**Πανεπιστήμιο Πελοποννήσου**

**Σχολή Οικονομίας και Τεχνολογίας**

**Τμήμα Πληροφορικής και Τηλεπικοινωνιών**

**Π.Μ.Σ. στην Επιστήμη Υπολογιστών**

**Μεταπτυχιακή Εργασία**

**<Δημιουργία Εφαρμογής για Android και IOS>**

**Βενέρης Νικόλαος Δημήτριος**

**ΑΜ: 2022202002003**

**Πέππας Κωνσταντίνος**

**<Τρίπολη, Σεπτέμβρης 2023>**

**Πίνακας Περιεχομένων / Table of Contents**

[**II.** **Ευρετήριο Εικόνων / Table of Figures** 4](#_Toc145872274)

[**III.** **Εκτεταμένη Περίληψη** 4](#_Toc145872275)

[**IV.** **Extended abstract** 4](#_Toc145872276)

[1. Εισαγωγή / Introduction 5](#_Toc145872277)

[1. Η Ψηφιακή Επανάσταση 5](#_Toc145872278)

[2. Η Εξάπλωση των Εφαρμογών 5](#_Toc145872279)

[3. Στόχος Εργασίας 5](#_Toc145872280)

[2. Βασικοί Όροι 6](#_Toc145872281)

[1. Android vs IOS 6](#_Toc145872282)

[2. Android 6](#_Toc145872283)

[3. IOS 7](#_Toc145872284)

[4. Δυσκολία δημιουργίας εφαρμογών για Android και IOS 7](#_Toc145872285)

[5. Cross-platform frameworks 8](#_Toc145872286)

[6. React Native 8](#_Toc145872287)

[7. Flutter 9](#_Toc145872288)

[8. Επιλογή Flutter έναντι React Native 10](#_Toc145872289)

[3. Δομικά στοιχεία της εφαρμογής 10](#_Toc145872290)

[1. Google FireBase 11](#_Toc145872291)

[2. State Εφαρμογής 12](#_Toc145872292)

[3. Είδη State Management 12](#_Toc145872293)

[4. 3. Περιγραφή της Εφαρμογής 13](#_Toc145872294)

[1. Είσοδος στην εφαρμογή 13](#_Toc145872295)

[2. Προβολή λίστας προπονητικών προγραμμάτων/ προπονήσεων 16](#_Toc145872296)

[3. Δημιουργία/ επεξεργασία προπονητικού προγράμματος 19](#_Toc145872297)

[4. Πραγματοποίηση προπόνησης. 22](#_Toc145872298)

[5. Λίστα ασκήσεων 25](#_Toc145872299)

[6. Διαχείριση του Ιστορικού 26](#_Toc145872300)

[7. Επεξεργασία Προσωπικού Προφίλ 27](#_Toc145872301)

[**V.** **Συμπεράσματα – Μελλοντικές κατευθύνσεις / Conclusions – Future Directions** 29](#_Toc145872302)

[**VI.** **Βιβλιογραφία / References** 30](#_Toc145872303)

[1. Βιβλία 30](#_Toc145872304)

[2. Διαδίκτυο 30](#_Toc145872305)

[**VII.** **Παραρτήματα / Appendixes** 30](#_Toc145872306)

[1. Τοποθεσία κώδικα 30](#_Toc145872307)

# **Ευρετήριο Εικόνων / Table of Figures**

[Εικόνα 1 – Δομή React και React Native. 8](#_Toc145870280)

[Εικόνα 2 - Δομή Flutter. 9](#_Toc145870281)

[Εικόνα 3 - Σελίδα σύνδεσης. 13](#_Toc145870282)

[Εικόνα 4 - Αρχικοποίηση δεδομένων. 14](#_Toc145870283)

[Εικόνα 5 - Εγγραφή βάσης δεδομένων. 14](#_Toc145870284)

[Εικόνα 6 - Κενό ιστορικό προπονήσεων. 15](#_Toc145870285)

[Εικόνα 7 - Λίστα προπονήσεων. 16](#_Toc145870286)

[Εικόνα 8 - Επιπλέον επιλογές προπόνησης. 17](#_Toc145870287)

[Εικόνα 9 - Μήνυμα διαγραφής προπόνησης. 18](#_Toc145870288)

[Εικόνα 10 - Δημιουργία νέας προπόνησης. 19](#_Toc145870289)

[Εικόνα 11 - Προπόνηση με συμπληρωμένα πεδία. 20](#_Toc145870290)

[Εικόνα 12 - Μήνυμα σφάλματος. 21](#_Toc145870291)

[Εικόνα 13 - Έναρξη προπόνησης. 22](#_Toc145870292)

[Εικόνα 14 - Σελίδα πραγματοποίησης προπόνησης. 22](#_Toc145870293)

[Εικόνα 15 - Δήλωση άσκησης ως ολοκληρωμένη. 23](#_Toc145870294)

[Εικόνα 16 - Χρονόμετρο αποθεραπείας. 23](#_Toc145870295)

[Εικόνα 17 - Λίστα ασκήσεων. 24](#_Toc145870296)

[Εικόνα 18 - Ιστορικό προπονήσεων με εγγραφές. 25](#_Toc145870297)

[Εικόνα 19 - Επιπλέον πληροφορίες από το ιστορικό προπονήσεων. 26](#_Toc145870298)

[Εικόνα 20 - Σελίδα χρήστη. 27](#_Toc145870299)

[Εικόνα 21 - Σελίδα επεξεργασίας προσωποποιημένων δεδομένων χρήστη. 28](#_Toc145870300)

# **Εκτεταμένη Περίληψη**

Η παρούσα εργασία αναλύει την ανάπτυξη και τη δομή μιας κινητής εφαρμογής υγείας και ευεξίας που δημιουργήθηκε με χρήση του cross-platform framework Flutter. Η εφαρμογή, γνωστή ως "Workout Tracker," λειτουργεί ως ισχυρό εργαλείο παρακολούθησης και βελτίωσης της φυσικής κατάστασης των χρηστών. Αυτό επιτρέπει στους χρήστες να δημιουργούν προσαρμοσμένα προπονητικά προγράμματα, να καταγράφουν την πρόοδό τους και να διαχειρίζονται τα προσωπικά τους προφίλ.

Κατά τη διάρκεια της εργασίας, εξετάστηκε η επιλογή μεταξύ δύο δημοφιλών frameworks, του React και του Flutter, και αναλύθηκαν οι λόγοι που οδήγησαν στην επιλογή του τελευταίου. Επιπλέον, διερευνήθηκαν οι όροι Android και ΙOS, και εξηγήθηκε η απόφαση για τη χρήση ενός cross-platform framework όπως το Flutter και όχι για την χρήση native εργαλειών.

Επιπροσθέτως, αναλύθηκε η χρήση του Google Firebase και η σημασία της σε αυτό το πλαίσιο, καθώς και η επιλογή ενός state management εργαλείου για τη διαχείριση της κατάστασης της εφαρμογής. Τέλος, πραγματοποιήθηκε μια σύντομη παρουσίαση της εφαρμογής που αναπτύχθηκε, παρέχοντας μια επισκόπηση των βασικών χαρακτηριστικών και λειτουργιών της.

# **Extended abstract**

This thesis analyzes the development and structure of a mobile health and wellness application created using the cross-platform framework Flutter. The application, known as "Workout Tracker," functions as a powerful tool for monitoring and improving users' physical fitness. This allows users to create customized workout programs, track their progress, and manage their personal profiles.

During the course of the work, the choice between two popular frameworks, React and Flutter, was examined, and the reasons for selecting the latter were discussed. Additionally, the Android and iOS platforms were explored, and the decision to use a cross-platform framework like Flutter rather than native tools was explained.

Furthermore, the use of Google Firebase was analyzed, along with its significance in this context. The selection of a state management tool for managing the application's state was also discussed. Finally, a brief overview of the developed application was provided, giving an overview of its key features and functionalities.

