίτιδα είναι η φλεγμονή του περικαρδίου, του λεπτού υμένα που περιβάλλει την καρδιά. Συχνά προκαλεί πόνο στο στήθος, ο οποίος επιδεινώνεται με την αναπνοή ή την κίνηση. Μπορεί να προκληθεί από λοιμώξεις, αυτοάνοσα νοσήματα ή άλλους παράγοντες.

8. Συγγενείς Καρδιοπάθειες

Οι συγγενείς καρδιοπάθειες είναι δομικές ανωμαλίες της καρδιάς που υπάρχουν από τη

γέννηση. Μπορεί να περιλαμβάνουν ανωμαλίες στις βαλβίδες, τις κοιλίες ή τα αγγεία της καρδιάς και συχνά απαιτούν χειρουργική ή φαρμακευτική αντιμετώπιση.

Πρόληψη και Διαχείριση

Η υιοθέτηση ενός υγιεινού τρόπου ζωής, όπως η ισορροπημένη διατροφή, η τακτική άσκηση και η αποφυγή του καπνίσματος, μπορεί να μειώσει τον κίνδυνο καρδιακών παθήσεων. Επιπλέον, η χρήση έξυπνων ρολογιών για την παρακολούθηση του καρδιακού παλμού και της HRV μπορεί να βοηθήσει στην έγκαιρη ανίχνευση πιθανών προβλημάτων και στη βελτίωση της γενικής υγείας της καρδιάς. [26], [27]

2.2.3 Αρρυθμίες

2.2.3.1 Γενικά

Οι καρδιακές αρρυθμίες είναι διαταραχές του φυσιολογικού καρδιακού ρυθμού, που μπορεί να επηρεάζουν τη συχνότητα, τη κανονικότητα ή την αγωγή των ηλεκτρικών σημάτων στην καρδιά. Οι αρρυθμίες μπορεί να είναι αβλαβείς ή να υποδηλώνουν σοβαρές καρδιακές παθήσεις. Συνήθως, ανιχνεύονται μέσω ηλεκτροκαρδιογραφήματος (ECG), ενώ ορισμένα σύγχρονα smartwatches μπορούν να παρέχουν προειδοποιήσεις για πιθανές αρρυθμίες. [28]

Οι κύριες κατηγορίες αρρυθμιών περιλαμβάνουν:

- Ταχυκαρδίες (αυξημένος καρδιακός ρυθμός >100 παλμούς/λεπτό)
- Βραδυκαρδίες (χαμηλός καρδιακός ρυθμός <60 παλμούς/λεπτό)
- Έκτακτες συστολές (παλμοί που προκύπτουν πρόωρα ή εκτός σειράς)
- Κολπικές και κοιλιακές διαταραχές αγωγής

2.2.3.2 Τύποι Αρρυθμιών

Κολπικές Αρρυθμίες

Αφορούν διαταραχές στη λειτουργία των κόλπων, που είναι το αρχικό σημείο παραγωγής των ηλεκτρικών σημάτων της καρδιάς.

- Κολπική μαρμαρυγή (Atrial Fibrillation - AFib)

Μπορεί να είναι χρόνια ή παροξυσμική και χαρακτηρίζεται από ακανόνιστες και χαοτικές ηλεκτρικές εκφορτίσεις στους κόλπους, οδηγώντας σε ασύγχρονη σύσπαση των κοιλιών. Επιπλέον, συνδέεται με αυξημένο κίνδυνο εγκεφαλικού επεισοδίου λόγω σχηματισμού θρόμβων. Η θεραπεία της περιλαμβάνει αντιπηκτικά, αντιαρρυθμικά φάρμακα και σε ορισμένες περιπτώσεις κατάλυση με καθετήρα (ablation).

- Κολπικός πτερυγισμός (Atrial Flutter)

Παρόμοιο με την κολπική μαρμαρυγή, αλλά με πιο οργανωμένη και επαναλαμβανόμενη ηλεκτρική δραστηριότητα. Η αγωγή μπορεί να περιλαμβάνει φάρμακα ή καθετηριακή κατάλυση.

Κοιλιακές Αρρυθμίες

Οι κοιλιακές αρρυθμίες προέρχονται από τις κοιλίες της καρδιάς και μπορεί να είναι σοβαρές ή απειλητικές για τη ζωή.

- Κοιλιακή ταχυκαρδία (Ventricular Tachycardia - VT)

Χαρακτηρίζεται από γρήγορους, τακτικούς καρδιακούς παλμούς που προέρχονται από τις κοιλίες και αν παραμείνει για παρατεταμένο διάστημα, μπορεί να οδηγήσει σε αιμοδυναμική αστάθεια. Σε σοβαρές περιπτώσεις μάλιστα, απαιτείται απινίδωση ή εμφύτευση απινιδωτή (ICD).

- Κοιλιακή μαρμαρυγή (Ventricular Fibrillation - VFib)

Ακανόνιστη και χαοτική ηλεκτρική δραστηριότητα στις κοιλίες, που τις εμποδίζει να αντλήσουν αίμα. Θεωρείται επείγουσα κατάσταση που απαιτεί άμεση απινίδωση (defibrillation) για να αποτραπεί η καρδιακή ανακοπή.

Κολποκοιλιακές και Διαταραχές Αγωγής

Περιλαμβάνουν δυσλειτουργίες στη μετάδοση των ηλεκτρικών σημάτων μέσω του συστήματος αγωγής της καρδιάς.

- Κολποκοιλιακός αποκλεισμός (AV Block)

Πρόκειται για διαταραχή στην αγωγή των ηλεκτρικών σημάτων από τους κόλπους προς τις κοιλίες μέσω του κολποκοιλιακού κόμβου (AV node). Μπορεί να είναι ήπιος (με καθυστέρηση στην αγωγή) ή σοβαρός (με πλήρη αποκλεισμό), προκαλώντας βραδυκαρδία και πιθανή συγκοπή. Σε σοβαρές περιπτώσεις απαιτείται εμφύτευση βηματοδότη.

- Σύνδρομο νοσούντος φλεβόκομβου (Sick Sinus Syndrome - SSS)

Είναι η δυσλειτουργία του φλεβόκομβου (SA node), που οδηγεί σε εναλλαγές μεταξύ ταχυκαρδίας και βραδυκαρδίας και συχνά απαιτείται η τοποθέτηση βηματοδότη.

2.2.3.3 Ανίχνευση

Η διάγνωση των αρρυθμιών γίνεται με διάφορους τρόπου;:

- Ηλεκτροκαρδιογράφημα (ECG): Βασική διαγνωστική εξέταση για την ανίχνευση αρρυθμιών.
- Holter Monitor: Φορητή συσκευή που καταγράφει συνεχώς τον καρδιακό ρυθμό για 24-48 ώρες.
- Ηλεκτροφυσιολογική Μελέτη (EPS): Καθετηριακή εξέταση για τη χαρτογράφηση της ηλεκτρικής δραστηριότητας της καρδιάς.
- Smartwatches και φορητές ECG συσκευές: Ορισμένες έξυπνες συσκευές μπορούν να ανιχνεύσουν βασικές αρρυθμίες, αλλά δεν αντικαθιστούν την ιατρική διάγνωση.

2.2.3.4 Συμπτώματα και Διαχείριση

Τα συμπτώματα των αρρυθμιών ποικίλλουν από ήπια έως σοβαρά και περιλαμβάνουν αίσθημα παλμών, ζάλη και αίσθημα λιποθυμίας, δύσπνοια, συγκοπή (λιποθυμία) και πόνο

στο στήθος.

Η αντιμετώπιση των αρρυθμιών εξαρτάται από τον τύπο και τη σοβαρότητά τους:

- Φαρμακευτική αγωγή: Αντιαρρυθμικά, β-αναστολείς, αποκλειστές διαύλων ασβεστίου, αντιπηκτικά.
- Ηλεκτρική καρδιοανάταξη: Εφαρμογή ηλεκτρικού σοκ για την αποκατάσταση του φυσιολογικού ρυθμού.
- Καθετηριακή κατάλυση (Ablation): Χρήση ραδιοσυχνοτήτων ή κρυοθεραπείας για την εξάλειψη παθολογικών ηλεκτρικών οδών.
- Βηματοδότης (Pacemaker): Για ασθενείς με σοβαρή βραδυκαρδία ή σύνδρομο νοσούντος φλεβόκομβου.
- Εμφυτεύσιμος καρδιομετατροπέας-απινιδωτής (ICD): Προλαμβάνει τον αιφνίδιο καρδιακό θάνατο σε ασθενείς με επικίνδυνες κοιλιακές αρρυθμίες.

Η έγκαιρη διάγνωση και η κατάλληλη θεραπεία είναι κρίσιμες για την πρόληψη επιπλοκών και την προστασία της καρδιακής λειτουργίας.

2.2.4 AV Block

2.2.4.1 Γενικά

Το AV Block (Κολποκοιλιακός αποκλεισμός) ανήκει στην κατηγορία των αρρυθμιών, καθώς πρόκειται για διαταραχή στη μετάδοση των ηλεκτρικών σημάτων από τους κόλπους προς τις κοιλίες της καρδιάς μέσω του κολποκοιλιακού κόμβου (AV node). Κανονικά, το ηλεκτρικό σήμα που ξεκινά από τον φλεβόκομβο (SA node) ταξιδεύει μέσω των κόλπων και στη συνέχεια περνά στον κολποκοιλιακό κόμβο (AV node), από όπου κατευθύνεται προς τις κοιλίες. Στο AV Block, αυτή η μετάδοση επιβραδύνεται ή ακόμα και διακόπτεται, με αποτέλεσμα ανωμαλίες στον καρδιακό ρυθμό. [29]

2.2.4.2 Τύποι AV Block

1. Πρώτου βαθμού (First-degree AV Block)

- Το ηλεκτρικό σήμα καθυστερεί περισσότερο από το φυσιολογικό, αλλά τελικά φτάνει στις κοιλίες.
- Συχνά είναι ασυμπτωματικό και ανιχνεύεται τυχαία σε ένα ηλεκτροκαρδιογράφημα.
- Δεν απαιτείται θεραπεία στις περισσότερες περιπτώσεις.

2. Δευτέρου βαθμού (Second-degree AV Block)

- Ορισμένα ηλεκτρικά σήματα δεν φτάνουν στις κοιλίες, προκαλώντας παροδικές απώλειες καρδιακών παλμών.
- Διακρίνεται σε:
 - Mobitz Type I (Wenckebach): Προοδευτική επιβράδυνση της αγωγής μέχρι που ένας παλμός δεν μεταδίδεται. Συνήθως είναι καλοήθης.

- Mobitz Type II: Ξαφνική αποτυχία στη μετάδοση ενός παλμού χωρίς προηγούμενη επιβράδυνση. Είναι πιο σοβαρή μορφή και μπορεί να απαιτήσει θεραπεία.

3. Τρίτου βαθμού (Third-degree AV Block - πλήρης αποκλεισμός)

- Καμία ηλεκτρική ώθηση δεν μεταδίδεται από τους κόλπους στις κοιλίες.
- Η καρδιά βασίζεται σε πιο αργούς εφεδρικούς ρυθμούς από κατώτερα κέντρα αγωγής, οδηγώντας σε σημαντική μείωση της καρδιακής συχνότητας.
- Συχνά απαιτείται βηματοδότης για να αποκατασταθεί ο φυσιολογικός ρυθμός.

2.2.4.3 Ανίχνευση

Το AV Block ανιχνεύεται μέσω ηλεκτροκαρδιογραφήματος (ECG), το οποίο μπορεί να γίνει είτε σε ιατρικό περιβάλλον είτε μέσω φορητών συσκευών, όπως τα έξυπνα ρολόγια με δυνατότητα καταγραφής ECG. Ορισμένα smartwatches μπορούν να καταγράψουν βασικά ηλεκτρικά σήματα και να ανιχνεύσουν αρρυθμίες, αλλά η διάγνωση απαιτεί επιβεβαίωση από καρδιολόγο.

2.2.4.4 Συμπτώματα και Διαχείριση

Ανάλογα με τη σοβαρότητα, το AV Block μπορεί να είναι ασυμπτωματικό ή να προκαλεί συμπτώματα όπως ζάλη, αίσθημα κόπωσης, συγκοπή (λιποθυμία) και βραδυκαρδία. Η αντιμετώπιση εξαρτάται από τον τύπο και τη σοβαρότητα του αποκλεισμού. Σε σοβαρές περιπτώσεις, η εμφύτευση βηματοδότη είναι η κύρια θεραπευτική επιλογή.

2.3 Υπολογισμός Αρρυθμιών από Single-Channel ECG

2.3.1 Γενικά

Η ανίχνευση αρρυθμιών από ένα μονοκαναλικό ηλεκτροκαρδιογράφημα (single-channel ECG) βασίζεται στην ανάλυση της χρονικής διάρκειας και της μορφολογίας των ηλεκτρικών σημάτων της καρδιάς. Παρόλο που ένα μονοκάναλο ECG δεν προσφέρει την ίδια ακρίβεια με ένα πολύκαναλο, μπορεί να ανιχνεύσει σημαντικές διαταραχές του καρδιακού ρυθμού.

Η ανάλυση αρρυθμιών από single-channel ECG μέσω έξυπνων ρολογιών είναι ένα ισχυρό εργαλείο για την παρακολούθηση του καρδιακού ρυθμού και την ανίχνευση ανωμαλιών όπως κολπική μαρμαρυγή, έκτακτες συστολές και βραδυκαρδία. Αν και δεν αντικαθιστά τη διάγνωση από έναν ειδικό, μπορεί να παρέχει πολύτιμες πληροφορίες για τη γενική υγεία της καρδιάς και να βοηθήσει στην έγκαιρη αναγνώριση πιθανών καρδιακών προβλημάτων. [30], [31]

2.3.2 Διαδικασία Ανάλυσης Αρρυθμιών σε Single-Channel ECG

2.3.2.1 Εντοπισμός του R-Peak και Υπολογισμός του RR-Interval

Το πρώτο βήμα στην ανάλυση ενός ECG είναι η ανίχνευση των κορυφών R στο σύμπλεγμα QRS. Οι αποστάσεις μεταξύ διαδοχικών R-peaks σχηματίζουν τα RR-Intervals, τα οποία

χρησιμοποιούνται για τον υπολογισμό της καρδιακής συχνότητας και τη διάγνωση ανωμαλιών.

Υπολογισμός καρδιακής συχνότητας (HR):

$$HR = \frac{60}{\text{Μέσος όρος } RR} \left(\frac{\text{παλμοί}}{\text{λεπτό}}, bpm \right)$$

Όπου το RR-Interval είναι σε δευτερόλεπτα.

2.3.2.2 Ανίχνευση Ανώμαλων Καρδιακών Ρυθμών

Χρησιμοποιώντας την ανάλυση των RR-Intervals, μπορούμε να ανιχνεύσουμε διάφορους τύπους αρρυθμιών:

1. Φυσιολογικός Ρυθμός (Normal Sinus Rhythm - NSR)

- Σταθερό HR μεταξύ 60-100 bpm.
- Ομοιόμορφα RR-Intervals.
- Κανονικό σχήμα κυματομορφών P-QRS-T.

2. Φλεβοκομβική Βραδυκαρδία (Sinus Bradycardia)

- $HR < 60$ bpm.
- Σταθερά RR-Intervals αλλά με μεγάλη απόσταση μεταξύ των R-peaks.
- Μπορεί να είναι φυσιολογική σε αθλητές ή αποτέλεσμα φαρμάκων.

3. Φλεβοκομβική Ταχυκαρδία (Sinus Tachycardia)

- $HR > 100$ bpm.
- Κανονική ακολουθία P-QRS-T, αλλά με μικρότερα RR-Intervals.
- Συχνά σχετίζεται με άγχος, άσκηση ή αφυδάτωση.

4. Κολπική Μαρμαρυγή (Atrial Fibrillation - AFib)

- Ακανόνιστα RR-Intervals (καμία σταθερότητα στον ρυθμό).
- Απουσία διακριτών κυμάτων P.
- Ανιχνεύεται με μεταβλητότητα HRV και ακανόνιστους παλμούς.
- Smartwatches όπως το Apple Watch και το Fitbit υποστηρίζουν την ανίχνευση AFib με αλγόριθμους που βασίζονται σε ανωμαλίες των RR-Intervals.

5. Extrasystoles (Έκτακτες Συστολές - PVCs & PACs)

- Ξαφνική εμφάνιση πρόωρων παλμών.
- Μπορεί να είναι πρόωρες κολπικές συστολές (PACs) ή πρόωρες κοιλιακές συστολές (PVCs).

- Τα PVCs έχουν ευρύ σύμπλεγμα QRS, ενώ τα PACs έχουν κανονικό QRS αλλά εμφανίζονται πρόωρα.

6. Κολποκοιλιακός Αποκλεισμός (AV Block)

- Πρώτου βαθμού: Μεγάλη διάρκεια PR-διαστήματος (> 200 ms).
- Δευτέρου βαθμού: Ορισμένα QRS σύμπλοκα λείπουν.
- Τρίτου βαθμού (πλήρης αποκλεισμός): Ασύνδετα P-waves και QRS.

7. Κουλιακή Ταχυκαρδία (Ventricular Tachycardia - VTach)

- HR > 150 bpm, με ευρεία QRS σύμπλοκα.
- Εάν διαρκεί >30 sec, είναι απειλητική για τη ζωή.

8. Κουλιακή Μαρμαρυγή (Ventricular Fibrillation - VFib)

- Χαοτικό μοτίβο χωρίς οργανωμένο ρυθμό.
- Χρειάζεται άμεση ιατρική παρέμβαση.

2.3.3 Περιορισμοί του Single-Channel ECG

Το ηλεκτροκαρδιογράφημα από ένα κανάλι, δεν παρέχει πλήρη εικόνα της ηλεκτρικής δραστηριότητας της καρδιάς, όπως τα 12-κάναλα ιατρικά ECG. Υπάρχει πιθανότητα λανθασμένης ανίχνευσης αρρυθμίες λόγω κίνησης του χεριού, ατελούς επαφής του αισθητήρα ή παρεμβολών. Επιπλέον, δεν μπορεί να εντοπίσει ισχαιμικές αλλαγές που ανιχνεύονται μόνο με πλήρη καρδιολογική εξέταση.

