Παρουσίαση και Ανάλυση Κώδικα (2): Συνοπτική παρουσίαση του κώδικα που χρησιμοποιήθηκε για την δημιουργία της εφαρμογής.

Επικοινωνία με firebase

Για να μπορέσει η εφαρμογή μας να επικοινωνήσει με την πλατφόρμα του Firbase της Google, αρχικά χρειάζεται η εγκατάσταση των απαραίτητων pod files με την βοήθεια του terminal.

A picture containing graphical user interface

Description automatically generated

Ακολούθως, το Xcode και το firebase κάνουν εύκολη στην ζωή στον προγραμματιστή, αφού παρέχουν την δυνατότητα εγκατάστασης ενός αρχείου που περιέχει όλες τις απαραίτητες πληροφορίες για να γίνει η σύνδεση μεταξύ του Cloud και του ρολογιού. Πιο συγκεκριμένα, το GoogleService-Info.plist παράγεται στην πλατφόρμα της firebase, προστίθεται στο project του Xcode και είναι εξιδεικευμένο για την κάθε εφαρμογή. Όπως φαίνεται στην φωτογραφία περιέχει διάφορες μεταβλητές με τις τιμές τους απαραίτητες για την σύνδεση της εφαρμογής.

A screenshot of a computer

Description automatically generated with medium confidence

Ύστερα από τα παραπάνω βήματα, ο προγραμματιστής το μόνο που έχει να κάνει είναι να καλέσει την συνάρτηση init() κατά την εκκίνηση της εφαρμογής.

Text

Description automatically generated with medium confidence

Πλέον η σύνδεση έχει επιτευχθεί και η κλήση των συναρτήσεων του firebase είναι εφικτή εφόσον έχει προηγουμένως συμπεριληφθεί η κατάλληλη βιβλιοθηκή.

Εγγραφή νέων χρηστών

Αρχικά βλέπουμε πως εμφανίζεται στην οθόνη της εφαρμογής.

A screenshot of a phone

Description automatically generated with medium confidence

Ακολουθώντας το documentation του firebase, πραγματοποιούμε την εγγραφή νέων χρηστών στην εφαρμογή μας μόνο με email και password. Για τον σκοπό αυτό, κάνουμε import τις κατάλληλες βιβλιοθήκες όπως φαίνονται στην εικόνα.

Text

Description automatically generated

Στην συνέχεια καλούμε την συνάρτηση createUser(withEmail: , password:), δίνοντας φυσικά τα κατάλληλα ορίσματα όπως φαίνεται στην παρακάτω εικόνα.Text

Description automatically generatedΣε περίπτωση ανεπιτυχούς εγγραφής, που μπορεί να οφείλεται σε αρκετούς λόγους όπως λανθασμένη διεύθυνση email, σε κωδικό με λιγότερο από 6 χαρακτήρες ή και ακόμα και σφάλμα δικτύου, τότε απλώς εκτυπώνουμε το σφάλμα. Σε επιτυχημένη προσπάθεια, ακολουθεί μια σειρά ενεργειών για τον καθορισμό των βασικών παραμέτρων του χρήστη. Όπως φαίνεται στις γραμμές 42-49 αποθηκεύουμε μόνιμα στην συσκευή κάποιες μεταβλητές όπως τα credentials του χρήστη. Αυτό γίνεται, διότι η πληκτρολόγηση στο ρολόι είναι δύσκολη και έτσι ο χρήστης θα μπορεί να συνδέεται κατευθείαν κάθε φορά που ανοίγει την εφαρμογή. Για να επιτευχθεί αυτό θέτουμε κάποιες μεταβλητές με τις τιμές τους στο UserDefaults.standard και στην συνέχεια καλούμε την εντολή .synchronize(). Στις γραμμές 50 και 51 σώζουμε σε μεταβλητές ένα reference της βάσης όπου θα αποθηκεύσουμε τα δεδομένα μας καθώς και το ID του χρήστη. Έτσι, είμαστε σε θέση με την εντολή ref.child("users/\(userID ?? "N/A")/email").setValue(email) (γραμμές 52-55) να αποθηκεύσουμε στην βάση πληροφορίες για τον χρήστη σε μια διεύθυνση που περιέχει το ID του που είναι μοναδικό. Το τελευταίο στάδιο είναι να κάνουμε true την μεταβλητή mainViewActive η οποία ενεργοποιεί αυτόματα την μετάβαση στο κύριο μενού της εφαρμογής (γραμμή 60).

Σύνδεση χρηστών

Η σύνδεση των χρηστών ακολουθεί παρόμοια υλοποίηση με την εγγραφή. Παρακάτω παρουσιάζεται η οθόνη που εμφανίζεται στον χρήστη.

A screenshot of a phone

Description automatically generated with medium confidence

Η βασική διαφοροποίηση είναι η μέθοδος που καλείται που είναι η createUser(withEmail: , password:) δίνοντας τα κατάλληλα ορίσματα.