## Εισαγωγή / Introduction

Στην εποχή μιας τεράστιας τεχνολογικής προόδου, οι κινητές συσκευές έχουν γίνει οι μηχανισμοί μέσω των οποίων έχουμε πρόσβαση, αλληλεπιδρούμε και εξατομικεύουμε τον ψηφιακό μας κόσμο. Αυτή η βαθιά μεταμόρφωση έχει αναδείξει την ανάπτυξη κινητών εφαρμογών στο προσκήνιο του ψηφιακού τοπίου, καθορίζοντας τον τρόπο που ζούμε, εργαζόμαστε, και συνδεόμαστε.

### Η Ψηφιακή Επανάσταση

Η έναρξη της ανάπτυξης κινητών συσκευών σηματοδότησε μια επανάσταση που διαμόρφωσε οριστικά τον τρόπο που αλληλεπιδρούμε με την τεχνολογία. Οι κινητές συσκευές, από τα πρώτα κινητά τηλέφωνα έως τα εξελιγμένα smartphones και tablets της σήμερον ημέρας, έχουν εξελιχθεί από απλά εργαλεία επικοινωνίας σε πολυδιάστατους πύργους παραγωγικότητας, ψυχαγωγίας, και πληροφόρησης. Έχουν γεφυρώσει γεωγραφικές αποστάσεις, ξεπεράσει χρονικούς περιορισμούς, και έχουν ανοίξει νέες διαστάσεις της ανθρώπινης αλληλεπίδρασης.

### Η Εξάπλωση των Εφαρμογών

Στην καρδιά αυτής της επανάστασης βρίσκεται η εξάπλωση των κινητών εφαρμογών, μικρά κομμάτια λογισμικού που εξυπηρετούν μια πληθώρα αναγκών και επιθυμιών. Οι εφαρμογές έχουν εισάγει ένα επίπεδο εξατομίκευσης και διευκόλυνσης που κάποτε φαινόταν αδιανόητο. Έχουν μας επιτρέψει να κουβαλήσουμε τη δύναμη του υπολογισμού στην τσέπη μας και έχουν αναδιαμορφώσει βιομηχανίες, από την υγεία μέχρι την εκπαίδευση και τη διασκέδαση και έχουν κάνει τομείς όπως αυτός της υγείας πολύ πιο προσβάσιμους. Έτσι, ο κόσμος μπορεί να αναπτύσσει τις γνώσεις του συνεχώς και με μεγάλη ευκολία, να έχει περισσότερες επιλογές διασκέδασης, αλλά το σημαντικότερο, έχει την δυνατότητα να ενημερώνεται και να κάνει ενέργειές για την σωστή διατήρηση της υγείας του!

### Στόχος Εργασίας

Ο στόχος αυτής της εργασίας επικεντρώνεται στη δημιουργία μιας χρήσιμης για Android και Ios που αφορά την ανάπτυξη ενός "Workout Tracker". Η εφαρμογή αυτή έχει ως στόχο να προσφέρει στους χρήστες τη δυνατότητα να καταγράφουν, να παρακολουθούν και να βελτιώνουν την φυσική τους κατάσταση μέσω προπονητικών προγραμμάτων με τα οποία θα μπορούν να αλληλεπιδρούν.

Πιο συγκεκριμένα, οι στόχοι της εργασίας είναι οι εξής:

1. Σύνδεση των χρηστών. Οι χρήστες θα έχουν την δυνατότητα να συνδέονται (login) με ένα μοναδικό αναγνωριστικό και να αποθηκεύουν τα δεδομένα τους στο cloud.
2. Εξατομίκευση των προπονητικών προγραμμάτων: Η εφαρμογή θα παρέχει τη δυνατότητα εξατομίκευσης των προπονητικών προγραμμάτων, επιτρέποντας στους χρήστες να προσαρμόζουν τις προπονήσεις τους στις ατομικές τους ανάγκες και στόχους.
3. Παρακολούθηση της προόδου: Οι χρήστες θα έχουν τη δυνατότητα να παρακολουθούν την πρόοδό τους, καταγράφοντας τα βάρη, τις επαναλήψεις, και τον χρόνο που δαπανούν σε κάθε άσκηση.
4. Επεξεργασία προσωποποιημένων δεδομένων. Οι χρήστες θα έχουν την δυνατότητα να επεξεργάζονται ορισμένα προσωποποιημένα στοιχεία τους, όπως εικόνες, σημειώσεις κ.α.

Ένας έμμεσος στόχος είναι η προώθηση της φυσικής κατάστασης και της υγείας των χρηστών μέσω της παροχής ενός εργαλείου που θα τους εμπνέει να είναι πιο δραστήριοι και υγιείς.

Αυτοί οι στόχοι αντικατοπτρίζουν τη φιλοδοξία της εργασίας να δημιουργήσει μια εφαρμογή που όχι μόνο θα προσφέρει λειτουργικότητα, αλλά θα ενισχύσει επίσης την υγεία και την ευεξία των χρηστών μέσω της παρακολούθησης και της βελτίωσης της φυσικής τους κατάστασης

## Βασικοί Όροι

Σε αυτό το κεφάλαιο θα αναπτυχθούν ορισμένοι όροι που θα αναφερθούν κατά την διάρκεια της αναφοράς.

### Android vs IOS

Τα smartphones είναι από τις πιο διαδεδομένες συσκευές στον κόσμο. Μέσω των εφαρμογών τους ο κόσμος μπορεί να έρθει σε επαφή με μια πληθώρα τομέων όπως αυτοί την υγείας, της διασκέδασης και της εκπαίδευσης.

Δύο από τις πιο βασικές και διαδεδομένες πλατφόρμες λειτουργικών συστημάτων για smartphones είναι το Android και το IOS.

### Android

Το Android είναι μια πλατφόρμα ανοιχτού κώδικα για κινητές συσκευές, αναπτύχθηκε από τη Google και ανήκει στο Open Handset Alliance. Στόχος του είναι να προωθήσει την καινοτομία στον κλάδο της κινητής τηλεφωνίας και να προσφέρει στους χρήστες μια πιο προσιτή και βελτιωμένη εμπειρία στη χρήση κινητών συσκευών. Παρέχει μια πλήρη στοίβα λογισμικού για κινητές συσκευές, διευκολύνοντας τους προγραμματιστές στη δημιουργία εφαρμογών χρησιμοποιώντας το Android SDK. Οι χρήστες απολαμβάνουν μια φιλική προς τον χρήστη εμπειρία από την αρχή, με δυνατότητα προσαρμογής των τηλεφώνων τους. Οι κατασκευαστές επωφελούνται από την ολοκληρωμένη λύση του Android για τη λειτουργία των συσκευών τους.

Το Android είναι ανοιχτού κώδικα, από χαμηλού επιπέδου modules μέχρι εφαρμογές και έχει άδειες επιχειρηματικής χρήσης όπως Apache/MIT. Οι προγραμματιστές έχουν πρόσβαση στον πηγαίο κώδικα της πλατφόρμας και οι κατασκευαστές μπορούν να προσαρμόσουν το Android ανάλογα με τις ανάγκες τους. Είναι σχεδιασμένο ειδικά για κινητές συσκευές, λαμβάνοντας υπόψη περιορισμούς όπως η διάρκεια της μπαταρίας και η περιορισμένη μνήμη και ταχύτητα. Μπορεί να τρέξει σε διάφορες φυσικές συσκευές, ανεξαρτήτως μεγέθους οθόνης, ανάλυσης ή τσιπσετ.

Η δημιουργία και συγγραφή εφαρμογών σε android μπορεί να γίνει σε οποιοδήποτε από τα διαδεδομένα λειτουργικά συστήματα και για την συγγραφή τους είναι εφικτή με την χρήση 2 προγραμματιστικών γλωσσών. Την Java ή την Kotlin.