2.4 Εντοπισμός AV Block από Single-Channel ECG

2.4.1 Γενικά

Ο εντοπισμός του Atrioventricular (AV) Block από μονοκάναλο ηλεκτροκαρδιογράφημα (single-channel ECG) μπορεί να γίνει με την ανάλυση της σχέσης μεταξύ των κυμάτων P και των συμπλεγμάτων QRS, καθώς και της διάρκειας του διαστήματος PR. [32]

2.4.2 Βήματα εντοπισμού

1. Εντοπισμός των κυμάτων P και QRS

- Το κύμα P αντιπροσωπεύει την κολπική δραστηριότητα.
- Το σύμπλεγμα QRS αντιπροσωπεύει τη διέγερση των κοιλιών.

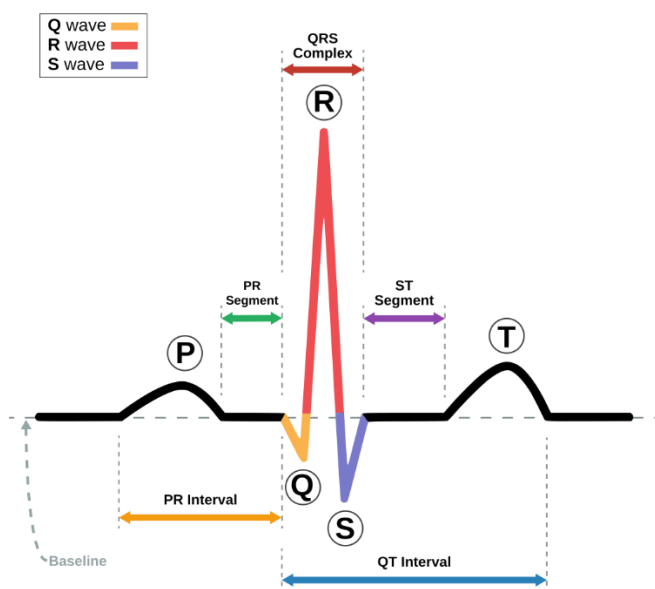
2. Ανάλυση του διαστήματος PR

- Φυσιολογικό διάστημα PR: 120-200 ms (3-5 μικρά τετράγωνα στο ΗΚΓ 25 mm/s).
- Παρατεταμένο PR > 200 ms μπορεί να υποδηλώνει AV Block.

3. Διάκριση των τύπων AV Block

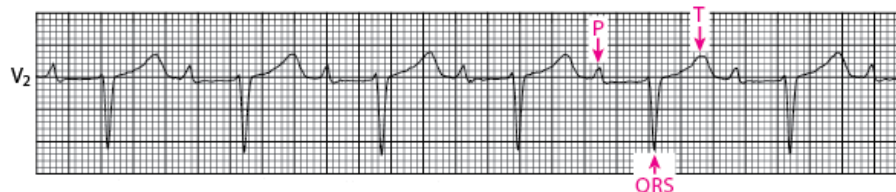
Πίνακας 1. Τύποι AV Block και χαρακτηριστικά τους

Τύπος AV Block	Κλινικά Χαρακτηριστικά στο Single-Channel ECG
1ου βαθμού	Σταθερό και παρατεταμένο PR (>200ms), χωρίς απώλεια QRS
2ου βαθμού Mobitz I (Wenckebach)	Σταδιακή αύξηση του PR μέχρι να χαθεί ένα QRS (πρόοδος)
2ου βαθμού Mobitz II	Απότομα απολεσθέντα QRS (χωρίς προοδευτική αύξηση του PR)
3ου βαθμού (Complete AV Block)	Καμία σχέση P-QRS, ανεξάρτητη εκπόλωση κόλπων και κοιλιών



Εικόνα 4. Ανάλυση του συμπλέγματος QRS σε ηλεκτροκαρδιογράφημα

Για τη διάγνωση ενός AV Block από ένα μονοκάναλο ΗΚΓ, πρέπει να εξετάσουμε τον αριθμό των κυμάτων P σε σχέση με τα QRS. Αν υπάρχουν περισσότερα P από QRS, είναι πιθανό να υπάρχει αποκλεισμός AV. Επίσης, αν το διάστημα PR είναι σταθερό ή μεταβαλλόμενο, μπορεί να μας βοηθήσει να διακρίνουμε μεταξύ Mobitz I και Mobitz II. Τέλος, αν οι κόλποι και οι κοιλίες αποπολώνονται ανεξάρτητα, τότε πρόκειται για πλήρη αποκλεισμό 3ου βαθμού. Στις παρακάτω εικόνες φαίνονται παραδείγματα ηλεκτροκαρδιογραφημάτων που παρουσιάζουν όλα τα ήδη κολποκοιλιακού αποκλεισμού.



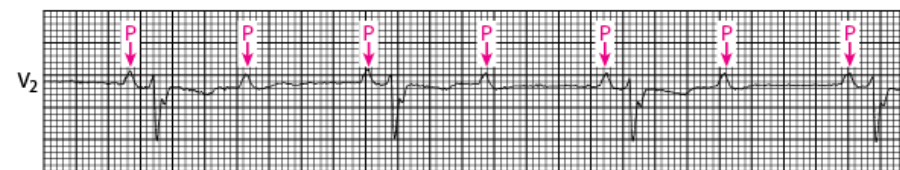
Εικόνα 5. Πρώτου βαθμού AV Block



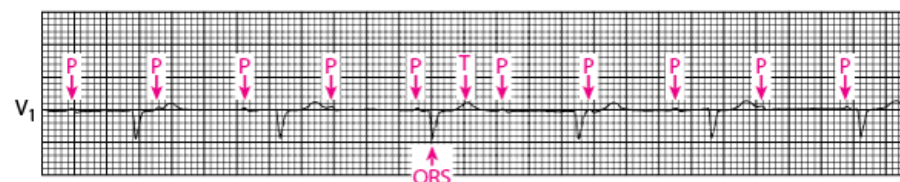
Εικόνα 6. Δεύτερου βαθμού Mobitz I (Wenckebach) AV Block



Εικόνα 7. Δεύτερου βαθμού Mobitz II AV Block



Εικόνα 8. Δεύτερου βαθμού (High Grade) AV Block



Εικόνα 9. Τρίτου βαθμού AV Block

2.5 Οξυγόνο στο αίμα

2.5.1 Γενικά

Το επίπεδο οξυγόνου στο αίμα αναφέρεται στην ποσότητα οξυγόνου που κυκλοφορεί μέσω της κυκλοφορίας του αίματος και διανέμεται στα κύτταρα του σώματος. Το οξυγόνο, το οποίο είναι ζωτικής σημασίας για την επιβίωση, εισέρχεται στο σώμα μέσω της αναπνοής και μεταφέρεται στους ιστούς μέσω της κυκλοφορίας του αίματος. Τα κύτταρα το χρησιμοποιούν για να παράγουν ενέργεια και στη συνέχεια αποβάλλουν διοξείδιο του άνθρακα, το οποίο απομακρύνεται από τον οργανισμό μέσω των πνευμόνων κατά την εκπνοή. Η ισορροπία των επιπέδων οξυγόνου ρυθμίζεται αυστηρά από το σώμα, καθώς η ανεπάρκειά του μπορεί να οδηγήσει σε σοβαρές επιπλοκές, επηρεάζοντας ιδιαίτερα τον εγκέφαλο και την καρδιά. Αν τα επίπεδα οξυγόνου μειωθούν υπερβολικά, μπορεί να υποδηλώνει προβλήματα στη λειτουργία των πνευμόνων ή του κυκλοφορικού συστήματος, γεγονός που απαιτεί άμεση προσοχή. [33]

2.5.2 Υποδειγματικές τιμές

Το επίπεδο κορεσμού του οξυγόνου στο αίμα μετράται σε ποσοστό (%) και αναφέρεται στην ποσότητα οξυγόνου που μεταφέρεται μέσω της αιμοσφαιρίνης στα ερυθρά αιμοσφαίρια. Οι φυσιολογικές τιμές κυμαίνονται ως εξής:

- **95-100%:** Φυσιολογικό επίπεδο οξυγόνου στο αίμα.
- **90-94%:** Ήπια υποξαιμία, μπορεί να απαιτεί παρακολούθηση, ιδιαίτερα σε άτομα με αναπνευστικά προβλήματα.
- **Κάτω από 90%:** Ενδεικτικό υποξαιμίας, απαιτεί ιατρική αξιολόγηση.
- **Κάτω από 80%:** Επικίνδυνα χαμηλό επίπεδο, μπορεί να οδηγήσει σε σοβαρές επιπλοκές όπως βλάβες σε ζωτικά όργανα.

Η μέτρηση του οξυγόνου γίνεται συνήθως με οξύμετρο (μέτρηση κορεσμού οξυγόνου – SpO₂) ή μέσω αιματολογικής εξέτασης αερίων αίματος (PaO₂).

2.5.3 Έλλειψη οξυγόνου στο αίμα

Η έλλειψη οξυγόνου στο αίμα, γνωστή ως υποξαιμία, μπορεί να προκαλέσει σοβαρά προβλήματα στον οργανισμό, καθώς το οξυγόνο είναι απαραίτητο για τη σωστή λειτουργία των κυττάρων και των οργάνων. Ανάλογα με τη σοβαρότητα της κατάστασης, τα συμπτώματα μπορεί να περιλαμβάνουν:

- Δύσπνοια ή δυσκολία στην αναπνοή
- Ταχυκαρδία ή αρρυθμίες
- Ζάλη ή αίσθημα σύγχυσης
- Κυάνωση (μπλε απόχρωση στα χείλη, τα νύχια ή το δέρμα)
- Πονοκέφαλος
- Κόπωση και αδυναμία
- Διαταραχές όρασης ή απώλεια συγκέντρωσης
- Σε σοβαρές περιπτώσεις, απώλεια συνείδησης

Η υποξαιμία μπορεί να οφείλεται σε διάφορες παθήσεις που επηρεάζουν την αναπνευστική ή την κυκλοφορική λειτουργία. Οι κύριες αιτίες περιλαμβάνουν:

- Χρόνια Αποφρακτική Πνευμονοπάθεια (ΧΑΠ): Περιορίζει τη ροή του αέρα στους πνεύμονες, δυσκολεύοντας την ανταλλαγή αερίων.
- Άσθμα: Φλεγμονή και στένωση των αεραγωγών, μειώνοντας την ποσότητα οξυγόνου που φτάνει στους πνεύμονες.
- Πνευμονία: Λοίμωξη των πνευμόνων που μειώνει την ικανότητα πρόσληψης οξυγόνου.

- Πνευμονική εμβολή: Απόφραξη αγγείου στους πνεύμονες, που εμποδίζει τη ροή του αίματος και την οξυγόνωση.
- Ύπνικη άπνοια: Διακοπές της αναπνοής κατά τον ύπνο που μειώνουν τα επίπεδα οξυγόνου.
- Καρδιακή ανεπάρκεια: Η καρδιά δεν αντλεί αποτελεσματικά το αίμα, μειώνοντας την οξυγόνωση των ιστών.
- Αναιμία: Χαμηλή ποσότητα αιμοσφαιρίνης, η οποία είναι απαραίτητη για τη μεταφορά του οξυγόνου.
- Πνευμονική ίνωση: Βλάβη στους πνεύμονες που μειώνει την ικανότητα πρόσληψης και διάχυσης οξυγόνου.

Η χαμηλή οξυγόνωση του αίματος μπορεί να οδηγήσει σε σοβαρές επιπλοκές και απαιτεί ιατρική αξιολόγηση. Ανάλογα με την αιτία, η αντιμετώπιση μπορεί να περιλαμβάνει οξυγονοθεραπεία ή θεραπεία της υποκείμενης πάθησης.

2.6 Ανάλυση Βιοηλεκτρικής Εμπέδησης (BIA)

2.6.1 Γενικά

Η Ανάλυση Βιοηλεκτρικής Εμπέδησης (BIA) είναι μια τεχνική που χρησιμοποιείται για τη μέτρηση της σύστασης του σώματος, συμπεριλαμβανομένου του ποσοστού σωματικού λίπους, της μυϊκής μάζας και της συνολικής ενυδάτωσης. Η μέθοδος λειτουργεί στέλνοντας μικρά ηλεκτρικά ρεύματα μέσω του σώματος και μετρώντας την αντίσταση (εμπέδηση) που παρουσιάζουν οι ιστοί του σώματος στο ρεύμα. [34]

2.6.2 Βασικές αρχές λειτουργίας της BIA

Όπως είναι ευρέως διαδεδομένο, το νερό στο σώμα είναι καλός αγωγός του ηλεκτρικού ρεύματος, ενώ το λίπος προσφέρει μεγαλύτερη αντίσταση, έτσι μέσω της διαφοράς αυτής, υπολογίζεται το ποσοστό λίπους και η μυϊκή μάζα. Οι σύγχρονες συσκευές BIA μπορούν να παρέχουν αναλυτικά δεδομένα για το ποσοστό λίπους, το μυϊκό ιστό και το επίπεδο ενυδάτωσης του σώματος.

2.6.3 Χρήσεις της BIA

Η Ανάλυση Βιοηλεκτρικής Εμπέδησης χρησιμοποιείται για τον υπολογισμό βασικών δεικτών, όπως το ποσοστό λίπους, η μυϊκή μάζα, η οστική μάζα, τα συνολικά υγρά του σώματος και ο Βασικός Μεταβολικός Ρυθμός (BMR). Αυτά τα δεδομένα είναι ιδιαίτερα χρήσιμα για την παρακολούθηση της φυσικής κατάστασης, τη διαχείριση του βάρους και τη βελτίωση της αθλητικής απόδοσης. Παρακολουθεί την ενυδάτωση του σώματος και μπορεί να εκτιμήσει τον μεταβολικό ρυθμό του κάθε οργανισμού. Η BIA μπορεί επίσης να βοηθήσει στη δημιουργία εξατομικευμένων προγραμμάτων διατροφής και άσκησης, παρέχοντας χρήσιμες πληροφορίες για τη γενική υγεία και τον μεταβολισμό του χρήστη.

2.6.4 Παράγοντες που επηρεάζουν την ακρίβεια της BIA

Η ακρίβεια των μετρήσεων BIA στο Galaxy Watch 4 επηρεάζεται από διάφορους παράγοντες, οι οποίοι μπορούν να οδηγήσουν σε αποκλίσεις στα αποτελέσματα. Ένας από τους κυριότερους είναι η κατανάλωση υγρών πριν τη μέτρηση. Η ποσότητα νερού στο σώμα μπορεί να επηρεάσει τη βιοηλεκτρική αγωγιμότητα, συνεπώς η υπερβολική ή μειωμένη πρόσληψη υγρών πριν από τη μέτρηση μπορεί να αλλοιώσει τις τιμές της λιπώδους και άλυτης μάζας. Επιπλέον, η ηλεκτρολυτική ισορροπία του οργανισμού, δηλαδή τα επίπεδα ηλεκτρολυτών, όπως το νάτριο και το κάλιο, επηρεάζουν την αγωγιμότητα του σώματος. Αν υπάρχει ανισορροπία (π.χ. μετά από έντονη άσκηση ή αφυδάτωση), τα αποτελέσματα μπορεί να είναι λιγότερο ακριβή. Η ώρα της μέτρησης σίγουρα παίζει βασικό ρόλο. Οι μετρήσεις είναι πιο ακριβείς όταν γίνονται το πρωί, πριν την κατανάλωση τροφής ή υγρών. Κατά τη διάρκεια της ημέρας, η πρόσληψη τροφής και η κατακράτηση υγρών μπορούν να αλλάξουν τη σύνθεση του σώματος, επηρεάζοντας την αγωγιμότητα και κατ' επέκταση την ακρίβεια της μέτρησης. Γενικά, για την καλύτερη ακρίβεια, συνιστάται οι μετρήσεις να γίνονται υπό σταθερές συνθήκες, ιδανικά την ίδια ώρα κάθε μέρα και πριν από την κατανάλωση φαγητού ή υγρών.

2.7 Φωτοπληθυσμογραφία (PPG)

2.7.1 Γενικά

Η Φωτοπληθυσμογραφία (PPG) είναι μια οπτική τεχνική που χρησιμοποιείται για τη μέτρηση αλλαγών στη ροή του αίματος με τη χρήση φωτός. Η μέθοδος αυτή είναι ευρέως διαδεδομένη σε έξυπνα ρολόγια, fitness trackers και ιατρικές συσκευές, καθώς επιτρέπει την καταγραφή βασικών βιολογικών σημάτων, όπως ο καρδιακός ρυθμός και ο κορεσμός οξυγόνου (SpO_2). [35]

2.7.2 Βασικές αρχές λειτουργίας της PPG

Μια LED πηγή φωτός (συνήθως πράσινου, κόκκινου ή υπέρυθρου φάσματος) εκπέμπει φως στο δέρμα και ένας φωτοαισθητήρας ανιχνεύει το ανακλώμενο ή το διαπερασμένο φως μέσω των αγγείων. Με βάση την απορρόφηση του φωτός, υπολογίζονται οι αλλαγές στη ροή του αίματος, οι οποίες σχετίζονται με τον καρδιακό ρυθμό και άλλες παραμέτρους.

2.7.3 Χρήσεις της PPG

- Μέτρηση καρδιακού ρυθμού (HR): Χρησιμοποιείται σε αθλητικές εφαρμογές και ιατρικές διαγνώσεις.
- Κορεσμός οξυγόνου στο αίμα (SpO_2): Χρησιμοποιείται για την εκτίμηση της υποξαιμίας και την παρακολούθηση ασθενών με αναπνευστικά προβλήματα.
- Ανίχνευση αρρυθμιών: Κάποια προηγμένα συστήματα PPG μπορούν να αναγνωρίσουν αρρυθμίες όπως την κολλική μαρμαρυγή.
- Παρακολούθηση στρες και HRV (μεταβλητότητα καρδιακού ρυθμού): Χρησιμοποιείται σε εφαρμογές ευεξίας και υγείας.