A screenshot of a computer

Description automatically generated with medium confidenceΕπίσης, γίνεται έλεγχος των παραμέτρων του χρήστη βάσει των οποίων γίνονται οι μετρήσεις σε σχέση με αυτά που έχουν αποθηκευτεί στην βάση του firebase. Σε περίπτωση αλλαγής ανανεώνονται οι τιμές. Ο λόγος που γίνεται αυτό είναι γιατί μπορεί ο γιατρός μας ο οποίος ελέγχει τις τιμές των μετρήσεων να θέλει για παράδειγμα να λαμβάνει πιο συχνά μετρήσεις από τον ασθενή, οπότε και να μειώνει τον χρόνο ειδοποίησης νέας μέτρησης. Η ανάγνωση των τιμών γίνεται με την εντολή ref.child("/users/\(userID)/repeat\_time").getData(completion: { error, snapshot in όπου στην ουσία δίνουμε ένα reference της βάσης, το κατάλληλο path όπου βρίσκονται τα δεδομένα που θέλουμε (μέσα κρύβεται και το user ID) και την εντολή getData. Σε περίπτωση επιτυχίας τα δεδομένα βρίσκονται στην μεταβλητή snapshot την οποία και συγκρίνουμε με τις τιμές που έχουμε αποθηκεύσει στην συσκευή και αν χρειαστεί τις ανανεώνουμε (έλεγχος if … else γραμμές 62-68).

Text, application, chat or text message

Description automatically generated

Άδεια για ειδοποιήσεις, καταγραφή μετρήσεων

Η Apple για να περιορίσει την ανεξέλεγκτη χρήση των προσωπικών δεδομένων χωρίς την συναίνεση του χρήστη, επιβάλει στον προγραμματιστή να ζητάει άδεια με σκοπό την είσοδο και επεξεργασία ευαίσθητων δεδομένων. Για τον σκοπό αυτό, είναι αναγκαία τα αιτήματα για άδεια τόσο μέτρησης των ζωτικών σημάτων που πραγματεύεται η εφαρμογή (καρδιακοί παλμοί οξυγόνο κλπ), όσο και για την παραγωγή ειδοποιήσεων.

Στις πρώτες δύο φωτογραφίες παρουσιάζεται το κομμάτι κώδικα που ζητάει άδεια για δημιουργία ειδοποιήσεων μέσω της κλήσης της συνάρτησης requestAuthorization() στο UNUserNotificationCenter, καθώς και ο τρόπος που φαίνεται στον χρήστη.

Graphical user interface, text, application, chat or text message

Description automatically generated

Text

Description automatically generated

Αντίστοιχα, βλέπουμε το αίτημα για άδεια μέτρησης από την εφαρμογή των ζωτικών λειτουργειών του χρήστη και τον κώδικα που το υλοποιεί. Συγκεκριμένα, επιτυγχάνεται μέσω της κλήσης της συνάρτησης requestAuthorization() αυτή τη φορά στο HKHealthStore για συγκεκριμένα μεγέθη (quantities).

A screenshot of a phone

Description automatically generated with medium confidenceA screenshot of a cell phone

Description automatically generated with medium confidenceA screenshot of a cell phone

Description automatically generated with medium confidence

Text

Description automatically generated

Δημιουργία ειδοποιήσεων

Στην περίπτωση που ο χρήστης παραχωρήσει το δικαίωμα στην εφαρμογή να του στέλνει ειδοποιήσεις, τότε παρέχεται η δυνατότητα στον χρήστη ή ακόμα και στον γιατρό να προγραμματίζει ειδοποιήσεις ανά κάποιο διάστημα ωρών. Έτσι, υπενθυμίζεται στον χρήστη πότε πρέπει να λάβει μέτρηση και παράλληλα έχει την δυνατότητα να ρυθμίζει εκείνος ανά πόσες ώρες να του έρχεται ειδοποίηση μέσω των ρυθμίσεων όπως φαίνεται στην παρακάτω εικόνα.

A screenshot of a phone

Description automatically generated with medium confidence

Για να γίνει εφικτό αυτό, καλούμε την συνάρτηση func scheduleNotification(repeat\_time: Float) δίνοντας ως παράμετρο το χρονικό διάστημα μεταξύ 2 διαδοχικών ειδοποιήσεων. Εκεί ορίζουμε το περιεχόμενο του notification, δηλαδή τον τίτλο, το μήνυμα, τον ήχο κλπ (γραμμές 42-46) και στην συνέχεια δημιουργούμε ένα επαναλαμβανόμενο trigger κάθε όσες ώρες λάβαμε ως παράμετρο (γραμμή 48). Τέλος, σπρώχνουμε το αίτημα στο UNUserNotificationCenter του ρολογιού (γραμμές 51-54).

Text

Description automatically generated

Οι παραπάνω ενέργειες έχουν ως αποτέλεσμα να εμφανιστεί στον χρήστη η παρακάτω ειδοποίηση.