### IOS

Το iOS (πρώην iPhone OS) είναι ένα κινητό λειτουργικό σύστημα που αναπτύχθηκε από την Apple Inc. Και απευθύνεται αποκλειστικά για τις συσκευές της Apple. Είναι το δεύτερο πιο διαδεδομένο κινητό λειτουργικό σύστημα στον κόσμο, μετά το Android. Το iOS αποτελεί τη βάση για τρία άλλα λειτουργικά συστήματα που δημιουργήθηκαν από την Apple: το iPadOS, το tvOS και το watchOS. Στην πλειονότητα του δεν είναι open source σε αντίθεση με τo Android αν και ορισμένα τμήματά του είναι ανοικτού κώδικα σύμφωνα με την άδεια χρήσης Apple Public Source και άλλες άδειες. Αποκαλύφθηκε το 2007 για το πρώτο iPhone και από τότε έχει επεκταθεί για να υποστηρίζει άλλες συσκευές της Apple, όπως το iPod Touch και το iPad.

Η δημιουργία και συγγραφή εφαρμογών σε IOS μπορεί να γίνει αποκλειστικά μόνο από συσκευές της Apple που έχουν το αντίστοιχο λογισμικό και για την συγγραφή τους είναι εφικτή με την χρήση 2 προγραμματιστικών γλωσσών. Την Swift ή την Objective-C.

Οι γλώσσες αυτές χαρακτηρίζονται και ως **Native** καθώς έχουν απευθείας επαφή με την συσκευή, το λειτουργικό και τα εργαλεία του.

### Δυσκολία δημιουργίας εφαρμογών για Android και IOS

Από τα παραπάνω συμπεραίνουμε ότι για τη δημιουργία μιας εφαρμογής για συσκευές Android και iOS απαιτεί γνώση δύο διαφορετικών γλωσσών προγραμματισμού: Java ή Kotlin για το Android και Swift ή Objective-C για το iOS. Η πολυπλοκότητα αυτού του εγχειρήματος εξαρτάται από διάφορους παράγοντες, συμπεριλαμβανομένων των παρακάτω:

* Επίπεδο εμπειρίας: Η εμπειρία του προγραμματιστή στις δύο πλατφόρμες είναι σημαντικός παράγοντας. Ένας προγραμματιστής που είναι εξοικειωμένος με μία από τις δύο γλώσσες προγραμματισμού θα χρειαστεί χρόνο για να μάθει τη δεύτερη.
* Εργαλεία και πλατφόρμες: Και οι δύο πλατφόρμες προσφέρουν τα δικά τους εργαλεία ανάπτυξης, όπως το Android Studio για Android και το Xcode για iOS. Ο προγραμματιστής πρέπει να είναι εξοικειωμένος με αυτά τα εργαλεία.
* Σχεδιασμός διεπαφής: Οι διεπαφές χρήστη διαφέρουν ανάμεσα στα δύο λειτουργικά συστήματα, και ο προγραμματιστής πρέπει να προσαρμόσει τον σχεδιασμό της εφαρμογής σε κάθε πλατφόρμα.
* Λειτουργικότητα: Σε ορισμένες περιπτώσεις, η εφαρμογή ενδέχεται να απαιτεί λειτουργικότητες που είναι διαφορετικές ανάμεσα στις δύο πλατφόρμες. Αυτό μπορεί να αυξήσει την πολυπλοκότητα της ανάπτυξης.

Για να αντιμετωπιστούν αυτά τα προβλήματα, υπάρχουν εργαλεία που επιτρέπουν την ανάπτυξη εφαρμογών για και τα δύο λειτουργικά συστήματα χρησιμοποιώντας μία κοινή γλώσσα προγραμματισμού, όπως το Flutter της Google ή το React Native της Facebook. Αυτές Τα εργαλεία αυτά (Cross-platform frameworks) επιτρέπουν στους προγραμματιστές να αναπτύξουν εφαρμογές για και τα δύο λειτουργικά συστήματα με μία κοινή γλώσσα προγραμματισμού και ένα κοινό κώδικα.

### Cross-platform frameworks

Τα Cross-platform frameworks είναι εργαλεία ανάπτυξης λογισμικού που επιτρέπουν στους προγραμματιστές να γράφουν κώδικα μία φορά και να τον αναπτύσσουν σε πολλά λειτουργικά συστήματα ή πλατφόρμες. Αυτά τα πλαίσια είναι ιδιαίτερα χρήσιμα για τη δημιουργία κινητών εφαρμογών που μπορούν να τρέχουν τόσο σε συσκευές Android όσο και σε συσκευές iOS, καθώς και για τη δημιουργία εφαρμογών επιφανείας εργασίας που μπορούν να λειτουργούν σε διάφορα λειτουργικά συστήματα όπως Windows, macOS και Linux.

Οι σημαντικότεροι λόγοι για την επιλογή τους είναι:

* Γρηγορότερος προγραμματισμός καθώς χρησιμοποιείται μόνο μια γλώσσα προγραμματισμού.
* Μικρότερα έξοδα καθώς μια εταιρεία δεν χρειάζεται να αναπτύξει ομάδες και για τις 2 πλατφόρμες.
* Συνοχή μεταξύ πλατφορμών καθώς διατηρούνται οι ίδιες λειτουργικότητες και το ίδιο UI.

Ωστόσο, παρά τα θετικά τους, τα Frameworks αυτά έχουν και μερικά αρνητικά:

* Χαμηλότερη απόδοση καθώς μια native γλώσσα έχει απευθείας επικοινωνία με την συσκευή και τα frameworks απατούν πάντα μια γέφυρα επικοινωνίας για να επικοινωνούν με το λειτουργικό.
* Αργή απόκτηση updates. Μετά από κάποιο update, η αφομοίωσή του από ένα framework είναι σχετικά αργή.

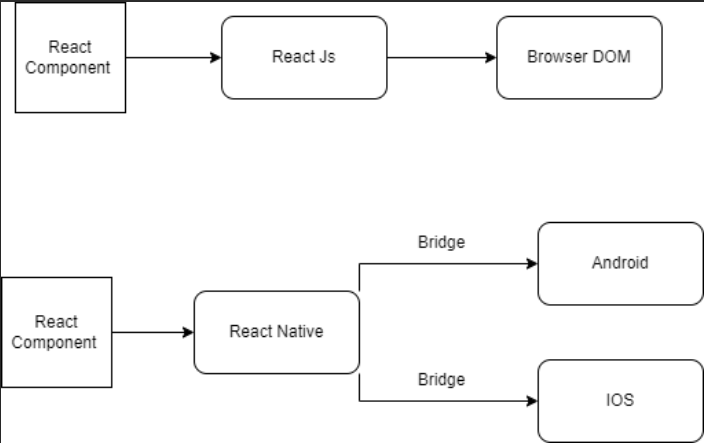
Αν και υπάρχουν πολλά τέτοια εργαλεία για την ανάπτυξη android και IOS εφαρμογών όπως τα Ionic, Cordova, Xamarin, Maui (Ο διάδοχος του Xamarin), δύο ήταν τα frameworks τα οποία εξετάστηκαν λόγω της τεράστιας κοινότητας που το καθένα έχει αλλά και της δημοτικότητας τους. Το Flutter της Google και η React Native της Facebook.