2.7.3 Πλεονεκτήματα της PPG:

Η τεχνολογία PPG προσφέρει σημαντικά πλεονεκτήματα στη μέτρηση βιοσημάτων, καθώς είναι μια μη επεμβατική μέθοδος που χρησιμοποιεί φως για την ανίχνευση μεταβολών στη ροή του αίματος, καθιστώντας τη εύκολη στη χρήση και κατάλληλη για καθημερινή παρακολούθηση. Ένα από τα κύρια πλεονεκτήματά της είναι η συνεχής και μακροχρόνια παρακολούθηση, επιτρέποντας στους χρήστες να παρακολουθούν την καρδιακή τους δραστηριότητα, το επίπεδο οξυγόνου στο αίμα και άλλους ζωτικούς δείκτες καθ' όλη τη διάρκεια της ημέρας. Επιπλέον, η PPG έχει ενσωματωθεί σε έξυπνες συσκευές, όπως smartwatches και fitness trackers, παρέχοντας υγειονομική παρακολούθηση σε πραγματικό χρόνο χωρίς την ανάγκη για εξειδικευμένο ιατρικό εξοπλισμό, διευκολύνοντας έτσι την προληπτική υγεία και τη διαχείριση χρόνιων παθήσεων.

2.7.5 Παράγοντες που επηρεάζουν την ακρίβεια της PPG

Η ακρίβεια της εξαρτάται από διάφορους παράγοντες, όπως η κίνηση του χρήστη. Η φυσική δραστηριότητα, όπως η άσκηση ή οι απότομες κινήσεις του χεριού, μπορεί να προκαλέσει θόρυβο στο σήμα, καθιστώντας τη μέτρηση λιγότερο ακριβή. Η αυξημένη ροή αίματος και οι αλλαγές στην πίεση του αγγειακού συστήματος επηρεάζουν την ανίχνευση του σφυγμού. Επιπλέον, η μελάγχρωση του δέρματος μπορεί να διαταράσσει την απορρόφηση και την ανάκλαση του φωτός που χρησιμοποιείται στη μέτρηση. Σε πιο σκούρους τόνους δέρματος ή σε περιπτώσεις αυξημένης μελάγχρωσης, η ένταση του λαμβανόμενου σήματος μπορεί να είναι χαμηλότερη, επηρεάζοντας την ακρίβεια. Τέλος, ο ισχυρός περιβαλλοντικός φωτισμός, όπως το άμεσο ηλιακό φως ή τα λαμπερά τεχνητά φώτα, μπορεί να προκαλέσει παρεμβολές στο σήμα PPG, αυξάνοντας τον θόρυβο της μέτρησης και μειώνοντας την αξιοπιστία των αποτελεσμάτων. Για τη βελτίωση της ακρίβειας, συνιστάται η χρήση της PPG σε ήρεμες συνθήκες, με σωστή εφαρμογή της συσκευής στον καρπό και αποφυγή έντονου φωτισμού.

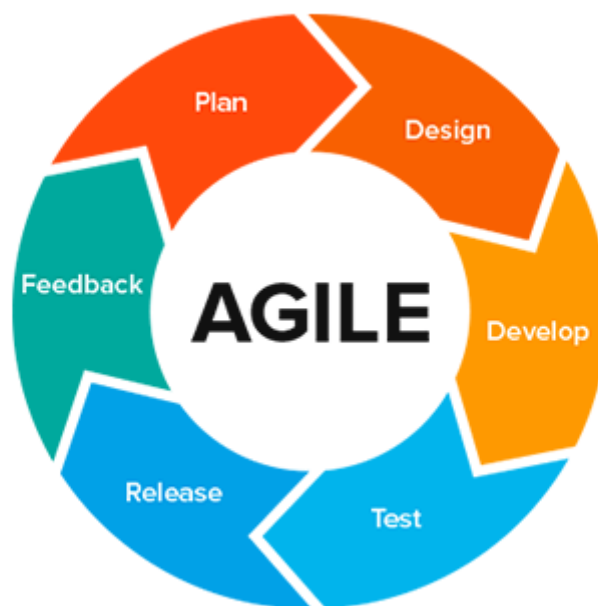
Κεφάλαιο 3. Δομή της εφαρμογής και μεθοδολογία υλοποίησής της

3.1 Εισαγωγή

Σε αυτό το κεφάλαιο, γίνεται εκτενής αναφορά στα βασικά στοιχεία της εφαρμογής και στην αλληλεπίδρασή τους. Επιπλέον, περιγράφεται η μεθοδολογία που ακολουθήθηκε για την ανάπτυξη του λογισμικού, η οποία βασίστηκε στην Agile μέθοδο, εξασφαλίζοντας ευελιξία και προσαρμοστικότητα. Τέλος, γίνεται αναφορά στους απαραίτητους πόρους και στα έξοδα που απαιτήθηκαν για την ολοκλήρωση του έργου. Στην υποενότητα που ακολουθεί, θα παρουσιαστεί αναλυτικά η εφαρμογή αυτής της μεθόδου.

3.2 Ευέλικτη μεθοδολογία (Agile)

Η ανάπτυξη της εφαρμογής για το Galaxy Watch, η οποία εκμεταλλεύεται τα δεδομένα υγείας από τους αισθητήρες και επεξεργάζεται το ECG για την ανίχνευση αρρυθμιών και AV Block, βασίστηκε στην ευέλικτη μεθοδολογία. Η Agile προσέγγιση διασφαλίζει τη σταδιακή εξέλιξη της εφαρμογής, την προσαρμοστικότητα στις ανάγκες των χρηστών και τη συνεχή βελτίωση. Η ανάπτυξη ακολούθησε τα έξι βασικά βήματα της Agile προσέγγισης. [36], [37]



Εικόνα 10. Ευέλικτη μεθοδολογία

1. Προγραμματισμός (Plan)

Το πρώτο βήμα αφορούσε τον καθορισμό των βασικών λειτουργιών της εφαρμογής. Οι στόχοι περιλάμβαναν τη συλλογή και επεξεργασία δεδομένων υγείας από τους αισθητήρες του ρολογιού, την ανάλυση του ECG για την ανίχνευση αρρυθμιών και AV Block, τη χρήση ειδοποιήσεων για ενημέρωση του χρήστη και την αποστολή γραφημάτων μέσω email. Κατά τον σχεδιασμό, δόθηκε έμφαση στη λειτουργικότητα, την ακρίβεια των υπολογισμών και

την αυτονομία της εφαρμογής, καθώς όλα τα δεδομένα επεξεργάζονται τοπικά χωρίς τη χρήση εξωτερικής βάσης δεδομένων.

2. Σχεδίαση (Design)

Βάσει των απαιτήσεων που ορίστηκαν, σχεδιάστηκε η αρχιτεκτονική της εφαρμογής. Δόθηκε ιδιαίτερη προσοχή στη βέλτιστη αξιοποίηση των υπολογιστικών πόρων του Galaxy Watch, καθώς οι υπολογισμοί γίνονται τοπικά. Ο σχεδιασμός περιλάμβανε τον ορισμό των αλγορίθμων για την ανάλυση των σημάτων ECG, τη διαχείριση των ειδοποιήσεων και την αποστολή των δεδομένων στον χρήστη μέσω email.

3. Ανάπτυξη (Develop)

Η ανάπτυξη της εφαρμογής πραγματοποιήθηκε σε επαναληπτικά στάδια, όπου κάθε νέο χαρακτηριστικό προστίθετο και ενσωματωνόταν σταδιακά στο σύστημα. Χρησιμοποιήθηκε το GitHub για τη διαχείριση των εκδόσεων, επιτρέποντας την ασφαλή αποθήκευση του κώδικα και την εύκολη αναίρεση πιθανών σφαλμάτων.

4. Δοκιμή (Test)

Μετά από κάθε νέα υλοποίηση, πραγματοποιήθηκαν εκτεταμένες δοκιμές για να διασφαλιστεί η ακρίβεια των υπολογισμών και η σωστή λειτουργία της εφαρμογής. Οι δοκιμές περιλάμβαναν τη λήψη δεδομένων από τους αισθητήρες, την ανάλυση των σημάτων ECG σε διαφορετικές συνθήκες και την αξιολόγηση της αποτελεσματικότητας των ειδοποιήσεων και της αποστολής email.

5. Διάθεση (Release)

Μετά την επιτυχή ολοκλήρωση των δοκιμών, η εφαρμογή τέθηκε σε χρήση, επιτρέποντας σε πραγματικούς χρήστες να τη δοκιμάσουν. Η διάθεση πραγματοποιήθηκε σε περιορισμένο αριθμό χρηστών για αρχική αξιολόγηση, ώστε να διασφαλιστεί η ομαλή λειτουργία πριν από μια ευρύτερη κυκλοφορία.

6. Ανάδραση (Feedback)

Η αξιολόγηση της απόδοσης της εφαρμογής βασίστηκε στις παρατηρήσεις των χρηστών και στα δεδομένα που συλλέχθηκαν από τις δοκιμές. Τα σχόλια που λήφθηκαν βοήθησαν στη βελτιστοποίηση των αλγορίθμων ECG, στη βελτίωση της διαχείρισης ειδοποιήσεων και στη διόρθωση τυχόν σφαλμάτων.

3.3 Δομή της εφαρμογής

Η παρούσα διπλωματική εργασία επικεντρώνεται στην ανάπτυξη μιας εφαρμογής για το Galaxy Watch, η οποία αξιοποιεί τα δεδομένα υγείας από τους ενσωματωμένους αισθητήρες του ρολογιού, με κύριο στόχο την επεξεργασία του ΗΚΓ (ECG) για την ανίχνευση αρρυθμιών και AV Block. Η εφαρμογή λειτουργεί αυτόνομα, χωρίς σύνδεση σε βάση δεδομένων, πραγματοποιώντας όλους τους υπολογισμούς τοπικά στη συσκευή. Επιπλέον, ενσωματώνει σύστημα ειδοποιήσεων, προειδοποιώντας τον χρήστη σε περίπτωση

εντοπισμού ανωμαλιών στον καρδιακό ρυθμό. Παράλληλα, παρέχει τη δυνατότητα αποστολής email με εικόνες και γραφήματα των αποτελεσμάτων, επιτρέποντας στον χρήστη να έχει πλήρη εικόνα της κατάστασής του. Η ανάπτυξη της εφαρμογής ακολούθησε την Agile μεθοδολογία, εξασφαλίζοντας την προοδευτική εξέλιξή της μέσω συνεχών βελτιώσεων και αξιολογήσεων, ώστε να ανταποκρίνεται με ακρίβεια στις ανάγκες του χρήστη.

3.4 Επικοινωνία εφαρμογής με το Samsung Health

Η υπηρεσία Samsung Health επιτρέπει την άμεση πρόσβαση στα δεδομένα υγείας του χρήστη, επιτρέποντας την ζωντανή καταγραφή και ανάλυση μετρήσεων από τους ενσωματωμένους αισθητήρες του Galaxy Watch. Η εφαρμογή που αναπτύχθηκε πραγματοποιεί σε πραγματικό χρόνο λήψη και επεξεργασία δεδομένων, όπως καρδιακοί παλμοί, ECG, επίπεδα δραστηριότητας, συστάσεις σώματος και κορεσμός οξυγόνου στο αίμα (SpO₂).

Ειδικά για το ηλεκτροκαρδιογράφημα (ECG), το σήμα λαμβάνεται άμεσα από τον αισθητήρα και επεξεργάζεται τοπικά, προκειμένου να ανιχνευθούν αρρυθμίες και AV Block, χωρίς τη χρήση εξωτερικής βάσης δεδομένων ή αποθήκευσης ιστορικών μετρήσεων. Οι αισθητήρες του ρολογιού ενεργοποιούνται κατά τη χρήση της εφαρμογής, εξασφαλίζοντας ότι όλες οι μετρήσεις είναι ζωντανές και δυναμικές, αντί για στατικές ανακτήσεις δεδομένων.

Ο χρήστης ενημερώνεται άμεσα μέσω ειδοποιήσεων, ενώ τα αποτελέσματα των αναλύσεων αποστέλλονται οπτικοποιημένα μέσω email με τη μορφή γραφημάτων, παρέχοντας του πλήρη εικόνα των καταγραφών του, δίνοντας τη δυνατότητα στον χρήστη για περαιτέρω ανάλυση.

3.5 Κόστος ανάπτυξης της διπλωματικής εργασίας

Η ανάπτυξη της εφαρμογής πραγματοποιήθηκε σε οικοσύστημα Microsoft Windows, με χρήση του Android Studio ως περιβάλλον ανάπτυξης και του Samsung Galaxy Watch 4 ως την κύρια συσκευή δοκιμών. Παρότι το Android Studio παρέχει Wear OS emulators, αυτοί δεν περιλαμβάνουν τους απαραίτητους αισθητήρες, με αποτέλεσμα η κατοχή πραγματικής συσκευής Galaxy Watch 4 να είναι απαραίτητη για τη σωστή λειτουργία και αξιολόγηση της εφαρμογής. Επιπλέον, απαιτήθηκε και Samsung smartphone για τη σύνδεση του ρολογιού στο διαδίκτυο και την επικοινωνία του με το Samsung Health, καθώς το ρολόι από μόνο του δεν έχει δυνατότητα αυτόνομης πρόσβασης στο διαδίκτυο χωρίς σύνδεση με κινητό.

Το Android Studio είναι ένα δωρεάν εργαλείο ανάπτυξης, ωστόσο, η πρόσβαση στα δεδομένα υγείας μέσω του Samsung Health SDK απαιτούσε εγγραφή ως Samsung Developer, η οποία είναι δωρεάν αλλά περιλαμβάνει διαδικασία έγκρισης για τη χρήση ορισμένων API. Για την πλήρη αξιοποίηση των δυνατοτήτων της εφαρμογής, το ρολόι και το κινητό έπρεπε να είναι ενημερωμένα στην τελευταία έκδοση λογισμικού, γεγονός που

μερικές φορές απαιτήσε πρόσθετες ρυθμίσεις ή αναβαθμίσεις. Όσον αφορά την κυκλοφορία της εφαρμογής, η Google επιτρέπει τη δοκιμαστική χρήση και ανάπτυξη εφαρμογών δωρεάν, αλλά για την ανάρτησή της στο Google Play Store, απαιτείται η πληρωμή ενός εφάπαξ ποσού των 25\$ για τη δημιουργία λογαριασμού προγραμματιστή. Επιπλέον, η χρήση ειδοποιήσεων και άλλων cloud υπηρεσιών (όπως αποστολή email μέσω εξωτερικών API) ενδέχεται να απαιτεί πρόσθετο κόστος, ανάλογα με τον όγκο δεδομένων και τη συχνότητα χρήσης.

Συνολικά, το κόστος ανάπτυξης της εφαρμογής παρέμεινε χαμηλό, με τις κύριες δαπάνες να αφορούν την απόκτηση των απαραίτητων συσκευών δοκιμών (Galaxy Watch 4 και Samsung smartphone), τη διαχείριση των δικαιωμάτων χρήσης του Samsung Health SDK, καθώς και το ενδεχόμενο κόστος για τη διάθεση της εφαρμογής στο Google Play Store.

Free \$0/mo	Essentials Starting \$19.95/mo	Pro Starting at \$89.95/mo	Premier Custom pricing
Send 100 emails/day for free	Send 50,000 - 100,000 emails/mo	Send 100,000 - 2,500,000 emails/mo	Customize your plan for exactly what you need
Start for free	Start for free	Start for free	Contact sales
No credit card, no commitment.	You can upgrade any time after your free account is set up.	You can upgrade any time after your free account is set up.	Need to send more? Talk to us for custom, high volume pricing.
	Taxes and overages may apply.	Taxes and overages may apply.	
What's Included:	What's Included:	What's Included:	What's Included:
✓ Analytics & deliverability optimization ✗ No access to dedicated IPs ✗ No subuser management ✗ No single sign-on (SSO)	✓ Analytics & deliverability optimization ✗ No access to dedicated IPs ✗ No subuser management ✗ No single sign-on (SSO)	✓ Analytics & deliverability optimization ✓ Dedicated IPs included ✓ Subuser management ✓ Single sign-on (SSO)	✓ Analytics & deliverability optimization ✓ Dedicated IPs included ✓ Subuser management ✓ Single sign-on (SSO)
Additional teammates: 1	Additional teammates: 1	Additional teammates: 1,000	Additional teammates: 1,000
Event webhook: 1	Event webhook: 2	Event webhook: 5	Event webhook: 5
Email validation: 0	Email validation: 0	Email validation: 2,500	Email validation: 5,000
Dynamic templates: Editor	Dynamic templates: Editor	Dynamic templates: Editor and testing	Dynamic templates: Editor and testing

Εικόνα 11. Πλάνο κοστολόγησης SendGrid

Κεφάλαιο 4. Παρουσίαση και ερμηνεία κώδικα

4.1 Εισαγωγή

Για την υλοποίηση όλων των προαναφερθεισών λειτουργιών και την επίτευξη των επιθυμητών αποτελεσμάτων, ήταν απαραίτητη η ανάπτυξη κώδικα, η οποία αποτέλεσε τον κύριο άξονα της παρούσας διπλωματικής εργασίας. Το παρόν κεφάλαιο εστιάζει στην ανάλυση του κώδικα που χρησιμοποιήθηκε για την εφαρμογή η οποία τρέχει σε Galaxy Watch 4, ανεπτυγμένη στο Android Studio, καθώς και στον τρόπο με τον οποίο η εφαρμογή αλληλεπιδρά με το Samsung Health για την ανάκτηση και επεξεργασία δεδομένων υγείας. Ειδικότερα, αναλύονται οι αλγόριθμοι που εφαρμόστηκαν για την τοπική επεξεργασία των σημάτων ECG, την ανίχνευση αρρυθμιών και AV Block, καθώς και η διαχείριση των ειδοποιήσεων και της αποστολής αποτελεσμάτων μέσω email.