A screenshot of a phone

Description automatically generated with medium confidence

Λήψη τιμών

Αρχικοποίηση του Workout Session

Έχοντας λάβει από τον χρήστη έγκριση ώστε η εφαρμογή μας να χρησιμοποιεί τους αισθητήρες, ο προγραμματιστής έχει την δυνατότητα να κάνει αίτηση στο healthkit για να λάβει τις τιμές των μετρήσεων για τους καρδιακούς παλμούς, το οξυγόνο, το HRV και την απόσταση. Όπως έχουμε αναφέρει η Apple δεν παρέχει άμεση πρόσβαση στους αισθητήρες και κρατά κρυπτογραφημένη την υλοποίηση της. Έτσι, ο μοναδικός τρόπος είναι η προώθηση ενός request για τις τιμές που ενδιαφερόμαστε και το healthkit απαντά στο αίτημα, όταν έχει διαθέσιμες τιμές. Για τον σκοπό αυτό δημιουργούμε ένα workout session κατά την διάρκεια του οποίου λαμβάνονται συνεχώς οι μετρήσεις που θέλουμε. Πιο συγκεκριμένα, αρχικοποιούμε το workout session δίνοντάς του ως παραμέτρους τον τύπο άσκησης που κάνει ο χρήστης αφού τον επιλέξει ο ίδιος από την λίστα δραστηριοτήτων όπως φαίνεται στην εικόνα.Graphical user interface, text, application, chat or text message

Description automatically generated

Text

Description automatically generated

Έναρξη του Workout Session

Εφόσον ο χρήστης επιλέξει να εκκινήσει την μέτρηση, ξεκινάει ταυτόχρονα και ένας listener που περιμένει από το HealthStore να αποστείλει τιμές για τα αιτήματα που πραγματοποιήσαμε. Συνεπώς, δουλεία του προγραμματιστή είναι να ακούει τις απαντήσεις, να αναγνωρίζει ποιο μέγεθος αφορά, και στην συνέχεια να αποθηκεύει την τιμή που έλαβε μαζί με την χρονική στιγμή που έφτασε. Όλες αυτές οι τιμές αποθηκεύονται ανά κατηγορία και περιμένουν να εκπληρωθεί η συνθήκη τερματισμού για να αποθηκευτούν στην βάση δεδομένων του firebase (σειρά 135). Στο παρακάτω παράδειγμα κώδικα φαίνεται ο διακόπτης switch που περιέχει όλες τις περιπτώσεις μετρήσεων που λαμβάνει το πρόγραμμα από το HealthStore, καθώς και το παράδειγμα της τιμής για τον καρδιακό παλμό. Παράλληλα, ανάλογα τον τύπο, η τιμή που λαμβάνουμε υπόκεινται και στις ανάλογες μετατροπές μονάδας (σειρές 125-126). Με αντίστοιχο τρόπο γίνεται η συλλογή των δεδομένων και για τις υπόλοιπες μετρήσεις (οξυγόνο, HRV, απόσταση).Text

Description automatically generated

Προβολή των μετρήσεων

Κατά την διάρκεια του Workout Session κάθε τιμή που έρχεται από τους αισθητήρες προβάλλεται ταυτόχρονα στην οθόνη του ρολογιού όπως παρουσιάζεται παρακάτω. Ο κώδικας που υλοποιεί το συγκεκριμένο view και ενημερώνει και τις αντίστοιχές τιμές φαίνεται στην επόμενη φωτογραφία.

Graphical user interface, text, application, chat or text message

Description automatically generated

Text

Description automatically generated

Λήξη της μέτρησης

Κάθε φορά που ο χρήστης εκκινεί μια νέα διαδικασία μέτρησης απαιτείται να συμπληρωθεί ένας αριθμός τιμών για να ολοκληρωθεί η καταγραφή. Συγκεκριμένα, έχουμε ορίσει μια παράμετρο, τον αριθμό μετρήσεων ανά καταγραφή που αποτελεί και την συνθήκη τερματισμού. Πιο συγκεκριμένα, αν η μεταβλητή αυτή έχει τιμή 10, τότε το measurement θα τελειώσει όταν το ρολόι λάβει από το healthkit 10 μετρήσεις για το οξυγόνο, 10 για τους καρδιακούς παλμούς και 10 για το HRV αντίστοιχα. Τότε και μόνο τότε θα καλέσει την συνάρτηση end() (γραμμή 137) που τερματίζει την διαδικασία. Ταυτόχρονα, κρατάμε έναν μετρητή για τον κάθε τύπο μέτρησης και όποτε αυτός φτάσει την τιμή του measurement\_counter εγγράφει τις τιμές στην βάση δεδομένων στο firebase (γραμμές 133-135).Text

Description automatically generated

Την μεταβλητή αυτή τερματισμού των μετρήσεων μπορεί να την καθορίσει τόσο ο χρήστης μέσω των ρυθμίσεων, όσο και κάποιος απομακρυσμένος διαχειριστής (γιατρός) όπου ύστερα μέσα από το cloud θα ανανεωθεί και η τιμή στην εφαρμογή.

Graphical user interface, text, application

Description automatically generated