### React Native

Η React είναι μια open-source βιβλιοθήκη της JavaScript για τη δημιουργία διεπαφών χρήστη. Δημιουργήθηκε από το Facebook το 2013 και έχει γνωρίσει ενθουσιώδη υποδοχή από ένα ευρύ κοινό προγραμματιστών Σχεδιάστηκε για τις ανάγκες της ομάδας ανάπτυξης του Facebook και είναι ιδιαίτερα κατάλληλη για πολύπλοκες web εφαρμογές που ασχολούνται εκτενώς με την αλληλεπίδραση του χρήστη και την αλλαγή δεδομένων. Το 2015, η ομάδα της React ανακοίνωσε τη React Native η οποία χρησιμοποιεί τη React για να στοχεύσει πλατφόρμες διαφορετικές όπως το ΙOS και το Android, υλοποιώντας μια λεγόμενη "γέφυρα" (bridge) μεταξύ του JavaScript και της πλατφόρμας (IOS, Android). Η γέφυρα αυτή, είναι το μέσο επικοινωνίας μεταξύ του επιπέδου της Javascript και της πλατφόρμας. Λειτουργεί δηλαδή σαν ο μεταφραστής που μεταφέρει και μεταφράζει μηνύματα από και προς τα 2 περιβάλλοντα

Ένα από τα χαρακτηριστικά της React είναι το Virtual DOM. Το Virtual DOM δρα ως ένα επίπεδο μεταξύ του πως ο προγραμματιστής θέλει να απεικονίζονται οι διεπαφές χρήστη και της διαδικασίας που χρησιμοποιείται για την πραγματική απεικόνισή τους στη σελίδα. Για να απεικονίσουν αλληλεπιδραστικές διεπαφές χρήστη σε έναν περιηγητή, οι προγραμματιστές πρέπει να επεξεργαστούν το DOM του περιηγητή, το οποίο είναι μια δαπανηρή διαδικασία και οι υπερβολικές εγγραφές στο DOM έχουν σημαντικό αντίκτυπο στην απόδοση. Αντί να απεικονίσει απευθείας τις αλλαγές στη σελίδα, το React υπολογίζει τις απαραίτητες αλλαγές χρησιμοποιώντας μια εκδοχή του DOM στη μνήμη. Στη συνέχεια, επανυπολογίζει το ελάχιστο ποσοστό της εφαρμογής σας που χρειάζεται να υπολογιστεί.

Τα κύρια πλεονεκτήματα του Virtual DOM είναι η απόδοση και το abstraction του, που επιτρέπουν στη React να απεικονίσει πέρα από το DOM του περιηγητή/browser. Γι’ αυτό το λόγο, η Facebook παρουσίασε το React Native για την απεικόνιση στις κινητές συσκευές (IOS και Android). Σε αυτό το πλαίσιο, η "native" σημαίνει ότι η React δημιουργεί προβολές χρησιμοποιώντας τα native εργαλεία της πλατφόρμας. Αυτό είναι δυνατό χάρη στη "γέφυρα", που παρέχει στη React μια διεπαφή προς τα εργαλεία της native πλατφόρμας.

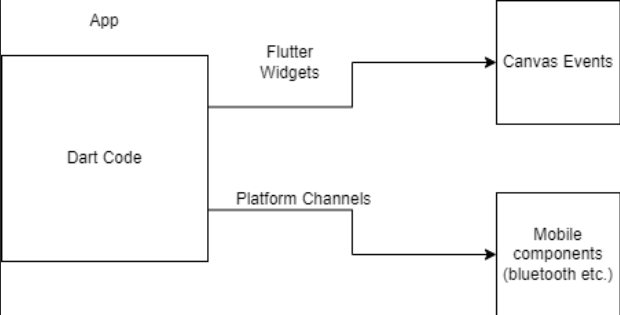


Εικόνα 1 – Δομή React και React Native.

### Flutter

Το Flutter είναι ένα open source σύνολο εργαλείων ανάπτυξης λογισμικού διεπαφής χρήστη (UI) που δημιουργήθηκε από την Google το 2017. Το Flutter χρησιμοποιείται για την κατασκευή εφαρμογών χρησιμοποιώντας τη γλώσσα προγραμματισμού Dart, η οποία επίσης αναπτύσσεται από την Google.

Στο Flutter, η διεπαφή χρήστη (UI) χτίζεται χρησιμοποιώντας μια βασισμένη σε widgets προσέγγιση. Τα widgets είναι τα οικοδομικά στοιχεία μιας εφαρμογής Flutter. Όλα στο Flutter είναι ένα widget, συμπεριλαμβανομένης της εφαρμογής ίδιας, κουμπιών, κειμένου, εικόνων, διατάξεων και ακόμη και ολόκληρης της οθόνης της εφαρμογής. Τα widgets μπορούν να συνδυαστούν και να αποτελέσουν πολύπλοκα στοιχεία διεπαφής χρήστη. Ένα από τα κορυφαία χαρακτηριστικά του Flutter είναι το Hot Reload που επιτρέπει στους προγραμματιστές να κάνουν αλλαγές στον κώδικά τους και να δουν αμέσως τα αποτελέσματα στην τρέχουσα εφαρμογή χωρίς να χάνουν την κατάσταση της εφαρμογής. Αυτό επιταχύνει σημαντικά τη διαδικασία ανάπτυξης και επιτρέπει γρήγορη επανάληψη. Το Flutter διαθέτει τη δική του μηχανή απόδοσης που ονομάζεται "Skia," μια ισχυρή βιβλιοθήκη 2D γραφικών. Η Skia είναι υπεύθυνη για το σχεδιασμό όλων των στοιχείων στην οθόνη και είναι υψηλά βελτιστοποιημένη για ομαλές και αποδοτικά animations και transitions. Επίσης, το Flutter παρέχει μια σειρά συνδέσεων για κάθε πλατφορμα που επιτρέπουν στον κώδικά της Dart να επικοινωνεί με την ανωτέρω πλατφόρμα (iOS, Android, web κτλ.). Αυτές οι συνδέσεις επιτρέπουν στο Flutter να έχει πρόσβαση σε native λειτουργίες και διεπαφές πλατφόρμας. Τέλος, το Flutter είναι γνωστό για τις υψηλές του δυνατότητες απόδοσης. Διότι μεταγλωττίζεται σε φυσικό κώδικα ARM (για κινητά) και JavaScript (για web), μπορεί να επιτύχει απόδοση κοντά στην απόδοση των φυσικών εφαρμογών (Native).



Εικόνα 2 - Δομή Flutter.

### Επιλογή Flutter έναντι React Native

Όπως αναφέραμε πριν, η React χρησιμοποιεί μια γέφυρα η οποία λειτουργεί σαν ένας τρόπος επικοινωνίας με τα στοιχεία διεπαφής χρήστη της αντίστοιχης πλατφόρμας. Η χρήση της γέφυρας είναι πολύ χρήσιμη αλλά μπορεί και να κάνει την εφαρμογή πιο αργή όταν δομικά στοιχεία καλούνται πολύ συχνά ή ταυτόχρονα. Επίσης, η χρήση JavaScript και σε συνδυασμό με την γέφυρα είναι πιο αργή από την χρήση της Dart. Το flutter μπορεί να τρέξει με περισσότερα fps (Μαξ 120 έναντι 60 της react). Τέλος, όλα όσα αναφέρθηκαν σχετικά με την επιλογή έχουν να κάνουν και λόγω προσωπικής εμπειρίας με το flutter, καθώς οι διαφορές μεταξύ Flutter και React Native στην εφαρμογή που δημιουργήθηκε δεν θα ήταν εμφανείς μιας είναι και τα 2 πολύ γρήγορα εργαλεία.

## Δομικά στοιχεία της εφαρμογής

Η επιτυχία μιας κινητής εφαρμογής βασίζεται σε μεγάλο βαθμό στα δομικά της στοιχεία, τα οποία καθορίζουν την απόδοση, τη λειτουργικότητα και την αξιοπιστία της. Στο πλαίσιο αυτου του κεφαλαίου, θα εξεταστούν τα δομικά στοιχεία της εφαρμογής, η οποία εκτός από Flutter αναπτύχθηκε το Google Firebase για την αποθήκευση δεδομένων στο cloud. Τέλος θα αναπτυχθεί και η state management τεχνολογία που δημιουργήθηκε.

