ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ
ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΜΑΚΕΔΟΝΙΑΣ
ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ
ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
& ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΥΠΟΛΟΓΙΣΜΟΣ ΜΕΤΑΒΛΗΤΟΤΗΤΑΣ ΚΑΡΔΙΑΚΟΥ ΡΥΘΜΟΥ ΣΕ ΠΡΑΓΜΑΤΙΚΟ ΧΡΟΝΟ ΜΕΣΩ ΦΩΤΟΠΛΗΘΥΣΜΟΓΡΑΦΙΑΣ ΚΟΚΚΙΝΗΣ ΑΚΤΙΝΟΒΟΛΙΑΣ

Ανάλυση της σχέσης με την πνευματική υγεία και ανάπτυξη εφαρμογής σε έξυπνο ρολόι για υπολογισμό άγχους

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

της

ΤΣΑΠΑΛΗ ΜΑΡΙΑ ΑΝΝΑ

Επιβλέπων: Αγγελίδης Παντελής

Καθηγητής

ΚΟΖΑΝΗ/ΣΕΠΤΕΜΒΡΙΟΣ/2024

HELLENIC DEMOCRACY
UNIVERSITY OF WESTERN MACEDONIA
SCHOOL OF ENGINEERING
DEPARTMENT OF ELECTRICAL
& COMPUTER ENGINEERING
ece.uowm.gr

REAL-TIME CALCULATION OF HEART RATE VARIABILITY USING RED PHOTOPLETHYSMOGRAPHY SIGNALS

Analysis of Its Relationship with Mental Health and Development of an Application on Smartwatch for Anxiety Assessment

THESIS

TSAPALI MARIA ANNA

SUPERVISOR: Angelidis Pantelis

Professor

KOZANI/SEPTEMBER/2024

ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ
ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΜΑΚΕΔΟΝΙΑΣ
ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ
ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
& ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΔΗΛΩΣΗ ΜΗ ΛΟΓΟΚΛΟΠΗΣ ΚΑΙ ΑΝΑΛΗΨΗΣ ΠΡΟΣΩΠΙΚΗΣ ΕΥΘΥΝΗΣ

Δηλώνω ρητά ότι, σύμφωνα με το άρθρο 8 του Ν. 1599/1986 και τα άρθρα 2,4,6 παρ. 3 του Ν. 1256/1982, η παρούσα Διπλωματική Εργασία με τίτλο " ΥΠΟΛΟΓΙΣΜΟΣ ΜΕΤΑΒΛΗΤΟΤΗΤΑΣ ΚΑΡΔΙΑΚΟΥ ΡΥΘΜΟΥ ΣΕ ΠΡΑΓΜΑΤΙΚΟ ΧΡΟΝΟ ΜΕΣΩ ΦΩΤΟΠΛΗΘΥΣΜΟΓΡΑΦΙΑΣ ΚΟΚΚΙΝΗΣ ΑΚΤΙΝΟΒΟΛΙΑΣ ΚΑΙ ΣΥΣΧΕΤΙΣΜΟΣ ΜΕ ΤΗΝ ΠΝΕΥΜΑΤΙΚΗ ΥΓΕΙΑ" καθώς και τα ηλεκτρονικά αρχεία και πηγαίοι κώδικες που αναπτύχθηκαν ή τροποποιήθηκαν στα πλαίσια αυτής της εργασίας και αναφέρονται ρητώς μέσα στο κείμενο που συνοδεύουν, και η οποία έχει εκπονηθεί στο Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών του Πανεπιστημίου Δυτικής Μακεδονίας, υπό την επίβλεψη του μέλους του Τμήματος κ. Αγγελίδη Παντελή αποτελεί αποκλειστικά προϊόν προσωπικής εργασίας και δεν προσβάλλει κάθε μορφής πνευματικά δικαιώματα τρίτων και δεν είναι προϊόν μερικής ή ολικής αντιγραφής, οι πηγές δε που χρησιμοποιήθηκαν περιορίζονται στις βιβλιογραφικές αναφορές και μόνον. Τα σημεία όπου έχω χρησιμοποιήσει ιδέες, κείμενο, αρχεία ή / και πηγές άλλων συγγραφέων, αναφέρονται ευδιάκριτα στο κείμενο με την κατάλληλη παραπομπή και η σχετική αναφορά περιλαμβάνεται στο τμήμα των βιβλιογραφικών αναφορών με πλήρη περιγραφή. Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τον συγγραφέα. Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον συγγραφέα και μόνο.

Copyright (C) Τσάπαλη Μαρία Άννα & Αγγελίδης Παντελής, 2024, Κοζάνη

Υπογραφή Φοιτητή: _____

Περίληψη

Η παρούσα διπλωματική εργασία ασχολείται με τον υπολογισμό της μεταβλητότητας του καρδιακού ρυθμού (HRV) σε πραγματικό χρόνο μέσω της χρήσης φωτοπληθυσμογραφίας (PPG) κόκκινης ακτινοβολίας. Η HRV είναι ένας σημαντικός βιοδείκτης της λειτουργίας του αυτόνομου νευρικού συστήματος, καθώς και της σωματικής και ψυχικής υγείας. Χαρακτηρίζεται από τις χρονικές διακυμάνσεις μεταξύ των καρδιακών παλμών και μπορεί να παρέχει πληροφορίες για την ισορροπία του συμπαθητικού και παρασυμπαθητικού συστήματος, συνδέοντας έτσι τον καρδιακό ρυθμό με τη συναισθηματική κατάσταση ενός ατόμου.

Η εργασία εστιάζει στην ανάπτυξη μίας εφαρμογής για έξυπνα ρολόγια (smartwatches) που χρησιμοποιεί σήματα PPG για τους υπολογισμούς δεικτών της HRV και των επιπέδων άγχους. Οι δείκτες HRV, όπως οι SDNN, RMSSD και pNN50, υπολογίζονται με βάση τα διαστήματα RR, τα οποία εξάγονται από το PPG σήμα μέσω αλγορίθμων όπως ο Automatic Multiscale Peak Detection (AMPD). Επιπρόσθετα, φίλτρα όπως το Butterworth χρησιμοποιούνται για την απομάκρυνση θορύβου και τη βελτίωση της ακρίβειας των μετρήσεων.

Η εφαρμογή επιτρέπει την πραγματοποίηση μετρήσεων διάρκειας 60 ή 120 δευτερολέπτων, προσφέροντας γρήγορη αξιολόγηση των επιπέδων άγχους σε καθημερινές συνθήκες. Τα δεδομένα που συλλέγονται συγκρίνονται με βασικές τιμές για την εκτίμηση του στρες, χρησιμοποιώντας βάσεις δεδομένων όπως αυτή της EliteHRV, αλλά και δείγματος συμμετεχόντων. Συμπερασματικά, μετά από έρευνα για την επιβεβαίωση των αποτελεσμάτων, καταλήγει ότι η HRV μπορεί να χρησιμοποιηθεί ως αξιόπιστος δείκτης της ψυχικής ευεξίας και ότι εφαρμογές όπως αυτή που αναπτύχθηκε μπορούν να προσφέρουν ουσιαστική βοήθεια στον έλεγχο του άγχους στην καθημερινότητα ενός ατόμου.

Λέξεις Κλειδιά

Μεταβλητότητα καρδιακού ρυθμού (HRV), φωτοπληθυσμογραφία (PPG), έξυπνες φορητές συσκευές, διαχείριση άγχους, αυτόνομο νευρικό σύστημα, αλγόριθμοι επεξεργασίας σήματος, εφαρμογές υγείας, φίλτρο Butterworth.

Abstract

This thesis focuses on the real-time calculation of heart rate variability (HRV) using red photoplethysmography (PPG). HRV is an important biomarker of the autonomic nervous system's function, as well as physical and mental health. It is characterized by the time variations between heartbeats and can provide information about the balance of the sympathetic and parasympathetic systems, thus linking heart rate to an individual's emotional state.

This project centers on the development of a smartwatch application that uses PPG signals to calculate HRV indicators and stress levels. HRV indicators, such as SDNN, RMSSD, and pNN50, are computed based on RR intervals, which are extracted from the PPG signal through algorithms like Automatic Multiscale Peak Detection (AMPD). Additionally, filters such as the Butterworth filter are used to remove noise and enhance measurement accuracy.

The application allows for measurements of 60 or 120 seconds, offering a quick assessment of stress levels in everyday situations. The collected data are compared with baseline values for stress estimation, using databases like EliteHRV and a sample of participants. In conclusion, after research to validate the results, it is determined that HRV can be used as a reliable indicator of mental well-being, and applications like the one developed can offer substantial support in managing stress in daily life.

Keywords

Heart rate variability (HRV), photoplethysmography (PPG), wearable devices, stress management, autonomic nervous system, signal processing algorithms, health applications, Butterworth filter.

Ευχαριστίες

Θα ήθελα να εκφράσω τις θερμές μου ευχαριστίες στην εταιρεία Vidavo και ιδιαιτέρως στην κα. Χίτζιου Θ. για την εξαιρετική συνεργασία και την πολύτιμη βοήθειά τους καθ' όλη τη διάρκεια της διπλωματικής εργασίας. Επίσης, ευχαριστώ τον επιβλέποντα καθηγητή μου, κο. Αγγελίδη, για την καθοδήγηση και την υποστήριξή του, καθώς και την εταιρεία Samsung για την παροχή των έξυπνων ρολογιών που χρησιμοποιήθηκαν για την υλοποίηση της εφαρμογής. Θερμές ευχαριστίες στον κο. Παπακυριαζή Η. για την ευγενική παροχή του Android κινητού τηλεφώνου. Τέλος, ευχαριστώ όλους τους συμμετέχοντες που προσφέρθηκαν εθελοντικά και συνέβαλαν καθοριστικά στην επιβεβαίωση των αποτελεσμάτων της έρευνας.

ΚΟΖΑΝΗ/ΜΗΝΑΣ/2024

Περιεχόμενα

ΥΠΟΛΟΓΙΣΜΟΣ ΜΕΤΑΒΛΗΤΟΤΗΤΑΣ ΚΑΡΔΙΑΚΟΥ ΡΥΘΜΟΥ ΣΕ ΠΡΑΓΜΑΤΙΚΟ ΧΡΟΝΟ ΜΕΣΩ ΦΩΤΟΠΛΗΘΥΣΜΟΓΡΑΦΙΑΣ ΚΟΚΚΙΝΗΣ ΑΚΤΙΝΟΒΟΛΙΑΣ 1

REAL-TIME CALCULATION OF HEART RATE VARIABILITY USING RED PHOTOPLETHYSMOGRAPHY SIGNALS 3

Περίληψη 7

Abstract 8

Ευχαριστίες 9

Περιεχόμενα 11

Κατάλογος Σχημάτων 14

Κατάλογος Εικόνων 15

Κατάλογος Πινάκων 16

Πρόλογος 17

Κεφάλαιο 1: Εισαγωγή 18

1.1 ΑΝΤΙΚΕΙΜΕΝΟ ΤΗΣ ΔΙΠΛΩΜΑΤΙΚΗΣ 18

1.2 ΟΡΓΑΝΩΣΗ ΤΟΥ ΤΟΜΟΥ 18

Κεφάλαιο 2: Θεωρητικό υπόβαθρο 20

2.1 ΕΙΣΑΓΩΓΙΚΕΣ ΕΝΝΟΙΕΣ 20

2.1.1 Συμπαθητικό και παρασυμπαθητικό σύστημα 20

2.1.2 Σύμπλεγμα QRS 20

2.1.3 Χρονικά διαστήματα RR και NN 21

2.1.4 Φίλτρο και φιλτράρισμα ενός σήματος 22

2.1.5 Φίλτρο Butterworth 22

2.1.6 Κανονικοποίηση τιμών 23

2.1.7 Διακριτός Μετασχηματισμός Fourier 24

2.1.8 Γρήγορος Μετασχηματισμός Fourier 25

2.1.9 Φάσμα Ισχύος 26

2.2 ΕΙΣΑΓΩΓΗ ΣΤΗΝ ΜΕΤΑΒΛΗΤΟΤΗΤΑ ΤΟΥ ΚΑΡΔΙΑΚΟΥ ΡΥΘΜΟΥ (HRV) 26

2.2.1 Ορισμός της HRV και σημασία της στην ανάλυση της λειτουργίας του ανθρώπινου νευρικού συστήματος 27

Κεφάλαιο 3: Ανάλυση Θέματος 33
3.1 ΑΝΑΣΚΟΠΗΣΗ ΤΗΣ ΒΙΒΛΙΟΓΡΑΦΙΑΣ 33
3.2 ΘΕΩΡΗΤΙΚΟ ΠΛΑΙΣΙΟ 34
3.3 ΕΡΕΥΝΗΤΙΚΕΣ ΕΡΩΤΗΣΕΙΣ ΚΑΙ ΥΠΟΘΕΣΕΙΣ 35
Κεφάλαιο 4: Σχεδίαση Θέματος 37
4.1 ΑΝΑΛΥΣΗ ΑΝΑΓΚΩΝ ΚΑΙ ΣΤΟΧΩΝ 37
4.2 ΣΧΕΔΙΑΣΜΟΣ ΕΡΕΥΝΗΤΙΚΟΥ ΠΛΑΝΟΥ 37
4.2.1 Αλγόριθμοι, μεθοδολογίες και έρευνες 37
4.2.2 Εξοπλισμός και εφαρμογές 38
4.2.3 Δομή και οργάνωση κώδικα 39
4.2.4 Επιβεβαίωση αποτελεσμάτων 40
Κεφάλαιο 5: Υλοποίηση 42
5.1 ΕΠΙΣΚΟΠΗΣΗ ΤΕΧΝΟΛΟΓΙΩΝ ΚΑΙ ΕΡΓΑΛΕΙΩΝ 42
5.1.1 Έξυπνα Ρολόγια Samsung Galaxy Watch 5 42
5.1.2 Χρήση του Samsung Privileged Health SDK 42
5.1.3 Χρήση Android Studio Code και Kotlin για την ανάπτυξη εφαρμογής σε smartwatch 43
5.2 ΑΝΑΛΥΣΗ ΚΩΔΙΚΑ ΥΛΟΠΟΙΗΣΗΣ ΤΗΣ ΕΦΑΡΜΟΓΗΣ 44
5.2.1 Εκκίνηση και Βασικά Στοιχεία της Εφαρμογής 44
5.2.2 Διεπαφή Χρήστη 49
5.2.3 Σύνδεση με Samsung Health Platform, Άδειες Χρήστη και Καταγραφή Δεδομένων 60
5.2.4 Αλγόριθμος Υπολογισμού της HRV από σήμα PPG 65
Κεφάλαιο 6: Αποτελέσματα 85
6.1 ΕΙΣΑΓΩΓΗ 85
6.2 ΛΕΠΤΟΜΕΡΕΙΕΣ ΣΥΜΜΕΤΕΧΟΝΤΩΝ 85
6.3 ΕΠΙΣΚΟΠΗΣΗ ΣΥΛΛΟΓΗΣ ΔΕΔΟΜΕΝΩΝ 87
6.3.1 Στάδιο Προετοιμασίας 87
6.3.2 Διαδικασία μέτρησης 88
6.3.3 Καταγραφή Δεδομένων 88

6.4 ΣΥΓΚΡΙΣΗ ΜΕΤΡΗΣΕΩΝ 60S ΚΑΙ 120S 88
6.4.1 Περιγραφή της Μεθόδου 88
6.4.2 Ανάλυση Αποτελεσμάτων 88
6.5 ΣΥΓΚΡΙΣΗ ΜΕΤΑΞΥ ΕΦΑΡΜΟΓΩΝ 89
6.5.1 Περιγραφή της Μεθόδου 89
6.5.2 Ανάλυση Αποτελεσμάτων 90
6.6 ΣΥΓΚΡΙΣΗ ΜΕ PSS SCORE 91
6.6.1 Περιγραφή της Μεθόδου 91
6.6.2 Ανάλυση Αποτελεσμάτων 91
Κεφάλαιο 7: Επίλογος 93
7.1 ΣΥΜΠΕΡΑΣΜΑΤΑ 93
7.2 ΜΕΛΛΟΝΤΙΚΕΣ ΕΠΕΚΤΑΣΕΙΣ 93
Παραρτήματα 95
ΠΑΡΑΡΤΗΜΑ 1: ΕΡΩΤΗΣΕΙΣ ΑΝΤΙΛΗΨΗΣ ΆΓΧΟΥΣ 95
Βιβλιογραφία 97
Συντομογραφίες - Αρκτικόλεξα - Ακρωνύμια 100
Απόδοση ξενόγλωσσων όρων 101

Κατάλογος Σχημάτων

Σχήμα 1 - Διαδικασία Συλλογής Δεδομένων από smartwatch μέσω του Samsung Privileged Health SDK....43
Σχήμα 2- Διαγράμματα Φυσιολογικών τιμών HRV δεικτών ανάλογα με το φύλο και την ηλικιακή ομάδα....48
Σχήμα 3- Κατανομή φύλου συμμετεχόντων....85
Σχήμα 4- Κατανομή ηλικιακών ομάδων συμμετεχόντων....86
Σχήμα 5- Κατανομή ωρών ύπνου συμμετεχόντων....86
Σχήμα 6- Κατανομή επιπέδου άσκησης συμμετεχόντων....87
Σχήμα 7- Κατανομή κατανάλωσης καφεΐνης ή αλκοόλ συμμετεχόντων....87
Σχήμα 8 - Διάγραμμα αποκλίσεων τιμών ανά δείκτη HRV μεταξύ συνεχόμενων μετρήσεων διαφορετικής διάρκειας (60s και 120s)....89
Σχήμα 9 - Διάγραμμα αποκλίσεων τιμών ανά δείκτη HRV μεταξύ της εφαρμογής μας και μιας παρόμοιας ευρέως χρησιμοποιούμενης....90

Κατάλογος Εικόνων

Εικόνα 1: Μορφή του συμπλόκου QRS....21
Εικόνα 2 - Χρονικό Διάστημα RR....21
Εικόνα 3 - Φίλτρο Butterworth για διαφορετικές τάξεις....23
Εικόνα 4 - Τύπος κανονικοποίησης Ελαχίστου – Μεγίστου....24
Εικόνα 7 – Κατά την έναρξη της εφαρμογής/εισαγωγή στοιχείων του χρήστη....50
Εικόνα 8 - Εμφάνιση μηνύματος σφάλματος κατά την πληκτρολόγηση μη αποδεκτών τιμών από τον χρήστη....50
Εικόνα 9 - Οθόνη εμφάνισης σφάλματος κατά τη μέτρηση....53
Εικόνα 10 - Αρχικό Μενού Εφαρμογής / πριν την μέτρηση....54
Εικόνα 11 - Όψη Εφαρμογής μετά την μέτρηση....54
Εικόνα 12 - Κουμπί αλλαγής τιμής μέτρησης με διαφορετικές τιμές ανάλογα με των αριθμό των πατημάτων....55
Εικόνα 13 - Οθόνη Καταγραφής Σήματος....59
Εικόνα 14 - Οθόνη Επεξεργασίας Δεδομένων....60
Εικόνα 15 - Κριτήριο Επιλογής Τοπικού Μεγίστου....74
Εικόνα 16 - Τύπος Υπολογισμού παραμέτρου b....76
Εικόνα 17 - Τύπος Υπολογισμού p2....76

Κατάλογος Πινάκων

Πίνακας 1: Ελάχιστη Χρονική Περίοδος Απαιτούμενη για την Εκτίμηση των Μετρήσεων HRV 29
Πίνακας 2 - Αντιστοίχιση επιπέδου άγχους με χρώμα 65
Πίνακας 3 - Συγκεντρωτικός πίνακας αποκλίσεων τιμών ανά δείκτη HRV μεταξύ συνεχόμενων μετρήσεων διαφορετικής διάρκειας (60s και 120s) 89
Πίνακας 4 - Συγκεντρωτικός πίνακας αποκλίσεων τιμών ανά δείκτη HRV μεταξύ της εφαρμογής μας και μιας παρόμοιας ευρέως χρησιμοποιούμενη 90
Πίνακας 6 - Συγκεντρωτικός πίνακας αποκλίσεων τιμών άγχους μεταξύ της εφαρμογής μας και του PSS Score 91

Πρόλογος

Η παρούσα διπλωματική εργασία εκπονήθηκε στο πλαίσιο των σπουδών μου στο Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών του Πανεπιστημίου Δυτικής Μακεδονίας. Το αντικείμενο της εργασίας αφορά τον υπολογισμό της μεταβλητότητας του καρδιακού ρυθμού (HRV) σε πραγματικό χρόνο μέσω φωτοπληθυσμογραφίας (PPG) και την ανάπτυξη μιας εφαρμογής για έξυπνα ρολόγια (smartwatches) με στόχο την εκτίμηση των επιπέδων άγχους.

Η ιδέα για αυτή την έρευνα προέκυψε από την αυξανόμενη σημασία της χρήσης φορητών συσκευών για την παρακολούθηση της υγείας και τη διαχείριση του άγχους στην καθημερινότητα. Με την τεχνολογία να παίζει όλο και μεγαλύτερο ρόλο στη διατήρηση της ευεξίας, θεώρησα σημαντικό να διερευνήσω τον τρόπο με τον οποίο οι φορητές συσκευές μπορούν να προσφέρουν ουσιαστική βοήθεια στον έλεγχο του στρες μέσω ακριβών και άμεσων μετρήσεων της HRV.

Η εργασία αυτή αποτελεί προσωπικό επίτευγμα αλλά και μια ευκαιρία να συνεισφέρω στην επιστημονική έρευνα για τη βελτίωση της ποιότητας ζωής μέσω τεχνολογίας.

Κεφάλαιο 1: Εισαγωγή

Η εισαγωγή αυτού του κεφαλαίου παρέχει μια επισκόπηση των βασικών εννοιών και του αντικειμένου της διπλωματικής εργασίας. Ειδικότερα, αναλύεται η σημασία της μεταβλητότητας του καρδιακού ρυθμού και οι λόγοι για τους οποίους αποτελεί έναν σημαντικό δείκτη υγείας και ευεξίας.

1.1 Αντικείμενο της διπλωματικής

Η **μεταβλητότητα του καρδιακού ρυθμού**, ή **HRV (Heart Rate Variability)**, ορίζεται ως η διακύμανση στα χρονικά διαστήματα μεταξύ των καρδιακών παλμών για μια συγκεκριμένη χρονική περίοδο [1]. Αποτελεί έναν σημαντικό δείκτη υγείας, διότι αντανακλά την ισορροπία και την ευεξία του ανθρώπινου νευρικού συστήματος. Ακριβέστερα, αντιπροσωπεύει την ικανότητα του ανθρώπινου οργανισμού να προσαρμόζεται σε διάφορες φυσιολογικές και παθολογικές καταστάσεις. Η υψηλή μεταβλητότητα του καρδιακού ρυθμού σχετίζεται συνήθως με καλή υγεία και φυσική κατάσταση, ενώ χαμηλές τιμές μπορεί να υποδεικνύουν προβλήματα στο νευρικό σύστημα. Επιπλέον, η HRV έχει συσχετιστεί με διάφορες πτυχές της ψυχοφυσιολογίας και της συναισθηματικής κατάστασης ενός ατόμου. Έρευνες έχουν δείξει ότι υψηλές τιμές δεικτών της HRV συνδέονται με μειωμένο άγχος, καλύτερη αντιμετώπιση απαιτητικών καταστάσεων και ανώτερες επιδόσεις σε καθημερινές δραστηριότητες [2]. Από την άλλη πλευρά, οι χαμηλές τιμές μπορεί να συσχετιστούν με ψυχολογικές διαταραχές, κατάθλιψη και υψηλά επίπεδα στρες.

Μέσω της χρήσης σήματος φωτοπληθυσμογραφίας (PPG), μπορεί να πραγματοποιηθεί ο υπολογισμός της HRV σε πραγματικό χρόνο με αρκετά ικανοποιητική ακρίβεια, ανοίγοντας νέους δρόμους για την αντιμετώπιση διαφόρων προβλημάτων υγείας και τη βελτίωση της ψυχολογικής ευεξίας [3].

Γι αυτόν τον λόγο, η παρούσα εργασία στοχεύει στη μελέτη της HRV και τη χρήση της φωτοπληθυσμογραφίας για τον άμεσο υπολογισμό της, καθώς και στον συσχετισμό της με την πνευματική υγεία και συγκεκριμένα με τον υπολογισμό του άγχους.

1.2 Οργάνωση του τόμου

Το έγγραφο είναι οργανωμένο σε επτά κύρια κεφάλαια, καθένα από τα οποία συμβάλλει στη διαμόρφωση μιας ολοκληρωμένης εικόνας για τον υπολογισμό της μεταβλητότητας του καρδιακού ρυθμού και την ανάπτυξη μιας εφαρμογής για την αξιολόγηση του άγχους.

- Στο **Κεφάλαιο 1** παρουσιάζεται η εισαγωγή, η οποία περιλαμβάνει τη θεματική της εργασίας, τους στόχους και τη σημασία της HRV ως δείκτη ψυχικής και σωματικής υγείας.

- Στο **Κεφάλαιο 2** αναπτύσσεται το θεωρητικό υπόβαθρο, όπου αναλύονται βασικές έννοιες που σχετίζονται με τη μεταβλητότητα του καρδιακού ρυθμού, τις τεχνολογίες που χρησιμοποιούνται για την καταγραφή του PPG σήματος, καθώς και την επεξεργασία σημάτων.

- Στο **Κεφάλαιο 3** παρουσιάζεται η ανάλυση του θέματος, που περιλαμβάνει τη βιβλιογραφική επισκόπηση, τις ερευνητικές ερωτήσεις και τις υποθέσεις της έρευνας.

- Στο **Κεφάλαιο 4** περιγράφεται η σχεδίαση του συστήματος. Αναλύονται τα εργαλεία, οι τεχνολογίες, οι αλγόριθμοι και η διαδικασία ανάπτυξης της εφαρμογής για έξυπνα ρολόγια, καθώς και η μεθοδολογία για τη συλλογή και ανάλυση δεδομένων.

- Στο **Κεφάλαιο 5** παρουσιάζεται η υλοποίηση της εφαρμογής, όπου περιγράφονται τα τεχνικά χαρακτηριστικά και η ανάπτυξη της σε περιβάλλον Android, με λεπτομέρειες σχετικά με τον κώδικα και τη διασύνδεση με τη συσκευή.

- Στο **Κεφάλαιο 6** αναλύονται τα αποτελέσματα της έρευνας. Εξετάζονται οι υπολογισμοί των δεικτών της HRV, η ακρίβειά τους και η συσχέτισή τους με τους δείκτες άγχους.

- Τέλος, το **Κεφάλαιο 7** περιλαμβάνει τα συμπεράσματα και τις προτάσεις για μελλοντική έρευνα και βελτιώσεις.

Κάθε κεφάλαιο συμβάλλει στην πληρέστερη κατανόηση του τρόπου με τον οποίο η HRV μπορεί να χρησιμοποιηθεί ως εργαλείο για την παρακολούθηση της ψυχικής και σωματικής υγείας μέσω έξυπνων φορητών συσκευών.

Κεφάλαιο 2: Θεωρητικό υπόβαθρο

Σε αυτό το κεφάλαιο παρουσιάζεται μια περιεκτική ανάλυση των βασικών θεωρητικών εννοιών, μοντέλων και πλαισίων που απαιτούνται για την κατανόηση και την εμβάθυνση του θέματος. Μέσω μιας λεπτομερούς ανασκόπησης της σχετικής βιβλιογραφίας και της ανάλυσης των υπαρχουσών θεωρητικών προσεγγίσεων, καθορίζει το πλαίσιο που θα καθοδηγήσει την επιστημονική προσέγγιση της συγκεκριμένης εργασίας και την απόδοση νοήματος στα ευρήματα της έρευνας.

2.1 Εισαγωγικές έννοιες

Η παρούσα ενότητα αποσκοπεί στην αποτελεσματική κατανόηση του θέματος, διευκρινίζοντας τις βασικές έννοιες που διέπουν την έρευνά.

2.1.1 Συμπαθητικό και παρασυμπαθητικό σύστημα

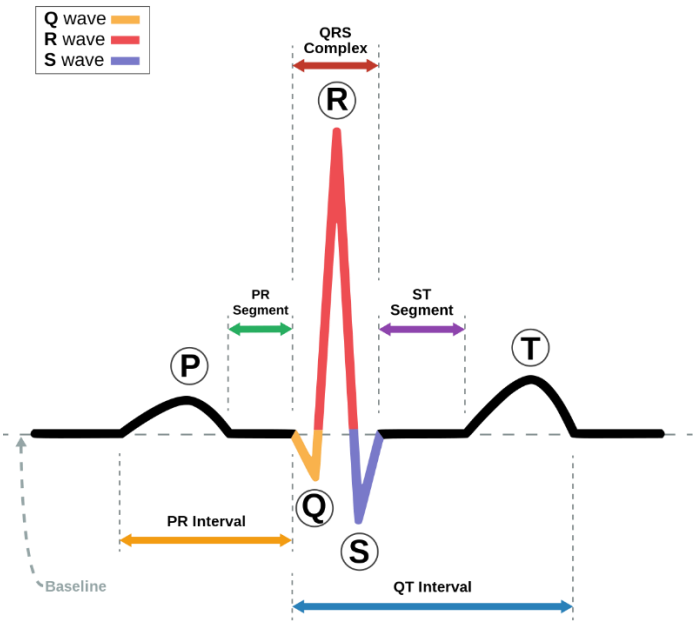
Το ανθρώπινο νευρικό σύστημα αποτελεί ένα διατεταγμένο δίκτυο νεύρων και κυττάρων, τα οποία συνεργάζονται για τη ρύθμιση όλων των ζωτικών λειτουργιών που εκτελούνται στο σώμα μας. Το συμπαθητικό νευρικό σύστημα (ΣΝΣ) και το παρασυμπαθητικό νευρικό σύστημα (ΠΣΝΣ) αποτελούν δύο σημαντικά συστατικά του αυτόνομου νευρικού συστήματος (ΑΝΣ) και συνδράμουν στη ρύθμιση των αυτόματων και αντανακλαστικών λειτουργιών του ανθρώπινου οργανισμού.

Το παρασυμπαθητικό νευρικό σύστημα (ΠΝΣ) ρυθμίζει τη λειτουργία των οργάνων και των αδένων κατά τη διάρκεια της ανάπαυσης, γνωστής και ως δραστηριότητας «ανάπαυσης και πέψης» ή «τροφοδοσίας και αναπαραγωγής» [4]. Ουσιαστικά, το ΠΝΣ διατηρεί τη σωστή λειτουργία των σωματικών διεργασιών, διατηρεί τον καρδιακό παλμό και την αρτηριακή πίεση σταθερά, ενώ ενθαρρύνει δραστηριότητες που σχετίζονται με την πέψη και τη σεξουαλική λειτουργία.

Το συμπαθητικό νευρικό σύστημα (ΣΝΣ) θεωρείται, ενδεχομένως, ακόμα πιο σημαντικό από το παρασυμπαθητικό νευρικό σύστημα, διότι ελέγχει την αντίδρασή "πολέμησε ή τρέξε" [4]. Στην περίπτωση μίας επικίνδυνης κατάστασης, είναι το ΣΝΣ που προετοιμάζει έναν οργανισμό με στόχο την επιβίωση, είτε πολεμώντας τον κίνδυνο, είτε τρέχοντας μακριά από αυτόν. Σε έναν δυνητικό κίνδυνο, το ΣΝΣ κατευθύνει την ενέργεια ενός ατόμου μακριά από μη-ουσιώδεις λειτουργίες (όπως το πεπτικό σύστημα) προς λειτουργίες που είναι ουσιώδεις για την επιβίωση.

2.1.2 Σύμπλεγμα QRS

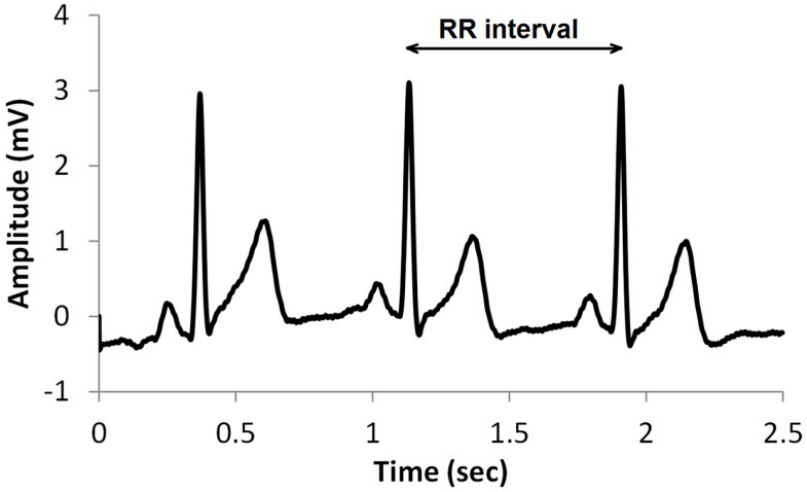
Το σύμπλεγμα QRS είναι ένας ιατρικός όρος που αναφέρεται στην ηλεκτρική δραστηριότητα της καρδιάς και παρατηρείται σε ένα τυπικό ηλεκτροκαρδιογράφημα. Αποτελείται από τρεις κύριες κυματομορφές, τα κύματα Q, R και S, και αντιπροσωπεύει τη σύσφιξη των κοιλιών της καρδιάς και την ανάγκη παροχής αίματος στο σώμα [5]. Το έπαρμα Q αντιστοιχεί στην έναρξη της διέγερσης των κοιλιών, το έπαρμα R στην πλήρη διέγερση των κοιλιών (εκπόλωση) και το έπαρμα S στο τέλος της διέγερσης.