4.2 Εφαρμογή στο smart watch

4.2.1 Άδεια για ειδοποιήσεις, καταγραφή μετρήσεων

Στην εφαρμογή που αναπτύχθηκε για το Galaxy Watch, ο προγραμματιστής είναι υποχρεωμένος να ζητήσει ρητή άδεια από τον χρήστη για την πρόσβαση και επεξεργασία των δεδομένων υγείας που συλλέγονται από τους αισθητήρες της συσκευής. Αυτό περιλαμβάνει τη μέτρηση και καταγραφή ζωτικών δεδομένων. Η εφαρμογή απαιτεί επίσης άδεια για την αποστολή ειδοποιήσεων στον χρήστη, υπενθυμίζοντάς του να πραγματοποιεί τις μετρήσεις του σε τακτά χρονικά διαστήματα. Αυτό διασφαλίζει ότι ο χρήστης παρακολουθεί συστηματικά την καρδιακή του υγεία. Στις εικόνες φαίνεται το μήνυμα που εμφανίζεται στη συσκευή για την παροχή της άδειας, καθώς και το αντίστοιχο τμήμα κώδικα που καθορίζει τη διαδικασία.

```
if (ActivityCompat.checkSelfPermission(getApplicationContext(), getString(R.string.BodySensors)) == PackageManager.PERMISSION_DENIED)
    requestPermissions(new String[]{Manifest.permission.BODY_SENSORS}, REQUEST_CODE);
else {
    permissionGranted = true;
    createConnectionManager();
}
```

Εικόνα 12. Κώδικας για ερώτηση άδειας Body Sensors

```
private void checkNotificationPermission() {
    if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.TIRAMISU) {
        if (ContextCompat.checkSelfPermission(context, Manifest.permission.POST_NOTIFICATIONS) != PackageManager.PERMISSION_GRANTED) {
            // Request the permission
            ActivityCompat.requestPermissions(
                activity, this,
                new String[]{Manifest.permission.POST_NOTIFICATIONS},
                NOTIFICATION_PERMISSION_REQUEST_CODE
            );
        }
    }
}
```

Εικόνα 13. Κώδικας για ερώτηση άδειας ειδοποιήσεων

Παρακάτω φαίνεται ο κώδικας που διαχειρίζεται την απάντηση του χρήστη και εμφανίζει αναδυόμενη ειδοποίηση στο κάτω μέρος της οθόνης αν του δόθηκε ή όχι η πρόσβαση.

```

@Override
public void onRequestPermissionsResult(int requestCode, String[] permissions, int[] grantResults) {
    super.onRequestPermissionsResult(requestCode, permissions, grantResults);
    if (requestCode == NOTIFICATION_PERMISSION_REQUEST_CODE) {
        if (grantResults.length > 0 && grantResults[0] == PackageManager.PERMISSION_GRANTED) {
            // Permission granted, proceed with scheduling notifications
            NotificationScheduler.scheduleDailyNotifications(context: this);
        } else {
            // Permission denied, show a message or disable notification functionality
            Toast.makeText(context: this, text: "Notification permission denied", Toast.LENGTH_SHORT).show();
        }
    }
}
}

```

Εικόνα 14. Κώδικας για διαχείριση ειδοποιήσεων

Για να μπορώ να χρησιμοποιήσω τα συγκεκριμένα permissions χρειάζεται να είναι όλα ορισμένα στο αρχείο AndroidManifest.xml . Παρακάτω φαίνονται τα όλα τα permission που απαιτήθηκαν στην εφαρμογή.

```

<uses-permission android:name="android.permission.WAKE_LOCK" />
<uses-permission android:name="android.permission.BODY_SENSORS" />
<uses-permission android:name="android.permission.ACTIVITY_RECOGNITION" />
<uses-permission android:name="android.permission.MANAGE_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.VIBRATE" />
<uses-permission android:name="android.permission.USE_EXACT_ALARM"/>
<uses-permission android:name="android.permission.RECEIVE_BOOT_COMPLETED" />
<uses-permission android:name="android.permission.POST_NOTIFICATIONS" />
<uses-permission android:name="android.permission.INTERNET" />

```

Εικόνα 15. Permissions στο AndroidManifest.xml

Οι οθόνες που εμφανίζονται στον χρήστη για την ερώτηση άδειας, ως αποτέλεσμα του παραπάνω κώδικα, φαίνονται παρακάτω.



Εικόνα 16. Ερώτηση άδειας ειδοποιήσεων



Εικόνα 17. Κουμπιά αποδοχής/απόρριψης άδειας



Εικόνα 18. Ερώτηση άδειας αισθητήρων

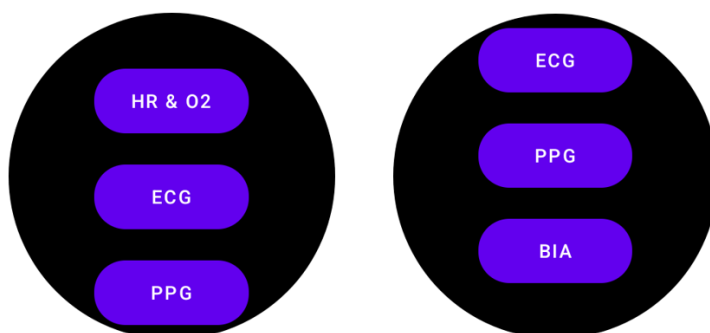
4.2.2 Περιήγηση

Η κεντρική οθόνη της εφαρμογής και η εναλλαγή μεταξύ των διαφορετικών οθονών γίνεται με τον κώδικα στην εικόνα (19). Η πρώτη οθόνη που εμφανίζεται όταν ο χρήστης ανοίγει την εφαρμογή διαχειρίζεται από τη MainActivity, η οποία είναι η μόνη που επιτρέπει σε άλλες εφαρμογές να την εκκινήσουν εξαιτίας του `android:exported="true"` και φαίνεται στην εικόνα (20).

```
<activity
    android:name=".MainActivity"
    android:exported="true">
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />

        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
</activity>
<activity
    android:name=".ECG.EcgActivity"
    android:exported="false" />
<activity
    android:name=".BIA.BiaActivity"
    android:exported="false" />
<activity
    android:name=".HR02.Hr02Activity"
    android:exported="false" />
<activity
    android:name=".PPG.PpgActivity"
    android:exported="false" />
<activity
    android:name=".ECG.EcgResults"
    android:exported="false" />
```

Εικόνα 19. Διάφορες οθόνες της εφαρμογής, κομμάτι κώδικα από *AndroidManifest.xml*



Εικόνα 20. Κεντρική οθόνη, μενού της εφαρμογής

Παρακάτω φαίνεται ο τρόπος που μεταβαίνουμε από την κεντρική οθόνη προς μια άλλη οθόνη. Πιο συγκεκριμένα, βρίσκουμε το κουμπί με συγκεκριμένο id, στην προκειμένη το `ecgbtn` και ορίζουμε έναν listener που ενεργοποιείται όταν ο χρήστης πατά το συγκεκριμένο κουμπί. Όταν πατηθεί, δημιουργείται ένα `Intent` που μεταφέρει τον χρήστη από τη

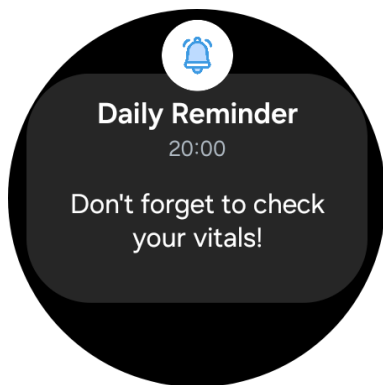
MainActivity στην επιθυμητή οθόνη, εδώ στην EcgActivity και τέλος καλείται startActivity(), το οποίο ανοίγει τη νέα οθόνη.

```
ecgbtn = findViewById(R.id.ecgbtn);
ecgbtn.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        Intent myint = new Intent( packageContext: MainActivity.this, EcgActivity.class);
        startActivity(myint);
    }
});
```

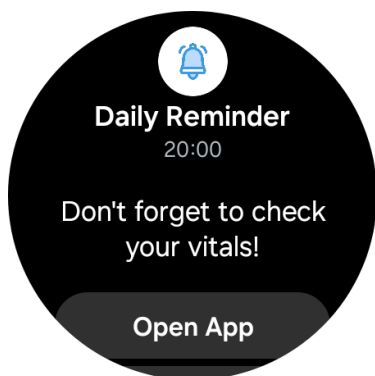
Εικόνα 21. Κώδικας για μετάβαση σε νέα δραστηριότητα μέσω κουμπιού

4.2.3 Δημιουργία ειδοποιήσεων

Η εφαρμογή, στην περίπτωση που της παραχωρήσει ο χρήστης το δικαίωμα για αποστολή ειδοποιήσεων, ήχου και δόνησης, στέλνει ειδοποίηση 2 φορές τη μέρα, μια το πρωί και μια το βράδυ για να του υπενθυμίσει να κάνει τις απαραίτητες μετρήσεις για τη βέλτιστη παρακολούθηση της υγείας του.



Εικόνα 22. Ειδοποίηση



Εικόνα 23. Εσωτερικό ειδοποίησης με επιλογή ανοίγματος της εφαρμογής

Στην MainActivity.java δημιουργείται κανάλι επικοινωνίας και αν υπάρχουν τα δικαιώματα προχωρά στον προγραμματισμό και αποστολή ειδοποιήσεων. Τα Notification Channels είναι απαραίτητα από το Android 8.0 (API 26+) για να κατηγοριοποιούνται οι ειδοποιήσεις

και να επιτρέπουν στο χρήστη να ελέγχει τις ρυθμίσεις τους.

```
// Create the notification channel
NotificationHelper.createNotificationChannel( context: this);

// Check for exact alarm permission
checkNotificationPermission();

// Schedule the first notification
NotificationScheduler.scheduleDailyNotifications( context: this);
```

Εικόνα 24. Δημιουργία καναλιού ειδοποιήσεων, έλεγχος permissions και προγραμματισμός ειδοποιήσεων

Στον παρακάτω κώδικα φαίνεται πως υπολογίζεται η επόμενη ώρα ειδοποίησης με βάση την τρέχουσα ωριαία παράδειγμα αν είναι πριν τις 10 π.μ., ρυθμίζει την ειδοποίηση για τις 10 π.μ. σήμερα.

```
// Schedule daily notifications at 10 AM and 8 PM
4 usages
public static void scheduleDailyNotifications(Context context) {
    long nextTriggerTime = getNextTriggerTime();
    scheduleAlarm(context, nextTriggerTime);
}

// Calculate the next occurrence of 10 AM or 8 PM
1 usage
private static long getNextTriggerTime() {
    Calendar now = Calendar.getInstance();
    Calendar tenAM = getTodayAt( hour: 10, minute: 0); // 10 AM
    Calendar eightPM = getTodayAt( hour: 20, minute: 0); // 8 PM (20:00)

    // If current time is before 10 AM, schedule 10 AM today
    if (now.before(tenAM)) {
        return tenAM.getTimeInMillis();
    }
    // If current time is before 8 PM, schedule 8 PM today
    else if (now.before(eightPM)) {
        return eightPM.getTimeInMillis();
    }
    // Otherwise, schedule 10 AM tomorrow
    else {
        tenAM.add(Calendar.DAY_OF_YEAR, 1);
        return tenAM.getTimeInMillis();
    }
}

// Create a Calendar instance for a specific hour/minute today
2 usages
private static Calendar getTodayAt(int hour, int minute) {
    Calendar calendar = Calendar.getInstance();
    calendar.set(Calendar.HOUR_OF_DAY, hour);
    calendar.set(Calendar.MINUTE, minute);
    calendar.set(Calendar.SECOND, 0);
    calendar.set(Calendar.MILLISECOND, 0);
    return calendar;
}
```

Εικόνα 25. Κώδικας για υπολογισμό κοντινότερης προγραμματισμένης ειδοποίησης

Για τον προγραμματισμό της ειδοποίησης ακριβώς την υπολογισμένη ώρα χρησιμοποιείται ο `AlarmManager`. Το `PendingIntent` επιτρέπει στο `AlarmManager` να εκτελέσει το `Intent`, που στοχεύει την `NotificationReceiver` κλάση, στο μέλλον. Το `setExactAndAllowWhileIdle` εξασφαλίζει ακριβή εκτέλεση ακόμα και αν η συσκευή είναι σε κατάσταση αναμονής ή σε αναστολή λειτουργίας.

```
private static void scheduleAlarm(Context context, long triggerTime) {
    AlarmManager alarmManager = (AlarmManager) context.getSystemService(Context.ALARM_SERVICE);
    Intent intent = new Intent(context, NotificationReceiver.class);
    PendingIntent pendingIntent = PendingIntent.getBroadcast(
        context,
        requestCode: 0,
        intent,
        flags: PendingIntent.FLAG_UPDATE_CURRENT | PendingIntent.FLAG_IMMUTABLE
    );
    if (alarmManager != null) {
        alarmManager.setExactAndAllowWhileIdle(
            AlarmManager.RTC_WAKEUP,
            triggerTime,
            pendingIntent
        );
    }
}
```

Εικόνα 26. Προγραμματίζει ειδοποίηση, ακόμα και σε κατάσταση εξοικονόμησης ενέργειας

Ο κώδικας που φαίνεται στην εικόνα () ορίζει έναν `Broadcast Receiver` που εμφανίζει μια ειδοποίηση όταν ενεργοποιηθεί από τον `AlarmManager` που προαναφέρθηκε και προγραμματίζει την επόμενη ειδοποίηση. Η `onReceive()` καλεί τη `showNotification(context)`, η οποία δημιουργεί και εμφανίζει μια ειδοποίηση με τίτλο "Daily Reminder", υψηλής προτεραιότητας, με δόνηση και ένα κουμπί "Open App" που ανοίγει την εφαρμογή αν το πατήσει ο χρήστης. Παράλληλα, προγραμματίζει την επόμενη ειδοποίηση για τις 10 π.μ. ή τις 8 μ.μ.

Η ειδοποίηση αποστέλλεται μέσω `notificationManager`, διασφαλίζοντας ότι ο χρήστης θα λαμβάνει τις υπενθυμίσεις του καθημερινά, ακόμα και αν η συσκευή είναι σε αναμονή.

```

public void onReceive(Context context, Intent intent) {
    // Show the notification
    showNotification(context);
    // Schedule the next notification (10 AM or 8 PM tomorrow)
    NotificationScheduler.scheduleDailyNotifications(context);
}

// usage
private void showNotification(Context context) {
    NotificationManager notificationManager = (NotificationManager) context.getSystemService(Context.NOTIFICATION_SERVICE);
    Intent appIntent = new Intent(context, MainActivity.class);
    appIntent.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP | Intent.FLAG_ACTIVITY_SINGLE_TOP);
    PendingIntent appPendingIntent = PendingIntent.getActivity(
        context,
        requestCode: 0,
        appIntent,
        flags: PendingIntent.FLAG_UPDATE_CURRENT | PendingIntent.FLAG_IMMUTABLE
    );
    NotificationCompat.Builder builder = new NotificationCompat.Builder(context, NotificationHelper.CHANNEL_ID)
        .setContentTitle("Daily Reminder")
        .setContentText("Don't forget to check your vitals!")
        .setSmallIcon(R.drawable.ic_notification)
        .setPriority(NotificationCompat.PRIORITY_HIGH)
        .setVibrate(new long[]{0, 500, 250, 500})
        .setAutoCancel(true)
        .addAction(
            R.drawable.ic_open_app,
            title: "Open App",
            appPendingIntent
        );
    if (notificationManager != null) {
        notificationManager.notify(1, builder.build());
    }
}
}

```

Εικόνα 27. Δημιουργία ειδοποίησης και εμφάνιση με ήχο, δόνηση και κουμπί για άνοιγμα της εφαρμογής

Ο κώδικας που φαίνεται παρακάτω ορίζει τη δημιουργία ενός καναλιού ειδοποιήσεων που επιτρέπει ειδοποιήσεις με ήχο, δόνηση και pop-up προβολή που είδαμε αρχικά να καλείται. Ορίζει υψηλή προτεραιότητα, προσαρμοσμένο μοτίβο δόνησης και τον προεπιλεγμένο ήχο συστήματος. Τέλος, το κανάλι εγγράφεται μέσω του NotificationManager, διασφαλίζοντας σωστή λειτουργία των ειδοποιήσεων.

```

public static void createNotificationChannel(Context context) {
    if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {
        CharSequence name = "Wear OS Notifications";
        String description = "Channel for Wear OS notifications";
        int importance = NotificationManager.IMPORTANCE_HIGH;

        NotificationChannel channel = new NotificationChannel(CHANNEL_ID, name, importance);
        channel.setDescription(description);

        channel.enableVibration(true);
        channel.setVibrationPattern(new long[]{0, 500, 250, 500}); // Custom vibration pattern

        AudioAttributes audioAttributes = new AudioAttributes.Builder()
            .setUsage(AudioAttributes.USAGE_NOTIFICATION)
            .setContentType(AudioAttributes.CONTENT_TYPE_SONIFICATION)
            .build();
        channel.setSound(Settings.System.DEFAULT_NOTIFICATION_URI, audioAttributes);

        channel.setLockscreenVisibility(Notification.VISIBILITY_PUBLIC);

        NotificationManager notificationManager = context.getSystemService(NotificationManager.class);
        notificationManager.createNotificationChannel(channel);
    }
}

```

Εικόνα 28. Δημιουργία καναλιού ειδοποιήσεων με δόνηση, ήχο και προβολή στην οθόνη κλειδώματος

4.2.4 Μέτρηση Ηλεκτροκαρδιογραφήματος (ECG) και Ανάλυση Καρδιακών Διαταραχών

Αφού η εφαρμογή ελέγχει αν έχει την απαραίτητη άδεια (BODY_SENSORS) για την πρόσβαση στους αισθητήρες της συσκευής, πραγματοποιείται η σύνδεση με την πλατφόρμα υγείας της Samsung και η επικοινωνία με τον ECG αισθητήρα.

```
public void connect(Context context) {  
    healthTrackingService = new HealthTrackingService(connectionListener, context);  
    healthTrackingService.connectService();  
}
```

Εικόνα 29. Σύνδεση με HealthTrackingService

Με την επιτυχή σύνδεση, αρχικοποιείται ο EcgListener, ο οποίος είναι υπεύθυνος για τη συλλογή και επεξεργασία των δεδομένων από τον ECG αισθητήρα.

```
TrackerDataNotifier.getInstance().addObserver(trackerDataObserver);  
  
ecgListener = new EcgListener(getApplicationContext());
```

Εικόνα 30. Προσθήκη Tracker και αρχικοποίηση EcgListener

Η μέθοδος initEcg(EcgListener ecgListener) αρχικοποιεί τον HealthTracker για τη λειτουργία ηλεκτροκαρδιογράφημα κατ' απαίτηση. Αρχικά, λαμβάνει τον κατάλληλο HealthTracker από την healthTrackingService, τον αναθέτει στον ecgListener μέσω setHealthTracker(ecgTracker) και στη συνέχεια καλεί setHandlerForBaseListener(ecgListener).

```
public void initEcg(EcgListener ecgListener) {  
    // Get HealthTracker for ECG_ON_DEMAND  
    HealthTracker ecgTracker = healthTrackingService.getHealthTracker(HealthTrackerType.ECG_ON_DEMAND);  
  
    // Pass it to ecgListener  
    ecgListener.setHealthTracker(ecgTracker);  
  
    // Set the handler for the listener  
    setHandlerForBaseListener(ecgListener);  
  
    Log.i("tag: \"Init ECG\", msg: \"ECG tracker initialized successfully.\");  
}
```

Εικόνα 31. Αρχικοποίηση ECG Tracker

Τα δεδομένα που λαμβάνονται περιλαμβάνουν:

ECG mV: Η ηλεκτρική δραστηριότητα της καρδιάς σε millivolts.