### Google FireBase

Το Firebase είναι μια πλατφόρμα ανάπτυξης εφαρμογών για κινητές συσκευές και ιστοσελίδες που αναπτύχθηκε από τη Firebase, Inc. και αργότερα αποκτήθηκε από τη Google το 2014. Παρέχει στους προγραμματιστές ένα σύνολο εργαλείων και υπηρεσιών για τη δημιουργία και διαχείριση εφαρμογών με πιο ευκολία και αποτελεσματικότητα. Η Firebase προσφέρει διάφορα χαρακτηριστικά και υπηρεσίες για να βοηθήσει τους προγραμματιστές σε καθήκοντα όπως η ανάπτυξη εφαρμογών, η διαχείριση βάσεων δεδομένων, η ταυτοποίηση, η αποθήκευση στο cloud και πολλά άλλα.

Ορισμένα από τα κύρια συστατικά και χαρακτηριστικά της Firebase περιλαμβάνουν:

1. Βάση Δεδομένων: Η Firebase παρέχει μια βάση δεδομένων NoSQL στο cloud που επιτρέπει στους προγραμματιστές να αποθηκεύουν και συγχρονίζουν δεδομένα σε πραγματικό χρόνο σε πολλούς πελάτες. Αυτό είναι ιδιαίτερα χρήσιμο για τη δημιουργία συνεργατικών και διαδραστικών εφαρμογών.
2. Ταυτοποίηση: Η Firebase προσφέρει ισχυρές υπηρεσίες ταυτοποίησης, συμπεριλαμβανομένης της ταυτοποίησης με email και κωδικό πρόσβασης, της ταυτοποίησης μέσω κοινωνικών δικτύων (π.χ., Google, Facebook, Twitter) και της ανώνυμης ταυτοποίησης. Αυτό απλοποιεί τη διαχείριση των χρηστών και εξασφαλίζει ασφαλή πρόσβαση στην εφαρμογή σας.
3. Hosting: Η Firebase Hosting επιτρέπει στους προγραμματιστές να αναπτύσσουν γρήγορα και ασφαλώς web εφαρμογές. Παρέχει ένα δίκτυο παράδοσης περιεχομένου (CDN) για γρήγορο και παγκόσμιο hosting και λειτουργεί ως πλατφόρμα για τη φιλοξενία στατικού και δυναμικού περιεχομένου.
4. Cloud Functions: Οι Firebase Cloud Functions επιτρέπουν τους χρήστες να εκτελουν λειτουργίες χωρίς διακομιστή σε απάντηση σε γεγονότα της Firebase ή αιτήματα HTTP. Αυτό επιτρέπει να επεκτείνουν και να προσαρμόζουν τη συμπεριφορά της εφαρμογής στην πλευρά του διακομιστή.
5. Cloud Firestore: Το Firestore είναι η επόμενη γενιά βάσης δεδομένων NoSQL της Firebase, προσφέροντας πιο προηγμένες δυνατότητες ερωτήσεων και μοντελοποίησης δεδομένων σε σύγκριση με τη Βάση Δεδομένων σε Πραγματικό Χρόνο.
6. Cloud Storage: Η Firebase Storage παρέχει κλιμακούμενη αποθήκευση στο cloud για περιεχόμενο που δημιουργούν οι χρήστες, όπως εικόνες, βίντεο και άλλα αρχεία.
7. Cloud Messaging: Το Firebase Cloud Messaging (FCM) επιτρέπει στους χρήστες να στέλνουν ειδοποιήσεις push στους χρήστες σε πλατφόρμες Android, iOS και web.
8. Αναλυτικά Δεδομένα: Τα Αναλυτικά Δεδομένα της Firebase παρέχουν εισαγωγές στη συμπεριφορά του χρήστη και την απόδοση της εφαρμογής, βοηθώντας τους χρήστες να λαμβάνουν αποφάσεις με βάση τα δεδομένα για τη βελτίωση της εφαρμογής τους.
9. Παρακολούθηση Απόδοσης: Η Παρακολούθηση Απόδοσης της Firebase βοηθά τους χρήστες να παρακολουθούν και να βελτιστοποιούν την απόδοση της εφαρμογής τους, συμπεριλαμβανομένων των χρόνων εκκίνησης της εφαρμογής, των αιτημάτων δικτύου και πολλών άλλων.

Η Firebase είναι μια ευέλικτη πλατφόρμα κατάλληλη για μια ευρεία γκάμα εφαρμογών, από μικρά προσωπικά έργα έως μεγάλες επιχειρησιακές λύσεις. Η ενσωμάτωσή της με άλλες υπηρεσίες του Google Cloud και η ευκολία χρήσης την καθιστούν μια δημοφιλή επιλογή μεταξύ των προγραμματιστών για τη δημιουργία και διαχείριση μοντέρνων εφαρμογών. Επίσης είναι πολύ σημαντικό να αναφερθεί πως η Firebase προσφέρει μια πολύ μεγάλη σουίτα εργαλείων για την διασύνδεση με Flutter εφαρμογές κάτι που πολλές φορές (και στην εφαρμογή μας) σημαίνει ότι δεν υπάρχει απαίτηση για backend.( <https://firebase.flutter.dev/>)

### State Εφαρμογής

Πριν αναφερθούμε στις διάφορες επιλογές που υπάρχουν για state management, καλό είναι να δοθεί ένας ορισμός για το τι είναι state management.

Η διαχείριση κατάστασης (state management) είναι ένα κρίσιμο κομμάτι της ανάπτυξης λογισμικού, ειδικά στο πλαίσιο του προγραμματισμού διεπαφής χρήστη (UI). Αναφέρεται στον χειρισμό των δεδομένων που ένα πρόγραμμα χρειάζεται να παρακολουθεί κατά την εκτέλεσή του. Στο πλαίσιο του προγραμματισμού του UI, η διαχείριση της κατάστασης περιλαμβάνει την παρακολούθηση μεταβλητών που μπορεί να επηρεάσουν την έξοδο μιας μεθόδου απεικόνισης (render) με την πάροδο του χρόνου. Αυτό μπορεί να περιλαμβάνει την είσοδο του χρήστη, τις ρυθμίσεις διαμόρφωσης ή τα δεδομένα που ανακτώνται από μια βάση δεδομένων. Η αποτελεσματική διαχείριση της κατάστασης εξασφαλίζει ότι μια εφαρμογή συμπεριφέρεται προβλέψιμα και είναι πιο εύκολη στον έλεγχο και την αποσφαλμάτωση. Μπορεί επίσης να βελτιώσει την απόδοση, αποτρέποντας περιττούς υπολογισμούς ή απεικονίσεις (renders).

### Είδη State Management

Υπάρχουν αρκετές στρατηγικές διαχείρισης στο Flutter, συμπεριλαμβανομένων των εξής:

1. SetState: Αυτή είναι η χαμηλού επιπέδου προσέγγιση που χρησιμοποιείται για την κατάσταση που αφορά συγκεκριμένα σε ένα widget. Είναι μια συνάρτηση που καλείται μέσα σε ένα StatefulWidget για τον τροποποιημένο του κατάσταση και την ανακατασκευή του.
2. Provider: Είναι μία προσθήκη πάνω δομικό στοιχείο InheritedWidget του flutter που το καθιστά πιο εύκολο να χρησιμοποιηθεί και να είναι πιο επαναχρησιμοποιήσιμο.
3. BLoC (Business Logic Component): Είναι ένα σύστημα διαχείρισης καταστάσεων που συνιστάται από τους προγραμματιστές της Google για το Flutter. Χωρίζει την επιχειρηματική λογική από το UI.
4. GetX: Είναι μια λύση διαχείρισης καταστάσεων που συνδυάζει χρήσιμες λειτουργίες με απλή σύνταξη για τους προγραμματιστές.
5. Riverpod: Λειτουργεί με παρόμοιο τρόπο με το Provider. Προσφέρει ασφάλεια στον χρόνο μεταγλώττισης (compile safety) και δυνατότητα δοκιμών χωρίς να εξαρτάται από το Flutter SDK.