ΕΙΚΟΝΑ 1: ΜΟΡΦΗ ΤΟΥ ΣΥΜΠΛΟΚΟΥ QRS
ΠΗΓΗ: ΠΡΟΣΑΡΜΟΓΗ ΑΠΟ [6]

2.1.3 Χρονικά διαστήματα RR και NN

Το διάστημα RR αναφέρεται στον χρόνο που μεσολαβεί μεταξύ δύο διαδοχικών κυμάτων R του συμπλέγματος QRS στο ηλεκτροκαρδιογράφημα (ECG) [7]. Είναι ένας σημαντικός στην αξιολόγηση της καρδιακής λειτουργίας και αντιπροσωπεύει την περίοδο μεταξύ δύο συνεχόμενων καρδιακών παλμών.



ΕΙΚΟΝΑ 2 - ΧΡΟΝΙΚΟ ΔΙΑΣΤΗΜΑ RR
ΠΗΓΗ: ΠΡΟΣΑΡΜΟΓΗ ΑΠΟ [8]

Αντίστοιχα, τα NN διαστήματα (Normal to Normal Intervals) ταυτίζονται με τα RR, αν από αυτά αφαιρεθούν οι εσφαλμένοι καρδιακοί παλμοί που έχουν καταγραφεί (θόρυβος) [9].

2.1.4 Φίλτρο και φιλτράρισμα ενός σήματος

Το φιλτράρισμα αποτελεί μία κρίσιμη διαδικασία στην επεξεργασία σημάτων, απαραίτητη για την απομόνωση ή εξάλειψη ανεπιθύμητων στοιχείων από ένα σήμα. Αυτό περιλαμβάνει την αφαίρεση θορύβου, την αποκατάσταση της αρχικής μορφής του σήματος και την ενίσχυση συγκεκριμένων χαρακτηριστικών. Η διαδικασία του φιλτραρίσματος μπορεί να εφαρμοστεί τόσο σε αναλογικά όσο και σε ψηφιακά σήματα, με στόχο τη βελτίωση της ποιότητας και της αξιοπιστίας των πληροφοριών που μεταφέρονται μέσω του σήματος [10].

Στα αναλογικά σήματα, τα φίλτρα συνήθως υλοποιούνται μέσω ηλεκτρονικών κυκλωμάτων, όπως αντιστάσεις, πυκνωτές και πηνία, τα οποία δημιουργούν διατάξεις που επιτρέπουν τη διέλευση συγκεκριμένων συχνοτήτων και την αποκοπή άλλων. Στα ψηφιακά σήματα, η υλοποίηση γίνεται μέσω αλγορίθμων που επεξεργάζονται τις αριθμητικές τιμές των δειγμάτων του σήματος. Αυτά τα ψηφιακά φίλτρα προσφέρουν μεγαλύτερη ευελιξία και ακρίβεια σε σύγκριση με τα αναλογικά, καθώς μπορούν να προσαρμοστούν σε ποικίλες απαιτήσεις και συνθήκες.

Οι βασικοί τύποι φίλτρων που χρησιμοποιούνται στην επεξεργασία σημάτων περιλαμβάνουν τα χαμηλοπερατά (low-pass), υψηλοπερατά (high-pass), ζωνοπερατά (band-pass) και ζωνοαποκοπής (band-stop) φίλτρα. Τα χαμηλοπερατά φίλτρα επιτρέπουν τη διέλευση των χαμηλών συχνοτήτων και αποκόπτουν τις υψηλές, ενώ τα υψηλοπερατά επιτρέπουν τη διέλευση των υψηλών συχνοτήτων και αποκόπτουν τις χαμηλές. Τα ζωνοπερατά φίλτρα επιτρέπουν τη διέλευση μιας συγκεκριμένης ζώνης συχνοτήτων και αποκόπτουν τις υπόλοιπες, ενώ τα ζωνοαποκοπής φίλτρα αποκόπτουν μια συγκεκριμένη ζώνη συχνοτήτων και επιτρέπουν τη διέλευση των υπόλοιπων.

Η εφαρμογή των φίλτρων στην επεξεργασία σημάτων μπορεί να βελτιώσει σημαντικά την ανάλυση και την ερμηνεία των δεδομένων, ειδικά σε περιοχές όπως η ιατρική απεικόνιση, η επικοινωνία και τα συστήματα ελέγχου.

2.1.5 Φίλτρο Butterworth

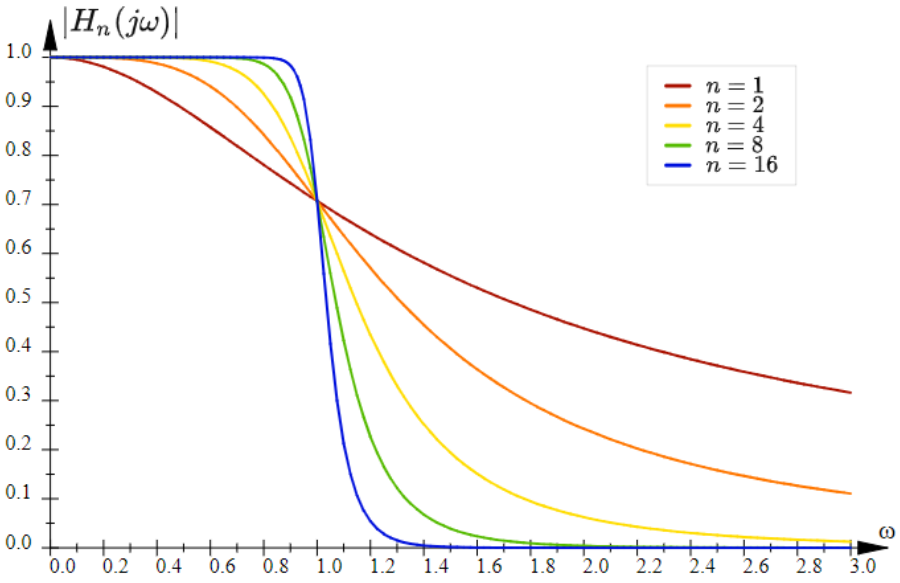
Το φίλτρο Butterworth αποτελεί μία από τις πιο διαδεδομένες κατηγορίες φίλτρων στην επεξεργασία σημάτων, γνωστό για την ομαλή απόκριση του στο πεδίο της συχνότητας. Εισήχθη από τον Βρετανό μηχανικό Stephen Butterworth το 1930 και χαρακτηρίζεται από τη μέγιστη επίπεδη απόκριση στο πεδίο της συχνότητας, χωρίς κυματισμούς (ripple) στην περιοχή διέλευσης ή αποκοπής.

Ο κύριος στόχος ενός φίλτρου Butterworth είναι να διατηρήσει μια ομαλή και επίπεδη απόκριση στην περιοχή διέλευσης (passband) και να εξασφαλίσει μια σταδιακή μείωση (roll-off) της ισχύος του σήματος στην περιοχή αποκοπής (stopband). Αυτή η χαρακτηριστική απόκριση, το καθιστά ιδιαίτερα κατάλληλο για εφαρμογές όπου η πιστότητα του σήματος είναι κρίσιμη, όπως σε σήματα φωτοπληθυσμογραφίας.

Η συνάρτηση μεταφοράς (transfer function) ενός φίλτρου Butterworth ορίζεται με τέτοιο τρόπο ώστε να εξασφαλίζει ότι η απόκριση του φίλτρου στο μέτρο της συχνότητας είναι όσο το δυνατόν πιο επίπεδη. Το μέτρο της συνάρτησης μεταφοράς δίνεται από τη σχέση:

$$|H(j\omega)| = \frac{1}{\sqrt{1 + \left(\frac{\omega}{\omega c}\right)^{2n}}}$$

όπου ω είναι η γωνιακή συχνότητα, ωc είναι η συχνότητα αποκοπής (cutoff frequency) και n είναι η τάξη του φίλτρου [11]. Όσο μεγαλύτερη είναι η τάξη του φίλτρου (δηλαδή, όσο μεγαλύτερη είναι η τιμή του n), τόσο πιο απότομη είναι η μετάβαση από την περιοχή διέλευσης στην περιοχή αποκοπής.



ΕΙΚΟΝΑ 3 - ΦΙΛΤΡΟ BUTTERWORTH ΓΙΑ ΔΙΑΦΟΡΕΤΙΚΕΣ ΤΑΞΕΙΣ
ΠΗΓΗ: ΠΡΟΣΑΡΜΟΓΗ ΑΠΟ [11]

2.1.6 Κανονικοποίηση τιμών

Η κανονικοποίηση ενός σήματος είναι μια διαδικασία στην επεξεργασία σημάτων που αποσκοπεί στη μετατροπή των τιμών του σήματος σε μια καθορισμένη κλίμακα, συνήθως μεταξύ -1 και 1 ή 0 και 1. Η κανονικοποίηση χρησιμοποιείται για να εξασφαλίσει ότι το σήμα έχει συνεπείς τιμές και να διευκολύνει τη σύγκριση μεταξύ διαφορετικών σημάτων ή την ανάλυση του ίδιου σήματος σε διαφορετικές χρονικές περιόδους. Υπάρχουν διάφορα ήδη κανονικοποίησης, όπως Ελάχιστου-Μέγιστου (Min-Max Normalization), Κανονικοποίηση Z-Score (Z-Score Normalization), Κανονικοποίηση με τον Μέγιστο (Max Normalization) κ.α.

Η κανονικοποίηση Ελάχιστου-Μέγιστου (η οποία χρησιμοποιείται στην συγκεκριμένη εργασία) πραγματοποιείται μέσω της εξής διαδικασίας [12]:

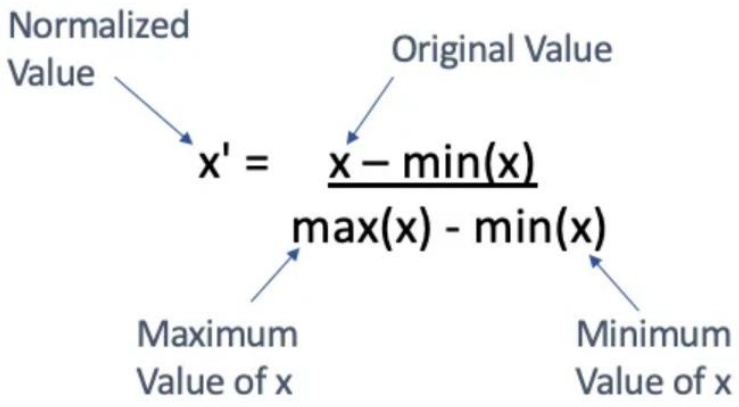
1. **Υπολογισμός των ελάχιστων και μέγιστων τιμών του σήματος:** Αυτές οι τιμές χρησιμοποιούνται για να προσδιοριστεί το εύρος του σήματος.

2. **Αφαίρεση της ελάχιστης τιμής από κάθε τιμή του σήματος:** Αυτό μετατοπίζει όλες τις τιμές του σήματος έτσι ώστε η ελάχιστη τιμή να είναι 0.

3. **Διαίρεση της διαφοράς με το εύρος του σήματος (μέγιστη τιμή - ελάχιστη τιμή):** Αυτό κλιμακώνει όλες τις τιμές ώστε να βρίσκονται μεταξύ 0 και 1.

4. **Προσαρμογή του αποτελέσματος ώστε να βρίσκεται στην επιθυμητή κλίμακα (π.χ. -1 έως 1):** Αυτό επιτυγχάνεται με την κατάλληλη γραμμική μετατροπή των τιμών.

Η κανονικοποίηση του σήματος βοηθά στην απομάκρυνση των διακυμάνσεων που οφείλονται σε διαφορετικές κλίμακες μέτρησης ή συνθήκες καταγραφής, επιτρέποντας μια πιο συνεπή και αξιόπιστη ανάλυση του σήματος.



ΕΙΚΟΝΑ 4 - ΤΥΠΟΣ ΚΑΝΟΝΙΚΟΠΟΙΗΣΗΣ ΕΛΑΧΙΣΤΟΥ – ΜΕΓΙΣΤΟΥ
ΠΗΓΗ: ΠΡΟΣΑΡΜΟΓΗ ΑΠΟ [12]

2.1.7 Διακριτός Μετασχηματισμός Fourier

Ο Διακριτός Μετασχηματισμός Fourier (DFT) [13] είναι ένας από τους πιο σημαντικούς μαθηματικούς μετασχηματισμούς στην επεξεργασία σήματος, που χρησιμοποιείται για τη μετατροπή ενός διακριτού σήματος από το πεδίο του χρόνου στο πεδίο της συχνότητας. Ουσιαστικά, παίρνει ένα σήμα στο πεδίο του χρόνου και το αναλύει στα συστατικά του ημίτονα και συνημίτονα διαφορετικών συχνοτήτων. Ο DFT μπορεί να εφαρμοστεί σε σήματα που αποτελούνται από N διακριτά δείγματα.

Η μαθηματική εξίσωση του DFT είναι:

$$X(k) = \sum_{n=0}^{N-1} x(n) \cdot e^{\frac{-2\pi k n j}{N}}, \qquad k = 0,1,2,\dots,N-1$$

$X(k)$: *η συνεισφορά της συχνότητας* k.

$x(n)$: *το αρχικό σήμα στο πεδίο του χρόνου.*

N: *ο αριθμός των δειγμάτων του σήματος.*

$e^{\frac{-2\pi knj}{N}}$: *το φάσμα των συχνοτήτων με τη μορφή ημιτονοειδών συναρτήσεων (εμφανίζεται σε πολική μορφή ως πραγματικό και φανταστικό μέρος).*

Ο DFT παράγει ένα σύνολο από σύνθετες τιμές που αντιστοιχούν σε συχνότητες από $k = 0$ έως $k = N - 1$. Κάθε $X(k)$ περιέχει πληροφορίες τόσο για το πλάτος όσο και για τη φάση της αντίστοιχης συχνότητας στο αρχικό σήμα. Ο δείκτης k, αναπαριστά κανονικοποιημένες συχνότητες, οι οποίες είναι κλάσματα της συχνότητας δειγματοληψίας f_s.

Οι πραγματικές συχνότητες που αντιστοιχούν στις κανονικοποιημένες συχνότητες υπολογίζονται με τον παρακάτω τύπο:

$$f_k = \frac{k}{N} f_s$$

f_k: *η συνεισφορά της συχνότητας k.*
N: *ο αριθμός των δειγμάτων του σήματος.*

k: *κανονικοποιημένη συχνότητα.*
f_s: *συχνότητα δειγματοληψίας.*

2.1.8 Γρήγορος Μετασχηματισμός Fourier

Ο Γρήγορος Μετασχηματισμός Fourier (FFT) [13] είναι ένας αλγόριθμος που χρησιμοποιείται για τον υπολογισμό του Διακριτού Μετασχηματισμού Fourier (DFT) και του αντιστρόφου αυτού, με έναν πιο αποδοτικό τρόπο από τον απευθείας υπολογισμό του.

Η βασική ιδέα του FFT είναι ότι αξιοποιεί την συμμετρία και την περιοδικότητα του DFT για να μειώσει τον υπολογιστικό φόρτο από $O(N^2)$ σε $O(N \cdot logN)$, όπου N είναι ο αριθμός των δειγμάτων του σήματος. Έτσι, είναι ιδιαίτερα χρήσιμος για την ανάλυση μεγάλων συνόλων δεδομένων και χρησιμοποιείται ευρέως σε εφαρμογές όπως η ψηφιακή επεξεργασία σήματος, η φασματική ανάλυση, η συμπίεση ήχου και εικόνας, και η επεξεργασία ιατρικών δεδομένων.

Συνοπτικά, η μεθοδολογία του αλγορίθμου είναι η εξής:

1. **Διαίρεση του Σήματος:**
   Το αρχικό σήμα χωρίζεται σε δύο υποσύνολα, αυτά με άρτια και περιττά δείγματα. Για παράδειγμα, αν έχουμε το σήμα $x(n) = [x0, x1, x2, x3]$, τα δείγματα διαχωρίζονται ως εξής:
   - Άρτια δείγματα: $[x0, x2]$
   - Περιττά δείγματα: $[x1, x3]$

2. **Υπολογισμός των Υποσημάτων:**
   Υπολογίζουμε τον DFT για τα υποσύνολα. Για κάθε υποσύνολο των $N/2$ δειγμάτων, εφαρμόζουμε τη διαδικασία του DFT. Έτσι, για τα άρτια δείγματα και τα περιττά δείγματα υπολογίζουμε:

$$X_{\text{even}}(k) = \sum_{n=0}^{N/2-1} x_{\text{even}}(n) \cdot e^{-j\frac{2\pi kn}{N/2}}$$

$$X_{\text{odd}}(k) = \sum_{n=0}^{N/2-1} x_{\text{odd}}(n) \cdot e^{-j\frac{2\pi kn}{N/2}}$$

X_{even}: *DFT των άρτιων δειγμάτων.*
X_{odd}: *DFT των περιττών δειγμάτων.*

3. **Συνδυασμός των Αποτελεσμάτων:**
   Συνδυάζουμε τα αποτελέσματα των άρτιων και περιττών δειγμάτων χρησιμοποιώντας τις εξής σχέσεις:

$$X(k) = X_{\text{even}}(k) + e^{-j\frac{2\pi k}{N}} \cdot X_{\text{odd}}(k)$$

$$X(k + N/2) = X_{\text{even}}(k) - e^{-j\frac{2\pi k}{N}} \cdot X_{\text{odd}}(k)$$

για $k = 0,1,\ldots,N/2 - 1$

2.1.9 Φάσμα Ισχύος

Το φάσμα ισχύος [13] είναι ένα θεμελιώδες εργαλείο στην ανάλυση σήματος, το οποίο παρέχει πληροφορίες για την κατανομή της ισχύος ενός σήματος ως προς τις διάφορες συχνότητες του. Στην επεξεργασία σήματος, χρησιμοποιείται για την κατανόηση της συνεισφοράς κάθε συχνότητας στο συνολικό σήμα.

Για τον υπολογισμό του φάσματος ισχύος χρησιμοποιούμε το τετράγωνο του μέτρου (magnitude) του αποτελέσματος του Διακριτού Μετασχηματισμού Fourier - X(k). Ο τύπος για τον υπολογισμό του μέτρου είναι ο εξής:

$$|X(k)| = \sqrt{\text{Re}\{X(k)\}^2 + \text{Im}\{X(k)\}^2}$$

,όπου Re{X(k)} είναι το πραγματικό μέρος και $\text{Im}\{X(k)\}$ είναι το φανταστικό μέρος του $X(k)$.

Επομένως, το φάσμα ισχύος $P(k)$ υπολογίζεται ως εξής:

$$P(k) = |X(k)|^2 = \text{R}\{X(k)\}^2 + \text{I}\{X(k)\}^2$$

Αυτή η διαδικασία παρέχει την κατανομή της ισχύος του σήματος σε διαφορετικές συχνότητες και μπορεί να χρησιμοποιηθεί για την ανάλυση και την ερμηνεία των χαρακτηριστικών του σήματος.

2.2 Εισαγωγή στην Μεταβλητότητα του Καρδιακού Ρυθμού (HRV)

Η παρούσα ενότητα παρουσιάζει μια συνοπτική επισκόπηση της σημασίας της HRV στο πλαίσιο της υγείας και της επιστήμης. Επιπλέον, διευκρινίζεται η σημασία της στην αξιολόγηση της υγείας του αυτόνομου νευρικού συστήματος και της καρδιαγγειακής λειτουργίας.

2.2.1 Ορισμός της HRV και σημασία της στην ανάλυση της λειτουργίας του ανθρώπινου νευρικού συστήματος

Το συμπαθητικό και παρασυμπαθητικό νευρικό σύστημα ασκούν επίδραση στη λειτουργία της καρδιάς και στον καρδιακό ρυθμό (HR). Η ισορροπία μεταξύ των δύο αυτών συστημάτων διαδραματίζει καθοριστικό ρόλο στη σταθερότητα του καρδιακού ρυθμού, η οποία μπορεί να μετρηθεί από τη μεταβλητότητα του καρδιακού ρυθμού. Πιο συγκεκριμένα, αποτελεί τη μεταβολή των χρονικών τμημάτων μεταξύ διαδοχικών καρδιακών παλμών [10] και αντανακλά τη λειτουργία του αυτόνομου νευρικού συστήματος, το οποίο επηρεάζει την καρδιαγγειακή λειτουργία.

Οι αλληλεπιδράσεις του αυτόνομου νευρικού συστήματος (ΑΝΣ) με τα διαστήματα R-R της καρδιάς αποτελούν ένα πολύπλοκο φαινόμενο που προέρχεται από τη νεύρωση της καρδιάς και των αγγείων [11]. Η λειτουργία της καρδιάς και του κυκλοφορικού συστήματος επηρεάζεται κυρίως από τη δραστηριότητα του ΑΝΣ, το οποίο αποτελείται από τα συμπαθητικά και παρασυμπαθητικά νεύρα. Ο μυελός είναι ο κύριος ρυθμιστής της αυτόματης αντίδρασης προς την καρδιά και τα αιμοφόρα αγγεία. Ειδικότερα, ο πυρήνας μονήρους δεσμίδας (nucleus tractus solitarius) στον μυελό λαμβάνει ερεθίσματα από αισθητήρια νεύρα και προκαλεί καρδιαγγειακές αντιδράσεις σε καταστάσεις όπως συναισθήματα και στρες, συμπεριλαμβανομένου και του σωματικού στρες, όπως η άσκηση. Από τον μυελό, το παρασυμπαθητικό νεύρο μεταφέρει ερεθίσματα προς την καρδιά, ενώ οι συμπαθητικές νευρικές ίνες ασκούν επιρροή στην καρδιακή λειτουργία. Η διέγερση από το συμπαθητικό νευρικό σύστημα οδηγεί σε αύξηση του καρδιακού ρυθμού, της εγρήγορσης και του άγχους, ενώ η παρασυμπαθητική διέγερση έχει ακριβώς το αντίθετο αποτέλεσμα.

2.2.2 Κλινικοί δείκτες μεταβλητότητας καρδιακού ρυθμού

Στο πλαίσιο αυτό παρουσιάζονται οι βασικοί δείκτες της HRV που προκύπτουν από την ανάλυση της μεταβλητότητας του καρδιακού ρυθμού και χρησιμοποιούνται ευρέως στην ιατρική έρευνα και την κλινική πράξη. Οι δείκτες αυτοί περιλαμβάνουν τα εξής:

- **SDNN (Standard Deviation of NN intervals):**

  Συνολική διακύμανση στην διάρκεια των NN διαστημάτων (διαδοχικά καρδιακά κύματα) κατά τη διάρκεια μιας συγκεκριμένης χρονικής περιόδου, προσφέροντας πληροφορίες για τη συνολική δραστηριότητα του αυτόνομου νευρικού συστήματος [12].

$$SDNN = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}\left(RR_i - Avg(RR)\right)^2}$$

RR_i: Το i-οστό διάστημα RR (ο χρόνος μεταξύ δύο διαδοχικών καρδιακών παλμών).

Avg (RR): Ο μέσος όρος όλων των RR διαστημάτων.

N: Ο συνολικός αριθμός των RR διαστημάτων που έχουν μετρηθεί.

- **RMSSD (Root Mean Square of Successive Differences):** M

  Μέσος όρος των τετραγώνων των διαφορών μεταξύ συνεχόμενων NN διαστημάτων. Αποτελεί μια ευαίσθητη μέτρηση για τις αλλαγές στον καρδιακό ρυθμό και συνήθως χρησιμοποιείται ως ένδειξη παρασυμπαθητικής δραστηριότητας [12].

$$RMSSD = \sqrt{Avg(RR_{i+1} - RR_i)^2}$$

RR_i: Το i-οστό διάστημα RR.

RR_{i+1}: Το επόμενο διάστημα RR μετά από το i-οστό.

N: Ο συνολικός αριθμός των RR διαστημάτων που έχουν μετρηθεί.

- **pNN50 (Percentage of successive NN intervals differing by more than 50 ms):** Ποσοστό των συνεχόμενων NN διαστημάτων που διαφέρουν περισσότερο από 50 ms [12].

$$pNN50(\%) = \frac{count|RR_{i+1} - RR_i| > 50ms}{N - 1}$$

RR_i: Το i-οστό διάστημα RR.

RR_{i+1}: Το επόμενο διάστημα RR μετά από το i-οστό.

N: Ο συνολικός αριθμός των RR διαστημάτων που έχουν μετρηθεί.

- **AMo (Amplitude Mode):**

  Αντιπροσωπεύει την ένταση του πιο συχνού RR διαστήματος στην κατανομή των διαστημάτων RR, υπολογιζόμενη ως ποσοστό. Πρακτικά, αντικατοπτρίζει το ποσοστό των RR διαστημάτων που έχουν ίδια τιμή με το πιο κοινό RR διάστημα. Ο συγκεκριμένος δείκτης χρησιμοποιείται συχνά ως ένδειξη της λειτουργικής

κατάστασης του αυτόνομου νευρικού συστήματος, με τις αυξήσεις του να συνδέονται με καταστάσεις αυξημένου στρες ή έντονης συμπαθητικής δραστηριότητας [16].

$$AMo = \frac{N_{Mo}}{N} \cdot 100\% \text{ [16]}$$

N_{Mo}: ο αριθμός των RR διαστημάτων που αντιστοιχούν στη συχνότερα εμφανιζόμενη τιμή (Mode) του RR διαστήματος.

N: ο συνολικός αριθμός των RR διαστημάτων που έχουν μετρηθεί.

- **MedSD (Median Standard Deviation):**

  Υπολογίζεται ως η τυπική απόκλιση των RR διαστημάτων γύρω από τη μέση τιμή τους και παρέχει μια εκτίμηση της σταθερότητας και της ομαλότητας του καρδιακού ρυθμού. Σε καταστάσεις στρες, οι τιμές του MedSD τείνουν να μειώνονται, υποδεικνύοντας αυξημένη καρδιακή σταθερότητα, αλλά και περιορισμένη ικανότητα του οργανισμού να προσαρμόζεται σε μεταβαλλόμενες συνθήκες [16].

$$MedSD = Median(|RR_{i+1} - RR_i|) \text{ [16]}$$

RR_i: Το i-οστό διάστημα RR.

RR_{i+1}: Το επόμενο διάστημα RR μετά από το i-οστό.

Αυτοί οι δείκτες παρέχουν πληροφορίες σχετικά με την κατάσταση του αυτόνομου νευρικού συστήματος και της καρδιαγγειακής λειτουργίας, και χρησιμοποιούνται ευρέως στην κλινική διάγνωση, την πρόβλεψη κινδύνου και την παρακολούθηση της ανταπόκρισης ενός οργανισμού σε θεραπευτικές παρεμβάσεις.

Η χρονική διάρκεια της μέτρησης είναι είτε 60 είτε 120 δευτερόλεπτα (ανάλογα με την επιλογή του χρήστη), το οποίο θεωρείται εξαιρετικά μικρή χρονική διάρκεια μέτρησης (Ultra Short-Term Measurement-UST). Με βάση έρευνες [14] για την ακρίβεια των UST μετρήσεων, έχει αποδειχτεί ότι οι παραπάνω δείκτες είναι έγκυροι και ακριβείς για αυτό το μήκος χρονικής διάρκειας, οπότε και γι αυτόν τον λόγο επιλέχτηκαν.

Πίνακας 1: Ελάχιστη Χρονική Περίοδος Απαιτούμενη για την Εκτίμηση των Μετρήσεων HRV

Πηγή: Προσαρμογή από [15]

Ελάχιστη UST διάρκεια μέτρησης	Δείκτης HRV
60s	pNN50, NN50, RMSSD, SDNN
120s	HRV Triangular Index

2.2.3 Στατιστικοί Δείκτες για την Επαλήθευση των Αποτελεσμάτων

Για την αξιολόγηση της ακρίβειας των υπολογισμών που σχετίζονται με τη μεταβλητότητα του καρδιακού ρυθμού και την πρόβλεψη του άγχους, χρησιμοποιήθηκαν δύο βασικοί στατιστικοί δείκτες, οι οποίοι αναλύονται παρακάτω.

- **Mean Absolute Error (MAE):**
  Αποτελεί το μέσο απόλυτο σφάλμα μεταξύ των πραγματικών τιμών HRV και των αντίστοιχων προβλεπόμενων τιμών (ή τιμών σύγκρισης σε σχέση με μία αξιόπιστη πηγή). Το MAE προσφέρει μια εύκολη στην κατανόηση μέτρηση της ακρίβειας του μοντέλου, καθώς εκφράζει τη μέση απόκλιση μεταξύ των πραγματικών και των προβλεπόμενων τιμών στη μονάδα μέτρησης των δεδομένων. Ο δείκτης αυτός χρησιμοποιήθηκε για να αξιολογηθεί η συνολική απόδοση του αλγορίθμου υπολογισμού της HRV σε σύγκριση με ήδη εγκεκριμένη εφαρμογή υπολογισμού των ίδιων δεικτών. Επισημαίνουμε εδώ ότι οι μικρότερες τιμές MAE υποδεικνύουν μεγαλύτερη ακρίβεια.

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i| \text{ [19]}$$

N: Ο συνολικός αριθμός των παρατηρήσεων.

y_i: Η πραγματική τιμή της παρατήρησης i.

$\hat{y}_i$: Η προβλεπόμενη τιμή της παρατήρησης i.

- **Root Mean Square Error (RMSE):**
  Δείκτης πιο ευαίσθητος σε μεγάλες αποκλίσεις (outliers), καθώς υπολογίζει το τετραγωνικό μέσο σφάλμα μεταξύ των προβλεπόμενων και των πραγματικών τιμών. Η τιμή του RMSE είναι πάντα μεγαλύτερη ή ίση του MAE και προσφέρει μια βαθύτερη ανάλυση της ακρίβειας του μοντέλου, ειδικά όταν οι αποκλίσεις από τις πραγματικές τιμές είναι σημαντικές. Στην παρούσα εργασία, το RMSE χρησιμοποιήθηκε για την αξιολόγηση της αξιοπιστίας του μοντέλου σε περιπτώσεις όπου μεγάλες αποκλίσεις θα μπορούσαν να επηρεάσουν την ακρίβεια των προβλέψεων.

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2} \text{ [20]}$$

N: Ο συνολικός αριθμός των παρατηρήσεων.

y_i*: Η πραγματική τιμή της παρατήρησης i.*

$\hat{y}_i$*: Η προβλεπόμενη τιμή της παρατήρησης i.*

Οι παραπάνω στατιστικοί δείκτες παρέχουν μια ολοκληρωμένη εικόνα της απόδοσης του μοντέλου και επιτρέπουν τη σύγκριση με άλλες μεθόδους ή έρευνες. Η χρήση τους συμβάλλει στην αξιολόγηση της αξιοπιστίας και της ακρίβειας των αποτελεσμάτων που προκύπτουν από την ανάλυση της HRV και τη συσχέτισή της με την ψυχική υγεία και το άγχος σε σύγκριση με άλλες εγκεκριμένες πηγές.

2.3 Χρήση της Φωτοπληθυσμογραφίας για τον υπολογισμό της HRV

Η Φωτοπληθυσμογραφία (PPG) μετρά την ποσότητα φωτός που απορροφάται ή ανακλάται από τους ανθρώπινους ιστούς. Αποτελεί μία οπτική κυματομορφή, η οποία συσχετίζεται με την αλλαγή του όγκου αίματος σε ένα μικρό δίκτυο αιμοφόρων αγγείων, που βρίσκονται σε κοντινή απόσταση από την επιφάνεια του δέρματος [16].

Τα PPG σήματα αποτελούν έναν εναλλακτικό τρόπο καθορισμού των καρδιακών παλμών, σε αντίθεση με τα παραδοσιακά ECG σήματα, τα οποία παρουσιάζουν προβλήματα καταγραφής και ανάλυσης. Στην ευρεία χρήση των σημάτων PPG συνείσφερε και η δυνατότητα καταγραφής τους από μη επεμβατικές μεθόδους και οικονομικούς αισθητήρες. Τέλος, από αρκετές μελέτες [17] έχουν αποδειχτεί αποτελεσματικά και έγκυρα ως μία εναλλακτική μέθοδος για την ανάλυση της HRV.

Τα σήματα PPG χωρίζονται ανάλογα με το μήκος της ακτινοβολίας που χρησιμοποιείται για την καταγραφή τους, όπως κόκκινη, πράσινη και υπεριώδη ακτινοβολία. Στην συγκεκριμένη εργασία χρησιμοποιείται κόκκινη ακτινοβολία, καθώς ο ανθρώπινος οργανισμός δεν απορροφά σχεδόν καθόλου την κόκκινη ακτινοβολία, πράγμα που επιτρέπει εισχώρηση στους ιστούς έως και 10 φορές μεγαλύτερη από τις υπόλοιπες (εκτός από την υπεριώδη).