Μέση Τιμή ECG (ECG mV Avg): Ο μέσος όρος των τελευταίων μετρήσεων για καλύτερη ακρίβεια.

Lead-Off Detection: Δείχνει αν τα ηλεκτρόδια έχουν τοποθετηθεί σωστά.

Ακολουθία δειγμάτων (Sequence): Αριθμός δειγματοληψίας του ECG.

PPG Green Data: Συνδυασμός δεδομένων PPG για καλύτερη ανάλυση του καρδιακού ρυθμού.

Threshold Max/Min Data mV: Μέγιστες και ελάχιστες τιμές του ECG σήματος.

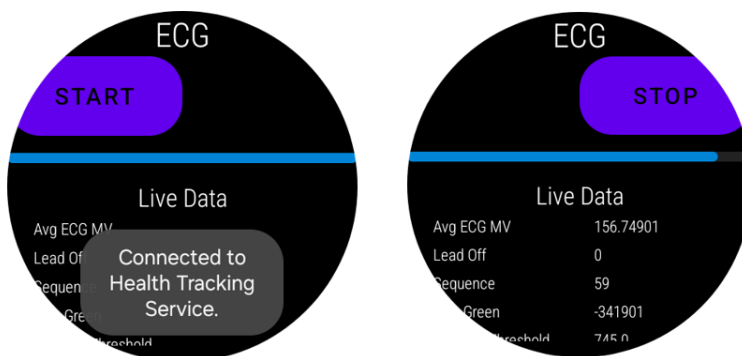
Παρα κάτω φαίνεται ο κώδικας που λαμβάνει τα δεδομένα από τον αισθητήρα.

```
private EcgData processEcgData(DataPoint dataPoint) throws IOException {  
    // Retrieve values from the dataPoint directly  
    float ecgmV = dataPoint.getValue(ValueKey.EcgSet.ECG_MV);  
    int ecgpppgreen = dataPoint.getValue(ValueKey.EcgSet.PPG_GREEN);  
    int sequence = dataPoint.getValue(ValueKey.EcgSet.SEQUENCE);  
    int leadoff = dataPoint.getValue(ValueKey.EcgSet.LEAD_OFF);  
    float maxthresholdmv = dataPoint.getValue(ValueKey.EcgSet.MAX_THRESHOLD_MV);  
    float minthresholdmv = dataPoint.getValue(ValueKey.EcgSet.MIN_THRESHOLD_MV);  
    saveDataToFile(ecgmV, ecgpppgreen, sequence, leadoff, maxthresholdmv, minthresholdmv);  
  
    // Log for debugging  
    Log.d(APP_TAG, "msg: " + dataPoint.toString());  
  
    return new EcgData(ecgmV, ecgpppgreen, sequence, leadoff, maxthresholdmv, minthresholdmv);  
}
```

Εικόνα 32. Ανάκτηση και αποθήκευση δεδομένων ECG

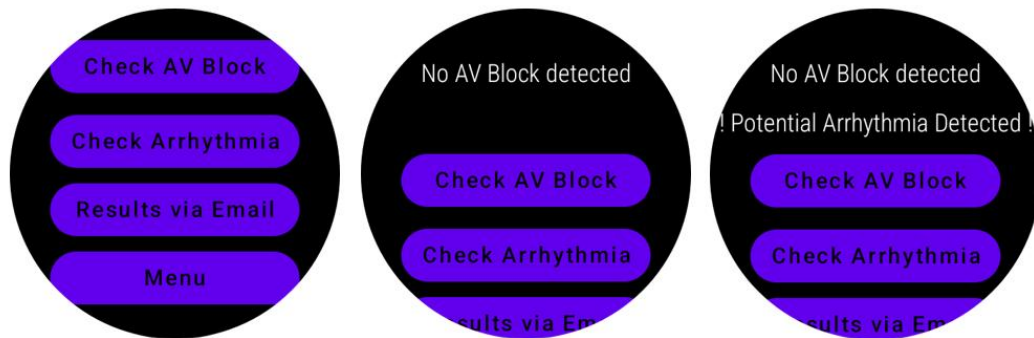
Επεξεργασία και Ανάλυση του ECG Σήματος

Κατά τη διάρκεια της μέτρησης, ένας TrackerDataObserver παρακολουθεί συνεχώς τις αλλαγές στα δεδομένα και ενημερώνει δυναμικά τη διεπαφή χρήστη. Αν το ρολόι είναι σωστά τοποθετημένο στο χέρι και ο χρήστης ακουμπά το δάχτυλό του στο πάνω κουμπί Home, η εφαρμογή εμφανίζει τις πραγματικές τιμές του ECG. Αν όμως κάποιο ηλεκτρόδιο έχει αποσυνδεθεί, εμφανίζεται μήνυμα σφάλματος, ζητώντας από τον χρήστη να επανατοποθετήσει τα δάχτυλά του στη σωστή θέση. Με το κουμπί STOP ο χρήστης μπορεί να σταματήσει τη μέτρηση, όμως προτείνονται τα 40 δευτερόλεπτα, τα οποία συμπληρώνονται στο progress bar, για να υπάρχει επαρκές δείγμα προς αξιολόγηση.

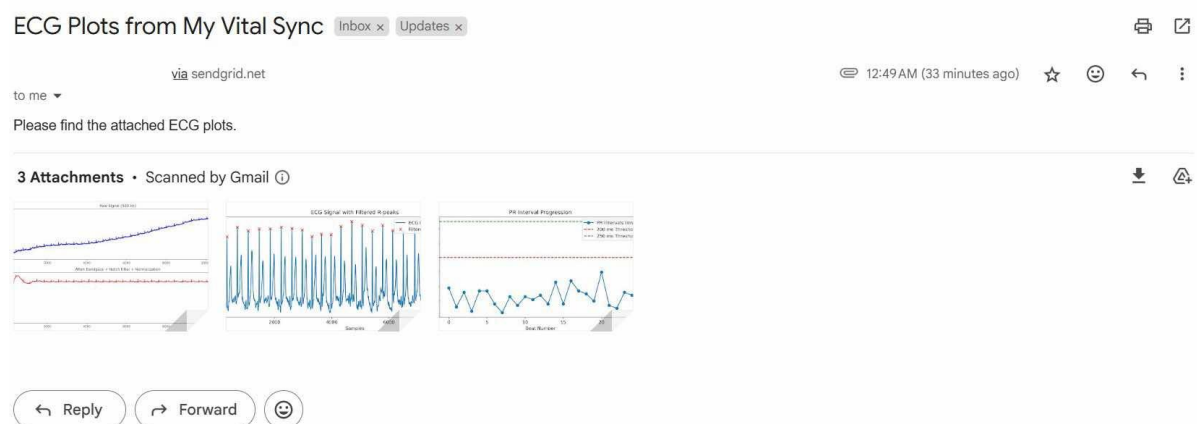


Εικόνα 33. Οθόνη μετρήσεων ECG και live εμφάνιση τιμών

Μετά το πάτημα του κουμπιού STOP, ο χρήστης μεταβαίνει σε μια καινούργια οθόνη όπου μπορεί να τσεκάρει το ηλεκτροκαρδιογράφημα του για AV Block και αρρυθμίες με το πάτημα των αντίστοιχων κουμπιών. Τα αποτελέσματά του μπορούν να σταλούν στο email του με το κουμπί Results via Email, ενώ παρέχεται και η δυνατότητα για επιστροφή στο αρχικό μενού.



Εικόνα 34. Οθόνες αποτελεσμάτων και κουμπιών για αποστολή αποτελεσμάτων μέσω email και για επιστροφή στο αρχικό μενού



Εικόνα 35. Email με τα αποτελέσματα που στέλνεται μέσω της εφαρμογής στον χρήστη

Πιο συγκεκριμένα, παρακάτω φαίνεται ο κώδικας που αποθηκεύει τα δεδομένα σε ένα αρχείο (ecg_data.txt) μετά την ολοκλήρωση της καταγραφής για να υποβληθούν αργότερα σε πολλαπλά στάδια επεξεργασίας για την αφαίρεση θορύβου, πριν την ανίχνευση αρρυθμίας και αν block με τη χρήση αλγορίθμων Pythοn, οι οποίοι εκτελούνται μέσα από την εφαρμογή με τη βοήθεια της βιβλιοθήκης Chaquopy.

```
private void saveDataToFile(float ecgmv, int ecgpppgreen, int sequence, int leadoff, float maxthresholdmv, float minthresholdmv) {
    if (bufferedWriter != null) {
        try {
            String data = String.format(Locale.US, format: "%.2f\n", ecgmv);
            bufferedWriter.write(data);
            bufferedWriter.flush();
        } catch (IOException e) {
            Log.e(APP_TAG, msg: "Error writing to file", e);
        }
    } else {
        Log.e(APP_TAG, msg: "BufferedWriter is not initialized");
    }
}

usage
public void closeFileWriter() {
    if (bufferedWriter != null) {
        try {
            bufferedWriter.close();
            Log.i(APP_TAG, msg: "BufferedWriter closed successfully");
        } catch (IOException e) {
            Log.e(APP_TAG, msg: "Error closing BufferedWriter", e);
        }
    }
}
}
```

Εικόνα 36. Αποθήκευση και κλείσιμο αρχείου για δεδομένα ECG

Η μέθοδος initializePython() ελέγχει αν η Python έχει ξεκινήσει και, αν όχι, την εκκινεί, επιτρέποντας την εκτέλεση Python κώδικα εντός του Android περιβάλλοντος. Στην συνέχεια, μέσα σε ένα νέο thread φορτώνεται το python module processecg.py

```
private void initializePython() {
    if (!Python.isStarted()) {
        Python.start(new AndroidPlatform(getApplicationContext()));
    }
}
```

Εικόνα 37. Αρχικοποίηση περιβάλλοντος Python στο Android

```
initializePython();

new Thread() -> {
    try {
        // Get the Python instance
        Python py = Python.getInstance();

        // Get the processecg.py module
        PyObject myScript = py.getModule(name: "processecg");

        PyObject result = myScript.callAttr(key: "main", ...args: getExternalFilesDir(Environment.DIRECTORY_DOWNLOADS) + "/ecg_data.txt");

        // Convert the result to a Java string
        String greeting = result.toString();

        Log.d(tag: "PythonScript", msg: "processecg.py finished");

        // Start the next activity after both scripts complete
        runOnUiThread() -> {
            Log.d(tag: "EcgActivity", msg: "Inside runOnUiThread");
            measurementProgress.setProgress(0);
            ecgStartButton.setEnabled(true);
            Intent intent = new Intent(packageContext: EcgActivity.this, EcgResults.class);
            startActivity(intent);
        };

    } catch (Exception e) {
        Log.e(tag: "PythonScript", msg: "Error in processecg.py: " + e.getMessage());
    }
}.start(); // Start processecg.py in its own thread
```

Εικόνα 38. Εκτέλεση Python Script για επεξεργασία ECG σε ξεχωριστό thread

Πιο συγκεκριμένα, το σήμα φιλτράρεται με χρήση butterworth bandpass filter και ύστερα notch φίλτρων για την απομάκρυνση του θορύβου και των παρεμβολών.

Μετά το φιλτράρισμα, το σήμα ομαλοποιείται μέσω Z-score normalization για τη σταθεροποίηση του, ώστε οι διακυμάνσεις του να μην επηρεάζονται από μεγάλες αλλαγές πλάτους και να αναγνωρίζονται πιο εύκολα οι R-peaks.

Τέλος, οπτικοποιείται τόσο το αρχικό σήμα, όσο και το επεξεργασμένο και αποθηκεύονται ως εικόνα τοπικά στη συσκευή, ώστε να είναι διαθέσιμα για να σταλούν αργότερα, αν ζητηθεί, με email.

```
def bandpass_filter(data, lowcut, highcut, fs, order=4):
    nyq = 0.5 * fs
    low = lowcut / nyq
    high = highcut / nyq
    b, a = butter(order, [low, high], btype='band')
    return filtfilt(b, a, data)

1 usage
def notch_filter(data, freq, fs, quality_factor=30):
    nyq = 0.5 * fs
    freq_normalized = freq / nyq
    b, a = iirnotch(freq_normalized, quality_factor)
    return filtfilt(b, a, data)

1 usage
def process_ecg(ecg_values, fs=500, powerline_freq=50, output_dir=None):

    ecg_array = np.array(ecg_values, dtype=float)

    # 1. Remove baseline wander and high-frequency noise
    ecg_filtered = bandpass_filter(ecg_array, lowcut: 0.5, highcut: 45, fs=fs)

    # 2. Remove 50 Hz powerline interference
    ecg_clean = notch_filter(ecg_filtered, powerline_freq, fs=fs)

    # 3. Normalization
    ecg_normalized = (ecg_clean - np.mean(ecg_clean)) / np.std(ecg_clean)

    plt.figure(figsize=(12, 6))
    plt.subplot(2, 1, 1)
    plt.plot(ecg_values, 'b', label='Raw ECG')
    plt.title(f"Raw Signal (500 Hz)")

    plt.subplot(2, 1, 2)
    plt.plot(ecg_normalized, 'r', label='Processed ECG')
    plt.title("After: Bandpass + Notch Filter + Normalization")
    plt.tight_layout()
```

Εικόνα 39. Φιλτράρισμα, επεξεργασία και οπτικοποίηση ECG σήματος

Όπως αναφέρθηκε νωρίτερα, μετά την ολοκλήρωση της επεξεργασίας ο χρήστης μεταβαίνει σε μια καινούργια οθόνη που του δίνεται η δυνατότητα να ελέγξει το επεξεργασμένο ECG του για τυχόν αρρυθμίες ή αν block, να στείλει με email τα αποτελέσματα στη διεύθυνση ηλεκτρονικού ταχυδρομείου του και να γυρίσει στο αρχικό μενού.

Ανίχνευση Καρδιακών Διαταραχών

Αν ο χρήστης επιλέξει το κουμπί Check Arrhythmia τότε σε ένα νέο thread το σήμα αναλύεται για την ανίχνευση R-Peaks και τον υπολογισμό PR-Διαστημάτων.

```
public void checkArrhythmia(View View) {
    initializePython();

    new Thread() -> {
        try {
            Python py2 = Python.getInstance();
            PyObject myScript2 = py2.getModule( name: "arrhythmiaDetection");
            PyObject result2 = myScript2.callAttr( key: "main",   __args: getExternalFilesDir(Environment.DIRECTORY_DOWNLOADS) + "/clean_ecg_500hz.txt");
            boolean flag2 = result2.toBoolean();
            arrhythmiaString = flag2 ? getString(R.string.Arrhythmia) : getString(R.string.NoArrhythmia);
            Log.d( tag: "PythonScript",   msg: "arrhythmiaDetection.py finished");
            ecgArrhythmiaResult.setText(arrhythmiaString);
        } catch (Exception e) {
            Log.e( tag: "PythonScript",   msg: "Error in arrhythmiaDetection.py: " + e.getMessage());
        }
    }).start();
}
```

Εικόνα 40. Έλεγχος αρρυθμίας μέσω Python Script

Πιο αναλυτικά, χρησιμοποιείται ο αυτόματος αλγόριθμος `nk.ecg_process()` της `NeuroKit2`, ο οποίος εντοπίζει τις κορυφές R του QRS συμπλέγματος. Με βάση τις θέσεις των R-peaks, υπολογίζονται τα RR-διαστήματα, δηλαδή ο χρόνος μεταξύ διαδοχικών καρδιακών παλμών.

```

def main(input_file):
    df = pd.read_csv(input_file, header=None, skiprows=1500)

    # Load ECG data
    ecg_signal = df.values.flatten()

    # Specify sampling rate
    sampling_rate = 500

    # Filter the ECG signal
    ecg_cleaned = nk.ecg_clean(ecg_signal, sampling_rate=sampling_rate, method="neurokit")

    # Detect R-peaks
    signals, info = nk.ecg_process(ecg_cleaned, sampling_rate=sampling_rate)
    r_peaks = info["ECG_R_Peaks"] #positions of R-peaks

    print(f"Initial number of detected R-peaks: {len(r_peaks)}")

    # Filter R-peaks by amplitude
    peak_amplitudes = signals["ECG_Clean"][r_peaks]

    # Calculate the mean amplitude.
    mean_amplitude = np.mean(peak_amplitudes)*0.7

    # Keep only R-peaks above the mean amplitude.
    r_peaks_filtered = r_peaks[peak_amplitudes > mean_amplitude]

    print(f"Number of R-peaks above the mean amplitude: {len(r_peaks_filtered)}")

    r_peaks = r_peaks_filtered

    if len(r_peaks) > 1:
        # RR-intervals in seconds
        rr_intervals = np.diff(r_peaks) / sampling_rate

        # Average Heart Rate (bpm)
        avg_hr = 60.0 / np.mean(rr_intervals)
    else:
        rr_intervals = []
        avg_hr = 0

    print(f"Average Heart Rate (bpm) [using filtered peaks]: {avg_hr:.2f}")

```

Εικόνα 41. Επεξεργασία ECG και υπολογισμός καρδιακού ρυθμού

Για την ανίχνευση αρρυθμιών υπολογίζεται ο καρδιακός ρυθμός (HR) με βάση τα RR-διαστήματα. Εντοπίζονται ανωμαλίες στην μεταβλητότητα των RR-διαστημάτων, οι οποίες μπορεί να υποδεικνύουν αρρυθμία. Αν ο ρυθμός είναι εκτός φυσιολογικών ορίων ($HR < 50$ bpm ή $HR > 120$ bpm), το σύστημα ειδοποιεί για πιθανή αρρυθμία.

```

# Flag potential arrhythmias
hr_lower_threshold = 50
hr_upper_threshold = 120
rr_std_threshold = 0.12 # threshold for RR interval std dev

arrhythmia_flag = False
arrhythmia_reasons = []

# Check heart rate range
if avg_hr < hr_lower_threshold or avg_hr > hr_upper_threshold:
    arrhythmia_flag = True
    arrhythmia_reasons.append(f"Heart rate out of range ({hr_lower_threshold}-{hr_upper_threshold} bpm).")

# Check RR interval standard deviation
if len(rr_intervals) > 1:
    rr_std = np.std(rr_intervals)
    if rr_std > rr_std_threshold:
        arrhythmia_flag = True
        arrhythmia_reasons.append(f"High RR-interval variability (std = {rr_std:.3f} s).")