Η επιλογή μεταξύ αυτών των στρατηγικών εξαρτάται από τις συγκεκριμένες ανάγκες κάθε εφαρμογής και την εξοικείωσή κάθε χρήστη με αυτές τις έννοιες. Για την συγκεκριμένη εφαρμογή χρησιμοποιήθηκε το εργαλείο Provider για τους εξής λόγους:

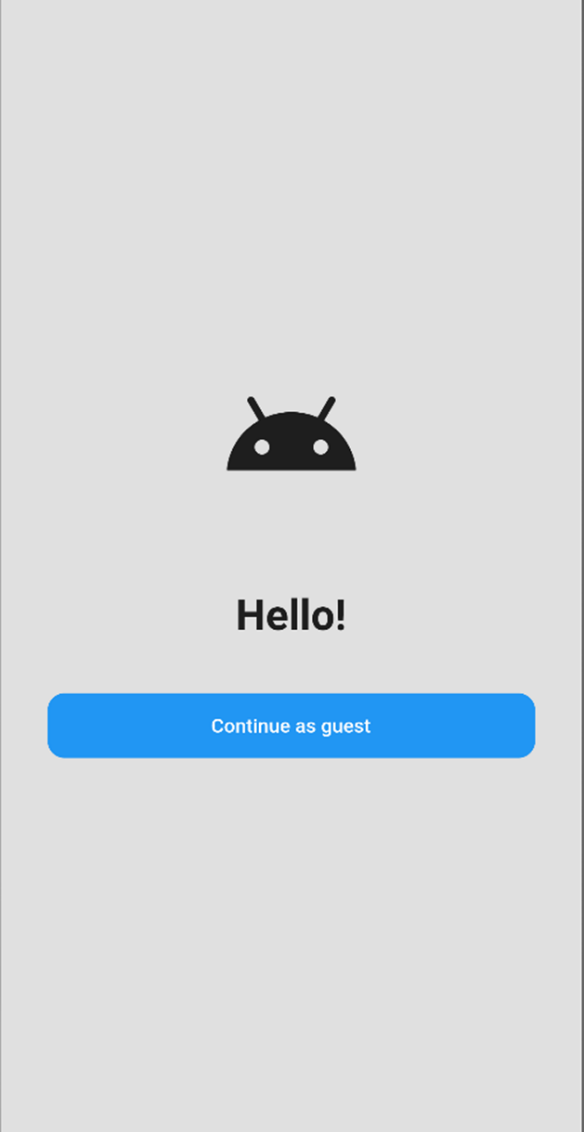
1. Απλότητα: Το Provider είναι σχετικά εύκολο να κατανοηθεί και δεν απαιτεί την εγγραφή πολλού κώδικα.
2. Ευελιξία: Το Provider μπορεί να χρησιμοποιηθεί με άλλες τεχνικές διαχείρισης κατάστασης όπως το BLoC, πράγμα που το καθιστά πιο ευέλικτο.
3. Αποδοτικότητα: Το Provider διασφαλίζει ότι η διεπαφή χρήστη σας είναι αποδοτική. Χρησιμοποιώντας το Provider, μόνο τα στοιχεία της διεπαφής που ακούνε για μια συγκεκριμένη πληροφορία θα ανακατασκευάζονται όταν αυτή η πληροφορία αλλάζει.
4. Υποστήριξη από την κοινότητα: Το Provider χρησιμοποιείται ευρέως στην κοινότητα του Flutter, πράγμα που σημαίνει ότι θα βρείτε πολλούς πόρους και υποστήριξη.
5. Αν και το Riverpod είναι βασισμένο στο Provider και προσφέρει πολλές παραπάνω δυνατότητες, είναι πιο περίπλοκο, απαιτεί περισσότερο χρόνο εκμάθησης και οι παραπάνω δυνατότητες δεν ήταν τόσο χρήσιμες.
6. Η χρήση του SetState και τα callbacks που απαιτεί μπορεί να οδηγήσει σε πολλά προβλήματα (bugs) και πολλά re-renders που δεν χρειάζονται.

## 3. Περιγραφή της Εφαρμογής

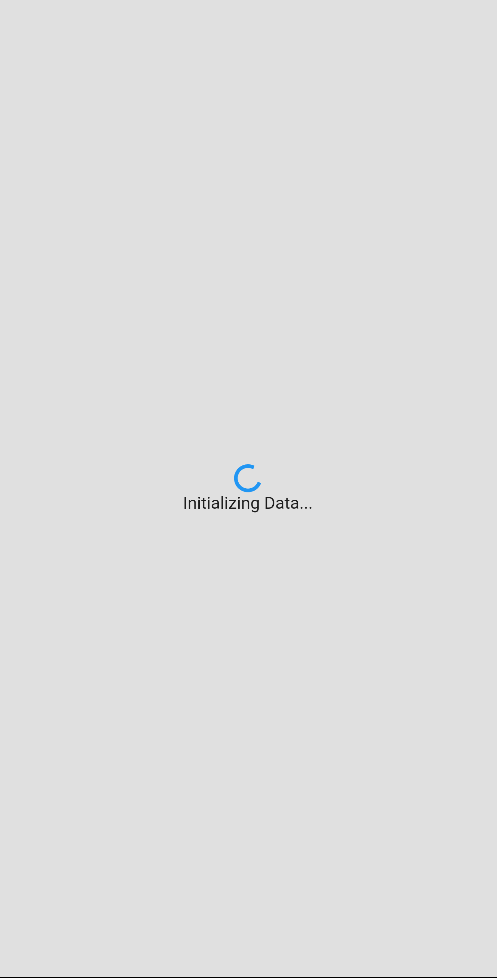
Σε αυτό το κεφάλαιο, θα εξετάσουμε αναλυτικά τη λειτουργικότητα της εφαρμογής μας, του "Workout Tracker." Θα εξηγήσουμε πώς οι χρήστες μπορούν να αλληλεπιδρούν με την εφαρμογή, ποιες λειτουργίες διατίθενται και πώς αυτές οδηγούν σε μια ομαλή και αποτελεσματική εμπειρία. Θα δώσουμε έμφαση στην προσαρμογή, τη δημιουργία προπονητικών προγραμμάτων, την καταγραφή προόδου και τη διαχείριση του προσωπικού προφίλ. Λόγω της απαίτησης της Apple για ύπαρξη Apple συσκευής έτσι ώστε να δημιουργηθεί η εφαρμογή για IOS, η εφαρμογή υλοποιήθηκε μόνο για Android.

### Είσοδος στην εφαρμογή

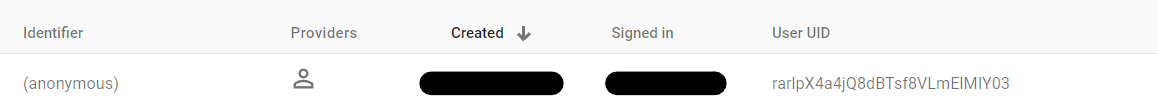
Η πρώτη επαφή του χρήστη με την εφαρμογή είναι η διαδικασία εισόδου. Μόλις ο χρήστης πατήσει “Continue as a guest” τότε δημιουργείται ένας ανώνυμος χρήστης με ένα τυχαίο όνομα χρήστη στο google Firebase και στη συνέχεια λαμβάνει μέρος μιας αρχικοποίηση, η οποία παράγει αυτόματα στην βάση δεδομένων στο google Firebase, ένα σύνολο βασικών ασκήσεων και κάποια έτοιμα προπονητικά προγράμματα.



Εικόνα 3 - Σελίδα σύνδεσης.