2.4 Συσχέτιση της HRV με την ψυχική υγεία

Ο Hans Selye ορίζει το άγχος ως "μια αντίδραση στην αλλαγή προκειμένου να διατηρηθεί η κατάσταση σταθερότητας που ο οργανισμός έχει διατηρήσει ενάντια σε ένα ερέθισμα που διαταράσσει την ψυχική και σωματική κατάστασή του" [18]. Το άγχος αποτελεί μια φυσιολογική αντίδραση του οργανισμού σε διάφορες προκλητικές καταστάσεις και περιστάσεις αβεβαιότητας. Ωστόσο, όταν το άγχος γίνει υπερβολικό ή διατηρείται για μεγάλα χρονικά διαστήματα, μπορεί να επηρεάσει αρνητικά την ψυχολογική και σωματική υγεία. Στη σημερινή κοινωνία, όπου οι απαιτήσεις είναι συχνά υψηλές, το άγχος μπορεί να είναι μια συχνή εμπειρία για πολλούς ανθρώπους.

Η συνεχής έκθεση σε υψηλά επίπεδα άγχους μπορεί να οδηγήσει σε διάφορα ψυχολογικά προβλήματα, όπως κατάθλιψη [19], αλλά και αρνητικές επιπτώσεις στη σωματική υγεία, συμπεριλαμβανομένων καρδιαγγειακών προβλημάτων [20] και ανοσολογικών αντιδράσεων. Σύμφωνα με τον Παγκόσμιο Οργανισμό Υγείας [21], ένας από τους πιο σημαντικούς παράγοντες που επηρεάζουν την υγεία παγκοσμίως, είναι οι ψυχικές διαταραχές και ιδιαίτερα οι διαταραχές άγχους και κατάθλιψης. Η πανδημία COVID-19 αποτέλεσε μια κρίσιμη περίοδο που επέφερε ανησυχητική αύξηση στον αριθμό των ατόμων που πάσχουν από αυτές τις διαταραχές. Συγκεκριμένα, παρατηρήθηκε αύξηση 26% στα ποσοστά της αγχώδους διαταραχής και 28% στα ποσοστά κατάθλιψης σε διάρκεια ενός μόνο έτους. Παρόλο που υπάρχουν αποτελεσματικές επιλογές πρόληψης και θεραπείας, η πλειονότητα των ατόμων με ψυχικές διαταραχές δεν έχουν πρόσβαση σε αποτελεσματική φροντίδα. Επιπλέον, πολλοί άνθρωποι αντιμετωπίζουν στιγματισμό, διακρίσεις και παραβιάσεις των ανθρωπίνων δικαιωμάτων ανά τον κόσμο. Η ανάδειξη αυτών των προκλήσεων αναδεικνύει την ανάγκη για περαιτέρω έρευνα και δράση σε παγκόσμιο επίπεδο, προκειμένου να υπάρξει μια πιο αποτελεσματική υποστήριξη σε όσους πάσχουν από ψυχικές διαταραχές. Γι αυτόν τον λόγο, η καταγραφή και η αξιολόγηση των επιπέδων άγχους αποτελούν σημαντικά εργαλεία για την κατανόηση της ψυχικής υγείας του ατόμου και την υποστήριξή της, πράγμα που αποτελεί τον στόχο της συγκεκριμένης εργασίας.

Οι άνθρωποι με υψηλή HRV συχνά επιδεικνύουν ενισχυμένη ικανότητα προσαρμογής σε διάφορες φυσιολογικές και περιβαλλοντικές αλλαγές. Αυτό συνήθως συνδέεται με χαμηλά επίπεδα άγχους και αυξημένη κατάσταση ευημερίας [22]. Αντίστοιχα, η μειωμένη HRV συχνά θεωρείται ένδειξη παρόντων ή μελλοντικών προβλημάτων υγείας, υποδηλώνοντας μειωμένη ανθεκτικότητα και ικανότητα για αντιμετώπιση δύσκολων καταστάσεων. Επιπλέον, χαμηλές τιμές καρδιακής μεταβλητότητας είναι πιο διαδεδομένες μεταξύ ατόμων με υψηλούς ρυθμούς καρδιακών παλμών στην κατάσταση ανάπαυσης. Αυτή η σχέση προέρχεται από το φαινόμενο όπου ο αυξημένος καρδιακός ρυθμός οδηγεί σε μικρότερα διαστήματα μεταξύ των παλμών, περιορίζοντας έτσι την εύρος της μεταβλητότητας.

Κεφάλαιο 3: Ανάλυση Θέματος

Η ενότητα ανάλυσης θέματος παρέχει μια σύντομη εισαγωγή στο πεδίο της έρευνας, αναδεικνύοντας τη σημασία του θέματος και διατυπώνοντας τις κύριες ερωτήσεις της μελέτης και τα επιθυμητά αποτελέσματα.

3.1 Ανασκόπηση της βιβλιογραφίας

Η μεταβλητότητα του καρδιακού ρυθμού (HRV) ορίζεται ως η μεταβολή των χρονικών τμημάτων μεταξύ διαδοχικών καρδιακών παλμών [10]. Όπως αναφέρθηκε και σε προηγούμενη ενότητα, αποτελεί έναν δείκτη υγείας που καθορίζει την ισορροπία μεταξύ συμπαθητικού και παρασυμπαθητικού νευρικού συστήματος. Επίσης, πολλές έρευνες έχουν συνδέσει τον συγκεκριμένο δείκτη με την ψυχική υγεία και τα επίπεδα άγχους [22].

Για τον υπολογισμό της HRV τυπικά χρησιμοποιούνται ηλεκτρόδια που τοποθετούνται σε συγκεκριμένα σημεία του σώματος (στήθος, χέρια και πόδια) και καταγράφουν την ηλεκτρική δραστηριότητα της καρδιάς, δηλαδή γίνεται με την καταγραφή ηλεκτροκαρδιογραφήματος (ECG) σε κάποια κλινική ή σε κάποιον ειδικό γιατρό [23]. Έπειτα γίνεται επεξεργασία του σήματος και εξαγωγή RR διαστημάτων, με σκοπό τον υπολογισμό της HRV. Σε αυτό το σημείο πρέπει να τονιστεί ότι για τον ακριβέστερο καθορισμό μίας μέσης τιμής HRV είναι απαραίτητη η καταγραφή σήματος διάρκειας 24 ωρών. Επιπλέον, για τον υπολογισμό της HRV είναι απαραίτητη η εξαγωγή των διαστημάτων RR, αλλά στην πράξη οποιοδήποτε σήμα μπορεί να παρέχει τα ακριβή χρονικά διαστήματα μεταξύ χρονικών παλμών θα μπορούσε να χρησιμοποιηθεί. Γι αυτόν τον λόγο, ενδείκνυται η χρήση σήματος PPG, καθώς όχι μόνο μπορεί να παρέχει τη μεταβλητότητα μεταξύ παλμών (Pulse Rate Variability – PRV), η οποία ουσιαστικά ταυτίζεται με την HRV, αλλά και οι αισθητήρες για την καταγραφή του υπάρχουν σε πολλά έξυπνα ρολόγια (smartwatches) και αποτελεί μία πιο οικονομική, γρήγορη και εύκολη εναλλακτική καταγραφής της HRV.

Σύμφωνα με το 38ο Διεθνές Συνέδριο της IEEE για την Μηχανική στην κοινότητα της Ιατρικής και της Βιολογίας, η χρήση PPG σήματος για τον υπολογισμό της HRV αποτελεί μία πολύ αξιόπιστη εναλλακτική με σημαντική συσχέτιση με την γνωστή μέθοδο με τη χρήση ECG σήματος σε ποσοστό μεγαλύτερο του 82% για συχνοτικούς και χρονικούς δείκτες της HRV [24]. Επίσης, με τη χρήση PPG αισθητήρων μπορεί να γίνει καταγραφή σήματος μικρής χρονικής διάρκειας, όπως 60 με 120 δευτερολέπτων, δηλαδή σε πραγματικό χρόνο. Τα παραπάνω χρονικά διαστήματα έχουν επιλεχτεί για την παρούσα εφαρμογή, καθώς με βάση την έρευνα των Shaffer κ.α. [15] οι δείκτες της HRV που υπολογίζονται είναι έγκυροι μόνο αν η χρονική διάρκεια του σήματος μέτρησης συμβαδίζει με τον Πίνακα 1 (Πίνακας 1: Ελάχιστη Χρονική Περίοδος Απαιτούμενη για την Εκτίμηση των Μετρήσεων HRV).

Μία από τις πιο απαιτητικές και δύσκολες διαδικασίες για τον υπολογισμό της HRV αποτελεί αυτή του εντοπισμού των κορυφών σε ένα PPG σήμα, με στόχο τον καθορισμό των καρδιακών παλμών και έπειτα της εξαγωγή των RR διαστημάτων, τα οποία οδηγούν και στο τελικό αποτέλεσμα. Γι αυτόν τον λόγο, μία αρκετά καλή λύση παρέχει η μέθοδος των Alqaraawi κ.ά. [3], η οποία αποτελεί βελτίωση του Automatic Multiscale Peak Detection (AMPD) [25] αλγορίθμου των Scholkmann κ.ά. Ο συγκεκριμένος αλγόριθμος είναι ιδανικός ακόμα και για θορυβώδη σήματα, καθώς γίνεται ανάλυση του τοπικού σκαλογραφήματος μεγίστου (Local

Maxima Scalogram). Με την παραπάνω μέθοδο, υπολογίζεται ένα ποσοστό πιθανότητας εμφάνισης μεγίστου (p1) και έπειτα με τη χρήση του μοντέλου Laplace και προηγούμενη γνώση από τα R μέγιστα (τα οποία υπολογίστηκαν με τη χρήση του p1) υπολογίζεται ένα δεύτερο ποσοστό εμφάνισης μεγίστου (p2). Τα δύο ποσοστά που υπολογίζονται παραπάνω συνδυάζονται ($p1 \cdot p2$) και προκύπτει η τελική συνδυασμένη πιθανότητα, που καθορίζει εάν τελικά ένα σημείο αποτελεί R μέγιστο. Για ακόμα μεγαλύτερη βελτίωση προτείνεται ο υπολογισμός ενός ακόμα ποσοστού πιθανότητας μεγίστου (p3), χρησιμοποιώντας το θεώρημα του Bayes, με το οποίο τα νέα PPG σήματα που καταγράφονται συνδυάζονται με μία εκτιμώμενη κατανομή πιθανοτήτων (προτού καταγραφεί κανένα σήμα ακόμα) και ενημερώνουν το ποσοστό πιθανότητας μεγίστου (p3). Έπειτα, συνδυάζονται οι πιθανότητες p1 και p3 για την εύρεση μεγίστων.

Για την βελτίωση της ακρίβειας του παραπάνω αλγορίθμου, γίνεται χρήση φίλτρων τύπου Butterworth με τη χρήση της βιβλιοθήκης IIRj του καθηγητή Bernd Porr [29]. Τα φίλτρα Butterworth είναι γνωστά για την ομαλή απόκριση συχνότητας και την απουσία κυματισμών στη ζώνη διέλευσης και αποκοπής. Η βιβλιοθήκη IIRj επιτρέπει την εύκολη και αποτελεσματική εφαρμογή αυτών των φίλτρων στα σήματα, προσφέροντας ένα ευρύ φάσμα δυνατοτήτων για τη διαχείριση θορύβου και ανεπιθύμητων συχνοτήτων στα δεδομένα.

Η σχέση του άγχους και της HRV έχει μελετηθεί αρκετά και έχει επιβεβαιωθεί από αρκετές δημοσιεύσεις στην βιβλιογραφία [1], [20]. Παρόλα αυτά, οι μελέτες για συγκεκριμένους δείκτες της HRV και ακριβείς τιμές αυτών για τον υπολογισμό του άγχους σε πραγματικό χρόνο είναι ελάχιστες. Πρόσφατη έρευνα από την Welltory ανέπτυξε έναν αλγόριθμο που βασίζεται στην ανάλυση της καρδιακής μεταβλητότητας για την αξιολόγηση των επιπέδων στρες. Αυτός ο αλγόριθμος χρησιμοποιεί δείκτες όπως οι AMo, pNN50, και MedSD, οι οποίοι υπολογίζονται από τις ακολουθίες των RR-διαστημάτων που καταγράφονται μέσω της εφαρμογής. Οι τιμές αυτές συγκρίνονται με τις ατομικές βασικές τιμές κάθε συμμετέχοντα για να υπολογιστεί το επίπεδο φυσιολογικού στρες σε ποσοστό [16].

3.2 Θεωρητικό πλαίσιο

Στην συγκεκριμένη ενότητα ενσωματώνονται έννοιες από τη φυσιολογία, τη ψυχολογία και την επεξεργασία σημάτων για τον καθορισμό της σχέση μεταξύ της μεταβλητότητας του καρδιακού ρυθμού σε πραγματικό χρόνο χρησιμοποιώντας σήματα φωτοπληθυσμογραφίας (PPG) και τη συσχέτισή της με δείκτες ψυχικής υγείας. Αντλώντας στοιχεία από δημοσιεύσεις σχετικά με το θέμα [23] [7] [10] [20] [13], οι οποίες υποστηρίζουν ότι η HRV αντανακλά την αλληλεπίδραση μεταξύ του αυτόνομου νευρικού συστήματος και της ρύθμισης των συναισθημάτων, υποθέτουμε ότι οι αλλαγές στα πρότυπα της HRV μπορεί να λειτουργούν ως δείκτες για διάφορες πτυχές της ψυχολογικής ευημερίας. Γι αυτόν τον λόγο, η ανάπτυξη μίας εφαρμογής σε smartwatch που θα υπολογίζει την HRV και το άγχος σε ένα σύντομο χρονικό διάστημα αποτελεί μία σημαντική βοήθεια για τον γρήγορο και εύκολο έλεγχο της υγείας ενός ατόμου σε καθημερινή βάση.

Ως συνέπεια των παραπάνω, χρησιμοποιούμε σήμα PPG και συγκεκριμένα κόκκινης ακτινοβολίας, για μεγαλύτερη ακρίβεια και μείωση σφαλμάτων. Το PPG σήμα αποτελεί μία καλή εναλλακτική για τον υπολογισμό της HRV σε σχέση με το σήμα ECG (συνήθης μέθοδος),

με συσχέτιση μεταξύ των δύο περίπου στο 82% ιδιαίτερα για δείκτες στο πεδίο του χρόνου [24]. Η διάρκεια της μέτρησης που δίνουμε τη δυνατότητα στον χρήστη να επιλέξει είναι είτε 60s είτε 120s, διότι για να θεωρείται έγκυρη η μέτρηση απαιτείται μία ελάχιστη χρονική διάρκεια καταγραφής σήματος, ανάλογα με τον δείκτη που υπολογίζουμε και πρέπει να συμβαδίζει με τον Πίνακα 1 (Πίνακας 1: Ελάχιστη Χρονική Περίοδος Απαιτούμενη για την Εκτίμηση των Μετρήσεων HRV).

Το φιλτράρισμα είναι μια κρίσιμη διαδικασία στην επεξεργασία σημάτων, ιδιαίτερα στην ανάλυση της καρδιακής μεταβλητότητας. Τα φίλτρα χρησιμοποιούνται για την απομάκρυνση του θορύβου και των ανεπιθύμητων συχνοτήτων από τα σήματα, διατηρώντας μόνο τις σημαντικές πληροφορίες. Υπάρχουν διάφοροι τύποι φίλτρων, όπως τα φίλτρα FIR (Finite Impulse Response) και τα φίλτρα IIR (Infinite Impulse Response). Τα φίλτρα Butterworth, τα οποία είναι μια κατηγορία φίλτρων IIR, είναι ιδιαίτερα δημοφιλή λόγω της ομαλής απόκρισής τους και της έλλειψης κυματισμών στη ζώνη διέλευσης και αποκοπής. Αυτή η ιδιότητα τα καθιστά ιδανικά για εφαρμογές όπου η ακρίβεια είναι ζωτικής σημασίας, όπως στην ανάλυση PPG σημάτων για την εξαγωγή δεικτών της HRV. Γι αυτόν τον λόγο έγινε χρήση προηγμένων βιβλιοθηκών όπως η IIRj του καθηγητή Bernd Porr [29], από όπου η εφαρμογή των φίλτρων Butterworth γίνεται εύκολα και αποτελεσματικά, επιτρέποντας την ακριβή επεξεργασία των σημάτων και την εξαγωγή αξιόπιστων δεδομένων για την περαιτέρω ανάλυση.

Για τον υπολογισμό της HRV χρησιμοποιούμε τον αλγόριθμο των Alqaraawi κ.ά. [3], για τον εντοπισμό των R μεγίστων και τον καθορισμό των RR διαστημάτων, καθώς είναι αποτελεσματικός ακόμα και για θορυβώδη σήματα. Παρόλα αυτά, ο αλγόριθμος εφαρμόζεται μέχρι και τον υπολογισμό της δεύτερης πιθανότητας (p2), διότι η διαδικασία πρέπει να πραγματοποιείται σε πραγματικό χρόνο, άρα δεν αποθηκεύονται δεδομένα τα οποία μπορούν να χρησιμοποιηθούν για την περαιτέρω βελτίωση της ακρίβειας. Αυτή η βελτίωση θα μπορούσε να ήταν αρνητική, καθώς θα οδηγούσε σε μεγάλη αύξηση του χρόνου επεξεργασίας, πράγμα που αντιτίθεται στον στόχο της εφαρμογής να παρέχει μία καλή μέτρηση σε σύντομο χρονικό διάστημα.

Όσον αφορά τον υπολογισμό του άγχους με βάση τους δείκτες HRV, η προσέγγιση που υιοθετούμε βασίζεται στον αλγόριθμο της Welltory [16], ο οποίος χρησιμοποιεί δείκτες όπως οι AMo, pNN50, και MedSD. Αυτοί οι δείκτες εξάγονται από τις ακολουθίες των RR-διαστημάτων που καταγράφονται μέσω της εφαρμογής, και οι τιμές τους συγκρίνονται με προκαθορισμένες βασικές τιμές: pNN50_baseline = 20.0 (με τη χρήση της βάσης δεδομένων της EliteHRV, θεωρούμε ως επίπεδο αναφοράς την μέση τιμή του φυσιολογικού εύρους τιμών [34]), AMo_baseline = 30.0 και MedSD_baseline = 50.0 (καθορίστηκαν από την ανάλυση δεδομένων που συλλέχθηκαν από ένα αντιπροσωπευτικό δείγμα χρηστών της εφαρμογής μας με βάση τις πρωινές μετρήσεις που καθορίζουν το επίπεδο αναφοράς ενός χρήστη). Η τιμή του φυσιολογικού στρες υπολογίζεται ως ποσοστό, λαμβάνοντας υπόψιν την απόκλιση από τις προκαθορισμένες τιμές, επιτρέποντας έτσι την ακριβή και προσαρμοσμένη αξιολόγηση του άγχους σε πραγματικό χρόνο, ανεξάρτητα από τη διάρκεια της μέτρησης που επιλέγει ο χρήστης.

3.3 Ερευνητικές Ερωτήσεις και Υποθέσεις

Τα δύο κύρια ερωτήματα που καθοδηγούν την παρούσα έρευνα είναι πρώτον, πώς μπορεί να υπολογιστεί η μεταβλητότητα του καρδιακού ρυθμού σε πραγματικό χρόνο μέσω σημάτων Φωτοπληθυσμογραφίας κόκκινης ακτινοβολίας και δεύτερον, πώς αυτή η μεταβλητότητα σχετίζεται με την πνευματική υγεία. Υποθέσεις που αντιμετωπίζονται σε αυτήν τη διατριβή περιλαμβάνουν την υπόθεση ότι η μεταβλητότητα του καρδιακού ρυθμού μπορεί να αντικατοπτρίζει την ψυχοσωματική κατάσταση ενός ατόμου και να προσφέρει ενδείξεις για την ψυχική του κατάσταση. Επιπλέον, υποθέτουμε ότι η εφαρμογή ενός βελτιωμένου αλγόριθμου AMPD στην ανάλυση των σημάτων PPG σε συνδυασμό με την εφαρμογή κατάλληλων φίλτρων για την προεπεξεργασία του σήματος, θα μας επιτρέψει να αξιοποιήσουμε αποτελεσματικά την πληροφορία αυτή για την εκτίμηση της πνευματικής υγείας ενός ατόμου. Μέσω αυτής της διερεύνησης, αναζητούμε τον τρόπο για την ανάπτυξη μίας νέας μεθόδου εκτίμησης της πνευματικής υγείας βασιζόμενη σε μη επεμβατικές τεχνολογίες παρακολούθησης του καρδιακού ρυθμού, όπως μια εφαρμογή σε smartwatch που θα μπορούσε να χρησιμοποιηθεί εύκολα στην καθημερινότητα ενός ατόμου και να παρέχει μία σύντομη αξιολόγηση της υγείας του.

Κεφάλαιο 4: Σχεδίαση Θέματος

Σε αυτήν την ενότητα, εξετάζουμε τη λεπτομερή διαδικασία πίσω από τον σχεδιασμό του θέματος της έρευνάς μας, αναδεικνύοντας τον λόγο, τους στόχους και τις μεθοδολογικές σκέψεις που υποστηρίζουν τη διαμόρφωσή του. Με έμφαση στην αντιμετώπιση εντοπισμένων αναγκών και στην προαγωγή του επιστημονικού διαλόγου, αυτή η ενότητα περιγράφει την συστηματική προσέγγιση που ακολουθήθηκε για την επινόηση και την επεξεργασία του ερευνητικού μας θέματος. Στόχος μας είναι να διαμορφώσουμε μια στέρεη βάση για τις προσπάθειες της μελέτης μας και να συμβάλουμε ουσιαστικά στο υπάρχον επιστημονικό πλαίσιο.

4.1 Ανάλυση Αναγκών και Στόχων

Παρατηρώντας την άνοδο του άγχους και των ψυχικών προβλημάτων τα τελευταία έτη, συμπεραίνουμε ότι απαιτείται ένα εργαλείο που θα διευκολύνει την αξιολόγηση της ψυχικής κατάστασης και της συνολικής ευημερίας του ατόμου, παρέχοντας μια γρήγορη και εύκολη μέτρηση. Η ανάγκη για μία τέτοια μέθοδο είναι ουσιαστική, καθώς η ευαισθησία στο θέμα της ψυχικής υγείας αυξάνεται και οι άνθρωποι αναζητούν ολοένα και περισσότερους τρόπους για να παρακολουθούν την κατάστασή τους καθημερινά και σε πραγματικό χρόνο.

Η ανάπτυξη μιας εφαρμογής για smartwatches ανταποκρίνεται στην ανάγκη για ένα πρακτικό και φορητό εργαλείο μέτρησης. Αυτές οι συσκευές, έχουν ενσωματωμένους αισθητήρες που μπορούν να μετρούν διάφορες φυσιολογικές παραμέτρους, καθιστώντας την παρακολούθηση της υγείας ευκολότερη και πιο προσιτή στο καθημερινό περιβάλλον του χρήστη. Η ανάπτυξη μιας τέτοιας εφαρμογής θα επιτρέψει στους χρήστες να έχουν πρόσβαση σε αξιόπιστες πληροφορίες σχετικά με την ψυχική τους κατάσταση, ενθαρρύνοντάς τους να λάβουν μέτρα για τη βελτίωσή της.

4.2 Σχεδιασμός Ερευνητικού Πλάνου

Η συγκεκριμένη ενότητα περιλαμβάνει τον σχεδιασμό μιας ολοκληρωμένης μεθοδολογίας για την αξιολόγηση της σχέσης μεταξύ της μεταβλητότητας του καρδιακού ρυθμού και της ψυχικής υγείας. Αυτό περιλαμβάνει την επιλογή αλγορίθμων ανάλυσης σημάτων Φωτοπληθυσμογραφίας και μεθόδων επεξεργασίας σήματος, τη χρήση συγκεκριμένου εξοπλισμού όπως το Samsung Galaxy Watch 4 και το Android Studio για την ανάπτυξη εφαρμογών, τη δομή και οργάνωση του κώδικα για την ανίχνευση του καρδιακού ρυθμού και τον υπολογισμό των δεικτών HRV, καθώς και τρόπους επαλήθευσης του αποτελέσματος.

4.2.1 Αλγόριθμοι, μεθοδολογίες και έρευνες

Κατά την επιλογή της μεθοδολογίας, λαμβάνεται υπόψιν η ανάγκη για μια ολοκληρωμένη προσέγγιση που θα επιτρέπει την αξιολόγηση της συνάφειας μεταξύ της μεταβλητότητας του καρδιακού ρυθμού και της ψυχικής υγείας. Αρχικά, προτείνεται η εφαρμογή διαφορετικών μεθόδων ανάλυσης σημάτων Φωτοπληθυσμογραφίας και αλγορίθμων επεξεργασίας σήματος. Συγκεκριμένα, εφαρμόζουμε μία διαδικασία προεπεξεργασίας του σήματος με τη χρήση φίλτρων τύπου Butterworth, έκτης τάξης, με υψηλή συχνότητα αποκοπής στα 5 Hz και χαμηλή στα 0.5 Hz. Οι παραπάνω τιμές επιλέχτηκαν λόγω της φύσης ενός φυσιολογικού σήματος, αλλά

και πειραματικά, έπειτα από δοκιμές στο σήμα που λαμβάναμε από τα έξυπνα ρολόγια της Samsung και την ανάλυση του σε Matlab. Η χαμηλή συχνότητα αποκοπής επιλέχθηκε για να αφαιρέσει τις πολύ χαμηλές συχνότητες από το σήμα, οι οποίες μπορεί να προέρχονται από αργές μεταβολές και θόρυβο περιβάλλοντος που δεν σχετίζονται με το καρδιακό σήμα. Αυτές οι χαμηλές συχνότητες συχνά περιλαμβάνουν κινήσεις του σώματος ή αλλαγές στον φωτισμό που επηρεάζουν το σήμα. Από την άλλη πλευρά, η υψηλή συχνότητα αποκοπής επιλέχθηκε για να αφαιρέσει τις υψηλές συχνότητες, οι οποίες μπορεί να περιλαμβάνουν θόρυβο από ηλεκτρομαγνητικές παρεμβολές ή άλλους παράγοντες που δεν σχετίζονται με την καρδιακή δραστηριότητα. Οι υψηλές συχνότητες πάνω από 5 Hz συνήθως δεν περιέχουν πληροφορίες για το καρδιακό σήμα και μπορούν να προκαλέσουν στρέβλωση στην ανάλυση της HRV [31]. Επιπλέον, οι παραπάνω τιμές επιλέχθηκαν πειραματικά μέσω δοκιμών και αξιολόγησης των αποτελεσμάτων σε πραγματικά δεδομένα. Η πειραματική διαδικασία περιλαμβάνει την εφαρμογή των φίλτρων σε διαφορετικά σύνολα δεδομένων και την αξιολόγηση της ακρίβειας της ανίχνευσης κορυφών και της μέτρησης των RR διαστημάτων. Η παραπάνω διαδικασία υλοποιήθηκε αρχικά σε Matlab και έπειτα σε Kotlin με τη χρήση της βιβλιοθήκης IIRj [29] με στόχο την εφαρμογή της μεθοδολογίας αυτής στο σήμα που λαμβάνεται κάθε φορά από τους αισθητήρες του ρολογιού.

Στην συνέχεια, χρησιμοποιούμε τον αλγόριθμο των Alqaraawi κ.ά. [3], για τον εντοπισμό των R μεγίστων και τον καθορισμό των RR διαστημάτων, ο οποίος λειτουργεί με την εκτίμηση της πιθανότητας εμφάνισης κορυφών καρδιακού παλμού και της μεταβλητότητας του καρδιακού ρυθμού. Η παραπάνω μεθοδολογία επιλέχτηκε, καθώς αποτελεί βελτίωση του αλγορίθμου του Scholkmann κ.ά. που ονομάζεται AMPD [25] και είναι αποδοτική ακόμα και για θορυβώδη σήματα. Τα PPG σήματα είναι ιδιαίτερα επιρρεπή στον θόρυβο, οπότε η επιλογή του κατάλληλου αλγόριθμου υπολογισμού μεγίστου αποτελεί πρωταρχική παράμετρο.

Επιπλέον, επιλέχτηκαν συγκεκριμένες χρονικές διάρκειες καταγραφής σήματος, με βάση μελέτες σχετικά με την εγκυρότητα των μετρήσεων (Πίνακας 1: Ελάχιστη Χρονική Περίοδος Απαιτούμενη για την Εκτίμηση των Μετρήσεων HRV), διότι για να θεωρηθεί ένα αποτέλεσμα ως έγκυρο, πρέπει να έχει προηγηθεί μία σύμφωνη με τις προδιαγραφές μέτρηση. Ακόμα, για τον υπολογισμό του άγχους, χρησιμοποιήσαμε τη μεθοδολογία της δημοσίευσης των Tyapochkin κ.α. [16], η οποία βασίζεται σε συγκεκριμένους δείκτες HRV όπως οι AMo, pNN50, και MedSD, προκειμένου να καθορίσουμε διακριτά επίπεδα στρες και να τα αξιολογήσουμε με ακρίβεια.

4.2.2 Εξοπλισμός και εφαρμογές

Στην παρούσα υποενότητα παρουσιάζονται ο εξοπλισμός και οι εφαρμογές που χρησιμοποιήθηκαν κατά την ανάπτυξη της εφαρμογής.

- **Android Studio**

  Το Android Studio είναι ένα ολοκληρωμένο περιβάλλον ανάπτυξης (IDE) που αναπτύχθηκε από την Google για τη δημιουργία εφαρμογών για το λειτουργικό σύστημα Android. Παρέχει ένα ευέλικτο και ισχυρό περιβάλλον προγραμματισμού που επιτρέπει στους προγραμματιστές να δημιουργήσουν υψηλής ποιότητας εφαρμογές για κινητές συσκευές Android. Το Android Studio περιλαμβάνει εργαλεία για σχεδιασμό διεπαφών χρήστη, επεξεργασία κώδικα, δοκιμή, ανάλυση και εκτέλεση εφαρμογών. Η

συγκεκριμένη εφαρμογή χρησιμοποιήθηκε για την ανάπτυξη της εφαρμογής σε smartwatch, δηλαδή για την υλοποίηση των αλγορίθμων, του κώδικα για την επικοινωνία με τους αισθητήρες του ρολογιού, αλλά και τις διεπαφές χρήστη.

- **Samsung Galaxy Watch 5**

  Το Samsung Galaxy Watch 5 είναι ένα έξυπνο ρολόι που διαθέτει ενσωματωμένους αισθητήρες καταγραφής PPG σήματος, σχεδιασμένο για την παρακολούθηση φυσιολογικών παραμέτρων, όπως ο καρδιακός ρυθμός και η αρτηριακή πίεση. Επιπλέον, παρέχει πολλές δυνατότητες για την παρακολούθηση της άσκησης, του ύπνου και πολλών άλλων δραστηριοτήτων που σχετίζονται με την υγεία. Η συνδεσιμότητά του με smartphone επιτρέπει στον χρήστη να λαμβάνει ειδοποιήσεις και να χειρίζεται τις εφαρμογές από τον καρπό του χεριού του.

  Για τη μελέτη της μεταβλητότητας του καρδιακού ρυθμού, χρησιμοποιήθηκαν οι αισθητήρες του ρολογιού για τη συλλογή σημάτων φωτοπληθυσμογραφίας (PPG) κόκκινης ακτινοβολίας. Αξιοσημείωτο είναι ότι όχι μόνο γίνεται η μέτρηση των σημάτων, αλλά η επεξεργασία τους και ο υπολογισμός της HRV πραγματοποιούνται απευθείας στη συσκευή.

- **Samsung Privileged Health SDK**

  Το Samsung Privileged Health SDK είναι ένα σύνολο εργαλείων ανάπτυξης λογισμικού (SDK) που παρέχεται από την Samsung για την ανάπτυξη εφαρμογών υγείας για τα έξυπνα ρολόγια της. Το συγκεκριμένο SDK παρέχει πρόσβαση σε ειδικές λειτουργίες και αισθητήρες του ρολογιού, ώστε οι προγραμματιστές να μπορούν να αναπτύξουν προηγμένες εφαρμογές που εκμεταλλεύονται πλήρως τις δυνατότητες της συσκευής για την παρακολούθηση της υγείας και της ευεξίας του χρήστη. Στην συγκεκριμένη εργασία χρησιμοποιείται για την διαχείριση των αισθητήρων, καθώς χωρίς αυτό δεν είναι δυνατή η χρήση τους για την καταγραφή του σήματος.

4.2.3 Δομή και οργάνωση κώδικα

Για τη δομή του κώδικα έχουμε δημιουργήσει 2 αρχεία, το MainActivity.kt και το HRVActivity.kt, τα οποία περιέχουν τον συνολικό κώδικα της εφαρμογής.

- **MainActivity.kt**

  Σε αυτό το αρχείο βρίσκεται ο κώδικας για την κύρια δραστηριότητα της εφαρμογής. Οι βασικές λειτουργίες είναι οι εξής:

  ***-Ρύθμιση του περιβάλλοντος εφαρμογής*:** Η μέθοδος onCreate είναι υπεύθυνη για τη δημιουργία του περιβάλλοντος εφαρμογής και την αρχικοποίηση των στοιχείων της διεπαφής χρήστη.

  ***-Εκκίνηση της μέτρησης HRV*:** Μέσω της μεθόδου ActivityPreview πραγματοποιείται η εκκίνηση της μέτρησης του καρδιακού ρυθμού μέσω της κλάσης HRVActivity.