```

Εικόνα 42. Ανίχνευση αρρυθμιών από καρδιακό ρυθμό και RR-διαστήματα

Τέλος, οπτικοποιείται το ECG με τα ανιχνευμένα R-peaks και αποθηκεύεται το γράφημα της αρρυθμίας (arrhythmia_plot.jpg) ώστε να μπορεί να σταλεί στον χρήστη αργότερα μέσω email.

```

# Visualize the signal and R-peaks
plt.figure(figsize=(12, 4))
plt.plot(signals["ECG_Clean"], label="ECG Cleaned")
plt.plot(r_peaks, signals["ECG_Clean"][r_peaks], "rx", label="Filtered R-peaks (above mean amplitude)")
plt.title("ECG Signal with Filtered R-peaks")
plt.xlabel("Samples")
plt.ylabel("Amplitude")
plt.legend()

output_dir = dirname(input_file)

if output_dir:
    plot_file = join(output_dir, "arrhythmia_plot.jpg")
    plt.savefig(plot_file, format='jpg', dpi=300)
    plt.close() # Close the plot to free up memory

# Print results
if arrhythmia_flag:
    print("\n>>> Potential Arrhythmia Detected <<<")
    print("Reasons:")
    for reason in arrhythmia_reasons:
        print(f" - {reason}")
    return True
else:
    print("\n>>> No obvious arrhythmia detected <<<")
    return False

```

Εικόνα 43. Οπτικοποίηση ECG σήματος και επισήμανση R-peaks

Αντίστοιχα, αν ο χρήστης επιλέξει το κουμπί Check AV Block τότε σε ένα καινούργιο thread το σήμα περνά ξανά από επεξεργασία. Πρώτα φιλτράρεται από butterworth ζωνοπερατό φίλτρο, το οποίο αφαιρεί συχνότητες εκτός του εύρους 0.5 - 50 Hz, κρατώντας μόνο τις χρήσιμες συνιστώσες του ECG. Μετά, χρησιμοποιείται το nk.signal_detrend από το NeuroKit2 για να αφαιρέσει την τάση βάσης (baseline drift), που μπορεί να οφείλεται σε κινήσεις ή αργές μεταβολές στο σήμα.

```

# Butterworth Bandpass filter
def bandpass_filter(signal, lowcut, highcut, fs, order=5):
    from scipy.signal import butter, filtfilt
    nyquist = 0.5 * fs
    low = lowcut / nyquist
    high = highcut / nyquist
    b, a = butter(order, [low, high], btype='band')
    filtered_signal = filtfilt(b, a, signal)
    return filtered_signal

def preprocess_ecg(ecg_signal, fs):
    ecg_filtered = bandpass_filter(ecg_signal, lowcut=0.5, highcut=50, fs=fs)
    # Detrending to clean ECG signal
    ecg_detrended = nk.signal_detrend(ecg_filtered)
    return ecg_detrended

def process_ecg(ecg_signal, fs):
    ecg_processed = nk.ecg_clean(ecg_signal, sampling_rate=fs)
    return ecg_processed

```

Εικόνα 44. Φιλτράρισμα και προεπεξεργασία σήματος ECG

Για την ανίχνευση πιθανού AV block αρχικά εντοπίζονται τα P-waves και οι R-κορυφές στο ECG σήμα για να υπολογιστούν τα PR διαστήματα (ο χρόνος μεταξύ P-wave και R-peak) και να κατηγοριοποιηθούν πιθανά AV block.

```

def detect_qrs(ecg_signal, fs):
    # Detect R peaks
    signals, info = nk.ecg_peaks(ecg_signal, sampling_rate=fs)
    r_peaks = info["ECG_R_Peaks"]
    return r_peaks

def detect_p_waves(ecg_signal, fs):
    # P wave detection
    signals, info = nk.ecg_process(ecg_signal, sampling_rate=fs)
    p_peaks = info["ECG_P_Peaks"] # Extract P wave peaks
    return p_peaks

def calculate_pr_intervals(p_peaks, r_peaks, fs):
    pr_intervals = []
    for r in r_peaks:
        # Find the closest preceding P wave
        p_candidates = [p for p in p_peaks if p < r]
        if p_candidates:
            p = max(p_candidates) # latest P wave before the R peak
            pr_intervals.append((r - p) / fs * 1000) # Convert to milliseconds
    return pr_intervals

def classify_av_block(pr_intervals):
    if not pr_intervals:
        return "No PR intervals detected", False

    avg_pr = np.mean(pr_intervals)

    if avg_pr > 300:
        return "Third-degree (complete) AV block", True

    elif avg_pr > 200:
        # Check for progressive PR prolongation (Wenckebach)
        if all(earlier < later for earlier, later in zip(pr_intervals, pr_intervals[1:])):
            return "Second-degree AV block (Type I - Wenckebach)", True
        elif any(pr > 250 for pr in pr_intervals):
            return "Second-degree AV block (Type II - Mobitz)", True
        else:
            return "First-degree AV block", True

    return "No AV block detected", False

```

Εικόνα 45. Ανίχνευση και ταξινόμηση AV Block

Τελευταίο βήμα είναι η οπτικοποίηση του PR διαστήματος και αποθήκευση του γραφήματος αποκλεισμού AV (avblock_plot.jpg) για ύστερη χρήση.

```
def plot_ecg_with_peaks(ecg_signal, r_peaks, p_peaks, fs):
    time = np.arange(len(ecg_signal)) / fs # Convert samples to time in seconds
    plt.figure(figsize=(12, 6))
    plt.plot(time, ecg_signal, label="ECG Signal")
    plt.plot(time[r_peaks], ecg_signal[r_peaks], "x", label="R Peaks", markersize=10)
    plt.plot(time[p_peaks], ecg_signal[p_peaks], "o", label="P Peaks", markersize=10)
    plt.xlabel("Time (s)")
    plt.ylabel("Amplitude")
    plt.title("ECG Signal with Detected Peaks")
    plt.legend()
    plt.show()

def plot_pr_intervals(pr_intervals):
    plt.figure(figsize=(8, 4))
    plt.plot(pr_intervals, marker="o", linestyle="--", label="PR Intervals (ms)")
    plt.axhline(y=200, color="r", linestyle="--", label="200 ms Threshold")
    plt.axhline(y=250, color="g", linestyle="--", label="250 ms Threshold")
    plt.xlabel("Beat Number")
    plt.ylabel("PR Interval (ms)")
    plt.title("PR Interval Progression")
    plt.legend()
    plt.show()
```

Εικόνα 46. Οπτικοποίηση σήματος ECG και PR διαστημάτων

Αποθήκευση και Αποστολή Αποτελεσμάτων

Μετά την ανάλυση, τα αποτελέσματα αποθηκεύονται και προβάλλονται στην EcgResults, όπου ο χρήστης μπορεί να δει αν ανιχνεύθηκε κάποια αρρυθμία ή AV block. Επιπλέον, η εφαρμογή στέλνει μέσω email όλα τα παραγόμενα διαγράμματα (ECG Plot, Arrhythmia Plot, AV Block Plot) στον χρήστη, προσφέροντας πλήρη αναφορά της μέτρησης.

Αρχικά απαιτήθηκε σύνδεση στο SendGrid ώστε να δωθεί API_Key για να παραχωρηθεί η δυνατότητα αποστολής email.

Όταν ο χρήστης πατήσει το κουμπί Results via Email τότε σε ένα νέο thread εκτελείται η `sentEmailResults(View View)`, για να μη παγώσει η κύρια εφαρμογή. Αρχικά ορίζεται η διαδρομή προς τον φάκελο που είναι αποθηκευμένα τα τρία αρχεία εικόνων και ύστερα ορίζεται ο παραλήπτης, το θέμα και το σώμα του email. Τέλος, καλείται η μέθοδος `sendEmailAttachments`, που φαίνεται στην εικόνα (48), επισυνάπτοντας τα τρία αρχεία εικόνων.

```

public void sendEmailResults(View View) {
    new Thread() -> {
        try {
            // Send email with all three ECG plots
            File downloadsDir = getExternalFilesDir(Environment.DIRECTORY_DOWNLOADS);
            File ecgPlotFile = new File(downloadsDir, child: "ecg_plot.jpg");
            File arrhythmiaPlotFile = new File(downloadsDir, child: "arrhythmia_plot.jpg");
            File avBlockPlotFile = new File(downloadsDir, child: "avblock_plot.jpg");

            String recipientEmail = "frosy.chri@gmail.com"; // Replace with the recipient's email
            String emailSubject = "ECG Plots from My Vital Sync";
            String emailBody = "Please find the attached ECG plots.";

            // Attach all three files
            SendGridEmailSender.sendEmailWithAttachments(recipientEmail, emailSubject, emailBody,
                new File[]{ecgPlotFile, arrhythmiaPlotFile, avBlockPlotFile});
        } catch (Throwable e) {
            Log.d( tag: "SendEmail", msg: "sendEmailResults");
        }
    }.start();
}

```

Εικόνα 47. Αποστολή αποτελεσμάτων ECG μέσω email με συνημμένα γραφήματα

Στην `sendEmailWithAttachments` αρχικά ορίζεται ο αποστολέας, ο παραλήπτης και το σώμα του email. Στη συνέχεια, δημιουργείται το αντιμείμενο email που περιέχει τον αποστολέα, τον παραλήπτη, το θέμα και το περιεχόμενο. Γίνεται επεξεργασία των αρχείων σε Base64 μορφή και δημιουργούνται Attachments για κάθε αρχείο, όπου ορίζεται το όνομα, ο τύπος MIME (image/jpeg) και το περιεχόμενο

Τέλος δημιουργείται ένα SendGrid αντικείμενο με το κλειδί API (`SENDGRID_API_KEY`) που είναι απόρρητο και μοναδικό για κάθε χρήστη. Ρυθμίζεται ένα Request αντικείμενο για αποστολή HTTP POST στο endpoint `mail/send`. Η `request.setBody(mail.build())` δημιουργεί το τελικό JSON payload και το email αποστέλλεται μέσω `sg.api(request)`.

Αν η αποστολή είναι επιτυχής, εκτυπώνεται το status code και το response body.

Αν συμβεί σφάλμα κατά την αποστολή ή την ανάγνωση αρχείων, καταγράφεται με `printStackTrace()`.

```

public static void sendEmailWithAttachments(String toEmail, String subject, String body, File[] attachmentFiles) {
    // Set up sender and recipient emails
    Email from = new Email("...@...");
    Email to = new Email(toEmail);

    // Create email content
    Content content = new Content("text/plain", body);

    // Create the email object
    Mail mail = new Mail(from, subject, to, content);

    // Attach all files
    for (File attachmentFile : attachmentFiles) {
        try {
            byte[] fileContent = Files.readAllBytes(attachmentFile.toPath());
            String encodedFile = Base64.getEncoder().encodeToString(fileContent);

            Attachments attachments = new Attachments();
            attachments.setFilename(attachmentFile.getName());
            attachments.setType("image/jpg");
            attachments.setContent(encodedFile);

            mail.addAttachments(attachments);
        } catch (IOException e) {
            e.printStackTrace();
            return;
        }
    }

    // Send the email using SendGrid
    SendGrid sg = new SendGrid(SENDGRID_API_KEY);
    Request request = new Request();

    try {
        request.setMethod(Method.POST);
        request.setEndpoint("mail/send");
        request.setBody(mail.build());
        Response response = sg.api(request);

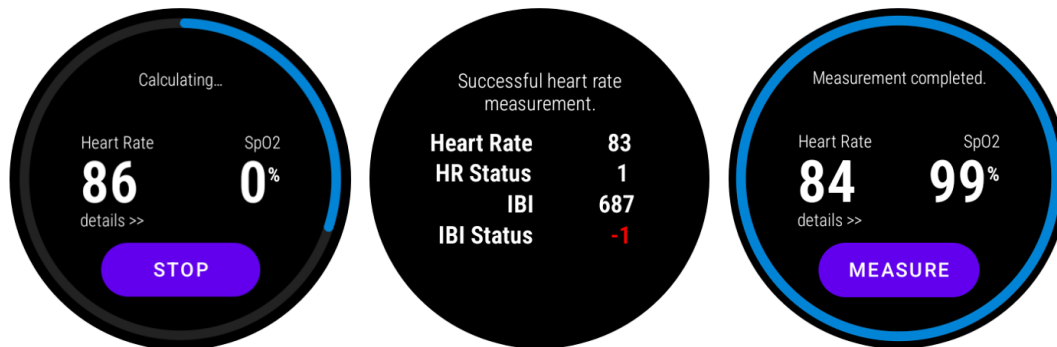
        System.out.println("Email sent! Status code: " + response.getStatusCode());
        System.out.println("Response body: " + response.getBody());
    } catch (IOException ex) {
        ex.printStackTrace();
    }
}

```

Εικόνα 48. Υλοποίηση αποστολής email με συνημμένα μέσω SendGrid σε Java

4.2.5 Λήψη μετρήσεων Καρδιακού ρυθμού (HR) και Κορεσμού Οξυγόνου (SpO²)

Η εφαρμογή παρέχει τη δυνατότητα στον χρήστη για καταγραφή καρδιακού ρυθμού και SpO₂, επιτρέποντάς του να ξεκινήσει ή να σταματήσει τη μέτρηση, ενώ παρέχει πληροφορίες όπως HR Status και IBI Status για την αξιολόγηση της ποιότητας της μέτρησης, τα οποία είναι διαθέσιμα, αν ο χρήστης πατήσει το details>> κάτω από τον μετρητή του καρδιακού ρυθμού. Για να πραγματοποιηθεί η μέτρηση SpO₂ ο χρήστης πρέπει να κρατά το δάχτυλό του στο πάνω κουμπί Home της συσκευής.



Εικόνα 49. Οθόνες μέτρησης HR και SpO₂

Πριν από κάθε μέτρηση, η εφαρμογή ελέγχει αν έχει παραχωρηθεί η απαιτούμενη άδεια (*BODY_SENSORS*) για την πρόσβαση στους αισθητήρες της συσκευής. Αν η άδεια δεν έχει δοθεί, ζητείται από τον χρήστη να την παραχωρήσει. Μόλις εξασφαλιστούν οι απαραίτητες άδειες, δημιουργείται ένας *ConnectionManager*, ο οποίος αναλαμβάνει τη σύνδεση με την πλατφόρμα υγείας της Samsung και την επικοινωνία με τους αισθητήρες καρδιακού ρυθμού και κορεσμού οξυγόνου.

```
TrackerDataNotifier.getInstance().addObserver(trackerDataObserver);

spO2Listener = new SpO2Listener();
heartRateListener = new HeartRateListener();

connectionManager.initSpO2(spO2Listener);
connectionManager.initHeartRate(heartRateListener);

heartRateListener.startTracker();
```

Εικόνα 50. Αρχικοποίηση και παρακολούθηση SpO₂ και καρδιακού ρυθμού

```

public void performMeasurement(View view) {
    if (isPermissionsOrConnectionInvalid()) {
        return;
    }

    if (!isMeasurementRunning.get()) {
        previousStatus = SpO2Status.INITIAL_STATUS;
        butStart.setText(R.string.StopLabel);
        txtSpo2.setText(R.string.SpO2DefaultValue);
        measurementProgress.setProgress(0);
        getWindow().addFlags(WindowManager.LayoutParams.FLAG_KEEP_SCREEN_ON);
        spO2Listener.startTracker();
        isMeasurementRunning.set(true);
        uiUpdateThread = new Thread(countDownTimer::start);
        uiUpdateThread.start();
    } else {
        butStart.setEnabled(false);
        isMeasurementRunning.set(false);
        getWindow().clearFlags(WindowManager.LayoutParams.FLAG_KEEP_SCREEN_ON);
        spO2Listener.stopTracker();
        final Handler progressHandler = new Handler(Looper.getMainLooper());
        progressHandler.postDelayed(() ->
            {
                butStart.setText(R.string.StartLabel);
                txtStatus.setText(R.string.StatusDefaultValue);
                measurementProgress.setProgress(0);
                butStart.setEnabled(true);
            }, delayMillis: MEASUREMENT_TICK * 2
        );
    }
}
}

```

Εικόνα 51. Έναρξη και διακοπή μέτρησης SpO2

Με την επιτυχή σύνδεση, αρχικοποιούνται οι HeartRateListener και SpO2Listener, οι οποίοι είναι υπεύθυνοι για τη λήψη και επεξεργασία των δεδομένων. Επίσης αρχικοποιείται και η μπάρα προόδου της μέτρησης του οξυγόνου. Τα δεδομένα που συλλέγονται περιλαμβάνουν τον καρδιακό ρυθμό (HR), το διάστημα μεταξύ καρδιακών παλμών (IBI), την ποιότητα μέτρησης του IBI (qIBI), καθώς και το ποσοστό κορεσμού οξυγόνου στο αίμα (SpO₂). Κατά τη διάρκεια της μέτρησης, ένας TrackerDataObserver παρακολουθεί συνεχώς τις αλλαγές στα δεδομένα και ενημερώνει δυναμικά τη διεπαφή χρήστη, εμφανίζοντας τις τρέχουσες τιμές στην οθόνη.

```

@Override
public void onHeartRateTrackerDataChanged(HeartRateData hrData) {
    HrO2Activity.this.runOnUiThread(() -> {
        heartRateDataLast = hrData;
        Log.i(APP_TAG, msg: "HR Status: " + hrData.status);
        if (hrData.status == HeartRateStatus.HR_STATUS_FIND_HR) {
            txtHeartRate.setText(String.valueOf(hrData.hr));
            Log.i(APP_TAG, msg: "HR: " + hrData.hr);
        } else {
            txtHeartRate.setText(getString(R.string.HeartRateDefaultValue));
        }
    });
}
}