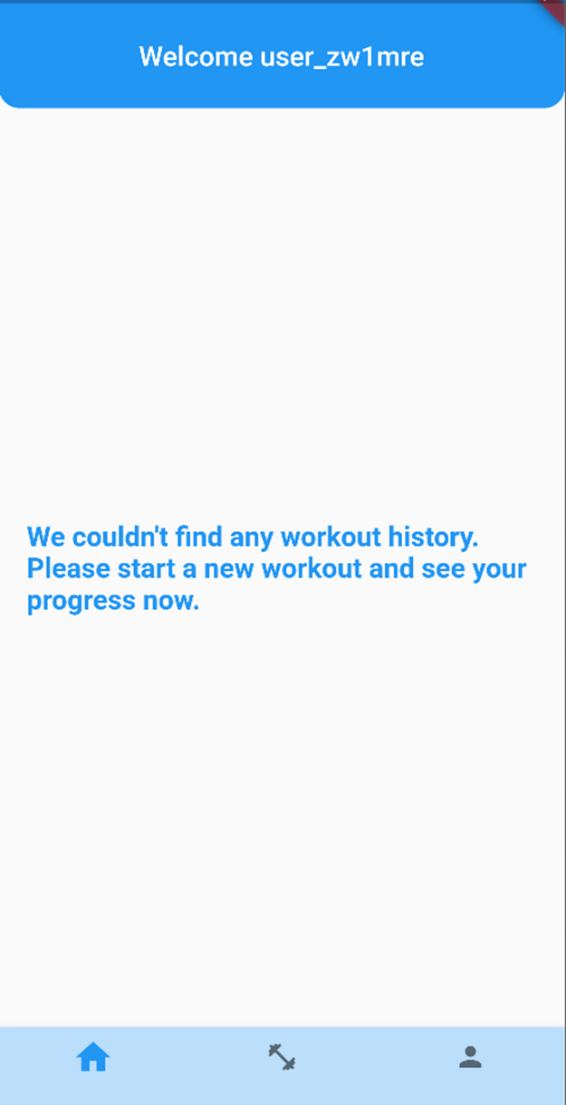


Εικόνα 4 - Αρχικοποίηση δεδομένων.



Εικόνα 5 - Εγγραφή βάσης δεδομένων.

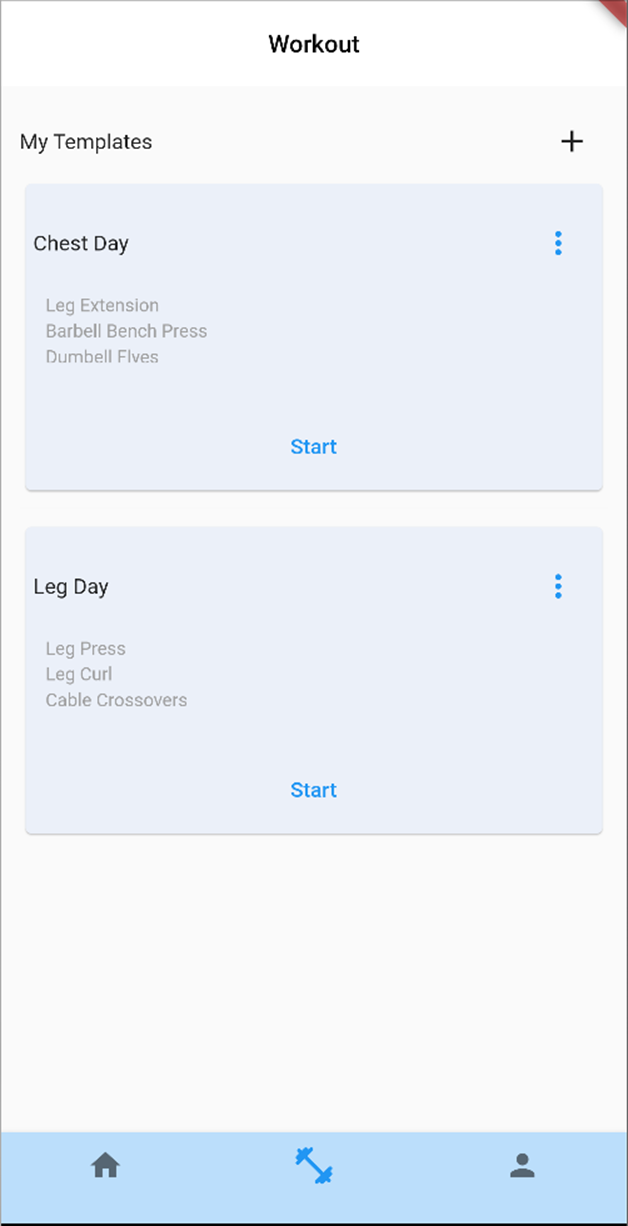
Μετά την είσοδο στην εφαρμογή ο χρήστης έρχεται σε επαφή με το ιστορικό καταγραφής προόδου το οποίο αρχικά είναι άδειο καθώς δεν έχει προχωρήσει στην χρήση κάποιου προπονητικού προγράμματος.



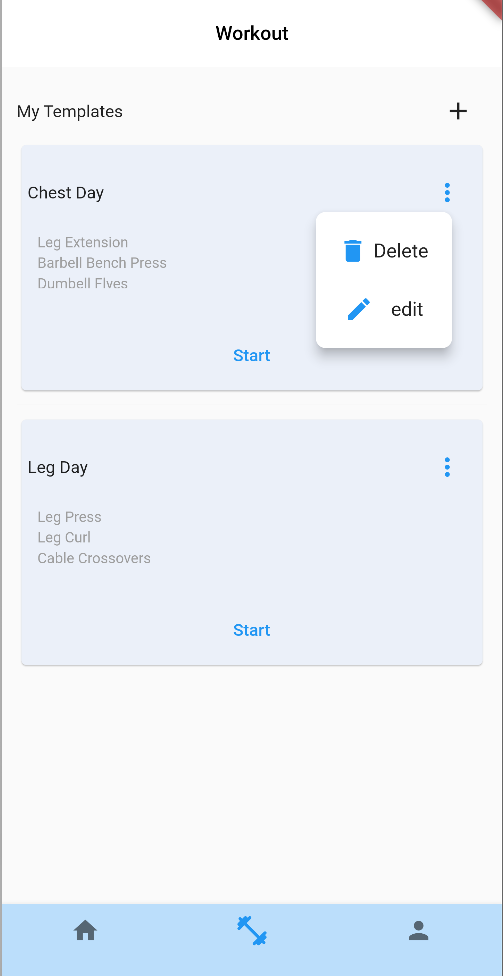
Εικόνα 6 - Κενό ιστορικό προπονήσεων.

### Προβολή λίστας προπονητικών προγραμμάτων/ προπονήσεων

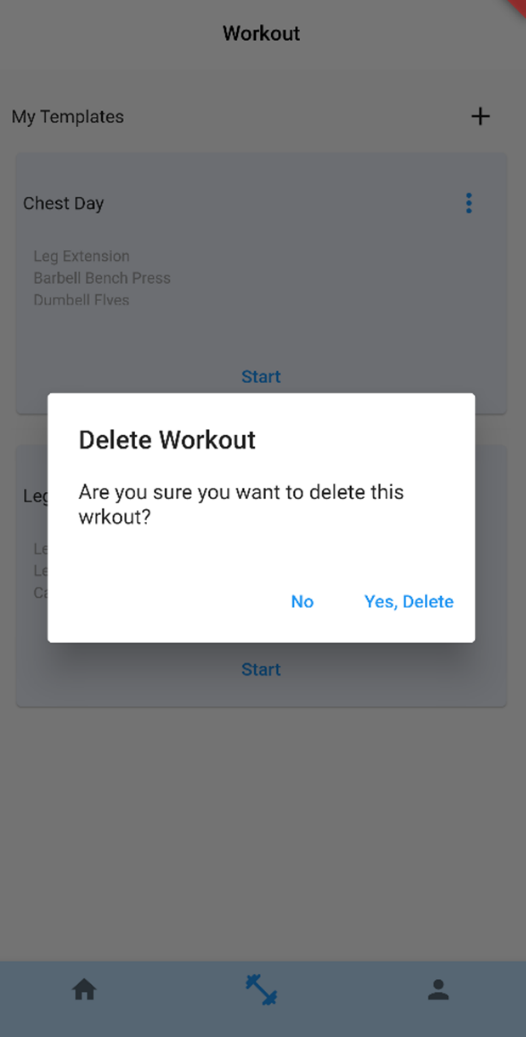
Κύριο στοιχείο της εφαρμογής είναι η δυνατότητα δημιουργίας προσαρμοσμένων προπονητικών προγραμμάτων. Μεταβαίνοντας το δεύτερο εικονίδιο της εφαρμογής ο χρήστης έρχεται σε επαφή με την λίστα των έτοιμων προπονητικών προγραμμάτων (που δημιουργήθηκαν κατά τη πρώτη σύνδεση του χρήστη), τα οποία ο χρήστης έχει τη δυνατότητα να επεξεργαστεί να διαγράψει. Επίσης έχει την δυνατότητα δημιουργήσει κάποιο νέο προπονητικό πρόγραμμα, καθώς και να πραγματοποιήσει κάποια προπόνηση.