- **HRVActivity.kt**

Σε αυτό το αρχείο, υλοποιείται η μεθοδολογία για την ανίχνευση της μεταβλητότητας του καρδιακού ρυθμού και τον υπολογισμό του άγχους. Ο κώδικας περιλαμβάνει τις εξής λειτουργίες:

***-Ρύθμιση Περιβάλλοντος*:** Η μέθοδος onCreate είναι υπεύθυνη για την αρχικοποίηση του περιβάλλοντος εφαρμογής και τη διαχείριση των δικαιωμάτων και των αδειών.

***-Καταγραφή σήματος*:** Γίνεται χρήση των Connection Listeners και Trackers για την επικοινωνία με το ρολόι και την ενεργοποίηση των αισθητήρων για καθορισμένη διάρκεια με σκοπό την καταγραφή του PPG σήματος. Ακόμα, υπάρχουν συναρτήσεις που αποδεσμεύουν τους αισθητήρες, όταν πλέον δεν είναι απαραίτητοι.

***-Υπολογισμός HRV*:** Η μέθοδος calculateHRVMetrics είναι υπεύθυνη για τον υπολογισμό των δεικτών της HRV, όπως οι RMSSD, SDNN, pNN50, AMo, και MedSD. Η διαδικασία περιλαμβάνει τον εντοπισμό μεγίστων στο σήμα, τον υπολογισμό των διαστημάτων RR, καθώς και το φιλτράρισμα των δεδομένων για την εξαγωγή των τελικών αποτελεσμάτων.

***-Διεπαφές χρήστη*:** Η HRVCalculator εμφανίζει το κυρίως μενού κατά την είσοδο στην εφαρμογή, αλλά και τα αποτελέσματα της μέτρησης. Ακόμα, οι μέθοδοι ReceivingDataScreen και ProcessingDataScreen εμφανίζουν τις αντίστοιχες οθόνες κατά τη διάρκεια της μέτρησης και της επεξεργασίας των δεδομένων αντίστοιχα. Στην εφαρμογή υπάρχουν και άλλες οθόνες που διευκολύνουν τη χρήση της και εμφανίζονται στα κατάλληλα σημεία.

***-Καθορισμός επιπέδων άγχους*:** Η μέθοδος calculateStressLevel υλοποιεί τη διαδικασία υπολογισμού των επιπέδων άγχους, βασιζόμενη στους δείκτες της HRV. Η μέθοδος αυτή συγκρίνει τις τιμές των δεικτών AMo, pNN50 και MedSD με προκαθορισμένες βασικές τιμές και υπολογίζει το άγχος ως ποσοστό, λαμβάνοντας υπόψιν τις αποκλίσεις από αυτές τις τιμές. Το συνολικό επίπεδο άγχους καθορίζεται ως ο μέσος όρος των επιμέρους τιμών άγχους που υπολογίζονται για κάθε δείκτη, με την τελική εκτίμηση να αποδίδεται σε μια κλίμακα από 0 έως 100%.

4.2.4 Επιβεβαίωση αποτελεσμάτων

Για την επιβεβαίωση των αποτελεσμάτων, προβλέπεται η διεξαγωγή ερευνητικής μελέτης σε άτομα με διαφορετικά προφίλ χρησιμοποιώντας ένα σύντομο ερωτηματολόγιο βασισμένο στο Perceived Stress Scale (PSS) του Sheldon Cohen [27] [28]. Το PSS είναι ένα ευρέως χρησιμοποιούμενο εργαλείο για τη μέτρηση της αντίληψης του στρες και αξιολογεί το βαθμό με τον οποίο κατατάσσονται οι καταστάσεις στη ζωή ως στρεσογόνες. Τα αποτελέσματα της έρευνας θα συγκρίνονται με τις μετρήσεις του καρδιακού ρυθμού που θα πραγματοποιούνται με τη χρήση της εφαρμογής που έχει υλοποιηθεί σε smartwatch.

Επιπλέον, θα πραγματοποιηθούν συγκριτικές μετρήσεις διάρκειας 60 και 120 δευτερολέπτων, προκειμένου να αξιολογηθεί η συνέπεια των αποτελεσμάτων. Αυτή η σύγκριση θα βοηθήσει να προσδιοριστεί αν η μικρότερη διάρκεια μέτρησης (60 δευτερόλεπτα) είναι εξίσου αξιόπιστη με τη μεγαλύτερη (120 δευτερόλεπτα).

Παράλληλα, θα γίνει σύγκριση των αποτελεσμάτων της εφαρμογής με ήδη εγκεκριμένες εφαρμογές που χρησιμοποιούνται ευρέως για τη μέτρηση δεικτών της HRV. Η διαδικασία αυτή θα επιτρέψει την αξιολόγηση της ακρίβειας της εφαρμογής σε σχέση με άλλες λύσεις της αγοράς, χρησιμοποιώντας δεδομένα από αντίστοιχους δείκτες, όπως SDNN και RMSSD.

Τέλος, θα πραγματοποιηθεί στατιστική ανάλυση των αποτελεσμάτων με τη χρήση δεικτών όπως το Mean Absolute Error (MAE) και το Root Mean Square Error (RMSE) για την εκτίμηση των αποκλίσεων μεταξύ των προβλεπόμενων και των πραγματικών τιμών της HRV, με σκοπό να διασφαλιστεί η αξιοπιστία των μετρήσεων.

Με την εφαρμογή της παραπάνω μεθοδολογίας, τίθεται ο στόχος η επιβεβαίωσης της αποτελεσματικότητας της εφαρμογής ως εργαλείου μέτρησης και παρακολούθησης της ψυχικής και σωματικής ευεξίας σε πραγματικό χρόνο.

Κεφάλαιο 5: Υλοποίηση

5.1 Επισκόπηση Τεχνολογιών και Εργαλείων

Στο πλαίσιο αυτής της ενότητας, θα εξεταστούν οι διάφορες τεχνολογίες και εργαλεία που επιλέχθηκαν για την υλοποίηση του έργου, καθώς και οι λόγοι που οδήγησαν σε αυτήν την επιλογή

5.1.1 Έξυπνα Ρολόγια Samsung Galaxy Watch 5

Η απόφαση να χρησιμοποιηθούν τα έξυπνα ρολόγια Samsung Galaxy Watch 5 προήλθε μετά από προσεκτική εξέταση σημαντικών κριτηρίων που αναδείχθηκαν ως καθοριστικά για την απρόσκοπτη πρόοδο της έρευνας. Πρώτον, τα εν λόγω ρολόγια διαθέτουν υψηλής ποιότητας αισθητήρες και τεχνολογία ανίχνευσης, τα οποία αποτελούν θεμέλιο για τη συλλογή ακριβών και αξιόπιστων δεδομένων. Συγκεκριμένα, διαθέτει αισθητήρες καταγραφής PPG ακατέργαστου σήματος κόκκινης ακτινοβολίας με αρκετά μεγάλη ακρίβεια (οπτικός αισθητήρας καρδιακού παλμού, αισθητήρας φωτός) [29]. Δεύτερον, η συμβατότητά τους με το Samsung Privileged Health SDK παρείχε τη δυνατότητα ανάπτυξης μιας πλήρως λειτουργικής εφαρμογής Android για τη συλλογή και ανάλυση των δεδομένων, προσφέροντας έτσι ένα ισχυρό εργαλείο για την επίτευξη των ερευνητικών στόχων. Τέλος, η ευρεία αποδοχή και η φήμη των Samsung Galaxy Watch 5 στην αγορά τους τα καθιστούν μια επιλογή που εγγυάται την προσήλωση στην επίτευξη σταθερών και αξιόπιστων αποτελεσμάτων.

5.1.2 Χρήση του Samsung Privileged Health SDK

Το Samsung Privileged Health SDK είναι μια πλατφόρμα λογισμικού που παρέχει εξειδικευμένες λειτουργίες για την παρακολούθηση της υγείας σε συσκευές Galaxy Watch 4 ή νεότερες εκδόσεις. Παρέχει τόσο ακατέργαστα δεδομένα από τον Samsung BioActive αισθητήρα όσο και επεξεργασμένα δεδομένα. Το SDK υποστηρίζει μια βελτιωμένη δυνατότητα παρακολούθησης της υγείας και επιτρέπει στις εφαρμογές να παρακολουθούν τα δεδομένα υγείας του χρήστη με ακρίβεια, χρησιμοποιώντας το επιταχυνσιόμετρο, το ακατέργαστο ECG (Ηλεκτροκαρδιογράφημα), το PPG (Φωτοπληθυσμογράφημα) και τους καρδιακούς παλμούς [30].

Η διαδικασία καταγραφής δεδομένων είναι αρκετά απλή και φαίνεται σχηματικά στην Εικόνα 3 (**Error! Reference source not found.**). Αρχικά γίνεται καταγραφή του σήματος μέσω του BioActive Sensor στο smartwatch και μέσω του Samsung Privileged Health SDK, μας δίνεται η δυνατότητα να διαχειριστούμε τα ακατέργαστα δεδομένα από το PPG σήμα κόκκινης ακτινοβολίας στην android εφαρμογή μας.

ΣΧΗΜΑ 1 - ΔΙΑΔΙΚΑΣΙΑ ΣΥΛΛΟΓΗΣ ΔΕΔΟΜΕΝΩΝ ΑΠΟ SMARTWATCH ΜΕΣΩ ΤΟΥ SAMSUNG PRIVILEGED HEALTH SDK
ΠΗΓΗ: ΠΡΟΣΑΡΜΟΓΗ ΑΠΟ [30]

Στον παρακάτω πίνακα περιγράφονται τα χαρακτηριστικά των δεδομένων που συλλέγονται μέσω του αισθητήρα.

Πίνακας 3 : Στοιχεία δεδομένων καταγραφής μέσω SAMSUNG PRIVILEGED HEALTH SDK

Πηγή: Προσαρμογή από [30]

Τύπος tracker σήματος	**Ακατέργαστο / Επεξεργασμένο σήμα**	**Περιγραφή**
PPG_RED	Ακατέργαστο	1 PpgRedSet σημείο δεδομένων (data point), κάθε 10 ms (100 hz).

5.1.3 Χρήση Android Studio Code και Kotlin για την ανάπτυξη εφαρμογής σε smartwatch

Η επιλογή να χρησιμοποιηθεί το περιβάλλον του Android Studio Code σε συνδυασμό με τη γλώσσα προγραμματισμού Kotlin για την ανάπτυξη της εφαρμογής σε smartwatch αποτελεί ένα στρατηγικό βήμα, καθώς απορρέει από τη σημαντική προηγούμενη εμπειρία και την πλήρη υποστήριξη που παρέχουν τα εν λόγω εργαλεία στον χώρο της ανάπτυξης λογισμικού. Το Android Studio Code, ως ένα από τα κορυφαία ενσωματωμένα περιβάλλοντα ανάπτυξης (IDE) για εφαρμογές Android, προσφέρει μια ολοκληρωμένη σουίτα εργαλείων για τη δημιουργία, τη δοκιμή και την εκτέλεση εφαρμογών Android. Το εύρος των λειτουργιών του περιλαμβάνει τη δυνατότητα δημιουργίας γρήγορων προτύπων, την υποστήριξη για πολλαπλές γλώσσες προγραμματισμού και τη δυνατότητα ενσωμάτωσης σετ εργαλείων για τη διαχείριση του κώδικα και τη διαμόρφωση των περιβαλλόντων εργασίας, όπως το Samsung Privileged Health SDK.

Από την άλλη πλευρά, η επιλογή της γλώσσας προγραμματισμού Kotlin συμβάλλει στην αύξηση της αποδοτικότητας της ανάπτυξης λογισμικού και τη βελτίωση της ποιότητας του κώδικα. Η Kotlin προσφέρει μια σύγχρονη, ευέλικτη και εκφραστική σύνταξη, ενώ ταυτόχρονα επιτρέπει την επαναχρησιμοποίηση του κώδικα και την εύκολη διαχείριση των εξαρτήσεων. Η υιοθέτηση της Kotlin επίσης επιτρέπει την επέκταση της εφαρμογής σε διάφορες πλατφόρμες και συσκευές, ενισχύοντας την συμβατότητα και την ευελιξία του λογισμικού, πράγμα που την καθιστά ιδανική επιλογή για την ανάπτυξη Android εφαρμογών σε smartwatch.

5.2 Ανάλυση Κώδικα Υλοποίησης της Εφαρμογής

Στην συγκεκριμένη ενότητα, γίνεται αναφορά στη δομή του κώδικα, τις βασικές λειτουργίες και τις αλληλεπιδράσεις μεταξύ των διάφορων συστατικών της εφαρμογής. Επίσης, θα εξεταστούν οι αρχές και οι πρακτικές προγραμματισμού που ακολουθήθηκαν κατά την ανάπτυξη του κώδικα, καθώς και οι αποφάσεις που λήφθηκαν κατά τη διάρκεια της διαδικασίας υλοποίησης.

5.2.1 Εκκίνηση και Βασικά Στοιχεία της Εφαρμογής

Για το αρχείο MainActivity.kt, γίνεται η δήλωση των πακέτων και η εισαγωγή των απαραίτητων βιβλιοθηκών.

```
import android.annotation.SuppressLint
import android.content.Intent
import android.os.Bundle
import androidx.activity.compose.setContent
import androidx.compose.foundation.layout.Arrangement
import androidx.compose.foundation.layout.Column
import androidx.compose.foundation.layout.fillMaxSize
import androidx.compose.runtime.Composable
import androidx.compose.runtime.LaunchedEffect
import androidx.compose.runtime.getValue
import androidx.compose.runtime.mutableStateOf
import androidx.compose.ui.Alignment
import androidx.compose.ui.Modifier
import androidx.compose.ui.tooling.preview.Preview
import androidx.fragment.app.FragmentActivity
import com.matsapali.hrvcalculator.presentation.theme.HRVCalculatorTheme
import kotlinx.coroutines.delay
```

Εδώ ορίζεται η κύρια δραστηριότητα της εφαρμογής και καλείται η μέθοδος setContent για τον καθορισμό του περιεχομένου της. Η συνάρτηση onCreate αποτελεί το σημείο εκκίνησης της εφαρμογής.

```
class MainActivity: FragmentActivity() {

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContent {
            HRVCalculatorTheme {
                ActivityPreview()
            }
        }
    }
```

Η συνάρτηση ActivityPreview αποτελεί μια σύνθετη συνάρτηση του Jetpack Compose που χρησιμοποιείται για τη δημιουργία του γραφικού περιβάλλοντος της εφαρμογής (UI), το οποίο εμφανίζεται σε προεπισκόπηση στο Android Studio. Ο σκοπός της συνάρτησης είναι να ορίσει τα οπτικά στοιχεία που θα εμφανιστούν κατά την εκτέλεση της εφαρμογής, επιτρέποντας μια οπτική αξιολόγηση μέσω της προεπισκόπησης (@Preview). Στο εσωτερικό της, χρησιμοποιούνται στοιχεία (Composables), όπως η διάταξη Column, για τη στοίχιση των περιεχομένων, ενώ γίνεται χρήση της συνάρτησης LaunchedEffect για την εκτέλεση μιας καθυστέρησης ενός δευτερολέπτου (delay), πριν ενεργοποιηθεί μια νέα δραστηριότητα (HRVActivity). Η νέα δραστηριότητα εκκινείται με τη χρήση ενός Intent, που θέτει την εφαρμογή σε νέο task, σηματοδοτώντας τη μετάβαση στο κύριο μέρος της λειτουργικότητας της εφαρμογής.

```
@SuppressLint("UnrememberedMutableState")
@Composable
@Preview
fun ActivityPreview() {
    HRVCalculatorTheme {
        Column(
            modifier = Modifier
                .fillMaxSize(),
            verticalArrangement = Arrangement.Center,
            horizontalAlignment = Alignment.CenterHorizontally
        ) {
            val currentTime by mutableStateOf(3)
            LaunchedEffect(key1 = currentTime) {
                delay(1000L)
                val intent = Intent(this@MainActivity, HRVActivity::class.java)
                intent.flags = Intent.FLAG_ACTIVITY_NEW_TASK
                this@MainActivity.startActivity(intent)
            }
        }
    }
}
```

Αντίστοιχα, για το αρχείο HRVActivity.kt γίνεται δήλωση των απαραίτητων πακέτων και βιβλιοθηκών.

```
import android.app.Activity
import android.content.Context
import android.os.Bundle
import android.os.Handler
import android.os.Looper
import android.util.Log
import android.view.WindowManager
import android.widget.Toast
import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import androidx.compose.foundation.background
import androidx.compose.foundation.layout.Arrangement
import androidx.compose.foundation.layout.Box
import androidx.compose.foundation.layout.Column
import androidx.compose.foundation.layout.Row
import androidx.compose.foundation.layout.fillMaxSize
```

```
import androidx.compose.foundation.layout.fillMaxWidth
import androidx.compose.foundation.layout.height
import androidx.compose.foundation.layout.padding
import androidx.compose.foundation.layout.width
import androidx.compose.foundation.rememberScrollState
import androidx.compose.foundation.shape.CircleShape
import androidx.compose.foundation.shape.RoundedCornerShape
import androidx.compose.foundation.verticalScroll
import androidx.compose.material3.DropdownMenu
import androidx.compose.material3.DropdownMenuItem
import androidx.compose.material3.TextField
import androidx.compose.material3.TextFieldDefaults
import androidx.compose.runtime.Composable
import androidx.compose.runtime.LaunchedEffect
import androidx.compose.runtime.getValue
import androidx.compose.runtime.mutableDoubleStateOf
import androidx.compose.runtime.mutableIntStateOf
import androidx.compose.runtime.mutableStateOf
import androidx.compose.runtime.remember
import androidx.compose.runtime.setValue
import androidx.compose.ui.Alignment
import androidx.compose.ui.Modifier
import androidx.compose.ui.draw.alpha
import androidx.compose.ui.draw.shadow
import androidx.compose.ui.graphics.Color
import androidx.compose.ui.text.TextStyle
import androidx.compose.ui.text.style.TextAlign
import androidx.compose.ui.tooling.preview.Preview
import androidx.compose.ui.unit.DpOffset
import androidx.compose.ui.unit.dp
import androidx.compose.ui.unit.sp
import androidx.wear.compose.foundation.lazy.ScalingLazyColumn
import androidx.wear.compose.material.Button
import androidx.wear.compose.material.ButtonDefaults
import androidx.wear.compose.material.CircularProgressIndicator
import androidx.wear.compose.material.Text
import com.paramsen.noise.Noise
import com.samsung.android.service.health.tracking.ConnectionListener
import com.samsung.android.service.health.tracking.HealthTracker
import
com.samsung.android.service.health.tracking.HealthTracker.TrackerEventListener
import com.samsung.android.service.health.tracking.HealthTrackerException
import com.samsung.android.service.health.tracking.HealthTrackingService
import com.samsung.android.service.health.tracking.data.DataPoint
import com.samsung.android.service.health.tracking.data.HealthTrackerType
import com.samsung.android.service.health.tracking.data.ValueKey
import uk.me.berndporr.iirj.Butterworth
import kotlin.math.abs
import kotlin.math.exp
import kotlin.math.round
import kotlin.math.sqrt
```

Για την κλάση HRVActivity, αρχικά δηλώνουμε τα εξής πεδία:

- **Σταθερές**: Άδειες, δικαιώματα (permissions), συχνότητα δειγματοληψίας (fs) και την σταθερά TAG για την εμφάνιση μηνυμάτων στην κονσόλα.

```
//Constants
private val permissions = arrayOf("android.permission.BODY_SENSORS")
private val REQUEST_ACCOUNT_PERMISSION = 100
private val TAG = HRVActivity::class.java.simpleName
private val fs = 100 //Sampling Frequency
private val handler = Handler(Looper.myLooper()!!)
```

- **Στοιχεία Μέτρησης:** Μεταβλητές για την αποθήκευση δεδομένων, όπως η διάρκεια της μέτρησης και τα δεδομένα που σχετίζονται με τη μέτρηση.

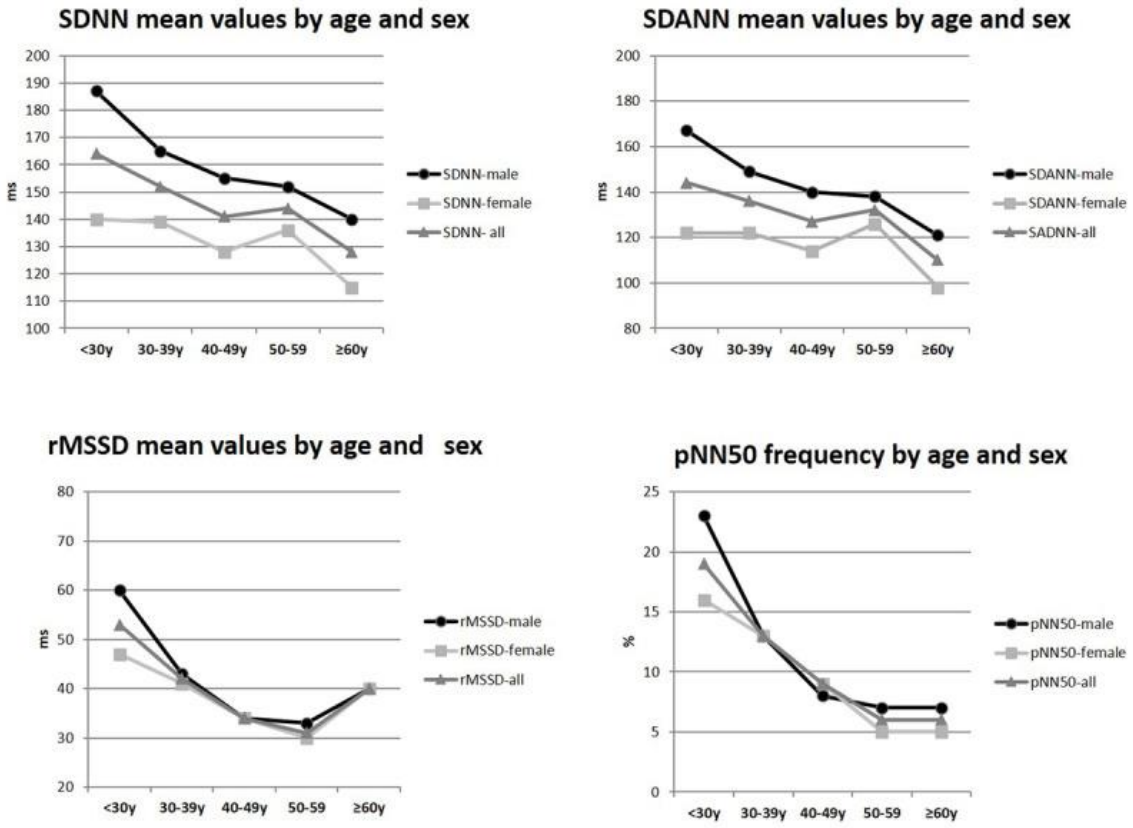
```
//Measurement details
private var collectionDurationMillis by mutableIntStateOf(60000)
private var startTimeMillis = 0L
private var time = mutableListOf<Long>()
private var ppgSignal = mutableListOf<Int>()
private var isHandlerRunning = false
```

- **Υπηρεσίες (Services) και Καταγραφείς (Trackers):** Υπηρεσίες παρακολούθησης υγείας (HealthTrackingService) και καταγραφείς υγείας (HealthTracker) για την καταγραφή των βιομετρικών δεδομένων.

```
//Services and Trackers
private var healthTrackingService: HealthTrackingService? = null
private var ppgRedTracker: HealthTracker? = null
```

- **Δομές για αποθήκευση δεδομένων:** Μεταβλητές για την αποθήκευση των τιμών για τους δείκτες της HRV, καθώς και δεδομένα που αφορούν τις κανονικές τιμές HRV για διάφορες ηλικιακές ομάδες.

  Οι πίνακες με τις αντίστοιχες φυσιολογικές τιμές προήλθαν από τα διαγράμματα της δημοσίευσης των Geovanini κ.ά. [31], όπως φαίνονται στην παρακάτω εικόνα.



ΣΧΗΜΑ 2- ΔΙΑΓΡΑΜΜΑΤΑ ΦΥΣΙΟΛΟΓΙΚΩΝ ΤΙΜΩΝ HRV ΔΕΙΚΤΩΝ ΑΝΑΛΟΓΑ ΜΕ ΤΟ ΦΥΛΟ ΚΑΙ ΤΗΝ ΗΛΙΚΙΑΚΗ ΟΜΑΔΑ
ΠΗΓΗ:ΠΡΟΣΑΡΜΟΓΗ ΑΠΟ [31]

```
//HRV ratios
private var RMSSD by mutableStateOf(0.0)
private var SDNN by mutableStateOf(0.0)
private var pNN50 by mutableStateOf(0.0)
private var hrvIndex by mutableStateOf(0f)
private var AMo by mutableDoubleStateOf(0.0)
private var MedSD by mutableDoubleStateOf(0.0)

//User personal info
private var age by mutableIntStateOf(23)
private var gender by mutableIntStateOf(1) //Male: 0, Female: 1
private var ageGroup by mutableIntStateOf(0)

//Data for normal ranges of HRV ratios
private val ageGroups = intArrayOf(30, 39, 49, 59, 60)
private val meanSDNN =
    arrayOf(intArrayOf(189, 164, 155, 152, 140), intArrayOf(140, 139,
129, 147, 115))
private val meanrMSSD = arrayOf(intArrayOf(60, 42, 33, 32, 40),
intArrayOf(48, 41, 33, 30, 40))
private val meanpNN50 = arrayOf(intArrayOf(23, 13, 8, 7, 7),
intArrayOf(16, 13, 9, 5, 5))
private var ageGroupIsFound by mutableStateOf(false)
```

- **Στοιχεία Διεπαφής Χρήστη (UI):** Μεταβλητές που χρησιμοποιούνται για την αλλαγή της κατάστασης του UI, όπως η ενεργοποίηση της κατάστασης υπολογισμού της HRV και η πρόοδος συλλογής δεδομένων.

```
//UI details
private var isHRVCalculated by mutableStateOf(false)
private var isProcessingData by mutableStateOf(false)
private var errorOccurred by mutableStateOf(false)
private var dataCollectionProgress by mutableIntStateOf(0)
private var exitWelcomePage by mutableStateOf(false)
private var isCollectingData by mutableStateOf(false)
private var stressLevel by
androidx.compose.runtime.mutableFloatStateOf(0f)
private var barColor by mutableStateOf(Color.DarkGray)
private var stressValue by mutableStateOf("Stress Level")
```

5.2.2 Διεπαφή Χρήστη

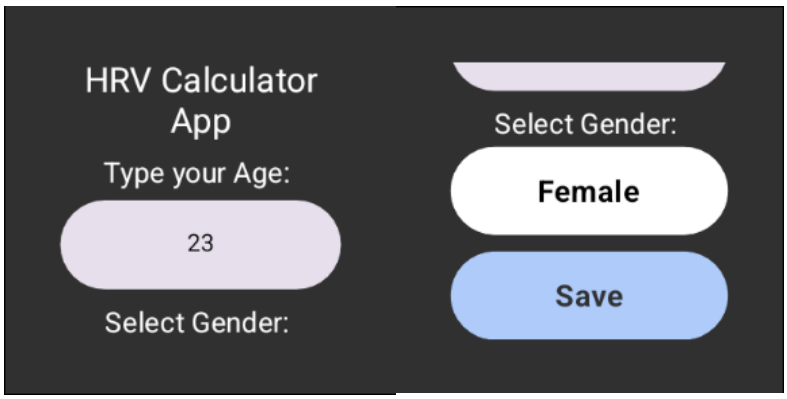
Για τη δημιουργία της διεπαφής χρήστη χρησιμοποιήθηκε η βιβλιοθήκη Compose Jetpack, η οποία παρέχει δυνατότητες δημιουργίας δυναμικών και ευέλικτων διεπαφών. Τα κύρια στοιχεία που υλοποιήθηκαν περιλαμβάνουν κουμπιά, κείμενο και κάποια γραφικά, τα οποία σχεδιάστηκαν για να προσφέρουν μια οικεία προς τον χρήστη και εύχρηστη διεπαφή. Σχεδιάστηκε με γνώμονα την ευκολία χρήσης και την επιθυμία για αισθητική απλότητα, ενώ έγινε προσεκτική χρήση των γραφικών στοιχείων για μια ευχάριστη εμπειρία χρήστη. Παρακάτω αναλύουμε τα στοιχεία που δημιουργούν το οπτικό αποτέλεσμα της εφαρμογής.

WelcomeScreen

Η συνάρτηση WelcomeScreen αποτελεί την αρχική διεπαφή της εφαρμογής και προσφέρει τη δυνατότητα εισαγωγής των βασικών στοιχείων του χρήστη που απαιτούνται για τον υπολογισμό της HRV. Πιο συγκεκριμένα, επιτρέπει στον χρήστη να εισάγει την ηλικία του και να επιλέξει το φύλο του, ώστε τα δεδομένα αυτά να χρησιμοποιηθούν στον υπολογισμό και την ανάλυση των επιπέδων HRV.

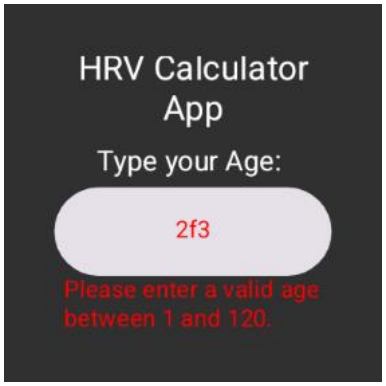
Η εισαγωγή της ηλικίας πραγματοποιείται μέσω της συνάρτησης TextField, η οποία δίνει τη δυνατότητα στον χρήστη να εισαγάγει την ηλικία του. Η τιμή που εισάγεται ελέγχεται για την εγκυρότητά της, δηλαδή πρέπει να βρίσκεται εντός του εύρους 1 έως 120 ετών. Αν η τιμή δεν είναι αποδεκτή, εμφανίζεται μήνυμα σφάλματος που προτρέπει την εισαγωγή μιας έγκυρης τιμής. Με αυτόν τον τρόπο, παρέχεται άμεση ανατροφοδότηση στον χρήστη, βοηθώντας τον να διορθώσει τυχόν σφάλματα στην εισαγωγή των δεδομένων του.

Ακόμα, προσφέρεται η δυνατότητα επιλογής φύλου μέσω ενός αναπτυσσόμενου μενού (DropdownMenu), μεταξύ δύο επιλογών: "Male" και "Female". Τέλος, δίνεται η δυνατότητα αποθήκευσης των παραπάνω δεδομένων με τη χρήση του κουμπιού "Save". Η εφαρμογή, μετά την χρήση της επιλογής αποθήκευσης, ελέγχει την εγκυρότητα των δεδομένων και μόνο τότε προχωρά στην επόμενη φάση, που περιλαμβάνει τον υπολογισμό της HRV, διασφαλίζοντας έτσι την ορθότητα των εισαγόμενων στοιχείων πριν την έναρξη της διαδικασίας μέτρησης.



ΕΙΚΟΝΑ 5 – ΚΑΤΑ ΤΗΝ ΕΝΑΡΞΗ ΤΗΣ ΕΦΑΡΜΟΓΗΣ/ΕΙΣΑΓΩΓΗ ΣΤΟΙΧΕΙΩΝ ΤΟΥ ΧΡΗΣΤΗ

Παρατηρούμε ότι η διεπαφή αλλάζει δυναμικά ανάλογα με την εγκυρότητα των εισαγόμενων δεδομένων, επιτρέποντας ή όχι τη συνέχιση της διαδικασίας.