```

Εικόνα 52. Ενημέρωση UI με δεδομένα HR

Παρακάτω φαίνεται ο κώδικας που παρακολουθεί και διαχειρίζεται δυναμικά την κατάσταση μιας μέτρησης SpO2, ενημερώνοντας το UI και παρέχοντας feedback στον χρήστη για την κατάσταση της μέτρησης, της συσκευής και των αισθητήρων. Πιο συγκεκριμένα, Αν το status του SpO2 είναι CALCULATING, εμφανίζει ότι η μέτρηση βρίσκεται σε εξέλιξη. Σε περίπτωση DEVICE_MOVING ή LOW_SIGNAL, εμφανίζει ειδοποιήσεις στο κάτω μέρος της οθόνης με Toast. Όταν η μέτρηση ολοκληρωθεί, σταματά τον αισθητήρα, ενημερώνει το UI με την τιμή SpO2, επαναφέρει το κουμπί έναρξης και τη γραμμή προόδου, και απενεργοποιεί τη διατήρηση της οθόνης ανοιχτής που υπήρχε καθ' όλη τη διάρκεια της μέτρησης SpO2.

```
@Override
public void onSpO2TrackerDataChanged(int status, int spO2Value) {
    if(status == previousStatus) {
        return;
    }
    previousStatus = status;
    Log.i(APP_TAG, "SP02 status: "+status);
    Log.i(APP_TAG, "SP02 value: "+spO2Value);
    switch (status) {
        case SpO2Status.CALCULATING:
            Log.i(APP_TAG, "Calculating measurement");
            runOnUiThread() -> {
                txtStatus.setText(R.string.StatusCalculating);
                txtStatus.invalidate();
            }
            break;
        case SpO2Status.DEVICE_MOVING:
            Log.i(APP_TAG, "Device is moving");
            runOnUiThread() -> {
                Toast.makeText(getApplicationContext(), R.string.StatusDeviceMoving, Toast.LENGTH_SHORT).show();
            }
            break;
        case SpO2Status.LOW_SIGNAL:
            Log.i(APP_TAG, "Low signal quality");
            runOnUiThread() -> {
                Toast.makeText(getApplicationContext(), R.string.StatusLowSignal, Toast.LENGTH_SHORT).show();
            }
            break;
        case SpO2Status.MEASUREMENT_COMPLETED:
            Log.i(APP_TAG, "Measurement completed");
            isMeasurementRunning.set(false);
            spO2Listener.stopTracker();
            runOnUiThread() -> {
                txtStatus.setText(R.string.StatusCompleted);
                txtStatus.invalidate();
                txtSpO2.setText(String.valueOf(spO2Value));
                txtSpO2.invalidate();
                btnStart.setText(R.string.StartLabel);
                measurementProgress.setProgress(measurementProgress.getMax(), animate: true);
                getWindow().clearFlags(WindowManager.LayoutParams.FLAG_KEEP_SCREEN_ON);
            };
            break;
    }
}
```

Εικόνα 53. Ενημέρωση κατάστασης και UI για SpO2 μέτρηση

Ο κώδικας που φαίνεται στην εικόνα (53) λαμβάνει και ενημερώνει την κατάσταση και την τιμή του SpO2 από ένα DataPoint. Αρχικά, ανακτά την κατάσταση της μέτρησης και αν δεν υπάρχει διαθέσιμη τιμή, ορίζεται ως CALCULATING. Στη συνέχεια, ορίζει την τιμή του SpO2 σε 0 αν δεν είναι διαθέσιμη. Αν η μέτρηση έχει ολοκληρωθεί, ανακτά την τιμή SpO2 από τα

δεδομένα και την αποθηκεύει μόνο αν είναι έγκυρη και ενημερώνει τους παρατηρητές (notifySpO2TrackerObservers).

```
public void updateSpO2(DataPoint dataPoint) {
    // Retrieve the status value from the dataPoint
    Integer statusValue = dataPoint.getValue(ValueKey.SpO2Set.STATUS);
    int status = statusValue != null ? statusValue : SpO2Status.CALCULATING;

    int spo2Value = 0; // Default to 0 if not applicable
    // Only set spo2Value if the status indicates completion
    if (status == SpO2Status.MEASUREMENT_COMPLETED) {
        Integer spo2ValueFromDataPoint = dataPoint.getValue(ValueKey.SpO2Set.SPO2);
        if (spo2ValueFromDataPoint != null) {
            spo2Value = spo2ValueFromDataPoint;
        }
    }

    // Notify observers with the retrieved values
    TrackerDataNotifier.getInstance().notifySpO2TrackerObservers(status, spo2Value);
    Log.d(APP_TAG, dataPoint.toString());
}
```

Εικόνα 54. Ενημέρωση δεδομένων SpO2 και ειδοποίηση trackers

Ανεξάρτητα από τη μέτρηση SpO2 ο αισθητήρας για τα HR δεδομένα τρέχει συνεχώς και ενημερώνει το UI του χρήστη. Πιο κάτω φαίνεται ο κώδικας που λαμβάνει αυτά τα δεδομένα (HR, HR_STATUS, IBI και IBI_STATUS) και εξάγει την πιο πρόσφατη τιμή. Στη συνέχεια, ενημερώνονται οι παρατηρητές (notifyHeartRateTrackerObservers) ώστε να ειδοποιηθούν για τα νέα δεδομένα.

```
public void readValuesFromDataPoint(DataPoint dataPoint) {
    final HeartRateData hrData = new HeartRateData();

    // Retrieve values from the dataPoint directly
    hrData.status = (Integer) dataPoint.getValue(ValueKey.HeartRateSet.HEART_RATE_STATUS);
    hrData.hr = (Integer) dataPoint.getValue(ValueKey.HeartRateSet.HEART_RATE);

    // Retrieve the IBI list and IBI status list
    final List<Integer> hrIbiList = (List<Integer>) dataPoint.getValue(ValueKey.HeartRateSet.IBI_LIST);
    final List<Integer> hrIbiStatus = (List<Integer>) dataPoint.getValue(ValueKey.HeartRateSet.IBI_STATUS_LIST);

    // Safely extract the last IBI and IBI status values
    if (hrIbiList != null && !hrIbiList.isEmpty()) {
        hrData.ibi = hrIbiList.get(hrIbiList.size() - 1);
    }

    if (hrIbiStatus != null && !hrIbiStatus.isEmpty()) {
        hrData.qIbi = hrIbiStatus.get(hrIbiStatus.size() - 1);
    }

    // Notify observers
    TrackerDataNotifier.getInstance().notifyHeartRateTrackerObservers(hrData);

    // Log for debugging
    Log.d(APP_TAG, "msg: " + "Processed DataPoint: " + dataPoint.toString());
}
```

Εικόνα 55. Ανάγνωση και επεξεργασία δεδομένων HR

Ανάλογα με την ποιότητα των δεδομένων, η εφαρμογή προσαρμόζει την εμφάνιση της μέτρησης. Αν τα δεδομένα είναι αξιόπιστα, εμφανίζονται σε λευκό χρώμα, ενώ σε περιπτώσεις χαμηλής ποιότητας δεδομένων ή αποτυχίας της μέτρησης, χρησιμοποιείται κόκκινο χρώμα για να ειδοποιηθεί ο χρήστης. Αν δεν υπάρχει διαθέσιμη μέτρηση, εμφανίζεται η προεπιλεγμένη ένδειξη (--). Ο κώδικας που ρυθμίζει την εμφάνιση που προαναφέρθηκε σύμφωνα με το status των δεδομένων φαίνεται παρακάτω.

```
private void updateUi(HeartRateData hrData) {
    txtHeartRateStatus.setText(String.valueOf(hrData.status));
    setStatus(hrData.status);

    if (hrData.status == HeartRateStatus.HR_STATUS_FIND_HR) {
        txtHeartRate.setText(String.valueOf(hrData.hr));
        txtHeartRateStatus.setTextColor(Color.WHITE);

        txtIbi.setText(String.valueOf(hrData.ibi));
        txtIbiStatus.setText(String.valueOf(hrData.qIbi));
        txtIbiStatus.setTextColor((hrData.qIbi == 0) ? Color.WHITE : Color.RED);
    } else {
        txtHeartRate.setText("0");
        txtHeartRateStatus.setTextColor(Color.RED);
        txtIbi.setText("0");
        txtIbiStatus.setText("1");
        txtIbiStatus.setTextColor(Color.RED);
    }
}
```

Εικόνα 56. Ενημέρωση του UI με τα δεδομένα HR

Η εφαρμογή ενσωματώνει επίσης μηχανισμούς διαχείρισης σφαλμάτων, οι οποίοι βασίζονται στις καταστάσεις HeartRateStatus και SpO2Status. Αν ανιχνευθεί κάποιο πρόβλημα, όπως κίνηση της συσκευής, χαμηλό σήμα, αποσύνδεση του αισθητήρα ή χαμηλή αξιοπιστία δεδομένων, ο χρήστης ενημερώνεται μέσω ειδοποιήσεων και κατάλληλων μηνυμάτων στην οθόνη.

```

private void setStatus(int status) {
    Log.i(APP_TAG, "msg: " + status);
    int stringId = "Initial heart rate measuring state.";
    switch (status) {
        case HeartRateStatus.HR_STATUS_FIND_HR:
            stringId = "Successful heart rate measurement.";
            break;
        case HeartRateStatus.HR_STATUS_NONE:
            break;
        case HeartRateStatus.HR_STATUS_ATTACHED:
            stringId = "Wearable is attached.";
            break;
        case HeartRateStatus.HR_STATUS_DETECT_MOVE:
            stringId = "Wearable move detection.";
            break;
        case HeartRateStatus.HR_STATUS_DETACHED:
            stringId = "Wearable is detached.";
            break;
        case HeartRateStatus.HR_STATUS_LOW_RELIABILITY:
            stringId = "PPG signal is weak.";
            break;
        case HeartRateStatus.HR_STATUS_VERY_LOW_RELIABILITY:
            stringId = "PPG signal is too weak.";
            break;
        case HeartRateStatus.HR_STATUS_NO_DATA_FLUSH:
            stringId = "flush() is called but no data.";
            break;
    }

    txtStatus.setText(getString(stringId));
}

```

Εικόνα 57. Ενημέρωση κατάστασης μετρητή HR

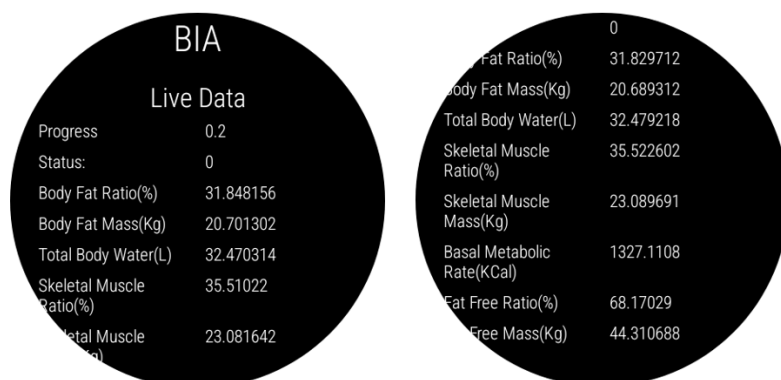
Σε περίπτωση που η μέτρηση ολοκληρωθεί επιτυχώς, το αποτέλεσμα εμφανίζεται κανονικά στην οθόνη, ενώ αν προκύψει σφάλμα, η μέτρηση τερματίζεται και ο αισθητήρας απενεργοποιείται για εξοικονόμηση ενέργειας.

Η εφαρμογή έχει επίσης προβλέψει μηχανισμούς αυτόματης διακοπής και αποσύνδεσης των αισθητήρων. Όταν η μέτρηση ολοκληρωθεί ή διακοπεί από τον χρήστη, η σύνδεση τερματίζεται, οι παρατηρητές αφαιρούνται και οι αισθητήρες σταματούν να λειτουργούν, ώστε να μην καταναλώνουν πόρους της συσκευής. Αν προκύψει αποσύνδεση από την πλατφόρμα υγείας ή η Samsung Health Tracking Service δεν είναι εγκατεστημένη ή ενημερωμένη, η εφαρμογή εμφανίζει στον χρήστη τα αντίστοιχα μηνύματα σφάλματος, προτρέποντάς τον να προβεί στις απαραίτητες ενέργειες.

Με αυτόν τον τρόπο, η εφαρμογή εξασφαλίζει ακριβείς μετρήσεις καρδιακού ρυθμού και κορεσμού οξυγόνου, με δυναμική ενημέρωση της διεπαφής χρήστη, διαχείριση σφαλμάτων και αποδοτική διαχείριση των πόρων της συσκευής, προσφέροντας μια αξιόπιστη και ομαλή εμπειρία χρήσης

4.2.6 Λήψη μετρήσεων BIA

Ο χρήστης έχει πρόσβαση και σε live μέτρηση BIA κρατώντας 2 δάχτυλα σε επαφή με καθένα από τα 2 κουμπιά της συσκευής.



Εικόνα 58. Οθόνη μετρήσεων BIA

Πριν από κάθε μέτρηση, η εφαρμογή πραγματοποιεί έναν έλεγχο αδειών, διασφαλίζοντας ότι έχει την απαιτούμενη άδεια (BODY_SENSORS) για την πρόσβαση στους αισθητήρες της συσκευής. Αν η άδεια δεν έχει δοθεί, η εφαρμογή ζητάει από τον χρήστη να την παραχωρήσει. Εφόσον οι άδειες είναι εξασφαλισμένες, δημιουργείται ένας ConnectionManager, ο οποίος αναλαμβάνει τη σύνδεση με την πλατφόρμα υγείας της Samsung, επιτρέποντας στην εφαρμογή να επικοινωνεί με τον βιοηλεκτρικό αισθητήρα ανάλυσης σύστασης σώματος. Ο κώδικας είναι παρόμοιος με τις προηγούμενες δύο υλοποιήσεις γι αυτό δεν παρατίθεται.

Μόλις η σύνδεση ολοκληρωθεί, αρχικοποιείται ο BiaListener, ο οποίος είναι υπεύθυνος για τη συλλογή και επεξεργασία των δεδομένων της μέτρησης από τον αισθητήρα. Τα δεδομένα που λαμβάνονται περιλαμβάνουν βασικές βιομετρικές παραμέτρους, όπως:

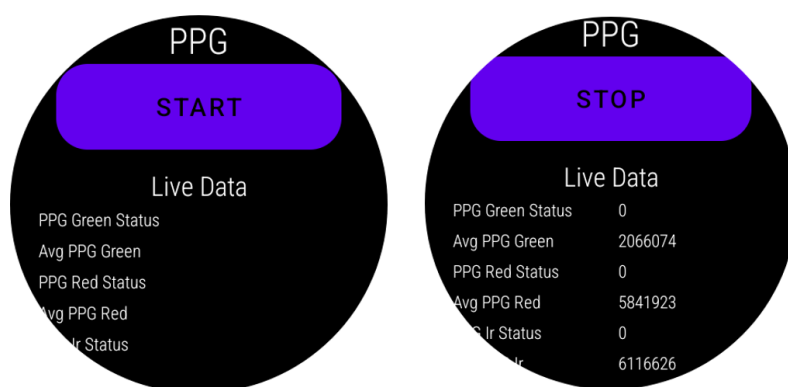
- Βασικός μεταβολικός ρυθμός (BMR) – Η ενέργεια που καταναλώνει το σώμα σε ηρεμία.
- Μάζα σωματικού λίπους (Body Fat Mass) – Η συνολική ποσότητα λίπους στο σώμα.
- Ποσοστό σωματικού λίπους (Body Fat Ratio) – Η αναλογία του λίπους σε σχέση με το συνολικό βάρος.
- Άλιπη μάζα σώματος (Fat-Free Mass, FFM) – Το συνολικό βάρος του σώματος χωρίς το λίπος.
- Ποσοστό άλιπης μάζας (Fat-Free Ratio) – Η αναλογία της άλιπης μάζας σε σχέση με το συνολικό βάρος.
- Σκελετική μυϊκή μάζα (Skeletal Muscle Mass) – Η συνολική ποσότητα των σκελετικών μυών.
- Ποσοστό σκελετικών μυών (Skeletal Muscle Ratio) – Η αναλογία της σκελετικής μυϊκής μάζας στο συνολικό βάρος.
- Συνολική ποσότητα νερού στο σώμα (Total Body Water) – Η ποσότητα του νερού που περιέχεται στο σώμα.

Για τη δυναμική ενημέρωση της διεπαφής χρήστη, χρησιμοποιείται ένας TrackerDataObserver, ο οποίος παρακολουθεί τις αλλαγές στα δεδομένα και τα εμφανίζει άμεσα στην οθόνη. Σε περίπτωση επιτυχημένης μέτρησης, οι τιμές ενημερώνονται και αποθηκεύονται, ενώ αν προκύψει κάποιο σφάλμα, όπως αποσύνδεση ηλεκτροδίων, ξηρότητα δέρματος ή χαλαρό ρολόι, η εφαρμογή εμφανίζει την κατάλληλη ειδοποίηση βασιζόμενη στις καταστάσεις σφαλμάτων του BiaStatus.

Για την ομαλή λειτουργία της εφαρμογής, η σύνδεση με τον αισθητήρα διακόπτεται αυτόματα όταν ο χρήστης κλείνει τη δραστηριότητα (BiaActivity), προκειμένου να αποφευχθεί η αχρείαστη κατανάλωση ενέργειας και πόρων της συσκευής. Παράλληλα, αν παρουσιαστεί κάποιο πρόβλημα σύνδεσης ή η πλατφόρμα υγείας της Samsung δεν είναι εγκατεστημένη ή ενημερωμένη, εμφανίζονται τα αντίστοιχα μηνύματα σφάλματος στον χρήστη.