ΕΙΚΟΝΑ 6 - ΕΜΦΑΝΙΣΗ ΜΗΝΥΜΑΤΟΣ ΣΦΑΛΜΑΤΟΣ ΚΑΤΑ ΤΗΝ ΠΛΗΚΤΡΟΛΟΓΗΣΗ ΜΗ ΑΠΟΔΕΚΤΩΝ ΤΙΜΩΝ ΑΠΟ ΤΟΝ ΧΡΗΣΤΗ

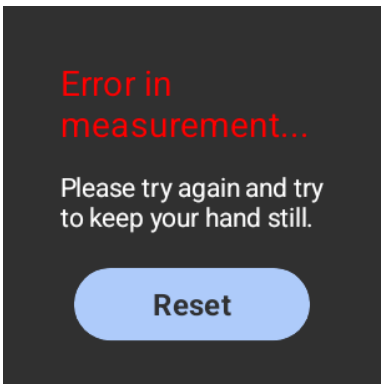
```
@Preview(showBackground = true, device = "id:wearos_small_round")
@Composable
fun WelcomeScreen() {
    var expanded by remember { mutableStateOf(false) }
    val genders = listOf("Male", "Female")
    val selectedGender = if (gender == 0) "Male" else "Female"
    val state = rememberScrollState()

    Column(
        modifier = Modifier
            .fillMaxSize()
            .padding(27.dp)
            .verticalScroll(state),
        horizontalAlignment = Alignment.CenterHorizontally,
        verticalArrangement = Arrangement.Center
    ) {

        Text(
            text = "HRV Calculator App",
            style = TextStyle(fontSize = 17.sp, textAlign =
TextAlign.Center),
            modifier = Modifier.padding(bottom = 8.dp)
        )
```

```
        Text(text = "Type your Age: ", color = Color.White)
        var ageInput by remember { mutableStateOf(if (age == 0) "" else
age.toString()) }
        var isAgeValid by remember { mutableStateOf(true) }

        TextField(
            value = ageInput,
            onValueChange = { newValue ->
                ageInput = newValue
                val newAge = newValue.toIntOrNull()
                if (newAge == null || newAge !in 1..120) {
                    isAgeValid = false
                } else {
                    age = newAge
                    isAgeValid = true
                }
            },
            shape = RoundedCornerShape(25.dp),
            textStyle = TextStyle(
                fontSize = 12.sp, textAlign = TextAlign.Center
            ),
            modifier = Modifier
                .padding(top = 5.dp)
                .height(44.dp)
                .fillMaxWidth(),
            colors = TextFieldDefaults.colors(
                focusedTextColor = Color.Black,
                unfocusedTextColor = Color.Black,
                focusedIndicatorColor = Color.Transparent,
                unfocusedIndicatorColor = Color.Transparent,
                errorIndicatorColor = Color.Transparent,
                errorTextColor = Color.Red
            ),
            isError = !isAgeValid
        )

        if (!isAgeValid) {
            Row(
                modifier = Modifier
                    .fillMaxWidth()
                    .padding(start = 5.dp),
                horizontalArrangement = Arrangement.Center
            ) {
                Text(
                    text = "Please enter a valid age between 1 and 120.",
                    color = Color.Red,
                    fontSize = 12.sp
                )
            }
        }

        Text(
            text = "Select Gender: ",
            color = Color.White,
            modifier = Modifier.padding(top = 8.dp)
        )

        Box(contentAlignment = Alignment.TopCenter) {
            Button(
                onClick = { expanded = true },
```

```
                shape = RoundedCornerShape(25.dp),
                colors = ButtonDefaults.buttonColors(
                    backgroundColor = Color.White, contentColor =
Color.Black
                ),
                modifier = Modifier
                    .padding(bottom = 8.dp, top = 3.dp)
                    .fillMaxWidth()
                    .height(44.dp)
            ) {
                Text(text = selectedGender) // Display selected gender
            }

            DropdownMenu(
                expanded = expanded,
                onDismissRequest = { expanded = false },
                offset = DpOffset(x = 22.dp, y = 10.dp),
                modifier = Modifier.background(color = Color.DarkGray)
            ) {
                genders.forEachIndexed { index, genderOption ->
                    DropdownMenuItem(text = { Text(text = genderOption) },
onClick = {
                        gender = index // Update gender based on selected
index
                        expanded = false
                    })
                }
            }

        }

        Button(
            onClick = {
                if (!ageGroupIsFound) findAgeGroup()
                exitWelcomePage = true
            }, enabled = isAgeValid, modifier = Modifier
                .height(44.dp)
                .fillMaxWidth()
        ) {
            Text(text = "Save")
        }
    }
}
```

ErrorScreen

Αποτελεί μία διεπαφή της εφαρμογής και εμφανίζεται όταν προκύψει σφάλμα κατά τη μέτρηση της HRV. Σχεδιάστηκε για να παρέχει άμεση ενημέρωση για το πρόβλημα που παρουσιάστηκε, καθώς και οδηγίες για την επανάληψη της διαδικασίας.
Εμφανίζει ένα μήνυμα σφάλματος, προειδοποιώντας με την ένδειξη "Error in measurement..." σε κόκκινο χρώμα, καθώς και οδηγίες που προτρέπουν τον χρήστη να επαναλάβει τη μέτρηση, διατηρώντας το χέρι του σταθερό. Επιπλέον, παρέχεται ένα κουμπί "Reset" που επανεκκινεί τη διαδικασία μέτρησης, καλώντας τη συνάρτηση reset. Με αυτόν τον τρόπο, διασφαλίζουμε ότι ο χρήστης ενημερώνεται άμεσα για το σφάλμα και μπορεί να επανεκκινήσει εύκολα τη μέτρηση.



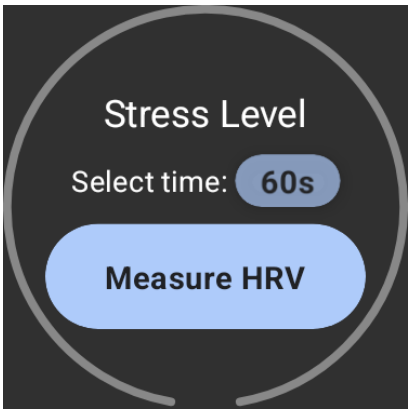
ΕΙΚΟΝΑ 7 - ΟΘΟΝΗ ΕΜΦΑΝΙΣΗΣ ΣΦΑΛΜΑΤΟΣ ΚΑΤΑ ΤΗ ΜΕΤΡΗΣΗ

```
@Composable
@Preview(showBackground = true, device = "id:wearos_small_round")
fun ErrorScreen() {
    Column(
        modifier = Modifier
            .fillMaxSize()
            .padding(22.dp),
        horizontalAlignment = Alignment.CenterHorizontally,
        verticalArrangement = Arrangement.Center
    ) {
        Text(
            text = "Error in measurement...",
            color = Color.Red,
            fontSize = 18.sp,
            modifier = Modifier.padding(7.dp)
        )
        Text(
            text = "Please try again and try to keep your hand still.",
            fontSize = 13.sp,
            modifier = Modifier.padding(7.dp)
        )
        Button(
            onClick = { reset() }, modifier = Modifier
                .padding(top = 11.dp)
                .width(120.dp)
        ) {
            Text(text = "Reset")
        }
    }
}
```

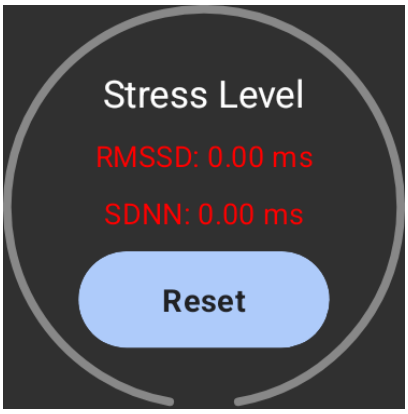
BeforeMeasurement / AfterMeasurement / HRVCalculator

Οι συναρτήσεις BeforeMeasurement και AfterMeasurement παρουσιάζουν τα δεδομένα πριν και μετά τη μέτρηση της HRV αντίστοιχα και αποτελούν ουσιαστικά μία προεπισκόπηση της HRVCalculator για τις διαφορετικές τιμές που μπορεί να πάρουν οι παράμετροί της. Η HRVCalculator διαχειρίζεται τη ροή της εφαρμογής ανάλογα με την κατάσταση της μέτρησης, παρέχοντας πληροφορίες για τα αποτελέσματα και την κατάσταση της διαδικασίας μέτρησης. Σε περίπτωση σφάλματος, καλείται η ErrorScreen. Σε διαφορετική περίπτωση, εμφανίζεται η WelcomeScreen αν ο χρήστης δεν έχει συμπληρώσει ήδη τα προσωπικά του στοιχεία ή η TextWithButton αν έχει

ολοκληρωθεί η αρχική εισαγωγή. Αυτή η δυναμική προσέγγιση εξασφαλίζει ότι η εφαρμογή ανταποκρίνεται κατάλληλα σε κάθε στάδιο της διαδικασίας.



ΕΙΚΟΝΑ 8 - ΑΡΧΙΚΟ ΜΕΝΟΥ ΕΦΑΡΜΟΓΗΣ / ΠΡΙΝ ΤΗΝ ΜΕΤΡΗΣΗ



ΕΙΚΟΝΑ 9 - ΌΨΗ ΕΦΑΡΜΟΓΗΣ ΜΕΤΑ ΤΗΝ ΜΕΤΡΗΣΗ

Παρατηρούμε ότι ανάλογα με την τιμή της μεταβλητής isHRVCalculated, αλλάζουν οι όψεις της HRVCalculator σε πριν και μετά την μέτρηση.

```
@Preview(showBackground = true, device = "id:wearos_small_round")
@Composable
fun BeforeMeasurement() {
    TextWithButton(RMSSD, SDNN, pNN50, false)
}

@Preview(showBackground = true, device = "id:wearos_small_round")
@Composable
fun AfterMeasurement() {
    TextWithButton(RMSSD, SDNN, pNN50, true)

}

@Composable
fun HRVCalculator() {
    LaunchedEffect(isHRVCalculated) {
        setContent {
            if (errorOccurred) {
```

```
                isProcessingData = false
                ErrorScreen()
            } else {
                if (!exitWelcomePage) WelcomeScreen()
                else if (exitWelcomePage) TextWithButton(RMSSD, SDNN,
pNN50, isHRVCalculated)
            }
        }
    }
}
```

TimeOfMeasurementSelection

Αυτή η συνάρτηση δημιουργεί ένα στοιχείο το οποίο με κάθε πάτημα εναλλάσσει τιμές μεταξύ 60 και 120s και με αυτόν τον τρόπο επιτρέπει στο χρήστη να επιλέξει τη διάρκεια της μέτρησης.

ΕΙΚΟΝΑ 10 - ΚΟΥΜΠΙ ΑΛΛΑΓΗΣ ΤΙΜΗΣ ΜΕΤΡΗΣΗΣ ΜΕ ΔΙΑΦΟΡΕΤΙΚΕΣ ΤΙΜΕΣ ΑΝΑΛΟΓΑ ΜΕ ΤΩΝ ΑΡΙΘΜΟ ΤΩΝ ΠΑΤΗΜΑΤΩΝ

```
@Composable
fun TimeOfMeasurementSelection() {
    // Define the options
    val options = listOf("60s", "120s")

    // Initialize selected option state
    var selectedOptionIndex by remember { mutableStateOf(0) }

    // Compose the DropdownMenu
    Column(
        modifier = Modifier
            .fillMaxWidth()
            .padding(8.dp),
        horizontalAlignment = Alignment.CenterHorizontally,
    ) {
        Row(
            verticalAlignment = Alignment.CenterVertically
        ) {
            Text(text = "Select time: ", color = Color.White)
            Button(
                modifier = Modifier
                    .height(25.dp)
                    .width(50.dp)
                    .shadow(
                        elevation = 4.dp,
                        shape = CircleShape,
                        clip = true
                    )
                    .alpha(0.7f) // Set opacity of shadows
                ,
```

```
                onClick = { // Increment the current index and reset to 0
if it exceeds the number of texts
                    selectedOptionIndex = (selectedOptionIndex + 1) %
options.size
                    // Update the variable based on the selected option
                    collectionDurationMillis =
                        if (options[selectedOptionIndex] == "60s") 60000
else 120000
                }
            ) {
                Text(text = options[selectedOptionIndex])
            }
        }
    }
}
```

TextWithButton

Η συγκεκριμένη συνάρτηση είναι υπεύθυνη για την εμφάνιση κειμένου και κουμπιών βάσει συγκεκριμένων συνθηκών. Ξεκινά με if-else συνθήκες για να καθορίσει ποιο περιεχόμενο να εμφανίσει, είτε πρόκειται για την πρόοδο της διαδικασίας, είτε για τις μετρήσεις της HRV. Αν η HRV έχει υπολογιστεί, παρουσιάζονται διάφοροι δείκτες, όπως οι RMSSD, SDNN, pNN50 και HRVi, μαζί με τους αντίστοιχους χρωματικούς κανόνες. Αν μία τιμή ξεπερνάει τις φυσιολογικές τιμές ανάλογα με την ηλικία και το φύλο του χρήστη, η τιμή εμφανίζεται με κόκκινο χρώμα, αλλιώς με πράσινο (αποτελεί την προεπιλεγμένη επιλογή). Επιπλέον, παρέχεται ένα κουμπί "Reset" για την επαναφορά του υπολογισμού της HRV και την εκτέλεση της συνάρτησης reset. Αν η HRV δεν έχει υπολογιστεί, εμφανίζεται ένα κουμπί "Measure HRV" για την έναρξη της διαδικασίας μέτρησης. Κατά τη διάρκεια της διαδικασίας συλλογής δεδομένων από τους αισθητήρες του ρολογιού, εμφανίζεται η οθόνη ReceivingDataScreen, ενώ κατά την επεξεργασία των δεδομένων εμφανίζεται η οθόνη ProcessingDataScreen.

```
@Composable
fun TextWithButton(RMSSD: Double, SDNN: Double, pNN50: Double,
isHRVCalculated: Boolean) {
    if (isCollectingData) {
        ReceivingDataScreen()
    } else if (isProcessingData) {
        ProcessingDataScreen()
    } else {
        CircularProgressIndicator(
            progress = stressLevel / 100,
            modifier = Modifier.fillMaxSize(),
            startAngle = 100f,
            endAngle = 80f,
            strokeWidth = 4.dp,
            indicatorColor = barColor,
            trackColor = Color.Gray
        )

        Column(
            modifier = Modifier
                .fillMaxSize()
```

```
                .padding(20.dp),
            horizontalAlignment = Alignment.CenterHorizontally,
            verticalArrangement = Arrangement.Center
        ) {
            Text(
                text = "Stress = ${stressLevel.toInt()} %",
                style = TextStyle(fontSize = 18.sp)
            )

            if (isHRVCalculated) {
                Box(
                    modifier = Modifier
                        .fillMaxWidth()
                        .padding(top = 10.dp)
                        .weight(1f)
                ) {
                    ScalingLazyColumn(
                        modifier = Modifier.fillMaxWidth(),
                        horizontalAlignment = Alignment.CenterHorizontally
                    ) {
                        // Define the colors for each metric
                        val rmssdColor = if (RMSSD <
meanrMSSD[gender][ageGroup]) Color.Red else Color.Green
                        val sdnnColor = if (SDNN <
meanSDNN[gender][ageGroup]) Color.Red else Color.Green
                        val pNN50Color = if (pNN50 <
meanpNN50[gender][ageGroup]) Color.Red else Color.Green
                        val AMoColor = Color.Green
                        val MedSDColor = Color.Green

                        // Add each HRV metric as an item
                        item {
                            Text("RMSSD: ${"%.2f".format(RMSSD)} ms",
color = rmssdColor, modifier = Modifier.padding(bottom = 8.dp))
                        }
                        item {
                            Text("SDNN: ${"%.2f".format(SDNN)} ms", color
= sdnnColor, modifier = Modifier.padding(bottom = 8.dp))
                        }
                        item {
                            Text("pNN50: ${"%.2f".format(pNN50)} %", color
= pNN50Color, modifier = Modifier.padding(bottom = 8.dp))
                        }
                        item {
                            Text("AMo: ${"%.2f".format(AMo)} %", color =
AMoColor, modifier = Modifier.padding(bottom = 8.dp))
                        }
                        item {
                            Text("MedSD: ${"%.2f".format(MedSD)} ms",
color = MedSDColor, modifier = Modifier.padding(bottom = 8.dp))
                        }

                        // Add HRV Index conditionally if collection
duration is 120 seconds
                        if (collectionDurationMillis == 12000) {
                            val hrvIndexColor = if (hrvIndex > 20.42)
Color.Red else Color.Green
                            item {
                                Text("HRVi: ${"%.2f".format(hrvIndex)}
ms", color = hrvIndexColor, modifier = Modifier.padding(bottom = 8.dp))
```

```
                                }
                            }
                        }
                    }

                    // Reset button
                    Button(
                        onClick = { reset() },
                        modifier = Modifier
                            .width(120.dp)
                            .padding(top = 16.dp)
                            .height(50.dp)
                    ) {
                        Text(text = "Reset")
                    }
                } else {
                    TimeOfMeasurementSelection()

                    Button(
                        onClick = {
                            Log.i(TAG, " setEventListener called ")
                            if (!isHandlerRunning) {
                                handler.post {

ppgRedTracker!!.setEventListener(trackerEventListener)
                                    isHandlerRunning = true
                                }
                            }
                            isCollectingData = true
                        },
                        modifier = Modifier
                            .fillMaxWidth()
                            .height(50.dp)
                    ) {
                        Text(text = "Measure HRV")
                    }
                }
            }
        }
    }
}
```

ReceivingDataScreen

Συνάρτηση υπεύθυνη για την εμφάνιση της οθόνης συλλογής δεδομένων κατά τη διάρκεια της διαδικασίας μέτρησης. Χρησιμοποιείται για την συνεχή ενημέρωση της προόδου της μέτρησης. Η συνάρτηση αυτή περιλαμβάνει ένα κείμενο που αναφέρει "Measuring HRV" και ένα κείμενο που δείχνει την πρόοδο της μέτρησης σε ποσοστιαία μορφή. Τα κείμενα εμφανίζονται κεντραρισμένα στην οθόνη με κατάλληλα περιθώρια για καλύτερη ορατότητα.

ΕΙΚΟΝΑ 11 - ΟΘΟΝΗ ΚΑΤΑΓΡΑΦΗΣ ΣΗΜΑΤΟΣ

```
@Preview(showBackground = true, device = "id:wearos_small_round")
@Composable
fun ReceivingDataScreen() {
    Column(
        modifier = Modifier
            .fillMaxSize()
            .padding(16.dp),
        horizontalAlignment = Alignment.CenterHorizontally,
        verticalArrangement = Arrangement.Center
    ) {
        Text(
            text = "Measuring HRV", modifier = Modifier.padding(bottom =
16.dp)
        )
        Text(
            text = "$dataCollectionProgress %", modifier =
Modifier.padding(bottom = 16.dp)
        )
    }
}
```

ProcessingDataScreen

Χρησιμοποιείται για να εμφανίσει την οθόνη επεξεργασίας κατά τη διάρκεια της επεξεργασίας των δεδομένων. Το κείμενο "Processing Data" προβάλλεται για να ενημερώσει τον χρήστη για την τρέχουσα διαδικασία.

ΕΙΚΟΝΑ 12 - ΟΘΟΝΗ ΕΠΕΞΕΡΓΑΣΙΑΣ ΔΕΔΟΜΕΝΩΝ

```
@Preview(showBackground = true, device = "id:wearos_small_round")
@Composable
fun ProcessingDataScreen() {
    Column(
        modifier = Modifier
            .fillMaxSize()
            .padding(16.dp),
        horizontalAlignment = Alignment.CenterHorizontally,
        verticalArrangement = Arrangement.Center
    ) {
        Text(
            text = "Processing Data", modifier = Modifier.padding(bottom =
16.dp)
        )
        Text(
            text = "It might take a while...", modifier =
Modifier.padding(bottom = 16.dp)
        )
    }
}
```

5.2.3 Σύνδεση με Samsung Health Platform, Άδειες Χρήστη και Καταγραφή Δεδομένων

Η ενότητα επικεντρώνεται στη διαδικασία σύνδεσης της εφαρμογής με την πλατφόρμα υγείας της Samsung, καθώς και στη διαχείριση των απαιτούμενων δικαιωμάτων χρήστη και στην καταγραφή δεδομένων υγείας. Κατά τη διάρκεια αυτής της διαδικασίας, περιγράφεται η σύνδεση με την υπηρεσία παρακολούθησης υγείας της Samsung, και ο τρόπος συλλογής των δεδομένων από το smartwatch.

onCreate:

Αποτελεί το αρχικό σημείο της εφαρμογής. Αρχικά ελέγχει για τις άδειες που απαιτούνται από την εφαρμογή, χρησιμοποιώντας την PermissionActivity.checkPermission. Εάν οι άδειες έχουν ήδη δοθεί, η μέθοδος setUp καλείται για να αρχικοποιήσει την εφαρμογή. Αν δεν έχουν δοθεί, εμφανίζεται ένα παράθυρο διαλόγου που ζητά τις άδειες από τον χρήστη.

```
override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    window.addFlags(WindowManager.LayoutParams.FLAG_KEEP_SCREEN_ON)

    if (PermissionActivity.checkPermission(this as Context,
this.permissions)) {
        Log.i(TAG, "onCreate Permission granted")
        setUp()
    } else {
        Log.i(TAG, "onCreate Permission not granted")
        PermissionActivity.showPermissionPrompt(
            (this as Activity), this.REQUEST_ACCOUNT_PERMISSION,
this.permissions
        )
    }
}
```

setUp

Προετοιμάζει το περιβάλλον της εφαρμογής για λειτουργία, συνδέοντας την εφαρμογή με το HealthTrackingService και θέτοντας την κύρια διεπαφή της εφαρμογής προς εμφάνιση. Αρχικά, δημιουργεί ένα αντικείμενο HealthTrackingService και με την εντολή connectService, συνδέεται με την υπηρεσία αυτή. Έπειτα καθορίζει την αρχική διεπαφή της εφαρμογής.

```
private fun setUp() {
    healthTrackingService = HealthTrackingService(connectionListener,
applicationContext)
    healthTrackingService!!.connectService()
    setContent {
        HRVCalculator()
    }
}
```

ConnectionListener

Αντικείμενο που χρησιμοποιείται για την αντιμετώπιση συμβάντων σύνδεσης με την υπηρεσία παρακολούθησης υγείας (Health Tracking Service). Περιέχει τρεις μεθόδους:

1. Η μέθοδος **onConnectionSuccess** καλείται όταν η σύνδεση με την υπηρεσία παρακολούθησης υγείας πραγματοποιείται επιτυχώς. Σε αυτήν την περίπτωση, εμφανίζεται ένα μήνυμα επιτυχίας και στη συνέχεια προσπαθεί να ξεκινήσει την καταγραφή δεδομένων μέσω του **ppgRedTracker**, αλλιώς αν αποτύχει εμφανίζει μήνυμα λάθους.

2. Σε περίπτωση που η σύνδεση αποτύχει, η μέθοδος **onConnectionFailed** καλείται, με την παράμετρο e να περιέχει την εξαίρεση που προκλήθηκε. Αν η εξαίρεση έχει λύση, επιχειρείται να λυθεί καλώντας τη μέθοδο resolve, η οποία θα προσπαθήσει να διορθώσει το πρόβλημα. Στη συνέχεια, εμφανίζεται ένα μήνυμα λάθους και η δραστηριότητα τερματίζεται μέσω της μεθόδου finish.

3. Σε περίπτωση τερματισμού σύνδεσης καλείται η **onConnectionEnded**, η οποία είναι κενή.

```
private val connectionListener: ConnectionListener = object :
ConnectionListener {
    override fun onConnectionSuccess() {
        Toast.makeText(
            applicationContext, "Connected to HSP", Toast.LENGTH_SHORT
        ).show()
        try {
            ppgRedTracker =
healthTrackingService!!.getHealthTracker(HealthTrackerType.PPG_RED)
        } catch (e: IllegalArgumentException) {
            runOnUiThread {
                Toast.makeText(
                    applicationContext, e.message, Toast.LENGTH_LONG
                ).show()
            }
            finish()
        }
    }

    override fun onConnectionEnded() {}
    override fun onConnectionFailed(e: HealthTrackerException) {
        if (e.hasResolution()) {
            e.resolve(this@HRVActivity)
        }
        runOnUiThread {
            Toast.makeText(
                applicationContext, "Unable to connect to HSP",
Toast.LENGTH_LONG
            ).show()
        }
        finish()
    }
}
```

TrackerEventListener

Υπεύθυνος για την ακρόαση συμβάντων που σχετίζονται με την καταγραφή δεδομένων από τον αισθητήρα PPG Red. Διακρίνονται τρία κύρια συμβάντα που μπορεί να αντιμετωπίσει:

1. **onDataReceived:** Αυτή η μέθοδος καλείται κάθε φορά που λαμβάνονται νέα δεδομένα από τον αισθητήρα υγείας. Τα νέα δεδομένα παρέχονται στη μέθοδο ως λίστα **DataPoint**. Ο κώδικας ελέγχει αν η λίστα δεδομένων δεν είναι κενή, και στη συνέχεια εκτελεί ενέργειες όπως η αποθήκευση δεδομένων για συγκεκριμένο χρονικό διάστημα (**collectionDurationMillis**), ο υπολογισμός προόδου συλλογής δεδομένων (dataCollectionProgress) και η επεξεργασία των δεδομένων (processBatchData).

2. **onFlushCompleted:** Καλείται όταν η διεργασία εκκαθάρισης (flush) ολοκληρωθεί.

3. **onError:** Καλείται σε περίπτωση σφάλματος κατά την επικοινωνία με τον αισθητήρα. Χρησιμοποιείται για την αντιμετώπιση και την εμφάνιση μηνυμάτων σχετικά με το σφάλμα που παρουσιάστηκε, όπως αν δεν δόθηκε άδεια από τον χρήστη για την κοινοποίηση των δεδομένων του.

```
private val trackerEventListener: TrackerEventListener = object :
TrackerEventListener {
    override fun onDataReceived(list: List<DataPoint>) {
    if (list.isNotEmpty()) {

        if (startTimeMillis == 0L) {
            // Initialize start time on the first data point
            startTimeMillis = System.currentTimeMillis()
        }
        val currentTimeMillis = System.currentTimeMillis()

        val elapsedTime = currentTimeMillis - startTimeMillis
        if (!isHRVCalculated) {
            // Accumulate data points
            for (dataPoint in list) {
                if (dataPoint.timestamp - startTimeMillis >= 0) { //Remove
negative time

ppgSignal.add(dataPoint.getValue(ValueKey.PpgRedSet.PPG_RED))
                    time.add((dataPoint.timestamp -
startTimeMillis).toInt())
                }
            }
            dataCollectionProgress =
                ((elapsedTime.toDouble() / collectionDurationMillis) *
100).toInt()
            if (elapsedTime >= collectionDurationMillis) {
                isCollectingData = false
                processBatchData()
            }
        }
    } else {
        Log.i(TAG, "onDataReceived List is zero")
    }
}


    override fun onFlushCompleted() {
        Log.i(TAG, " onFlushCompleted called")
    }

    override fun onError(trackerError: HealthTracker.TrackerError) {
        Log.i(TAG, " onError called")
        if (trackerError == HealthTracker.TrackerError.PERMISSION_ERROR) {
            runOnUiThread {
                Toast.makeText(
                    applicationContext, "Permissions Check Failed",
Toast.LENGTH_SHORT
```

```
                ).show()
            }
        }
        if (trackerError == HealthTracker.TrackerError.SDK_POLICY_ERROR) {
            runOnUiThread {
                Toast.makeText(
                    applicationContext, "SDK Policy denied",
Toast.LENGTH_SHORT
                ).show()
            }
        }
        isHandlerRunning = false
    }
}
```

stopTracking

Σταματά την καταγραφή των δεδομένων του σήματος από τον αισθητήρα PPG (Photoplethysmography). Εάν η παρακολούθηση μέσω του αισθητήρα κόκκινου φωτός (ppgRedTracker) είναι ενεργή, απενεργοποιείται ο event listener ώστε να σταματήσει η λήψη των δεδομένων. Τέλος, η μεταβλητή isHandlerRunning τίθεται σε false, δηλώνοντας ότι η διαδικασία παρακολούθησης έχει τερματιστεί.

```
// Function to stop tracking data
private fun stopTracking() {
    if (ppgRedTracker != null) {
        ppgRedTracker!!.unsetEventListener()  // Unset the event listener
to stop receiving data
    }
    isHandlerRunning = false
}
```

onDestroy

Καλείται όταν η δραστηριότητα (activity) καταστρέφεται και είναι υπεύθυνη για τον καθαρισμό των πόρων που χρησιμοποιούνται. Στην αρχή, η μέθοδος ελέγχει αν το αντικείμενο ppgRedTracker είναι ενεργό και, αν ναι, αποδεσμεύει οποιονδήποτε event listener έχει ρυθμιστεί, αποτρέποντας έτσι τυχόν αλληλεπιδράσεις με την καταστραμμένη δραστηριότητα. Στη συνέχεια, η μέθοδος καλεί την removeCallbacksAndMessages(null) για να αφαιρέσει όλες τις καθυστερημένες εργασίες και μηνύματα, αποτρέποντας την εκτέλεση τους. Επιπλέον, η μεταβλητή isHandlerRunning ρυθμίζεται σε false, υποδεικνύοντας ότι ο handler δεν είναι πλέον ενεργός. Τέλος, η μέθοδος ελέγχει αν η υπηρεσία παρακολούθησης υγείας (healthTrackingService) είναι ενεργή και, εάν είναι, καλεί την disconnectService για να την αποσυνδέσει, διασφαλίζοντας την ομαλή διακοπή της λειτουργίας της. Αυτές οι ενέργειες διασφαλίζουν ότι όλοι οι πόροι και οι συνδέσεις της δραστηριότητας αποδεσμεύονται σωστά, ελαχιστοποιώντας πιθανές διαρροές μνήμης και βελτιώνοντας τη γενική απόδοση της εφαρμογής.

```
override fun onDestroy() {
    super.onDestroy()
    if (ppgRedTracker != null) {
```

```
            ppgRedTracker!!.unsetEventListener()
        }
        handler.removeCallbacksAndMessages(null)
        isHandlerRunning = false

        if (healthTrackingService != null) {
            healthTrackingService!!.disconnectService()
        }
    }
```

5.2.4 Αλγόριθμος Υπολογισμού της HRV από σήμα PPG

processBatchData

Υπεύθυνη για την επεξεργασία του σήματος και τον υπολογισμό της HRV από τα δεδομένα που έχουν συλλεχθεί. Κατά την εκτέλεσή της, αρχικά, καλείται η συνάρτηση stopTracking, η οποία σταματάει την διαδικασία καταγραφής περαιτέρω στοιχείων από τον αισθητήρα του ρολογιού, αφού σε αυτό το σημείο η μέτρηση έχει ολοκληρωθεί. Η μεταβλητή isProcessingData ορίζεται σε true, υποδεικνύοντας ότι η επεξεργασία των δεδομένων έχει ξεκινήσει. Ακολουθεί φιλτράρισμα του σήματος μέσω της συνάρτησης filterPPGSignal, με σκοπό την απομάκρυνση θορύβου και ανεπιθύμητων συχνοτήτων. Έπειτα, προχωράμε στον υπολογισμό των δεικτών της HRV με τη χρήση της συνάρτησης calculateHRVMetrics.

Τέλος, το επίπεδο άγχους (stressLevel) εκτιμάται βάσει των δεικτών AMo, pNN50 και MedSD και αποδίδεται μέσω μιας χρωματικής κλίμακας.

Πίνακας 2 - Αντιστοίχιση επιπέδου άγχους με χρώμα

Ποσοστό Άγχους	**Χρώμα**
<30%	Μπλε
30-70%	Πράσινο
>70%	Κόκκινο

Αν κατά τη διάρκεια της μέτρησης και του υπολογισμού της HRV δεν έχει προκύψει κάποιο σφάλμα, οι μεταβλητές isHRVCalculated και isProcessingData ορίζονται κατάλληλα, υποδεικνύοντας ότι ο υπολογισμός ολοκληρώθηκε.

```
private fun processBatchData() {
    // Stop the tracker from receiving further data
    stopTracking()

    // Perform HRV calculation on the accumulated data
    isProcessingData = true
    filterPPGSignal()
```

```
    calculateHRVMetrics()

    // Calculate Stress Level
    stressLevel = calculateStressLevel(AMo, pNN50, MedSD)
    barColor = if (stressLevel < 30) {
        Color.Blue
    } else if (stressLevel > 70) {
        Color.Red
    } else {
        Color.Green
    }

    if (!errorOccurred) {
        isHRVCalculated = true
    }
    isProcessingData = false
}
```

findAgeGroup

Χρησιμοποιείται για τον εντοπισμό της ομάδας ηλικίας του χρήστη, με βάση την ηλικία που παρέχεται. Αρχικά, χρησιμοποιεί τη λίστα ageGroups, η οποία περιλαμβάνει τα όρια των ηλικιακών ομάδων. Αναζητά το πρώτο στοιχείο της λίστας που είναι μεγαλύτερο από την ηλικία του χρήστη και αν το βρει, η μεταβλητή ageGroup ορίζεται σε αυτή τη θέση. Αν δε βρεθεί καμία ηλικιακή ομάδα που να είναι μεγαλύτερη από την ηλικία του χρήστη, τότε η μεταβλητή ageGroup ορίζεται στην τελευταία θέση της λίστας ageGroups. Τέλος, η μεταβλητή ageGroupIsFound ορίζεται σε true, υποδεικνύοντας ότι η διαδικασία εύρεσης ηλικιακής ομάδας έχει ολοκληρωθεί.

```
private fun findAgeGroup() {
    ageGroup = ageGroups.indexOfFirst { it > age }
    if (ageGroup == -1)
        ageGroup = ageGroups.lastIndex
    ageGroupIsFound = true
}
```

calculateHRVMetrics

Εκτελεί τον υπολογισμό διαφόρων δεικτών που σχετίζονται με την HRV (Heart Rate Variability - Καρδιακή Ρυθμικότητα). Η μεθοδολογία στηρίζεται σε ένα μέρος της δημοσίευσης των Alqaraawi κ.ά. [3] όσον αφορά τον αλγόριθμο για τον εντοπισμό των μεγίστων.

Η διαδικασία ξεκινά με τον υπολογισμό της πρώτης παραμέτρου p1 μέσω της συνάρτησης calculateP1, η οποία αποτελεί την πιθανότητα ένα σημείο του σήματος (datapoint) να είναι μέγιστο. Έπειτα, γίνεται εντοπισμός των κορυφών στο σήμα PPG με χρήση της συνάρτησης peakDetection με βάση μόνο των πιθανοτήτων p1. Τα διαστήματα RR (RR intervals) υπολογίζονται μέσω της συνάρτησης calculateRRIntervals με τη χρήση των εντοπισμένων κορυφών.

Έπειτα, υπολογίζεται η δεύτερη παράμετρος p2 μέσω της συνάρτησης calculateP2, λαμβάνοντας υπόψιν τις κορυφές και τα διαστήματα RR που είχαν υπολογιστεί προηγουμένως. Η p2 βελτιώνει την ακρίβεια των αποτελεσμάτων που είχαν υπολογιστεί με βάση την p1, καθώς χρησιμοποιεί την κατανομή Laplace για να μοντελοποιήσει τις διαφορές μεταξύ γειτονικών RR-διαστημάτων στο PPG σήμα. Οι πιθανότητες p2 κανονικοποιούνται σε σχέση με τις τιμές των p1 πιθανοτήτων μέσω της normalizeAccordingTo. Αυτή η διαδικασία εξασφαλίζει ότι τα δύο είδη πιθανοτήτων είναι συγκρίσιμα και μπορούν να συνδυαστούν για να δημιουργηθεί ένας συνδυασμένος δείκτης πιθανότητας.

Ο συνδυασμένος δείκτης πιθανότητας (combinedPossibility) υπολογίζεται πολλαπλασιάζοντας τις πιθανότητες p1 και p2 για κάθε σημείο δεδομένων. Οι νέες κορυφές εντοπίζονται από αυτόν τον συνδυασμένο δείκτη και οι διαδοχικές κορυφές συγχωνεύονται μέσω της συνάρτησης mergeConsecutivePeaks για να αφαιρεθούν οι ψευδείς κορυφές (false peaks). Τα νέα διαστήματα RR υπολογίζονται από τις νέες κορυφές με τη κλήση της calculateRRIntervals.

Μια σημαντική πτυχή της διαδικασίας είναι ο υπολογισμός της μέσης περιόδου καρδιάς (heartPeriod), που υπολογίζεται με τη συνάρτηση calculateHeartPeriod. Αυτή η περίοδος χρησιμοποιείται για τον καθορισμό ενός εύρους αποδεκτών τιμών για τα διαστήματα RR, με το ελάχιστο αποδεκτό όριο να είναι 150 ms λιγότερο από την περίοδο της καρδιάς που υπολογίστηκε, ενώ η μέγιστη να είναι 150 ms περισσότερο. Τα διαστήματα RR που δεν εμπίπτουν σε αυτό το εύρος φιλτράρονται μέσω της συνάρτησης filterRRIntervals.

Ακολουθεί ο υπολογισμός των διαφορών μεταξύ των φιλτραρισμένων διαστημάτων RR με τη συνάρτηση calculateRRIntervalDifferences, οι οποίες είναι κρίσιμες για τον υπολογισμό των δεικτών της HRV. Οι βασικοί δείκτες που υπολογίζονται περιλαμβάνουν τους RMSSD, SDNN, pNN50, AMo και MedSD, οι οποίοι παρέχουν σημαντικές πληροφορίες για τη λειτουργία του αυτόνομου νευρικού συστήματος.

```
private fun calculateHRVMetrics() {

    val p1 = calculateP1()

    val peakIndices = peakDetection(p1)

    val rrIntervals = calculateRRIntervals(peakIndices)

    var p2 = calculateP2(peakIndices, rrIntervals)
    p2 = normalizeAccordingTo(p2, p1)

    // Calculate combined possibility
    val combinedPossibility = DoubleArray(ppgSignal.size)
    for (i in ppgSignal.indices) {
        combinedPossibility[i] = p1[i] * p2[i]
    }

    // Detect peaks from combined possibility
    var newPeakIndices = peakDetection(combinedPossibility)

    newPeakIndices = mergeConsecutivePeaks(newPeakIndices)
```

```
    val newRRIntervals = calculateRRIntervals(newPeakIndices)

    val heartPeriod = calculateHeartPeriod()

    val minValue = heartPeriod - 150
    val maxValue = heartPeriod + 150

    val filteredIntervals = filterRRIntervals(newRRIntervals, minValue,
maxValue)

    // Calculate the RR interval differences with filtering
    val rrDifferences = calculateRRIntervalDifferences(filteredIntervals)

    // Calculate metrics using the pre-calculated differences
    RMSSD = calculateRMSSD(rrDifferences)
    SDNN = calculateSDNN(filteredIntervals)  // SDNN still uses the
original intervals
    pNN50 = calculatePNN50(rrDifferences)
    AMo = calculateAMo(filteredIntervals)
    MedSD = calculateMedSD(filteredIntervals)
    if (collectionDurationMillis == 120000) {
        hrvIndex = calculateHRVTriangularIndex(filteredIntervals)
    }
}
```

filterPPGSignal

Η συνάρτηση επεξεργάζεται το σήμα φωτοπληθυσμογραφίας και αφαιρεί τον θόρυβο, διατηρώντας μόνο το φυσιολογικό σήμα.

Αρχικά, η συνάρτηση αφαιρεί τα πρώτα 500 στοιχεία δεδομένων από το σήμα PPG, καθώς και τις αντίστοιχες χρονικές στιγμές στον πίνακα του χρόνου. Αυτή η επιλογή γίνεται για να εξασφαλιστεί ότι οποιοσδήποτε αρχικός θόρυβος ή αστάθεια στο σήμα, που μπορεί να επηρεάσει αρνητικά την ανάλυση, απομακρύνεται. Το πλήθος των στοιχείων που αφαιρέθηκαν έχει επιλεχθεί πειραματικά.

Στη συνέχεια, ορίζεται η τάξη του φίλτρου σε 6 και τα στοιχεία προς αφαίρεση μετά την εφαρμογή του φίλτρου σε 1500. Η τάξη του φίλτρου (order), έχει επιλεγεί ως 6 για να εξασφαλιστεί η αποτελεσματική απόρριψη ανεπιθύμητων συχνοτήτων. Επίσης, η επιλογή των συγκεκριμένων τιμών προέκυψε έπειτα από παρατήρηση σε PPG σήματα που καταγράφηκαν με την χρήση smartwatch σε εθελοντές και την επεξεργασία τους σε Matlab με διάφορες τάξεις φίλτρων.

Για την εξαγωγή των χαρακτηριστικών από το σήμα PPG, χρησιμοποιούνται φίλτρα Butterworth. Το χαμηλοπερατό φίλτρο έχει επιλεγεί με υψηλή συχνότητα αποκοπής στα 5.0 Hz για να αφαιρέσει υψηλές συχνότητες που δεν σχετίζονται με τον καρδιακό ρυθμό. Αντίστοιχα, το υψηλοπερατό φίλτρο με χαμηλή συχνότητα αποκοπής στα 0.5 Hz χρησιμοποιείται για να αφαιρέσει τις πολύ χαμηλές συχνότητες, εξασφαλίζοντας ότι μόνο οι σχετικές με τον καρδιακό ρυθμό συχνότητες παραμένουν στο σήμα. Αυτή η διπλή διαδικασία φιλτραρίσματος συμβάλλει στη βελτίωση της ποιότητας του σήματος. Για την χρήση του συγκεκριμένου φίλτρου χρησιμοποιήθηκε η βιβλιοθήκη

του ερευνητή Bernd Porr [32], από όπου και χρησιμοποιήθηκαν οι συναρτήσεις που υλοποιούν τα παραπάνω φίλτρα.

Μετά το φιλτράρισμα, τα πρώτα δεδομένα του σήματος PPG φιλτράρονται ξανά, αυτή τη φορά για να απομακρυνθούν οποιαδήποτε κατάλοιπα από την προηγούμενη διαδικασία φιλτραρίσματος (order_cut = 1500 στοιχεία προς αφαίρεση). Αυτή η διαδικασία εξασφαλίζει ότι το σήμα PPG έχει καθαριστεί από το θόρυβο και διατηρεί μόνο τις φυσιολογικές πληροφορίες που είναι χρήσιμες για ανάλυση και επεξεργασία.

```
private fun filterPPGSignal() {
    filterFirstXDataPoints(ppgSignal, time, 500)
    val order = 6
    val order_cut = 1500

    //Frequencies between 0.5 - 5.0 Hz are allowed (physiological signals)
    val highCutoff = 5.0 // High cut-off frequency
    val lowCutoff = 0.5 // Low cut-off frequency

    val butterworth = Butterworth()
    butterworth.lowPass(order, fs.toDouble(), highCutoff)
    for (i in ppgSignal.indices) {
        ppgSignal[i] = butterworth.filter(ppgSignal[i].toDouble()).toInt()
    }

    val butterworth1 = Butterworth()
    butterworth1.highPass(order, fs.toDouble(), lowCutoff)
    for (i in ppgSignal.indices) {
        ppgSignal[i] =
butterworth1.filter(ppgSignal[i].toDouble()).toInt()
    }
    filterFirstXDataPoints(ppgSignal, time, order_cut)
}
```

filterRRIntervals

Έχει σχεδιαστεί για να φιλτράρει τα διαστήματα RR (RR intervals) με βάση καθορισμένες ελάχιστες και μέγιστες τιμές. Αυτό είναι ιδιαίτερα χρήσιμο για την απομάκρυνση ακραίων τιμών που μπορούν να επηρεάσουν την ανάλυση του σήματος και την ακρίβεια των δεικτών της HRV.

```
// Function to filter RR intervals based on a minimum and maximum value
threshold
private fun filterRRIntervals(
    rrIntervals: DoubleArray,
    minValue: Double,
    maxValue: Double
): DoubleArray {
    return rrIntervals.filter { it in minValue..maxValue }.toDoubleArray()
}
```

calculateRRIntervalDifferences

Υπολογίζει τις διαφορές μεταξύ διαδοχικών διαστημάτων RR (RR intervals), αφού πρώτα φιλτράρει τα διαστήματα αυτά βάσει καθορισμένων ελάχιστων και μέγιστων τιμών, χρησιμοποιώντας την filterRRIntervals.

```
// Function to calculate RR interval differences for filtered values
private fun calculateRRIntervalDifferences(
    rrIntervals: DoubleArray,
    minValue: Double,
    maxValue: Double
): DoubleArray {
    val filteredIntervals = filterRRIntervals(rrIntervals, minValue,
maxValue)
    if (filteredIntervals.size < 2) {
        println("Not enough rrIntervals to calculate the differences.")
        return doubleArrayOf()
    }

    val differences = DoubleArray(filteredIntervals.size - 1)
    for (i in 1 until filteredIntervals.size) {
        differences[i - 1] = filteredIntervals[i] - filteredIntervals[i -
1]
    }
    return differences
}
```

calculateHeartPeriod

Υπολογίζει την περίοδο του καρδιακού παλμού με βάση ένα σήμα PPG (Photoplethysmography), που δέχεται ως είσοδο, χρησιμοποιώντας τη μέθοδο του Γρήγορου Μετασχηματισμού Fourier (FFT) για την ανάλυση συχνοτήτων.

Αρχικά, το σήμα PPG που είναι σε μορφή ακέραιων αριθμών μετατρέπεται σε πίνακα αριθμών κινητής υποδιαστολής (float) για την περαιτέρω επεξεργασία. Για να διασφαλιστεί η ορθότητα του FFT, ελέγχεται εάν το μήκος του σήματος είναι άρτιο. Εάν δεν είναι, προστίθεται μια τιμή (0.0) στο τέλος του πίνακα για να γίνει άρτιος.

Έπειτα, αρχικοποιείται ένας πίνακας (Noise) που θα χρησιμοποιηθεί για τον μετασχηματισμό Fourier. Η συνάρτηση FFT εκτελείται πάνω στο σήμα, και τα αποτελέσματα αποθηκεύονται στον πίνακα fftResult. Για την υλοποίηση του FFT χρησιμοποιήθηκε η βιβλιοθήκη του Pär Nils Amsen [41].

Στην συνέχεια, υπολογίζεται το φάσμα ισχύος, το οποίο προκύπτει από το τετράγωνο του πραγματικού και φανταστικού μέρους των αποτελεσμάτων του FFT. Δεδομένου ότι ο μετασχηματισμός γίνεται από πραγματικές σε μιγαδικές συχνότητες, χρησιμοποιείται μόνο το μισό του φάσματος για τον υπολογισμό, καθώς το άλλο μισό είναι συμμετρικό.

Ακολούθως, η συνάρτηση εντοπίζει τη συχνότητα που έχει τη μέγιστη ισχύ στο φάσμα, η οποία αντιστοιχεί στην κυρίαρχη συχνότητα του καρδιακού παλμού. Ο υπολογισμός της συχνότητας γίνεται με βάση τη θέση της μέγιστης τιμής στο φάσμα και τη συχνότητα δειγματοληψίας fs.

Τέλος, η καρδιακή περίοδος υπολογίζεται ως το αντίστροφο της κυρίαρχης συχνότητας, και μετατρέπεται σε χιλιοστά του δευτερολέπτου (ms).

```
private fun calculateHeartPeriod(): Double {
    // Convert signal list from int to float
    var floatSignal = ppgSignal.map { it.toFloat() }.toFloatArray()

    // Ensure the length of floatSignal is even
    if (floatSignal.size % 2 != 0) {
        floatSignal = floatSignal.plus(0.0f)
    }

    // Initialize Noise for FFT
    val noise = Noise.real(floatSignal.size)

    // Create output array for FFT results
    val fftResult = FloatArray(floatSignal.size + 2)

    // Perform FFT
    noise.fft(floatSignal, fftResult)

    // Calculate power spectrum (magnitude squared of FFT results)
    // For real-to-complex FFT, use half of the spectrum, excluding the
Nyquist frequency
    val halfSize = floatSignal.size / 2
    val powerSpectrum = MutableList(halfSize) { 0.0 }
    // Calculate power spectrum (magnitude squared of FFT results)
    for (index in 0 until halfSize) {
        val realPart = fftResult[2 * index].toDouble()
        val imagPart = fftResult[2 * index + 1].toDouble()
        powerSpectrum[index] = realPart * realPart + imagPart * imagPart
    }

    // Find the center frequency
    val maxPowerIndex = powerSpectrum.indices.maxByOrNull {
powerSpectrum[it] } ?: 0
    val maxPowerFrequency = (maxPowerIndex * fs.toDouble()) /
floatSignal.size

    //Find the heart period
    return (1 / maxPowerFrequency) * 1000 //Convert to ms
}
```

calculateRMSSD

Υπολογίζει τον δείκτη RMSSD, ο οποίος αποτελεί μια μέτρηση της μεταβλητότητας του καρδιακού ρυθμού βασισμένη στις διαδοχικές διαφορές μεταξύ διαστημάτων RR. Η διαδικασία ξεκινά με τον έλεγχο αν ο πίνακας με τις διαφορές των RR διαστημάτων είναι κενός, και αν ναι, επιστρέφει 0.0. Στη συνέχεια, υπολογίζεται το άθροισμα των τετραγώνων των διαφορών, και η τελική τιμή του RMSSD προκύπτει από την τετραγωνική ρίζα του μέσου όρου αυτών των τετραγώνων (βλ. σελ.27). Το RMSSD αντανακλά την ποσότητα της μεταβλητότητας μεταξύ διαδοχικών διαστημάτων RR, δίνοντας έτσι μια εικόνα της αυτόνομης νευρικής δραστηριότητας.

```
// Function to calculate RMSSD
private fun calculateRMSSD(rrDifferences: DoubleArray): Double {
    if (rrDifferences.isEmpty()) return 0.0

    var sumOfSquares = 0.0
    for (diff in rrDifferences) {
        sumOfSquares += diff * diff
    }
    return sqrt(sumOfSquares / rrDifferences.size)
}
```

calculateSDNN

Υπολογίζει τον δείκτη SDNN, ο οποίος ορίζεται ως η τυπική απόκλιση των διαστημάτων RR και χρησιμεύει για να δώσει μια γενική μέτρηση της συνολικής μεταβλητότητας του καρδιακού ρυθμού. Η διαδικασία ξεκινά με την εύρεση του μέσου όρου των διαστημάτων RR και στη συνέχεια υπολογίζει το άθροισμα των τετραγώνων των αποκλίσεων από τον μέσο όρο. Το SDNN υπολογίζεται ως η τετραγωνική ρίζα του μέσου όρου των τετραγώνων αυτών των αποκλίσεων, προσαρμοσμένης με βάση τον αριθμό των διαστημάτων RR μειωμένο κατά ένα (βλ. σελ.27). Το SDNN παρέχει μια συνολική εικόνα της καρδιοαγγειακής μεταβλητότητας.

```
// Function to calculate SDNN
private fun calculateSDNN(rrIntervals: DoubleArray): Double {
    if (rrIntervals.isEmpty()) return 0.0

    val mean = rrIntervals.average()
    var sumOfSquares = 0.0
    for (rrInterval in rrIntervals) {
        val deviation = rrInterval - mean
        sumOfSquares += deviation * deviation
    }
    return sqrt(sumOfSquares / rrIntervals.size - 1)
}
```

calculatePNN50

Υπολογίζει το pNN50, το ποσοστό των διαδοχικών διαστημάτων RR που διαφέρουν κατά περισσότερα από 50 ms. Η συνάρτηση αρχικά μετρά τον αριθμό των διαφορών που υπερβαίνουν τα 50 ms και στη συνέχεια υπολογίζει το ποσοστό αυτών των διαφορών σε σχέση με το σύνολο των διαστημάτων RR. Το αποτέλεσμα, εκφρασμένο ως ποσοστό, παρέχει πληροφορίες για τη συχνότητα εμφάνισης μεγάλων διαφορών μεταξύ διαδοχικών διαστημάτων RR, υποδεικνύοντας τη διακύμανση του καρδιακού ρυθμού.

```
// Function to calculate pNN50
private fun calculatePNN50(rrDifferences: DoubleArray): Double {
    if (rrDifferences.isEmpty()) return 0.0

    var count = 0
    for (diff in rrDifferences) {
        if (abs(diff) > 50) {
            count++
```

```
            }
        }
        return (count.toDouble() / (rrDifferences.size + 1)) * 100
}
```

calculateMedSD

Έχει ως σκοπό τον υπολογισμό της διάμεσου των διαφορών των διαδοχικών χρονικών διαστημάτων RR, τα οποία αντιπροσωπεύουν τα χρονικά διαστήματα μεταξύ διαδοχικών καρδιακών παλμών [16]. Η διαδικασία ξεκινά με την κλήση της συνάρτησης calculateRRIntervalDifferences, η οποία υπολογίζει τις διαφορές μεταξύ κάθε ζεύγους διαδοχικών διαστημάτων RR. Στη συνέχεια, αυτές οι διαφορές μετατρέπονται σε απόλυτες τιμές και αποθηκεύονται σε έναν νέο πίνακα. Έπειτα, υπολογίζεται η διάμεσος αυτών των διαφορών με τη χρήση της συνάρτησης calculateMedian. Τελικά, επιστρέφεται η διάμεσος, η οποία αντιπροσωπεύει τη μέση διακύμανση των διαφορών των διαστημάτων RR, και αποτελεί μια ένδειξη της μεταβλητότητας του καρδιακού ρυθμού.

```
private fun calculateMedSD(rrIntervals: DoubleArray): Double {
    val differences = calculateRRIntervalDifferences(rrIntervals)
    val absoluteDifferences = differences.map { abs(it) }.toDoubleArray()
    val median = calculateMedian(absoluteDifferences)
    return median
}
```

calculateAMo

Υπολογίζει τον δείκτη AMo (Amplitude of Mode), που χρησιμοποιείται για την ανάλυση της μεταβλητότητας του καρδιακού ρυθμού, βασιζόμενο στα χρονικά διαστήματα RR [16]. Η διαδικασία ξεκινά με τον καθορισμό του μεγέθους του «κάδου» (bin size), το οποίο εδώ έχει οριστεί στα 50 ms. Στη συνέχεια, δημιουργείται ένας πίνακας κατανομής (histogram) με σκοπό την καταγραφή της συχνότητας εμφάνισης των χρονικών διαστημάτων RR εντός συγκεκριμένων τμημάτων.

Για κάθε RR διάστημα στον πίνακα rrIntervals, η συνάρτηση υπολογίζει σε ποιον κάδο ανήκει το διάστημα, διαιρώντας το με το μέγεθος του κάδου και μετατρέποντάς το σε ακέραιο αριθμό. Κατόπιν, αυξάνει την καταμέτρησή του συγκεκριμένου τμήματος κατά ένα στον πίνακα κατανομής. Μετά από την συμπλήρωση του πίνακα, η συνάρτηση αναζητά το τμήμα με τη μεγαλύτερη συχνότητα, που αντιστοιχεί στη συχνότερη τιμή των διαστημάτων RR.

Ο δείκτης AMo υπολογίζεται ως το ποσοστό του τμήματος με τη μεγαλύτερη συχνότητα σε σχέση με το συνολικό αριθμό των διαστημάτων RR και εκφράζει τη σχετική επικράτηση του πιο συχνού διαστήματος RR, δίνοντας μια ένδειξη της μεταβλητότητας του καρδιακού ρυθμού.

```
private fun calculateAMo(rrIntervals: DoubleArray): Double {
    val binSize = 50.0 // 50 ms bin size
    val histogram = mutableMapOf<Int, Int>()
```

```
    // Fill the histogram
    for (interval in rrIntervals) {
        val bin = (interval / binSize).toInt()
        histogram[bin] = histogram.getOrDefault(bin, 0) + 1
    }

    // Find the bin with the highest count (mode)
    val maxCount = histogram.values.maxOrNull() ?: 0

    // Calculate AMo as a percentage of the total number of RR intervals
    return (maxCount.toDouble() / rrIntervals.size) * 100
}
```

calculateP1

Υπολογίζει την πιθανότητα ένα σημείο του σήματος PPG να είναι μέγιστο, το οποίο αποτελεί βελτίωση του αλγόριθμου AMPD [25]. Δεν δέχεται κάποια παράμετρο, καθώς τα δεδομένα του σήματος αποτελούν στοιχείο της κλάσης και επιστρέφει έναν πίνακα τύπου double που κάθε τιμή του συμβολίζει την πιθανότητα το συγκεκριμένο σημείο του σήματος να είναι μέγιστο. Ουσιαστικά γίνεται ανάλυση σκαλογραφήματος τοπικού μεγίστου (Local Maxima Scalogram - LMS) στο PPG σήμα.

Θεωρούμε ως n το πλήθος όλων των στοιχείων του PPG σήματος και ως l = 2 * fs, τον αριθμό των επιπέδων στο σκαλογράφημα. Έπειτα, δημιουργούμε έναν πίνακα p1 με μήκος ίσο με το πλήθος των δειγμάτων του σήματος PPG, όπου θα αποθηκεύουμε την πιθανότητα να είναι το κάθε σημείο του σήματος μέγιστο. Στην συνέχεια εφαρμόζεται αλγόριθμος μεταβλητού παραθύρου για κάθε επίπεδο στο σκαλογράφημα από k=1 έως l-1 με windowSize = 2*k. Ακολούθως, εφαρμόζεται το κριτήριο τοπικού μεγίστου, που φαίνεται παρακάτω.

$$m_{k,i} = \begin{cases} 1 & x_{i-1} > x_{i-k-1} \text{ and } x_{i-1} > x_{i+k-1} \\ 0 & \text{otherwise} \end{cases}$$

ΕΙΚΟΝΑ 13 - ΚΡΙΤΗΡΙΟ ΕΠΙΛΟΓΗΣ ΤΟΠΙΚΟΥ ΜΕΓΙΣΤΟΥ
ΠΗΓΗ: ΠΡΟΣΑΡΜΟΓΗ ΑΠΟ [3]

Οπότε, στον κώδικα η παραπάνω μεθοδολογία εφαρμόστηκε με την αύξηση του p1 στο κάθε στοιχείο κατά 1 αν και μόνο αν ισχύει ppgSignal[j] > ppgSignal[j - 1] και ppgSignal[j] > ppgSignal[j + 1] με j=k+1 έως 2*k+1, όπου ppgSignal είναι ο πίνακας ακεραίων που περιέχει το σύνολο των σημείων του PPG σήματος και j συμβολίζει τη θέση του σημείου που εξετάζουμε. Στο τέλος, διαιρούμε το κάθε στοιχείο του p1 με το πλήθος των επιπέδων του σκαλογραφήματος l, για να προκύψει ο πίνακας με τις τελικές πιθανότητες, οποίος επιστρέφεται στο τέλος της συνάρτησης.

```
fun calculateP1(): DoubleArray {
    val n = ppgSignal.size
```

```
    val l = 2 * fs
    val m = Array(l) { IntArray(n) }
    val p1 = DoubleArray(n)

    for (k in 1 until l) {
        // Implement sliding window for local maximum detection
        for (i in k + 1 until n - k) {
            // Apply the comparison criterion to find local maxima within
the sliding window
            if (ppgSignal[i - 1] > ppgSignal[i - k - 1] && ppgSignal[i -
1] > ppgSignal[i + k - 1]) {
                m[k][i] = 1
            } else {
                m[k][i] = 0
            }
        }
    }

    for (i in 0 until n) {
        var sum = 0
        for (k in 0 until l) {
            sum += m[k][i]
        }
        p1[i] = ((sum.toDouble() / l))
    }

    return p1
}
```

calculateP2

Υπολογίζει μια δεύτερη πιθανότητα για κάθε σημείο του σήματος PPG να είναι μέγιστο, χρησιμοποιώντας την κατανομή Laplace. Δέχεται ως είσοδο μια λίστα με τους δείκτες των κορυφών (peakIndices) και έναν πίνακα με τα διαστήματα RR που υπολογίστηκαν προηγουμένως (rrIntervals). Επιστρέφει έναν πίνακα Double που περιέχει αυτή την πιθανότητα για κάθε σημείο του σήματος να είναι μέγιστο.

Αρχικά, η συνάρτηση ελέγχει αν υπάρχουν αρκετές κορυφές για να υπολογιστεί η τιμή P2. Αν οι κορυφές είναι λιγότερες από δύο, εμφανίζεται μήνυμα λάθος και η συνάρτηση επιστρέφει έναν πίνακα p2 με μηδενικές τιμές.

Στη συνέχεια, η συνάρτηση ξεκινά από την πρώτη ανιχνευμένη κορυφή και υπολογίζει το διάστημα χρόνου μεταξύ της τελευταίας ανιχνευμένης κορυφής και κάθε δείγματος (r[i] = time[i] - time[lastDetectedPeakIndex]). Από αυτή την τιμή αφαιρείται το τελευταίο RR διάστημα που υπολογίστηκε στον προηγούμενο κύκλο (d[i] = r[i] - rrIntervals[lastRRIntervalIndex]).

Η συνάρτηση συνεχίζει με τον υπολογισμό της παραμέτρου b (βαθμωτή παράμετρος). Αυτή η παράμετρος υπολογίζεται ως ο μέσος όρος της απόλυτης απόκλισης των τιμών του πίνακα d από τη διάμεσό τους. Η διάμεσος υπολογίζεται χρησιμοποιώντας τη συνάρτηση calculateMedian, και στη συνέχεια υπολογίζεται η μέση απόλυτη απόκλιση από αυτήν τη διάμεσο.

$$b = \frac{1}{K}\sum_{j=1}^{K}|d_j - \mu|$$

ΕΙΚΟΝΑ 14 - ΤΥΠΟΣ ΥΠΟΛΟΓΙΣΜΟΥ ΠΑΡΑΜΕΤΡΟΥ Β
ΠΗΓΗ: ΠΡΟΣΑΡΜΟΓΗ ΑΠΟ [3]

Τέλος, χρησιμοποιώντας την παράμετρο b, υπολογίζεται η πιθανότητα p2 για κάθε δείγμα στο σήμα PPG. Ο υπολογισμός αυτός βασίζεται στην κατανομή Laplace και εκφράζεται με τον τύπο:

$$p_2^{(i)} = \frac{1}{2b}\exp\left(-\frac{|d_i - \mu|}{b}\right)$$

ΕΙΚΟΝΑ 15 - ΤΥΠΟΣ ΥΠΟΛΟΓΙΣΜΟΥ P2
ΠΗΓΗ: ΠΡΟΣΑΡΜΟΓΗ ΑΠΟ [3]

Τέλος, επιστρέφει τον πίνακα p2 που περιέχει τις εκτιμήσεις πιθανότητας για κάθε δείγμα να είναι μέγιστο. Αυτός ο υπολογισμός συνεισφέρει στην ακριβέστερη εκτίμηση της HRV, ιδιαίτερα όταν το σήμα PPG είναι παραμορφωμένο με θόρυβο λόγω κίνησης.

```
private fun calculateP2(
    peakIndices: List<Int?>?, rrIntervals: DoubleArray
): DoubleArray {
    val d = DoubleArray(ppgSignal.size)
    val r = LongArray(ppgSignal.size)
    val p2 = DoubleArray(ppgSignal.size)
    var b = 0.0
    if (peakIndices!!.size <= 1) {
        println("Not enough peaks to calculate P2.")
        return p2
    }
    // Starting from the first detected peak
    var j = 0
    var k = 0 // Size of d vector
    var numberOfPeaks = 0 //Starting from the first detected peak
    var lastDetectedPeakIndex = 0
    var lastRRIntervalIndex = 0

    for (i in 0 until ppgSignal.size) {
        if (i == peakIndices[j]) { //If a peak is crossed
            lastDetectedPeakIndex = i
            numberOfPeaks++
            if (j + 1 < peakIndices.size) j++
        }
        if (numberOfPeaks > 2) {
            if (lastRRIntervalIndex + 1 < rrIntervals.size)
lastRRIntervalIndex++
        }
```

```
            r[i] = (time[i] - time[lastDetectedPeakIndex])
            d[i] = r[i] - rrIntervals[lastRRIntervalIndex]
            k++
        }
        val median = calculateMedian(d)
        for (a in 0 until k) {
            b += abs(d[a] - median)
        }
        b /= k
        for (i in 0 until ppgSignal.size) {
            p2[i] = round(((1.0 / (2.0 * b)) * exp(-(abs(d[i] - median) / b)))
* 1000.0) / 100.0
        }
        return p2
}
```

normalizeAccordingTo

Η συνάρτηση normalizeAccordingTo αποσκοπεί στη μετατροπή των πιθανοτήτων που περιέχονται στον πίνακα p2 στην ίδια κλίμακα με τις πιθανότητες του πίνακα p1. Οι πίνακες p1 και p2 αναπαριστούν πιθανότητες ότι συγκεκριμένα σημεία του σήματος αποτελούν κορυφές. Η διαδικασία κανονικοποίησης επιτρέπει τη σύγκριση ή τον συνδυασμό αυτών των πιθανοτήτων με βάση μια κοινή κλίμακα.

Η συνάρτηση αρχικά υπολογίζει τις ελάχιστες και μέγιστες τιμές για τους πίνακες p1 και p2. Ειδικότερα, η ελάχιστη και η μέγιστη τιμή του p1 χρησιμοποιούνται για τον καθορισμό της κλίμακας στην οποία θα κλιμακωθούν οι πιθανότητες του p2. Εάν οι πίνακες είναι κενοί, η συνάρτηση χρησιμοποιεί προκαθορισμένες τιμές (0.0 και 1.0) για να αποτρέψει σφάλματα.

Για κάθε τιμή του πίνακα p2, η συνάρτηση υπολογίζει την κανονικοποιημένη τιμή με βάση την κλίμακα του p1. Για την κανονικοποίηση γίνεται χρήση του παρακάτω τύπου:

$$normalizedP2[i] = \frac{p2[i] - \min P2}{maxP2 - minP2} \cdot (maxP1 - minP1) + minP1$$

Τέλος, η συνάρτηση επιστρέφει τον κανονικοποιημένο πίνακα normalizedP2, ο οποίος επιτρέπει την εύκολη σύγκριση και συνδυασμό των πιθανοτήτων από τους δύο πίνακες δεδομένων.

```
private fun normalizeAccordingTo(p2: DoubleArray, p1: DoubleArray):
DoubleArray {
    val minP1 = p1.minOrNull() ?: 0.0
    val maxP1 = p1.maxOrNull() ?: 1.0
    val minP2 = p2.minOrNull() ?: 0.0
    val maxP2 = p2.maxOrNull() ?: 1.0

    val normalizedP2 = DoubleArray(p2.size)

    for (i in p2.indices) {
        normalizedP2[i] = ((p2[i] - minP2) / (maxP2 - minP2)) * (maxP1 -
```

```
minP1) + minP1
    }

    return normalizedP2
}
```

calculateMedian

Υπολογίζει τη διάμεσο από έναν πίνακα αριθμών τύπου Double. Η διάμεσος είναι ένα στατιστικό στοιχείο που αντιπροσωπεύει την κεντρική τιμή μιας διανομής. Επιστρέφει τη διάμεσο των αριθμών, η οποία είναι η τιμή που χωρίζει τη διανομή σε δύο ίσα μέρη, με το 50% των τιμών να είναι μικρότερες ή ίσες και το άλλο 50% να είναι μεγαλύτερες ή ίσες από αυτήν.

Αρχικά, η συνάρτηση ελέγχει αν ο πίνακας των αριθμών είναι άδειος. Αν είναι, προκαλεί μια εξαίρεση (IllegalArgumentException) για να διασφαλίσει ότι η είσοδος περιέχει τουλάχιστον έναν αριθμό.

Στη συνέχεια, η συνάρτηση ταξινομεί τον πίνακα numbers σε αύξουσα σειρά χρησιμοποιώντας τη μέθοδο sorted. Αυτό είναι απαραίτητο γιατί η διάμεσος υπολογίζεται με βάση την ταξινομημένη σειρά των αριθμών.