4.2.7 Μέτρηση Φωτοπληθυσμογραφίας (PPG) – Πράσινο, Κόκκινο και Υπέρυθρο Φως

Τέλος, ο χρήστης μπορεί να μετρήσει τα PPG έχοντας και πάλι το δάχτυλό του στο επάνω κουμπί Home της συσκευής.



Εικόνα 59. Οθόνη μέτρησης PPG

Πριν από κάθε μέτρηση PPG (Φωτοπληθυσμογραφίας), η εφαρμογή ελέγχει αν έχει την απαραίτητη άδεια (BODY_SENSORS) για την πρόσβαση στους αισθητήρες. Αν η άδεια δεν έχει παραχωρηθεί, ζητείται από τον χρήστη να την επιτρέψει. Μόλις εξασφαλιστούν οι άδειες, δημιουργείται ένας ConnectionManager, ο οποίος αναλαμβάνει την επικοινωνία με την πλατφόρμα υγείας της Samsung και τη σύνδεση με τους αισθητήρες PPG.

Με την επιτυχή σύνδεση, αρχικοποιούνται οι PpgGreenListener, PpgRedListener και PpgIrListener, οι οποίοι είναι υπεύθυνοι για τη συλλογή και επεξεργασία των δεδομένων από τους αισθητήρες πράσινου, κόκκινου και υπέρυθρου φωτός αντίστοιχα. Τα δεδομένα που συλλέγονται περιλαμβάνουν την ένταση του ανακλώμενου φωτός από την κυκλοφορία του αίματος, η οποία χρησιμοποιείται για την εξαγωγή βιομετρικών παραμέτρων, όπως ο καρδιακός ρυθμός και ο κορεσμός οξυγόνου στο αίμα.

Κατά τη διάρκεια της μέτρησης, ένας TrackerDataObserver παρακολουθεί τις αλλαγές στα δεδομένα και ενημερώνει δυναμικά τη διεπαφή χρήστη, εμφανίζοντας τις τιμές σε πραγματικό χρόνο. Για κάθε τύπο μέτρησης, ο παρατηρητής ενημερώνει διαφορετικά στοιχεία της διεπαφής χρήστη:

- PPG Πράσινο (PpgGreenData): Συλλέγει δεδομένα από τον αισθητήρα πράσινου φωτός, τα οποία χρησιμοποιούνται κυρίως για τη μέτρηση του καρδιακού ρυθμού.
- PPG Κόκκινο (PpgRedData): Αξιοποιείται κυρίως για τη μέτρηση του κορεσμού οξυγόνου (SpO_2), καθώς το κόκκινο φως αλληλεπιδρά διαφορετικά με το οξυγονωμένο και μη οξυγονωμένο αίμα.
- PPG Υπέρυθρο (PpgIrData): Χρησιμοποιείται συχνά για την παρακολούθηση της κυκλοφορίας του αίματος και την ανίχνευση αλλαγών στη ροή.

Κατά τη συλλογή των δεδομένων, οι τιμές που λαμβάνονται επεξεργάζονται για τον υπολογισμό μέσων όρων και τη βελτίωση της ακρίβειας. Αν η ποιότητα του σήματος είναι χαμηλή ή υπάρχουν προβλήματα, όπως χαμηλό σήμα, αποσύνδεση του αισθητήρα ή υπερβολική κίνηση, εμφανίζονται τα αντίστοιχα μηνύματα σφάλματος στη διεπαφή χρήστη.

Η εφαρμογή προσφέρει επίσης μηχανισμούς διαχείρισης ενέργειας και αποσύνδεσης, ώστε να διασφαλίζεται η βέλτιστη απόδοση της συσκευής. Αν ο χρήστης σταματήσει τη μέτρηση, οι αισθητήρες απενεργοποιούνται αυτόματα και η σύνδεση με την πλατφόρμα υγείας της Samsung τερματίζεται. Σε περίπτωση αποτυχίας της σύνδεσης ή αν η πλατφόρμα Samsung Health δεν είναι εγκατεστημένη ή ενημερωμένη, εμφανίζεται μήνυμα σφάλματος στον χρήστη.

Με αυτόν τον τρόπο, η εφαρμογή διασφαλίζει αξιόπιστες και ενεργειακά αποδοτικές μετρήσεις PPG, προσφέροντας στον χρήστη άμεση ενημέρωση των δεδομένων, δυναμική προσαρμογή της διεπαφής χρήστη και διαχείριση σφαλμάτων για μια ομαλή εμπειρία χρήσης.

4.3 Συμπεράσματα και Ιδέες για αναβαθμίσεις της εφαρμογής

4.3.1 Τρέχουσα κατάσταση και δυνατότητες της εφαρμογής

Η εφαρμογή που αναπτύχθηκε για την παρούσα διπλωματική εργασία αποτελεί ένα ολοκληρωμένο σύστημα παρακολούθησης καρδιαγγειακής υγείας, καθώς καταγράφει και αναλύει ECG για την ανίχνευση αρρυθμιών και AV block, μετρά HR (καρδιακό ρυθμό) και SpO_2 (κορεσμό οξυγόνου στο αίμα) και χρησιμοποιεί BIA για ανάλυση σύστασης σώματος και PPG για καρδιακές μετρήσεις. Η δυνατότητα συνδυασμού αυτών των μετρήσεων την καθιστά ένα πολυδιάστατο εργαλείο με μεγάλη προοπτική για επεκτάσεις και βελτιώσεις.

4.3.2 Βελτίωση ακρίβειας και ανάλυσης δεδομένων

Μια σημαντική αναβάθμιση θα μπορούσε να είναι η ενσωμάτωση μηχανικής μάθησης για τη βελτίωση της ανάλυσης ECG. Η χρήση μοντέλων τεχνητής νοημοσύνης θα επέτρεπε την

αυτόματη ταξινόμηση αρρυθμιών όπως η κολπική μαρμαρυγή, οι έκτακτες κοιλιακές συστολές και τα διάφορα είδη AV block. Παράλληλα, η εφαρμογή θα μπορούσε να εφαρμόζει δυναμική προσαρμογή φίλτρων ανάλογα με το θόρυβο του σήματος, βελτιώνοντας έτσι την ακρίβεια των μετρήσεων. Επιπλέον, η ανάλυση της μεταβλητότητας καρδιακού ρυθμού (HRV) θα πρόσφερε χρήσιμες πληροφορίες για το επίπεδο άγχους και την αυτόνομη νευρική ρύθμιση του οργανισμού.

4.3.3 Επέκταση λειτουργιών υγείας και χρήση πρόσθετων αισθητήρων

Εκτός από τη βελτίωση των αλγορίθμων ανάλυσης, η εφαρμογή θα μπορούσε να επεκταθεί σε νέες λειτουργίες που θα βελτιώναν τη χρηστικότητά της. Η εκτίμηση του κινδύνου καρδιαγγειακών παθήσεων θα ήταν δυνατή με το συνδυασμό ECG, HRV, BIA και PPG, παρέχοντας μια πιο ολοκληρωμένη αξιολόγηση της καρδιακής υγείας. Επίσης, η προσθήκη μέτρησης του VO₂ max μέσω PPG θα έδινε μια ένδειξη της μέγιστης πρόσληψης οξυγόνου, που είναι ιδιαίτερα χρήσιμη για αθλητές και άτομα που επιθυμούν να παρακολουθούν τη φυσική τους κατάσταση. Μια άλλη ενδιαφέρουσα δυνατότητα θα ήταν η ανάλυση του ρυθμού αναπνοής με βάση τα δεδομένα ECG και PPG, κάτι που θα μπορούσε να συμβάλει στην ανίχνευση αναπνευστικών διαταραχών.

Επιπλέον, η ενσωμάτωση δεδομένων από το επιταχυνσιόμετρο και τον βηματομετρητή θα μπορούσε να προσφέρει ανίχνευση φυσικής δραστηριότητας και μεταβολών στη φυσιολογική κατάσταση του χρήστη. Για παράδειγμα, θα μπορούσε να συνδυάσει δεδομένα καρδιακού ρυθμού με επίπεδα δραστηριότητας για πιο ακριβή εκτίμηση της καρδιακής φόρτισης. Ο αισθητήρας μαγνητισμού θα μπορούσε επίσης να χρησιμοποιηθεί για αναγνώριση στάσης του σώματος ή ενδεχομένως για πιο ακριβή εντοπισμό θέσης σε συνδυασμό με το επιταχυνσιόμετρο.

4.3.4 Βελτίωση εμπειρίας χρήστη και διαδραστικότητας

Στον τομέα της εμπειρίας χρήστη, η εφαρμογή θα μπορούσε να υποστηρίξει real-time streaming του ECG με άμεση ανίχνευση αρρυθμιών, εμφανίζοντας ειδοποιήσεις σε περίπτωση ανωμαλιών. Ένα διαδραστικό dashboard με γραφήματα και αναφορές θα έκανε πιο εύκολη την κατανόηση των μετρήσεων, ενώ η δημιουργία προφίλ χρήστη με καταγραφή ιστορικών δεδομένων θα βοηθούσε στη δημιουργία πιο προσωποποιημένων αναλύσεων και συστάσεων.

4.3.5 Διασύνδεση με Cloud και Wearables

Τέλος, η εφαρμογή θα μπορούσε να ενισχυθεί με διασύνδεση με το cloud και APIs για ιατρική χρήση, επιτρέποντας την αποθήκευση δεδομένων και την κοινοποίησή τους σε γιατρούς για απομακρυσμένη αξιολόγηση. Η υποστήριξη για περισσότερες wearable συσκευές, όπως αυτές της Garmin ή του Apple Watch, θα διευρυνε τη βάση χρηστών της, ενώ η ενσωμάτωση με υπηρεσίες όπως το Google Fit ή το Apple Health θα διευκόλυνε τη διαχείριση των δεδομένων υγείας σε ένα ενιαίο περιβάλλον.

Βιβλιογραφία

- [1] Samsung, *Galaxy Watch 4 – Specifications*, [Online].
https://www.samsung.com/gr/watches/galaxy-watch/galaxy-watch4-classic-black-bluetooth-46mm-sm-r890nzkaeue/?srsId=AfmBOoowplAJ2udogRRZu-6eC4aWJQtp_DQKbaennPdBskfmx6_54FVc
- [2] Oracle, *Java*, [Online]. https://www.java.com/en/download/help/whatis_java.html
- [3] Google, *Android Studio*, [Online]. <https://developer.android.com/studio>
- [4] Python Software Foundation, *Python*, [Online]. <https://www.python.org/>
- [5] Samsung, *Samsung Health Sensor SDK*, [Online].
<https://developer.samsung.com/health/sensor/overview.html>
- [6] Google, *Android Developers: Wear*, [Online].
<https://developer.android.com/jetpack/androidx/releases/wear>
- [7] Google, *Android Developers: Activity*, [Online].
<https://developer.android.com/guide/components/activities/intro-activities>
- [8] Google, *Android Developers: AppCompatActivity*, [Online].
<https://developer.android.com/jetpack/androidx/releases/appcompat>
- [9] Google, *Android Developers: Material Components*, [Online].
<https://developer.android.com/develop/ui/compose/components>
- [10] Google, *Android Developers: ConstraintLayout*, [Online].
<https://developer.android.com/develop/ui/views/layout/constraint-layout>
- [11] Google, *Google Play Services: Wearable*, [Online].
<https://developers.google.com/android/reference/com/google/android/gms/wearable/Wearable>
- [12] Apache Software Foundation, *Apache Commons Math: The Apache Commons Mathematics Library*, [Online]. <https://commons.apache.org/proper/commons-math/>
- [13] Chaquopy, *Python for Android*, [Online]. <https://chaquo.com/chaquopy/>
- [14] WaveBeans, *JDSP: Java Digital Signal Processing Library*, [Online].
<https://github.com/WaveBeans/jDSP>
- [15] Google, *Android Developers: RecyclerView*, [Online].
<https://developer.android.com/guide/topics/ui/layout/recyclerview>
- [16] Twilio, *SendGrid Java Library*, [Online]. <https://github.com/sendgrid/sendgrid-java>
- [17] Apache Software Foundation, *Apache HttpClient: A Java library for HTTP communication*, [Online]. <https://hc.apache.org/httpcomponents-client-5.2.x/index.html>
- [18] K. Z. Summers, O. Agrippa, D. Ade-Odunlade, and K. A. Anie, "AI-driven prediction of vaso-occlusive crises in sickle cell disease: Precision through advanced machine-learning

frameworks and digital remote monitoring," *Blood*, Elsevier, 2024.

<https://www.sciencedirect.com/science/article/pii/S0006497124032695>

[19] J. Ernstsson, B. Svensson, P. Liuba, et al., "Validation of smartwatch electrocardiogram intervals in children compared to standard 12 lead electrocardiograms," *European Journal of Pediatrics*, Springer, 2024. <https://link.springer.com/article/10.1007/s00431-024-05648-7>

[20] A. Banerjee, "Artificial intelligence enabled mobile health technologies in arrhythmias—An opinion article on recent findings," *Frontiers in Cardiovascular Medicine*, 2025. <https://www.frontiersin.org/journals/cardiovascular-medicine/articles/10.3389/fcvm.2025.1548554/full>

[21] C. Breen and J. Hendriks, "Advancing cardiovascular health: photoplethysmography as a tool for electrocardiogram signal acquisition," *European Journal of Cardiovascular Nursing*, 2025. <https://academic.oup.com/eurjcn/advance-article-abstract/doi/10.1093/eurjcn/zvae176/7947834>

[22] J. I. V. de Oliveira, C. S. Lima, F. C. Bertocco, et al., "Assessment of Samsung Galaxy Watch4 PPG-based heart rate during light-to-vigorous physical activities," *IEEE Sensors Journal*, 2024. <https://ieeexplore.ieee.org/abstract/document/10543174/>

[23] F. Sarhaddi, K. Kazemi, I. Azimi, et al., "A comprehensive accuracy assessment of Samsung smartwatch heart rate and heart rate variability," *PLoS One*, 2022. <https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0268361>

[24] A. Hossain, A. Shaikh, and S. Hossain, "Comprehensive review on cardiovascular diseases: Coronary artery disease, arrhythmias, valvular disease, vascular disease, systemic hypertension, pericardial disease, and congenital heart disorders," *Circulation, American Heart Association*, 2024. https://www.ahajournals.org/doi/abs/10.1161/circ.150.suppl_1.4142343

[25] T. Pullaiah and S. Ojha, *Cardioprotective plants: Prevention and cure of hypertension and cardiovascular diseases*, Springer, 2024. https://www.researchgate.net/profile/Muhammad_Riaz4/publication/383295373_Plants_and_Phytochemicals_for_Prevention_and_Cure_of_Hypertension/links/670ab974a1215201916b1bc9/Plants-and-Phytochemicals-for-Prevention-and-Cure-of-Hypertension.pdf

[26] E. Faderin, O. G. Oginni, and B. Alade, "Telehealth innovations for cardiovascular disease management," *ResearchGate*, 2024. https://www.researchgate.net/publication/384734854_Telehealth_innovations_for_cardiovascular_disease_management

[27] S. Kiranyaz and M. Yamac, "Personalized and zero-shot electrocardiogram arrhythmia monitoring system," *Trepo*, 2024. <https://trepo.tuni.fi/bitstream/handle/10024/162192/DumanMert.pdf?sequence=2>

[28] A. K. Nagpal, A. Pundkar, A. Singh, and C. Gadkari, "Cardiac arrhythmias and their management: An in-depth review of current practices and emerging therapies," *Cureus*,

2024. <https://www.cureus.com/articles/283186-cardiac-arrhythmias-and-their-management-an-in-depth-review-of-current-practices-and-emerging-therapies.pdf>
- [29] A. H. Kashou, A. Goyal, T. Nguyen, and I. Ahmed, *Atrioventricular Block (Nursing)*, StatPearls, 2024. <https://www.ncbi.nlm.nih.gov/sites/books/NBK568758/>
- [30] A. Ç. Yumurtaş, D. I. Satti, J. M. H. Hui, and L. Pay, "Arrhythmias beyond atrial fibrillation detection using smartwatches: A systematic review," *Anatolian Journal of Cardiology*, 2023.
https://jag.journalagent.com/z4/download_fulltext.asp?pdire=anatoljcardiol&un=AJC-77096
- [31] J. Cao, L. Zhang, and B. Tang, "Application of different types of electrocardiogram monitoring in arrhythmias," *International Journal of Heart Rhythm*, 2024.
https://journals.lww.com/ijhr/fulltext/2024/01000/application_of_different_types_of.2.aspx?context=latestarticles
- [32] E. Fung, M. R. Järvelin, R. N. Doshi, and J. S. Shinbane, "Electrocardiographic patch devices and contemporary wireless cardiac monitoring," *Frontiers in Physiology*, 2015.
<https://www.frontiersin.org/journals/physiology/articles/10.3389/fphys.2015.00149/pdf>
- [33] S. H. Browne, M. Bernstein, and P. E. Bickler, "Evaluation of Pulse Oximetry Accuracy in a Commercial Smartphone and Smartwatch Device During Human Hypoxia Laboratory Testing," *Sensors*, 2025. <https://www.mdpi.com/1424-8220/25/5/1286>
- [34] E. Mazza, S. Maurotti, Y. Ferro, P. Doria, and M. Moraca, "Portable bioimpedance analyzer for remote body composition monitoring: A clinical investigation under controlled conditions," *Nutrition*, Elsevier, 2024.
<https://www.sciencedirect.com/science/article/pii/S0899900724001862>
- [35] R. Durbha and V. Koomson, "Chromasense-empowering health, empowering you," *IEEE Circuits and Systems Magazine*, 2024.
<https://ieeexplore.ieee.org/abstract/document/10639303/>
- [36] Wikipedia, *Ευέλικτη Μεθοδολογία*, [Online].
https://el.wikipedia.org/wiki/Ευέλικτη_μεθοδολογία
- [37] Indevlab, *What is Agile Development*, [Online]. <https://indevlab.com/blog/what-is-agile-development/>