Μετά την ταξινόμηση, η συνάρτηση υπολογίζει το μέγεθος του πίνακα sortedNumbers. Αν το μέγεθος είναι άρτιος αριθμός, τότε η διάμεσος είναι ο μέσος όρος των δύο κεντρικών αριθμών. Αυτό επιτυγχάνεται λαμβάνοντας τις τιμές από τις θέσεις size / 2 - 1 και size / 2 και υπολογίζοντας το μέσο όρο τους.

Αν το μέγεθος είναι περιττός αριθμός, η διάμεσος είναι ο αριθμός που βρίσκεται στη μέση της ταξινομημένης λίστας. Αυτός ο αριθμός βρίσκεται στη θέση size / 2.

```
private fun calculateMedian(numbers: DoubleArray): Double {
    if (numbers.isEmpty()) {
        throw IllegalArgumentException("The list of numbers cannot be
empty")
    }

    val sortedNumbers = numbers.sorted()
    val size = sortedNumbers.size

    return if (size % 2 == 0) {
        // If even, average the two middle numbers
        val middle1 = sortedNumbers[size / 2 - 1]
        val middle2 = sortedNumbers[size / 2]
        (middle1 + middle2) / 2.0
    } else {
        // If odd, return the middle number
        sortedNumbers[size / 2]
    }
}
```

calculateRRIntervals

Χρησιμοποιείται για τον υπολογισμό των διαστημάτων RR από μια λίστα δεικτών κορυφών που εντοπίζονται στο σήμα PPG. Δέχεται ως παράμετρο την λίστα δεικτών κορυφών του σήματος που είχε υπολογιστεί προηγουμένως και τον πίνακα τιμών χρονικής καταγραφής από κάθε datapoint. Επιστρέφει έναν πίνακα τύπου double που περιέχει διαστήματα RR.

Αρχικά, γίνεται έλεγχος για το αν η λίστα δεικτών κορυφών είναι διαθέσιμη και έχει επαρκή μέγεθος για τον υπολογισμό των διαστημάτων RR. Έπειτα, δημιουργείται ένας πίνακας rrIntervals με μήκος ίσο με το πλήθος των δεικτών κορυφών μειωμένο κατά 1. Κάθε διάστημα RR υπολογίζεται ως η διαφορά στο χρόνο μεταξύ των δύο συνεχόμενων κορυφών. Η συνάρτηση επιστρέφει έναν πίνακα που περιέχει τα διαστήματα RR.

```
private fun calculateRRIntervals(peakIndices: List<Int?>?, time:
List<Long>): DoubleArray {
    if (peakIndices == null || peakIndices.size <= 1) {
        println("Not enough peaks to calculate RR intervals.")
        return DoubleArray(0)
    }
    val rrIntervals = DoubleArray(peakIndices.size - 1)
    for (j in 0 until peakIndices.size - 1) if (peakIndices[j + 1]!! <
time.size) rrIntervals[j] =
        (time[peakIndices[j + 1]!!] - time[peakIndices[j]!!]).toDouble()
    return rrIntervals
}
```

peakDetection

Εντοπίζει τις κορυφές (peaks) με τη χρήση ενός πίνακα πιθανοτήτων (possibility) όπου οι τιμές του εκφράζουν την πιθανότητα κάθε σημείο να είναι κορυφή. Η συνάρτηση δέχεται ως είσοδο έναν πίνακα πιθανοτήτων από Double τιμές και επιστρέφει μια λίστα από Int, που περιέχει τους δείκτες των κορυφών.

Αρχικά, η συνάρτηση δημιουργεί μια κενή λίστα peakIndices για να αποθηκεύσει τους δείκτες των κορυφών που θα εντοπιστούν. Στη συνέχεια, προσδιορίζει τη μέγιστη τιμή στον πίνακα possibility χρησιμοποιώντας τη μέθοδο maxOrNull. Κατόπιν, για κάθε στοιχείο του πίνακα, ελέγχει αν η τιμή του είναι τουλάχιστον το 50% της μέγιστης τιμής. Εάν η συνθήκη αυτή ισχύει, ο δείκτης του στοιχείου προστίθεται στη λίστα peakIndices. Τελικά, η συνάρτηση επιστρέφει τη λίστα peakIndices, η οποία περιέχει τους δείκτες των σημείων που θεωρούνται κορυφές με βάση το καθορισμένο κριτήριο. Αυτή η διαδικασία είναι σημαντική για την ανάλυση των σημάτων PPG, καθώς επιτρέπει τον εντοπισμό των καρδιακών παλμών μέσω των κορυφών, που είναι απαραίτητοι για την εκτίμηση της μεταβλητότητας του καρδιακού ρυθμού. Η χρήση ενός συγκεκριμένου κριτηρίου ανίχνευσης βοηθά στην εξάλειψη των θορύβων και των μικρών διακυμάνσεων, εξασφαλίζοντας την ακρίβεια των ανιχνευόμενων κορυφών και βελτιώνοντας την ανάλυση, ιδιαίτερα σε θορυβώδη σήματα PPG.

```
private fun peakDetection(possibility: DoubleArray): List<Int> {
    val peakIndices: MutableList<Int> = ArrayList()
    // Find the maximum value
    val maxValue = possibility.maxOrNull()
    for (i in 0 until ppgSignal.size) {
```

```
        if (possibility[i] >= 0.5 * maxValue!!) {
            peakIndices.add(i)
        }
    }
    return peakIndices
}
```

mergeConsecutivePeaks

Συγχωνεύει διαδοχικές κορυφές σε μία ενιαία κορυφή. Δέχεται ως είσοδο μια λίστα peaks που περιέχει τους δείκτες των ανιχνευμένων κορυφών και επιστρέφει μια νέα λίστα με τους δείκτες των συγχωνευμένων κορυφών.

Αρχικά, η συνάρτηση ελέγχει αν η λίστα peaks είναι κενή και, αν αυτό ισχύει, επιστρέφει μια κενή λίστα. Στη συνέχεια, δημιουργεί μια κενή λίστα mergedPeaks για να αποθηκεύσει τους δείκτες των συγχωνευμένων κορυφών και αρχικοποιεί τη μεταβλητή startPeak με την πρώτη κορυφή της λίστας peaks.

Η συνάρτηση διατρέχει τη λίστα peaks και ελέγχει αν κάθε κορυφή είναι διαδοχική με την προηγούμενη. Εάν είναι, ενημερώνει τη μεταβλητή endPeak με τη νέα κορυφή και ελέγχει αν η τιμή του σήματος PPG στη νέα κορυφή είναι μεγαλύτερη από την τιμή στη μέγιστη κορυφή (maxPeakIndex) της τρέχουσας ομάδας διαδοχικών κορυφών. Εάν δεν είναι διαδοχική, προσθέτει τον δείκτη της μέγιστης κορυφής της τρέχουσας ομάδας στη λίστα mergedPeaks, και ξεκινά μια νέα ομάδα διαδοχικών κορυφών με την τρέχουσα κορυφή ως αρχή (startPeak) και τέλος (endPeak), ενημερώνοντας το maxPeakIndex με την τιμή της τρέχουσας κορυφής.

Τελικά, η συνάρτηση επιστρέφει τη λίστα mergedPeaks, η οποία περιέχει τους δείκτες των συγχωνευμένων κορυφών. Αυτή η διαδικασία βοηθά στην εξάλειψη των πολλαπλών κορυφών που μπορεί να οφείλονται σε μικρές διακυμάνσεις ή θόρυβο στο σήμα PPG, βελτιώνοντας έτσι την ακρίβεια του εντοπισμού των καρδιακών παλμών για την εκτίμηση της μεταβλητότητας του καρδιακού ρυθμού.

```
private fun mergeConsecutivePeaks(peaks: List<Int>): List<Int> {
    if (peaks.isEmpty()) return emptyList()

    val mergedPeaks = mutableListOf<Int>()
    var startPeak = peaks[0]
    var endPeak: Int
    var maxPeakIndex = startPeak

    for (i in 1 until peaks.size) {
        if (peaks[i] == peaks[i - 1] + 1) {
            endPeak = peaks[i]
            if (ppgSignal[endPeak] > ppgSignal[maxPeakIndex]) {
                maxPeakIndex = endPeak
            }
        } else {
            // Add the index of the max value of the current group
            mergedPeaks.add(maxPeakIndex)
            startPeak = peaks[i]
            endPeak = peaks[i]
            maxPeakIndex = startPeak
```

```
        }
    }
// Add the index of the max value of the last group
        mergedPeaks.add(maxPeakIndex)
        return mergedPeaks
    }
```

filterFirstXDataPoints

Ο σκοπός της συνάρτησης είναι η απομάκρυνση ενός προκαθορισμένου αριθμού αρχικών σημείων δεδομένων από τις λίστες ppgSignal και time. Δέχεται ως εισόδους τις λίστες ppgSignal και time, καθώς και τον αριθμό των σημείων δεδομένων που θα απομακρυνθούν (numberOfDatapointsToRemove). Αρχικά, ελέγχει αν το μέγεθος των λιστών είναι μικρότερο από τον αριθμό των σημείων που πρέπει να απομακρυνθούν. Εάν κάποια από τις λίστες έχει λιγότερα στοιχεία από αυτά που πρέπει να αφαιρεθούν, εμφανίζει ένα μήνυμα σφάλματος και τερματίζει την εκτέλεση.

Αν οι λίστες περιέχουν επαρκή αριθμό στοιχείων, η συνάρτηση προχωρά στην απομάκρυνση των σημείων αυτών. Χρησιμοποιεί έναν βρόχο for για να διατρέξει τις πρώτες numberOfDatapointsToRemove θέσεις των λιστών και απομακρύνει το πρώτο στοιχείο κάθε φορά χρησιμοποιώντας τη μέθοδο removeAt(0). Αυτή η διαδικασία εκτελείται και για τις δύο λίστες, ppgSignal και time, εξασφαλίζοντας ότι τα στοιχεία που αντιστοιχούν στο ίδιο χρονικό σημείο απομακρύνονται ταυτόχρονα και από τις δύο λίστες.

Η χρήση αυτής της συνάρτησης είναι σημαντική για την προεπεξεργασία των δεδομένων PPG, απομακρύνοντας πιθανώς μη αξιόπιστα αρχικά σημεία που μπορεί να περιέχουν θόρυβο ή να επηρεάζονται από την αρχική σταθεροποίηση του σήματος. Με αυτόν τον τρόπο, τα επόμενα βήματα ανάλυσης του σήματος, όπως η ανίχνευση κορυφών και η εκτίμηση της μεταβλητότητας του καρδιακού ρυθμού, μπορούν να πραγματοποιηθούν με μεγαλύτερη ακρίβεια.

```
private fun filterFirstXDataPoints(
    ppgSignal: MutableList<Int>, time: MutableList<Long>,
numberOfDatapointsToRemove: Int
) {
    if (ppgSignal.size < numberOfDatapointsToRemove || time.size <
numberOfDatapointsToRemove) {
        println("The lists do not contain enough elements to remove
$numberOfDatapointsToRemove data points.")
        return
    }

    for (i in 0 until numberOfDatapointsToRemove) {
        ppgSignal.removeAt(0)
        time.removeAt(0)
    }
}
```

calculateHRVTriangularIndex

Υπολογίζει τον τριγωνικό δείκτη της HRV (HRVTI) από μια λίστα με διαστήματα RR (RR intervals) του καρδιακού ρυθμού. Δέχεται ως παράμετρο έναν πίνακα τύπου double, στον οποίο περιέχονται οι τιμές των RR διαστημάτων και επιστρέφει τον τριγωνικό δείκτη της HRV σε μορφή float. Αρχικά, η συνάρτηση ελέγχει αν ο πίνακας rrIntervals είναι κενός. Σε αυτήν την περίπτωση, επιστρέφεται 0 ως αποτέλεσμα και εκτυπώνεται ένα μήνυμα σφάλματος. Αν ο πίνακας rrIntervals δεν είναι κενός, ο τριγωνικός δείκτης υπολογίζεται ως εξής:

1. Υπολογισμός του μέγιστου και ελάχιστου διαστήματος RR.
2. Διαίρεση του εύρους των διαστημάτων RR σε μικρά διαστήματα (bins) με πλάτος (1/128)s, το οποίο προτείνεται από την Ευρωπαϊκή Εταιρεία Καρδιολογίας [13].
3. Υπολογισμός του ιστογράμματος των διαστημάτων RR με βάση τα bins.
4. Υπολογισμός του ύψους του ιστογράμματος.
5. Υπολογισμός του HRVTI ως το πλήθος των διαστημάτων RR διαιρεμένο με το ύψος του ιστογράμματος.

```
    private fun calculateHRVTriangularIndex(
        rrIntervals: DoubleArray
    ): Float {
        return if (rrIntervals.isNotEmpty()) {
            var iHRV = 0f
            val maxInterval = rrIntervals.maxOrNull()
            val minInterval = rrIntervals.minOrNull()
            val binWidth = (1.0 / 128.0) * 1000
            if (maxInterval != null && minInterval != null) {
                val totalBins = (maxInterval - minInterval) / binWidth
                var binRanges = mutableListOf<Double>()
                binRanges.add(minInterval)
                for (i in rrIntervals.indices) {
                    if (minInterval + i * binWidth <= maxInterval) {
                        binRanges.add(minInterval + i * binWidth)
                    }
                }
                var histogram = IntArray(totalBins.toInt())
                for (interval in rrIntervals) {
                    for (i in 0 until binRanges.size - 1) {
                        if (interval >= binRanges[i] && interval <
binRanges[i + 1]) {
                            histogram[i]++
                        }
                    }
                }
                val height = histogram.maxOrNull()
                iHRV = (rrIntervals.size / height!!).toFloat()
            }
            iHRV


        } else {
            Log.i(TAG, "rrIntervals array is empty.")
            0f
        }

    }
```

calculateStressLevel

Έχει σχεδιαστεί για να εκτιμήσει το επίπεδο ψυχολογικού στρες μέσω της ανάλυσης τριών βασικών δεικτών καρδιοαγγειακής μεταβλητότητας: AMo, pNN50 και MedSD. Ο σχεδιασμός της συνάρτησης βασίζεται στις αρχές που περιγράφονται στη σχετική βιβλιογραφία [16] για την αξιολόγηση του φυσιολογικού στρες, χρησιμοποιώντας την ανάλυση μεταβλητότητας του καρδιακού ρυθμού.

Δέχεται ως είσοδο τις τρέχουσες τιμές των δεικτών AMo, pNN50 και MedSD και τις συγκρίνει με προκαθορισμένες βασικές τιμές, οι οποίες έχουν προκύψει από ανάλυση των τιμών των πρωινών μετρήσεων ενός δείγματος ατόμων που συμμετείχαν στην έρευνα για τους δείκτες AMo και MedSD, αλλά και στοιχεία από τη βάση δεδομένων της EliteHRV για τον δείκτη pNN50 [34]. Οι βασικές τιμές αντιπροσωπεύουν τον μέσο όρο των τιμών για το δείγμα που μετρήθηκε και συγκρίνονται με τις τρέχουσες τιμές για να εκτιμηθεί το επίπεδο στρες.

Για την υπολογισμένη τιμή του AMo, εάν η τρέχουσα τιμή είναι μεγαλύτερη από την βασική τιμή, το επίπεδο του στρες υπολογίζεται αυξάνοντας την τιμή του, προσθέτοντας την απόκλιση στο 0.5, διαφορετικά μειώνεται, αφαιρώντας την από το 0.5. Η τιμή του στρες περιορίζεται μεταξύ 0 και 1, ώστε να διασφαλιστεί ότι το αποτέλεσμα είναι έγκυρο. Παρόμοια, η τιμή του pNN50 επηρεάζει το επίπεδο στρες αντίστροφα, όπου μια αύξηση της τιμής δείχνει μείωση του στρες και μια μείωση δείχνει αύξηση του στρες. Για τον δείκτη MedSD, η μέθοδος είναι αντίστοιχη με αυτήν του pNN50, με τις τιμές του στρες να υπολογίζονται με βάση την απόκλιση από τη βασική τιμή.

Η συνολική εκτίμηση του επιπέδου στρες υπολογίζεται ως ο μέσος όρος των επιμέρους τιμών των τριών δεικτών, ο οποίος εκφράζεται σε ποσοστό.

```
private fun calculateStressLevel(AMo: Double, pNN50: Double, MedSD:
Double): Float {
    val AMo_baseline = 30.0     // Median AMo baseline
    val pNN50_baseline = 20.0    // Median pNN50 baseline
    val MedSD_baseline = 50.0    // Median MedSD baseline

    // Calculate stress for each metric based on how they deviate from the
baseline
    val AMo_stress = if (AMo > AMo_baseline) {
        (0.5 + ((AMo - AMo_baseline) / (2 * AMo_baseline))).coerceIn(0.0,
1.0)
    } else {
        (0.5 - ((AMo_baseline - AMo) / (2 * AMo_baseline))).coerceIn(0.0,
1.0)
    }

    val pNN50_stress = if (pNN50 > pNN50_baseline) {
        (0.5 - ((pNN50 - pNN50_baseline) / (2 *
pNN50_baseline))).coerceIn(0.0, 1.0)
    } else {
        (0.5 + ((pNN50_baseline - pNN50) / (2 *
pNN50_baseline))).coerceIn(0.0, 1.0)
```

```
    }

    val MedSD_stress = if (MedSD > MedSD_baseline) {
        (0.5 - ((MedSD - MedSD_baseline) / (2 *
MedSD_baseline))).coerceIn(0.0, 1.0)
    } else {
        (0.5 + ((MedSD_baseline - MedSD) / (2 *
MedSD_baseline))).coerceIn(0.0, 1.0)
    }

    // Calculate overall stress level as an average of these stress values
    return (((AMo_stress + pNN50_stress + MedSD_stress) / 3.0) *
100.0).toFloat()
}
```

reset

Χρησιμοποιείται για την επαναφορά των μεταβλητών στις αρχικές τους τιμές για τον επόμενο κύκλο μέτρησης.

```
private fun reset() {
    // Reset variables for the next batch
    ppgSignal.clear()
    time.clear()
    startTimeMillis = 0L
    isHRVCalculated = false
    isProcessingData = false
    errorOccurred = false
    hrvIndex = 0f
    RMSSD = 0.0
    SDNN = 0.0
    pNN50 = 0.0
    AMo = 0.0
    MedSD = 0.0
    stressValue = "Stress Level"
    stressLevel = 0f
    barColor = Color.DarkGray
    dataCollectionProgress = 0
    isCollectingData = false
    collectionDurationMillis = 60000
}
```

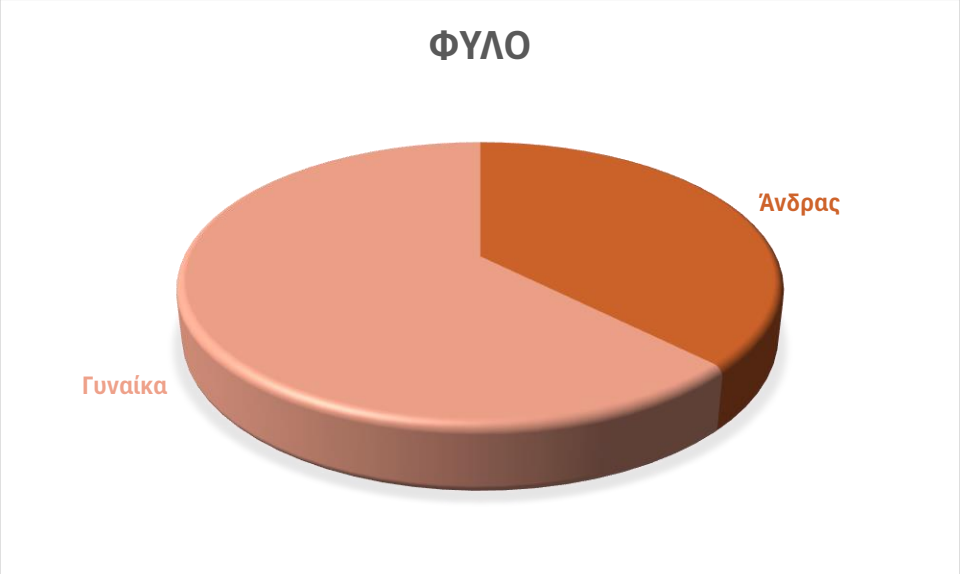
Κεφάλαιο 6: Αποτελέσματα

6.1 Εισαγωγή

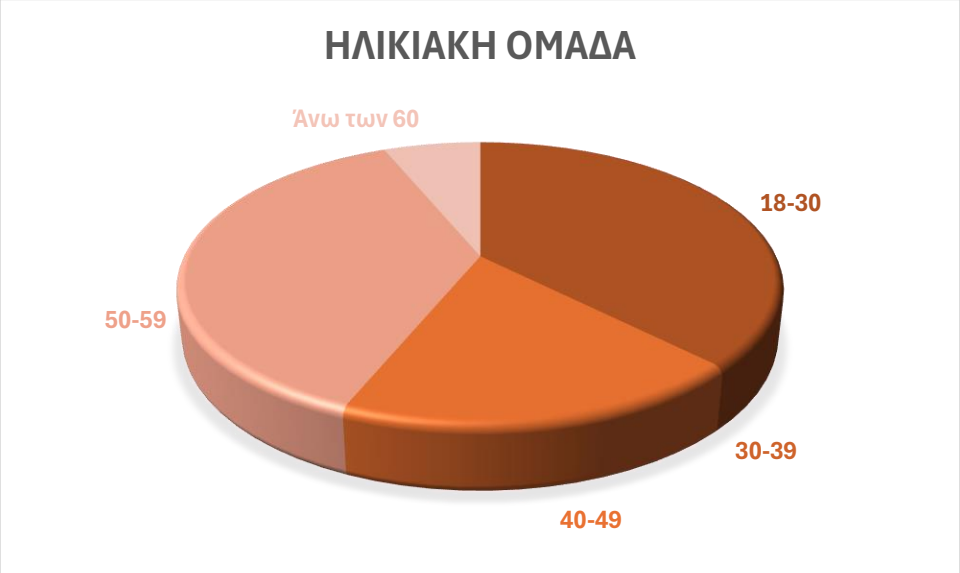
Ο σκοπός αυτής της ενότητας είναι να παρουσιαστούν τα ευρήματα από τις έξι συνεδρίες (τρεις των 60s και τρεις των 120s) καταγραφής της καρδιακής μεταβλητότητας (HRV) ανά άτομο σε μία μέρα που πραγματοποιήθηκαν με τη χρήση της εφαρμογής που αναπτύξαμε για έξυπνο ρολόι και μιας ευρέως χρησιμοποιούμενης εφαρμογής για τον ίδιο σκοπό. Η ανάλυση επικεντρώνεται στη σύγκριση της απόδοσης και της ακρίβειας των δύο εφαρμογών στη μέτρηση της HRV και των επιπέδων άγχους κατά τη διάρκεια της μέρας, αλλά και στη σταθερότητα των μετρήσεων μεταξύ δύο συνεχόμενων μετρήσεων διαφορετικής διάρκειας. Επιπλέον, γίνεται σύγκριση μεταξύ των αποτελεσμάτων αυτοαξιολόγησης τους στρες μέσω του PSS Score με τους υπολογισμούς της εφαρμογής μας.

6.2 Λεπτομέρειες Συμμετεχόντων

Στην έρευνα συμμετείχαν 16 άτομα, τα οποία συμμετείχαν σε 6 συνεδρίες καταγραφής καρδιακού ρυθμού, 2 μετρήσεις 3 φορές τη μέρα. Οι συμμετέχοντες επιλέχθηκαν με βάση κριτήρια που εξασφάλιζαν την ποικιλότητα και τη δυνατότητα αξιοπιστίας των αποτελεσμάτων. Στα παρακάτω γραφήματα παρουσιάζονται τα συνολικά χαρακτηριστικά των συμμετεχόντων, σχετικά με το φύλο, την ηλικιακή ομάδα, τις ώρες ύπνου, το επίπεδο δραστηριότητας και την κατανάλωση αλκοόλ ή καφεΐνης.



ΣΧΗΜΑ 3- ΚΑΤΑΝΟΜΗ ΦΥΛΟΥ ΣΥΜΜΕΤΕΧΟΝΤΩΝ



ΣΧΗΜΑ 4- ΚΑΤΑΝΟΜΗ ΗΛΙΚΙΑΚΩΝ ΟΜΑΔΩΝ ΣΥΜΜΕΤΕΧΟΝΤΩΝ

ΣΧΗΜΑ 5- ΚΑΤΑΝΟΜΗ ΩΡΩΝ ΥΠΝΟΥ ΣΥΜΜΕΤΕΧΟΝΤΩΝ

ΣΧΗΜΑ 6- ΚΑΤΑΝΟΜΗ ΕΠΙΠΕΔΟΥ ΑΣΚΗΣΗΣ ΣΥΜΜΕΤΕΧΟΝΤΩΝ

ΣΧΗΜΑ 7- ΚΑΤΑΝΟΜΗ ΚΑΤΑΝΑΛΩΣΗΣ ΚΑΦΕΪΝΗΣ Η ΑΛΚΟΟΛ ΣΥΜΜΕΤΕΧΟΝΤΩΝ

6.3 Επισκόπηση Συλλογής Δεδομένων

6.3.1 Στάδιο Προετοιμασίας

Κατά το στάδιο της προετοιμασίας, οι συμμετέχοντες εφοδιάστηκαν με δύο έξυπνα ρολόγια Samsung Watch 5. Το ένα ρολόι είχε εγκατεστημένη την εφαρμογή για τον υπολογισμό του άγχους και της HRV που αναπτύξαμε, ενώ το άλλο ρολόι είχε μια κοινώς χρησιμοποιούμενη εφαρμογή. Οι συσκευές φορέθηκαν ταυτόχρονα κατά τη διάρκεια των συνεδριών, επιτρέποντας την παράλληλη καταγραφή δεδομένων από τις δύο εφαρμογές. Επιπλέον, πραγματοποιήθηκε εναλλαγή των ρολογιών μεταξύ των χεριών σε κάθε συνεδρία, με στόχο την εξασφάλιση της αμεροληψίας στα αποτελέσματα.

6.3.2 Διαδικασία μέτρησης

Για τη διεξαγωγή της μελέτης, πραγματοποιήθηκαν συνολικά 6 μετρήσεις ανά ημέρα για όλους τους συμμετέχοντες. Συγκεκριμένα, έγιναν 2 μετρήσεις το πρωί, 2 το μεσημέρι και 2 το βράδυ. Οι πρωινές μετρήσεις χρησιμοποιήθηκαν για την εξαγωγή της βασικής τιμής (baseline). Συγκεκριμένα, υπολογίσαμε την διάμεσο των τιμών των μετρήσεων των 60s και των 120s για τους δείκτες Amo και MedSD αντίστοιχα και ως baseline ορίσαμε για τον κάθε δείκτη τον μέσο όρο μεταξύ των τιμών για τις δύο διαφορετικές χρονικές διάρκειες.

Οι συμμετέχοντες φορούσαν 2 ρολόγια, ένα στο δεξί και ένα στο αριστερό χέρι. Το ένα ρολόι ήταν εξοπλισμένο με την εφαρμογή που αναπτύξαμε για τη μελέτη, ενώ το άλλο με μία ευρέως χρησιμοποιούμενη εμπορική εφαρμογή. Εναλλάσσαμε τα ρολόγια στα χέρια των συμμετεχόντων, ώστε να εξασφαλιστεί η αμεροληψία στα αποτελέσματα.

Για κάθε μέτρηση πραγματοποιούνταν μία αρχική καταγραφή διάρκειας 60 δευτερολέπτων, όπου οι χρήστες ήταν καθιστοί με σταθερά τα χέρια τους, σε κατάσταση ηρεμίας. Έπειτα, ακολουθούσε η ίδια διαδικασία για τη μέτρηση των 120 δευτερολέπτων. Η χρονική διάρκεια μέτρησης της εμπορικής εφαρμογής ήταν σταθερή, περίπου στα 90 δευτερόλεπτα και για τις 2 μετρήσεις.

6.3.3 Καταγραφή Δεδομένων

Μετά την ολοκλήρωση κάθε δραστηριότητας, πραγματοποιήθηκε αυτόματη καταγραφή του καρδιακού ρυθμού (PPG σήμα) μέσω των δύο εφαρμογών. Οι συμμετέχοντες συμπλήρωσαν επίσης ένα σύντομο ερωτηματολόγιο (βλ. παράρτημα 1) για να εκτιμηθούν τα υποκειμενικά επίπεδα άγχους τους. Τα δεδομένα από τις εφαρμογές και τα ερωτηματολόγια συλλέχθηκαν για περαιτέρω ανάλυση.

Τα δεδομένα καταγραφής από την εφαρμογή που αναπτύξαμε αποθηκεύονταν αυτόματα (με τη χρήση μίας συνάρτησης που υλοποιήσαμε σε Kotlin) σε αρχείο στο ρολόι σε μορφή κατάλληλη για την εισαγωγή σε Matlab και μεταφέρονταν στον υπολογιστή για περαιτέρω ανάλυση. Τα δεδομένα από την εμπορική εφαρμογή αποθηκεύονταν στο αντίστοιχο app σε android κινητό, το οποίο ήταν συνδεδεμένο με το αντίστοιχο smartwatch.

6.4 Σύγκριση Μετρήσεων 60s και 120s

6.4.1 Περιγραφή της Μεθόδου

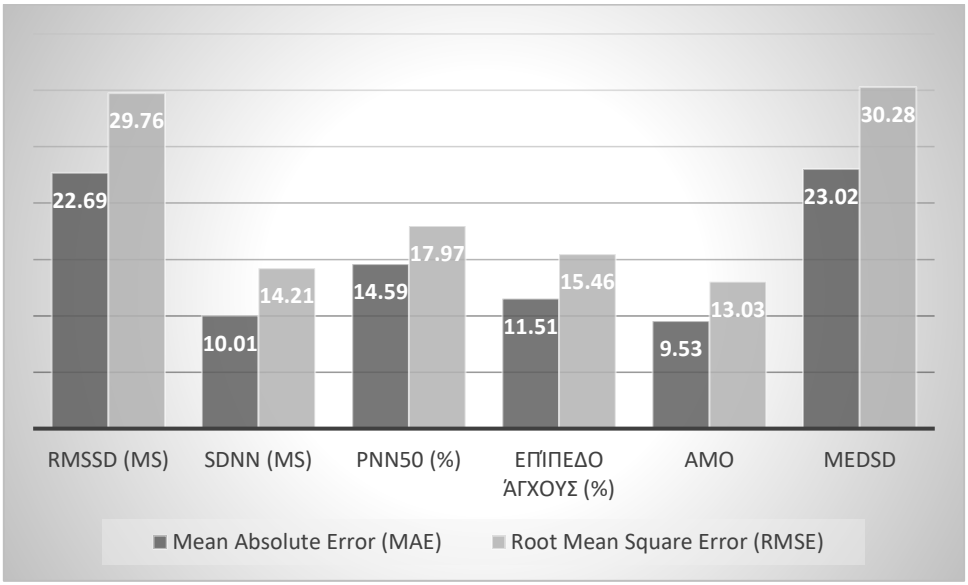
Η σύγκριση των μετρήσεων των 60 και 120 δευτερολέπτων πραγματοποιήθηκε για να εξετάσουμε τη διαφορά μεταξύ σύντομων και μεγαλύτερων χρονικών διαστημάτων καταγραφής δεδομένων. Συλλέχθηκαν μετρήσεις κατά τη διάρκεια της ημέρας από κάθε συμμετέχοντα σε διαφορετικές χρονικές στιγμές, και πραγματοποιήθηκαν 2 μετρήσεις το πρωί, 2 το μεσημέρι και 2 το βράδυ. Γενικότερα, οι μετρήσεις των 60 δευτερολέπτων παρέχουν γρήγορη ανατροφοδότηση για την κατάσταση του ατόμου, ενώ οι μετρήσεις των 120 δευτερολέπτων προσφέρουν μια πιο λεπτομερή εικόνα.

6.4.2 Ανάλυση Αποτελεσμάτων

Στους πίνακες και τα διαγράμματα που ακολουθούν, παρατηρούμε τις τιμές των αποκλίσεων των δύο συνεχόμενων μετρήσεων διαφορετικής διάρκειας. Πιο συγκεκριμένα, παρουσιάζεται η μέση απόλυτη τιμή του σφάλματος «Mean Absolute Error (MAE)» και τετραγωνικό μέσο σφάλμα «Root Mean Square Error (RMSE)» ανά δείκτη, σε έναν συγκεντρωτικό πίνακα, ο οποίος έχει προκύψει ως μέσος όρος των αντίστοιχων πινάκων από κάθε συνεδρία (πρωί, μεσημέρι και βράδυ).

Πίνακας 3 - Συγκεντρωτικός πίνακας αποκλίσεων τιμών ανά δείκτη HRV μεταξύ συνεχόμενων μετρήσεων διαφορετικής διάρκειας (60s και 120s)

	RMSSD (ms)	SDNN (ms)	pNN50 (%)	Επίπεδο Άγχους (%)	AMo	MedSD
Mean Absolute Error (MAE)	**22.69**	**10.01**	**14.59**	**11.51**	**9.53**	**23.02**
Root Mean Square Error (RMSE)	**29.76**	**14.21**	**17.97**	**15.46**	**13.03**	**30.28**



ΣΧΗΜΑ 8 - ΔΙΑΓΡΑΜΜΑ ΑΠΟΚΛΙΣΕΩΝ ΤΙΜΩΝ ΑΝΑ ΔΕΙΚΤΗ HRV ΜΕΤΑΞΥ ΣΥΝΕΧΟΜΕΝΩΝ ΜΕΤΡΗΣΕΩΝ ΔΙΑΦΟΡΕΤΙΚΗΣ ΔΙΑΡΚΕΙΑΣ (60S ΚΑΙ 120S)

Από τη σύγκριση, διαπιστώθηκε ότι οι μετρήσεις των 60 και 120 δευτερολέπτων έχουν πολύ μικρή απόκλιση μεταξύ τους, γεγονός που υποδηλώνει ότι και τα δύο χρονικά διαστήματα είναι αρκετά κοντά ως προς την ακρίβεια των αποτελεσμάτων. Αυτό υποδηλώνει ότι οι μετρήσεις των 60 δευτερολέπτων μπορούν να χρησιμοποιηθούν με σχεδόν την ίδια ακρίβεια με αυτές των 120 δευτερολέπτων, προσφέροντας μια ταχύτερη εναλλακτική χωρίς να θυσιάζεται η αξιοπιστία των δεδομένων.

6.5 Σύγκριση μεταξύ Εφαρμογών

6.5.1 Περιγραφή της Μεθόδου

ΥΠΟΛΟΓΙΣΜΟΣ ΜΕΤΑΒΛΗΤΟΤΗΤΑΣ ΚΑΡΔΙΑΚΟΥ ΡΥΘΜΟΥ ΣΕ ΠΡΑΓΜΑΤΙΚΟ ΧΡΟΝΟ ΜΕΣΩ ΦΩΤΟΠΛΗΘΥΣΜΟΓΡΑΦΙΑΣ ΚΟΚΚΙΝΗΣ ΑΚΤΙΝΟΒΟΛΙΑΣ

Η δεύτερη σύγκριση πραγματοποιήθηκε μεταξύ της παρούσας εφαρμογής και μιας ευρέως χρησιμοποιούμενης εφαρμογής υπολογισμού της HRV, η οποία επίσης συνδέεται με το ίδιο smartwatch και χρησιμοποιεί τους αισθητήρες για την καταγραφή σήματος φωτοπληθυσμογραφίας. Ο στόχος αυτής της σύγκρισης ήταν να ελέγξουμε την ακρίβεια της εφαρμογής που αναπτύξαμε σε σχέση με μια ήδη υπάρχουσα λύση. Οι μετρήσεις συλλέχθηκαν ταυτόχρονα από τους συμμετέχοντες φορώντας δύο ρολόγια, ένα σε κάθε χέρι, που έτρεχαν τις δύο εφαρμογές αντίστοιχα.

6.5.2 Ανάλυση Αποτελεσμάτων

Τα αποτελέσματα αποδεικνύουν ότι η προς μελέτη εφαρμογή παρείχε συγκρίσιμες μετρήσεις με την υπάρχουσα εφαρμογή, με μικρές αποκλίσεις που ήταν στατιστικά ασήμαντες. Οι μέσες τιμές των MAE και RMSE έδειξαν ότι οι δύο εφαρμογές ήταν πολύ κοντά σε απόδοση. Μεγαλύτερη απόκλιση παρατηρείται μόνο στον δείκτη RMSSD.

Πίνακας 4 - Συγκεντρωτικός πίνακας αποκλίσεων τιμών ανά δείκτη HRV μεταξύ της εφαρμογής μας και μιας παρόμοιας ευρέως χρησιμοποιούμενη

	RMSSD (ms)	SDNN (ms)	pNN50 (%)	Επίπεδο Άγχους (%)
Mean Absolute Error (MAE)	**45.17**	**21.65**	**29.73**	**22.03**
Root Mean Square Error (RMSE)	**51.74**	**25.8**	**34.05**	**26.89**

ΣΧΗΜΑ 9 - ΔΙΑΓΡΑΜΜΑ ΑΠΟΚΛΙΣΕΩΝ ΤΙΜΩΝ ΑΝΑ ΔΕΙΚΤΗ HRV ΜΕΤΑΞΥ ΤΗΣ ΕΦΑΡΜΟΓΗΣ ΜΑΣ ΚΑΙ ΜΙΑΣ ΠΑΡΟΜΟΙΑΣ ΕΥΡΕΩΣ ΧΡΗΣΙΜΟΠΟΙΟΥΜΕΝΗΣ

Τα αποτελέσματα είναι ικανοποιητικά από άποψη ακρίβειας, λαμβάνοντας υπόψιν ότι οι αποκλίσεις που παρατηρήθηκαν πιθανώς οφείλονται στους εξής παράγοντες:

- Πρώτον, η διάρκεια των μετρήσεων διέφερε μεταξύ των δύο εφαρμογών: η εφαρμογή προς αξιολόγηση πραγματοποιούσε μετρήσεις 60 και 120 δευτερολέπτων, ενώ η άλλη περίπου 90 δευτερολέπτων και στις δύο περιπτώσεις. Η διαφορετική διάρκεια ενδέχεται να επηρεάζει την ακρίβεια και τη συνέπεια των αποτελεσμάτων.

- Επιπλέον, στην άλλη εφαρμογή χρησιμοποιήθηκε ένας λογαριασμός για όλους τους συμμετέχοντες, γεγονός που θα μπορούσε να επηρεάζει τις τιμές των μετρήσεων, καθώς αναφέρεται ότι τα αποτελέσματα προσαρμόζονται με βάση προηγούμενες μετρήσεις. Αυτό ενδέχεται να οδηγεί σε προσαρμογές στα δεδομένα που δεν αντικατοπτρίζουν πλήρως τη φυσιολογική κατάσταση κάθε συμμετέχοντα τη δεδομένη στιγμή της μέτρησης.

6.6 Σύγκριση με PSS Score

6.6.1 Περιγραφή της Μεθόδου

Για να εξετάσουμε τη συσχέτιση μεταξύ των μετρήσεων μέσω της εφαρμογής που δημιουργήσαμε σε σχέση με το αντιληπτό στρες του κάθε χρήστη, πραγματοποιήσαμε σύγκριση των επιπέδων άγχους που υπολογίστηκαν με τα αποτελέσματα του PSS Score (Perceived Stress Scale) για κάθε συμμετέχοντα. Η σύγκριση αυτή μας επιτρέπει να διερευνήσουμε αν οι μετρήσεις του άγχους, όπως υπολογίζονται μέσω της εφαρμογής, συσχετίζονται και σε τι βαθμό με την αντίληψη κάθε ατόμου για το επίπεδο του στρες του. Για τον υπολογισμό του PSS Score, έγινε χρήση του αντίστοιχου ερωτηματολογίου (βλ. παράρτημα 1), όπου οι συμμετέχοντες έπρεπε να απαντήσουν με βάση μια κλίμακα (συνήθως από 0 έως 4) σε 10 ερωτήσεις, όπου το "0" αντιπροσωπεύει το "Ποτέ" και το "4" το "Πολύ Συχνά". Ο συνολικός βαθμός που προκύπτει από τις απαντήσεις αντικατοπτρίζει το αντιληπτό επίπεδο στρες του ατόμου, με υψηλότερες τιμές να υποδεικνύουν μεγαλύτερο αντιληπτό στρες.

6.6.2 Ανάλυση Αποτελεσμάτων

Τα αποτελέσματα της ανάλυσης δείχνουν μια μέτρια θετική συσχέτιση μεταξύ των επιπέδων άγχους που υπολογίστηκαν μέσω της εφαρμογής και του PSS Score, όπως φαίνεται από τα στατιστικά στοιχεία.

Πίνακας 5 - Συγκεντρωτικός πίνακας αποκλίσεων τιμών άγχους μεταξύ της εφαρμογής μας και του PSS Score

	Επίπεδο Άγχους (%) με PSS_10 Score
Mean Absolute Error (MAE)	**19.53**
Root Mean Square Error (RMSE)	**23.56**

Η θετική αυτή συσχέτιση υποδηλώνει ότι η εφαρμογή μας είναι σε θέση να αποτυπώνει με ικανοποιητική ακρίβεια το επίπεδο του άγχους ενός ατόμου, το οποίο αντανακλάται και στην

αντίληψη από τους ίδιους τους συμμετέχοντες, όπως καταγράφεται μέσω του PSS Score. Ωστόσο, είναι σημαντικό να επισημανθεί ότι το PSS Score προσφέρει μια γενική εκτίμηση του αντιληπτού άγχους σε βάθος χρόνου, καταγράφοντας το πώς αισθάνεται το άτομο για συγκεκριμένα γεγονότα ή καταστάσεις στη ζωή του. Αυτό σημαίνει ότι συνήθως δεν αντικατοπτρίζει το στιγμιαίο επίπεδο στρες που μπορεί να υπολογίζει η εφαρμογή κατά τη διάρκεια της μέτρησης, καθώς επικεντρώνεται σε μακροπρόθεσμες ψυχολογικές αντιδράσεις.

Επιπρόσθετα, πρέπει να ληφθεί υπόψιν ότι ορισμένοι συμμετέχοντες ενδέχεται να μην αντιλαμβάνονται τα φυσιολογικά σημάδια του στρες, ιδίως αν έχουν προσαρμοστεί σε μια συνεχόμενη κατάσταση έντασης ή πίεσης. Αυτό μπορεί να οδηγήσει σε διαφορές μεταξύ της υποκειμενικής αντίληψης του στρες και των αποτελεσμάτων που υπολογίζονται μέσω των βιομετρικών δεικτών της εφαρμογής.

Κεφάλαιο 7: Επίλογος

7.1 Συμπεράσματα

Η συγκεκριμένη διπλωματική εργασία είχε ως κεντρικό άξονα τον υπολογισμό της μεταβλητότητας του καρδιακού ρυθμού (HRV) σε πραγματικό χρόνο μέσω φωτοπληθυσμογραφίας (PPG), αξιοποιώντας αισθητήρες κόκκινης ακτινοβολίας και στη διερεύνηση της σχέσης της HRV με την ψυχική υγεία, ιδιαίτερα σε σχέση με τα επίπεδα άγχους. Κύριος στόχος ήταν η ανάπτυξη και η αξιολόγηση μιας εφαρμογής σε έξυπνο ρολόι (smartwatch), με σκοπό τον υπολογισμό των δεικτών της HRV και του στιγμιαίου επιπέδου άγχους ενός ατόμου με ακρίβεια και ταχύτητα, παρέχοντας μία εναλλακτική μέθοδο καταγραφής σε σχέση με τις παραδοσιακές μεθόδους.

Η μεθοδολογία που ακολουθήθηκε περιλάμβανε την ανάλυση σημάτων PPG και την εφαρμογή κατάλληλων αλγορίθμων για την εξαγωγή δεικτών όπως το RMSSD (Root Mean Square of Successive Differences), το SDNN (Standard Deviation of NN intervals) και το pNN50 (Percentage of successive NN intervals differing by more than 50 ms). Η επιλογή αυτών των δεικτών βασίστηκε στη βιβλιογραφία, η οποία έχει καταδείξει τη σημασία τους για την ακριβή αξιολόγηση της καρδιακής λειτουργίας και της ψυχικής κατάστασης. Το τεχνικό σκέλος της εφαρμογής υλοποιήθηκε με τη χρήση του Samsung Galaxy Watch 5 και του Samsung Privileged Health SDK, ενώ η ανάπτυξη του κώδικα έγινε μέσω του Android Studio χρησιμοποιώντας τη γλώσσα προγραμματισμού Kotlin.

Η έρευνα απέδειξε ότι η διάρκεια μέτρησης των 60 δευτερολέπτων παρέχει αξιόπιστα αποτελέσματα συγκρίσιμα με αυτά των 120 δευτερολέπτων, καθιστώντας την μέθοδο των σύντομων μετρήσεων ιδιαίτερα πρακτική για καθημερινή χρήση. Επιπλέον, η εφαρμογή παρουσίασε υψηλά επίπεδα ακρίβειας συγκριτικά με άλλη υπάρχουσα εγκεκριμένη εφαρμογή υπολογισμού δεικτών HRV, καθώς οι αποκλίσεις που παρατηρήθηκαν μεταξύ των αποτελεσμάτων ήταν στατιστικά ασήμαντες.

Η σύγκριση των αποτελεσμάτων του στιγμιαίου επιπέδου άγχους που υπολογίστηκε μέσω της εφαρμογής σε σχέση με τον Δείκτη Αντίληψης Άγχους (PSS Score) αποκάλυψε μία σημαντική συσχέτιση, επιβεβαιώνοντας τη δυνατότητα της εφαρμογής να εκτιμήσει τα επίπεδα άγχους σε πραγματικό χρόνο. Παρόλο που το PSS αντανακλά μακροπρόθεσμες ψυχολογικές καταστάσεις, ενώ οι δείκτες της HRV αποτυπώνουν στιγμιαία φυσιολογικά δεδομένα, η συσχέτιση μεταξύ τους αποδεικνύει τη χρησιμότητα της HRV ως δείκτη ψυχολογικής ευεξίας.

Συνοψίζοντας, τα αποτελέσματα της έρευνας επιβεβαιώνουν ότι η εφαρμογή που αναπτύχθηκε είναι ένα χρήσιμο εργαλείο για τη μέτρηση των δεικτών της HRV και την εκτίμηση του άγχους σε πραγματικό χρόνο, παρέχοντας μια πρακτική και μη επεμβατική λύση για την καθημερινή παρακολούθηση της ψυχικής και κατ' επέκταση σωματικής υγείας ενός ατόμου.

7.2 Μελλοντικές Επεκτάσεις

Στο πλαίσιο της μελλοντικής εξέλιξης της παρούσας εργασίας, υπάρχουν αρκετές κατευθύνσεις που μπορούν να ενισχύσουν τη λειτουργικότητα και τη συνεισφορά της εφαρμογής. Πρώτον, θα ήταν επωφελής η διεύρυνση του δείγματος των συμμετεχόντων, καθώς μία μελέτη με μεγαλύτερη και πιο ποικιλόμορφη πληθυσμιακή βάση θα προσέφερε βαθύτερη κατανόηση των διαφορών στην σχέση της HRV με το άγχος, ανάλογη με παράγοντες όπως η ηλικία, το φύλο και ο τρόπος ζωής. Έτσι, τα αποτελέσματα θα μπορούσαν να γενικευθούν σε ένα ευρύτερο φάσμα χρηστών.

Επιπλέον, η εφαρμογή θα μπορούσε να επεκταθεί με την ενσωμάτωση περισσότερων δεικτών HRV και τη χρήση πιο εξελιγμένων αλγορίθμων ανάλυσης, όπως μηχανικής μάθησης, που θα βελτιώσουν την ακρίβεια της ανάλυσης και την πρόβλεψη του άγχους σε πραγματικό χρόνο. Παράλληλα, θα ήταν χρήσιμη η δυνατότητα παρακολούθησης μακροπρόθεσμων μεταβολών, δίνοντας στους χρήστες τη δυνατότητα να κατανοούν πώς οι αλλαγές στον τρόπο ζωής ή το περιβάλλον τους επηρεάζουν την υγεία τους.

Η εξέλιξη της εφαρμογής θα μπορούσε να συμπεριλάβει την προσαρμογή της σε άλλες πλατφόρμες και συσκευές, αυξάνοντας έτσι τη διαθεσιμότητα της σε μεγαλύτερο κοινό. Τέλος, η σύνδεση με πλατφόρμες τηλεϊατρικής θα μπορούσε να προσφέρει τη δυνατότητα στους χρήστες να διαμοιράζονται τα δεδομένα τους με επαγγελματίες υγείας, διευκολύνοντας έτσι τη διάγνωση και παρακολούθηση ψυχολογικών καταστάσεων όπως το άγχος ή η κατάθλιψη.

Συνολικά, η εργασία αυτή συνεισφέρει σημαντικά στη διεπιστημονική περιοχή της τεχνολογίας και της υγείας, προσφέροντας μία νέα, πρακτική προσέγγιση για τη μέτρηση της HRV και τη διαχείριση του άγχους.

Παραρτήματα

Παράρτημα 1: Ερωτήσεις Αντίληψης Άγχους

Το παρόν παράρτημα περιλαμβάνει το ερωτηματολόγιο που χρησιμοποιήθηκε για την αξιολόγηση της αντίληψης άγχους των συμμετεχόντων κατά τη διάρκεια της έρευνας. Οι ερωτήσεις σχεδιάστηκαν για να διερευνήσουν τα συναισθήματα και τις σκέψεις των συμμετεχόντων κατά τη διάρκεια του τελευταίου μήνα. Σε κάθε ερώτηση, οι συμμετέχοντες κλήθηκαν να υποδείξουν πόσο συχνά αισθάνθηκαν ή σκέφτηκαν με έναν συγκεκριμένο τρόπο, κυκλώνοντας την κατάλληλη απάντηση σε μια κλίμακα από 0 (Ποτέ) έως 4 (Πολύ Συχνά).

Το ερωτηματολόγιο βασίζεται στην "Κλίμακα Αντίληψης Άγχους" (Perceived Stress Scale - PSS), η οποία αναπτύχθηκε και επικυρώθηκε από τους Cohen, Kamarck και Mermelstein (1983) [27] [28] και αποτελεί ένα διεθνώς αναγνωρισμένο εργαλείο για τη μέτρηση του άγχους. Το ερωτηματολόγιο έχει μεταφραστεί στα ελληνικά για διευκόλυνση των συμμετεχόντων.

ΕΡΩΤΗΣΕΙΣ ΑΝΤΙΛΗΨΗΣ ΑΓΧΟΥΣ

Οι ερωτήσεις σε αυτό το ερωτηματολόγιο αφορούν τα συναισθήματα και τις σκέψεις σας κατά τη διάρκεια του τελευταίου μήνα. **Σε κάθε περίπτωση, θα σας ζητηθεί να υποδείξετε, *κυκλώνοντας*, πόσο συχνά αισθανθήκατε ή σκεφτήκατε με έναν συγκεκριμένο τρόπο.**

Ονοματεπώνυμο _____

Ημερομηνία _____

Ηλικία _____ Φύλο *(Κυκλώστε)*: **Α** **Γ** Άλλο _____

0 = Ποτέ **1 = Σχεδόν Ποτέ** **2 = Μερικές Φορές** **3 = Αρκετά Συχνά** **4 = Πολύ Συχνά**

1.	Τον τελευταίο μήνα, πόσο συχνά ήσασταν αναστατωμένοι εξαιτίας κάποιου απροσδόκητου γεγονότος;	0	1	2	3	4
2.	Τον τελευταίο μήνα, πόσο συχνά αισθανθήκατε ότι δεν μπορούσατε να ελέγξετε τα σημαντικά πράγματα στη ζωή σας;	0	1	2	3	4
3.	Τον τελευταίο μήνα, πόσο συχνά αισθανθήκατε νευρικοί και "αγχωμένοι";	0	1	2	3	4
4.	Τον τελευταίο μήνα, πόσο συχνά αισθανθήκατε σίγουροι για την ικανότητά σας να αντιμετωπίσετε τα προσωπικά σας προβλήματα;	0	1	2	3	4
5.	Τον τελευταίο μήνα, πόσο συχνά αισθανθήκατε ότι τα πράγματα πήγαιναν όπως τα θέλατε;	0	1	2	3	4
6.	Τον τελευταίο μήνα, πόσο συχνά βρήκατε ότι δεν μπορούσατε να ανταπεξέλθετε σε όλα όσα έπρεπε να κάνετε;	0	1	2	3	4

7.	Τον τελευταίο μήνα, πόσο συχνά ήσασταν σε θέση να ελέγξετε τις ενοχλήσεις στη ζωή σας;	0	1	2	3	4
8.	Τον τελευταίο μήνα, πόσο συχνά αισθανθήκατε ότι όλα ήταν υπό έλεγχο;	0	1	2	3	4
9.	Τον τελευταίο μήνα, πόσο συχνά αισθανθήκατε θυμωμένοι λόγω πραγμάτων που ήταν εκτός του ελέγχου σας;	0	1	2	3	4
10.	Τον τελευταίο μήνα, πόσο συχνά αισθανθήκατε ότι οι δυσκολίες συσσωρεύονταν τόσο πολύ που δεν μπορούσατε να τις ξεπεράσετε;	0	1	2	3	4

Βιβλιογραφία

[1] C. Taylor, «Depression, heart rate related variables and cardiovascular disease.,» *Int. J. Psychophysiol.*, τόμ. 78, αρ. 1, pp. 80-88, 2010.

[2] R. Tiwari, R. Kumar, S. Malik, T. Raj και P. Kumar, «Analysis of Heart Rate Variability and Implication of Different Factors on Heart Rate Variability.,» *Curr Cardiol Rev.,* 2021.

[3] A. Alqaraawi, A. Alwosheel και A. Alasaad, «Heart rate variability estimation in photoplethysmography signals using Bayesian learning approach.,» *Healthc Technol Lett.,* 2016.

[4] K. Latham, «biology dictionary,» 20 February 2021. [Ηλεκτρονικό]. Available: https://biologydictionary.net/sympathetic-vs-parasympathetic-nervous-system/. [Πρόσβαση 14 April 2024].

[5] «Clinical Mnauals,» [Ηλεκτρονικό]. Available: https://ecgwaves.com/ecg-qrs-complex-q-r-s-wave-duration-interval/. [Πρόσβαση 14 April 2024].

[6] A. Anthony, *Schematic representation of a normal sinus rhythm ECG wave*, 2007.

[7] P. A. Lanfranchi και V. K. Somers, «Chapter 20 - Cardiovascular Physiology: Autonomic Control in Health and in Sleep Disorders,» σε *Principles and Practice of Sleep Medicine (Fifth Edition)*, Philadelphia, W.B. Saunders, 2011, pp. 226-236.

[8] C. David, T. Mika και J. Herbert, «How to Calculate Renyi Entropy from Heart Rate Variability, and Why it Matters for Detecting Cardiac Autonomic Neuropathy,» *Frontrontiers in Bioengineering and Biotechnology: Computational Physiology and Medicine,* τόμ. 2, 2014.

[9] Citi, L. a. Brown, E. a. Barbieri και Riccardo, «A Real-Time Automated Point-Process Method for the Detection and Correction of Erroneous and Ectopic Heartbeats,» *IEEE transactions on bio-medical engineering,* τόμ. 59, pp. 2828-37, 2012.

[10] B. a. G. P. a. T. J. Mulgrew, Digital signal processing: concepts and applications, Bloomsbury Publishing, 2002.

[11] P. P, «Butterworth Filters,» 15 July 2021. [Ηλεκτρονικό]. Available: https://tttapa.github.io/Pages/Mathematics/Systems-and-Control-Theory/Analog-Filters/Butterworth-Filters.html. [Πρόσβαση 25 July 2024].

[12] N. S. S. Kalpana, «Data Normalization With R,» Medium, 5 Ιούλιος 2020. [Ηλεκτρονικό]. Available: https://medium.com/swlh/data-normalisation-with-r-6ef1d1947970. [Πρόσβαση 26 Ιούλιος 2024].

[13] A. v. d. a. V. N. Enden, Discrete-time signal processing, Prentice-Hall, Inc., 1989.

[14] McCraty, R. a. Shaffer και Fred, «Heart rate variability: new perspectives on physiological mechanisms, assessment of self-regulatory capacity, and health risk,» *Global advances in health and medicine,* τόμ. 4, αρ. 1, pp. 46-61, 2015.

[15] Makivi, B. a. Niki, Djordjevi, M. a. Willis και M. S, «Heart Rate Variability (HRV) as a tool for diagnostic and monitoring performance in sport and physical activities.,» *Journal of Exercise Physiology Online,* τόμ. 16, αρ. 3, 2013.

[16] C. S., «The importance of time-domain HRV analysis in cardiac health prediction,» *Series Cardiol Res.,* τόμ. 4, αρ. 1, pp. 19-23, 2022.

[17] K. a. K. M. a. S. E. a. P. P. Tyapochkin, «Smartphone App Stress Assessments: Heart Rate Variability vs Perceived Stress in a Large Group of Adults,» αρ. 10.1101/2020.12.23.20247494, 2020.

[18] Shaffer, F. a. Meehan, M. Zachary, Zerr και L. Christopher, «A Critical Review of Ultra-Short-Term Heart Rate Variability Norms Research,» *Frontiers in Neuroscience*, τόμ. 14, 2020.

[19] Shaffer, F. a. Shearman, S. a. Meehan και M. Zachary, «The promise of ultra-short-term (UST) heart rate variability measurements,» *Biofeedback*, τόμ. 44, αρ. 4, pp. 229-233, 2016.

[20] C. a. M. K. Willmott, «Advantages of the Mean Absolute Error (MAE) over the Root Mean Square Error (RMSE) in Assessing Average Model Performance,» *Climate Research*, τόμ. 30, αρ. 10.3354/cr030079, p. 79, 2005.

[21] T. a. D. R. Chai, «Root mean square error (RMSE) or mean absolute error (MAE)?–Arguments against avoiding RMSE in the literature,» *Geoscientific Model Development*, τόμ. 7, αρ. 10.5194/gmd-7-1247-2014, pp. 1247-1250, 2014.

[22] M. Almarshad, M. Islam, S. Al-Ahmadi και A. BaHammam, «Diagnostic Features and Potential Applications of PPG Signal in Healthcare: A Systematic Review,» *Healthcare (Basel)*, τόμ. 10, αρ. 3, p. 547, 2022.

[23] G. Lu, F. Yang, J. Taylor και J. Stein, «A comparison of photoplethysmography and ECG recording to analyse heart rate variability in healthy subjects,» *J Med Eng Technol.*, τόμ. 33, αρ. 8, pp. 634-41, 2009.

[24] H. Selye, The Stress of Life, McGraw-Hill, 1976.

[25] R. Contrada και A. Baum, The Handbook of Stress Science: Biology, Psychology, and Health, Springer Publishing Company, 2010.

[26] A. Steptoe και M. Kivimaki, «Stress and cardiovascular disease,» *Nature Reviews Cardiology*, τόμ. 9, αρ. 6, pp. 360-370, 2012.

[27] «Mental disorders,» World Health Organization, 8 June 2022. [Ηλεκτρονικό]. Available: https://www.who.int/news-room/fact-sheets/detail/mental-disorders. [Πρόσβαση 17 April 2024].

[28] «Heart Rate Variability (HRV),» Cleveland Clinic, [Ηλεκτρονικό]. Available: https://my.clevelandclinic.org/health/symptoms/21773-heart-rate-variability-hrv. [Πρόσβαση 17 April 2024].

[29] A. Bailey, «What Is Heart Rate Variability & What Can It Tell You?,» verywell health, 15 February 2022. [Ηλεκτρονικό]. Available: https://www.verywellhealth.com/heart-rate-variability-5215411. [Πρόσβαση 18 April 2024].

[30] N. a. C. R. a. H. J. a. M. J. a. Q. I. a. G. L. a. C. P. Pinheiro, «Can PPG be used for HRV analysis?,» σε *2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, 2016.

[31] Scholkmann, F. a. Boss, J. a. Wolf και Martin, «An efficient algorithm for automatic peak detection in noisy periodic and quasi-periodic signals,» *Algorithms*, τόμ. 5, αρ. 4, pp. 588-603, 2012.

[32] B. Porr, «iirj,» [Ηλεκτρονικό]. Available: https://github.com/berndporr/iirj?tab=readme-ov-file. [Πρόσβαση 22 Ιούλιος 2024].

[33] T. F. o. t. E. S. o. C. a. t. N. A. S. o. P. a. Electrophysiology, «Heart rate variability: standards of measurement, physiological interpretation and clinical use.,» *Circulation*, τόμ. 93, αρ. 5, pp. 1043-65, 1996.

[34] «HRV Metrics: PNN50,» Elite HRV, 30 Νοέμβριος 2023. [Ηλεκτρονικό]. Available: https://help.elitehrv.com/article/344-hrv-metrics-pnn50. [Πρόσβαση 1 Σεπτέμβριος 2024].

[35] F. a. M. Y. a. D. A. a. K. L. C. N. a. S. R. a. T. M. Tueche, «Embedded Algorithm for QRS Detection Based on Signal Shape,» *IEEE Transactions on Instrumentation and Measurement*, τόμ. 70, pp. 1-12, 2021.

[36] S. a. W. G. Cohen, Perceived Stress in a Probability Sample of the United States, Newbury Park, CA: Sage, 1988.

[37] S. Cohen, T. Kamarck και R. Mermelstein, «The PSS Scale is reprinted with permission of the American Sociological Association,» *Journal of Health and Social Behavior*, τόμ. 24, pp. 386-396, 1983.

[38] «Galaxy Watch4 Classic,» Samsung, [Ηλεκτρονικό]. Available: https://www.samsung.com/gr/watches/galaxy-watch/galaxy-watch4-classic-black-bluetooth-46mm-sm-r890nzkaeue/. [Πρόσβαση 30 April 2024].

[39] «Samsung Privileged Health SDK,» Samsung, [Ηλεκτρονικό]. Available: https://developer.samsung.com/health/privileged/overview.html. [Πρόσβαση 30 April 2024].

[40] G. R. Geovanini, E. R. Vasques, R. de Oliveira Alvim, J. G. Mill, R. V. Andreão, B. K. Vasques, A. C. Pereira και J. E. Krieger, «Age and Sex Differences in Heart Rate Variability and Vagal Specific Patterns - Baependi Heart Study,» *Global heart*, τόμ. 15, αρ. 1, p. 71, 2020.

[41] P. N. Amsen, «Noise,» 2019. [Ηλεκτρονικό]. Available: https://github.com/paramsen/noise. [Πρόσβαση 5 September 2024].

[42] V. a. M. M. R. a. K. R. Madhuri, «Stress Management Using Artificial Intelligence,» σε *2013 Third International Conference on Advances in Computing and Communications*, 2013.

[43] J. a. N. A. Rodgers, «Thirteen Ways to Look at the Correlation Coefficient,» *American Statistician - AMER STATIST*, τόμ. 42, αρ. 10.1080/00031305.1988.10475524, pp. 59-66, 1988.

Συντομογραφίες - Αρκτικόλεξα - Ακρωνύμια

HRV	Heart Rate Variability
PPG	Photoplethysmography
SDNN	Standard Deviation of NN intervals
RMSSD	Root Mean Square of Successive Differences
pNN50	Percentage of successive NN intervals differing by more than 50 ms
MAE	Mean Absolute Error
RMSE	Root Mean Square Error
FFT	Fast Fourier Transform
DFT	Discrete Fourier Transform
AMo	Amplitude Mode
FIR	Finite Impulse Response
IIR	Infinite Impulse Response
PSS	Perceived Stress Scale
βλ.	Βλέπε
κ.ά.	Και άλλοι/άλλα
κο.	Κύριο

Απόδοση ξενόγλωσσων όρων

Απόδοση	Ξενόγλωσσος όρος
Heart Rate Variability	Μεταβλητότητα Καρδιακού Ρυθμού
Photoplethysmography	Φωτοπληθυσμογραφία
Standard Deviation of NN intervals	Τυπική Απόκλιση των NN διαστημάτων
Root Mean Square of Successive Differences	Ρίζα Μέσου Τετραγώνου Διαδοχικών Διαφορών
Percentage of successive NN intervals differing by more than 50 ms	Ποσοστό διαδοχικών NN διαστημάτων που διαφέρουν πάνω από 50 ms
Mean Absolute Error	Μέσο Απόλυτο Σφάλμα
Root Mean Square Error	Ρίζα Μέσου Τετραγώνου Σφάλματος
Fast Fourier Transform	Γρήγορος Μετασχηματισμός Fourier
Discrete Fourier Transform	Διακριτός Μετασχηματισμός Fourier
Amplitude Mode	Διαμόρφωση πλάτους
Ripple	Κυμάτωση
High-pass	Υψηλοπερατό
Low-pass	Χαμηλοπερατό
Band-pass	Ζωνοπερατό
Transfer Function	Συνάρτηση Μεταφοράς
Cutoff Frequency	Συχνότητα Αποκοπής

Min-Max Normalization	Κανονικοποίηση Ελαχίστου-Μεγίστου
Z-Score Normalization	Κανονικοποίηση Z-Score
Max Normalization	Κανονικοποίηση Μέγιστης Τιμής
order	Τάξη
False peaks	Ψευδείς Κορυφές
RR Intervals	Διαστήματα RR
Stress Level	Επίπεδο Άγχους
True	Αληθές
False	Ψευδές
Handler	Χειριστής
Event Listener	Ακροατής Συμβάντων
Activity	Δραστηριότητα
Reset	Επαναφορά
Error in Measurement	Σφάλμα στη Μέτρηση
Finite Impulse Response	Πεπερασμένη Κρουστική Απόκριση
Infinite Impulse Response	Άπειρη Κρουστική Απόκριση
Smartphone	Έξυπνο Τηλέφωνο
Perceived Stress Scale	Κλίμακα Αντιλαμβανόμενου Άγχους

Datapoint	Σημείο Δεδομένων
Composables	Σύνθετα Στοιχεία σε Kotlin
Nucleus Tractus Solitarius	Πυρήνας της Μονήρους Δεσμίδας
Column	Στήλη
Delay	Καθυστέρηση
Task	Εργασία
Male	Άνδρας
Female	Γυναίκα
Save	Αποθήκευση
Float	Αριθμός Κινητής Υποδιαστολής
Double	Αριθμός Διπλής Ακρίβειας
App	Εφαρμογή
Baseline	Βασική Τιμή
Possibility	Πιθανότητα
Bin Size	Μέγεθος Κάδου
Histogram	Ιστόγραμμα