Εικόνα 7 - Λίστα προπονήσεων.



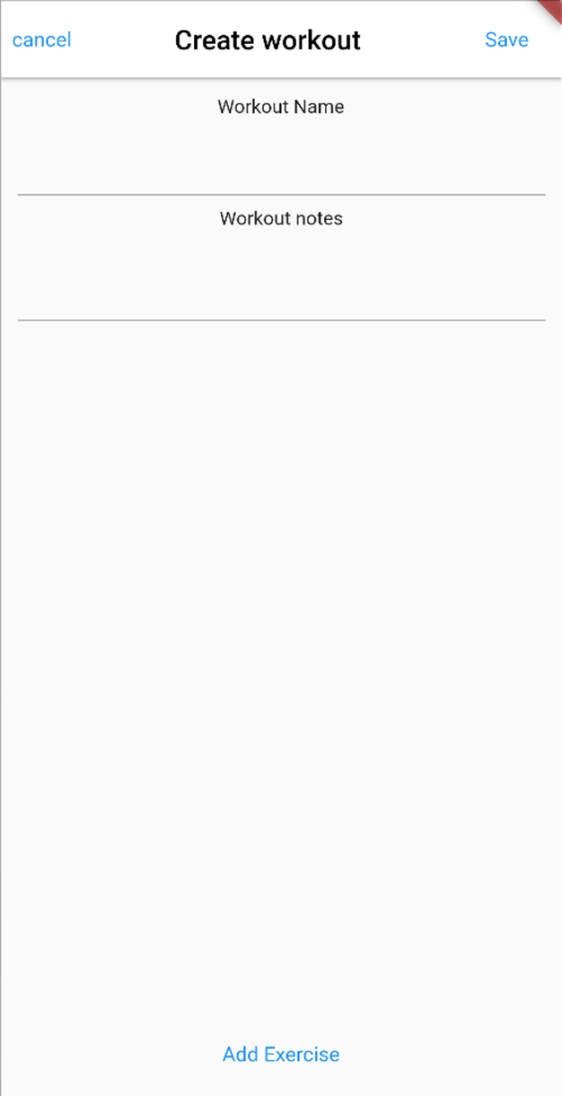
Εικόνα 8 - Επιπλέον επιλογές προπόνησης.



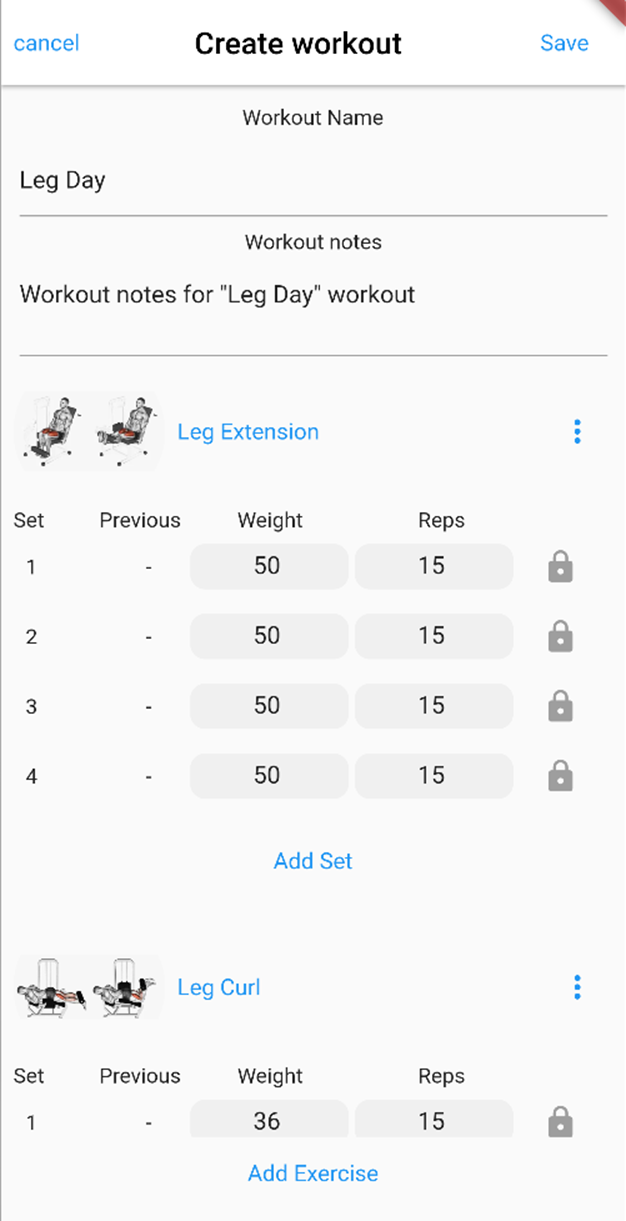
Εικόνα 9 - Μήνυμα διαγραφής προπόνησης.

### Δημιουργία/ επεξεργασία προπονητικού προγράμματος

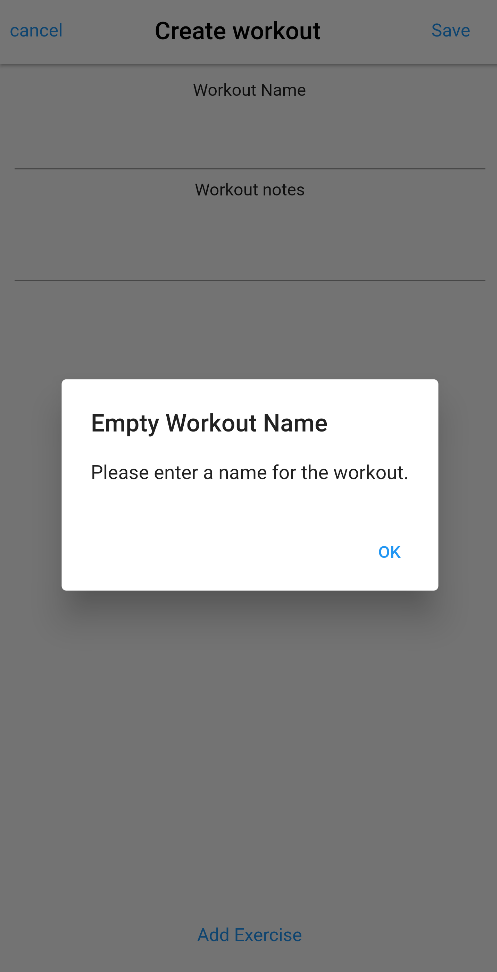
Ο χρήστης της εφαρμογής, κατά την δημιουργία και επεξεργασία κάποιου προπονητικού προγράμματος μπορεί να προσθέσει όνομα, κάποιες σημειώσεις καθώς και την λίστα ασκήσεων που θα πραγματοποιήσει κατά την διάρκεια την προπόνησης. Κάθε άσκηση, περιέχει έναν αριθμό σετ τα οποία περιέχουν τον σύνολο των βαρών σε κιλά, το σύνολο επαναλήψεων τον αριθμό του σετ καθώς και το σύνολο των προηγουμένων σετ (κιλά x επαναλήψεις) Αν δεν έχει εκτελέσει την συγκεκριμένη προπόνηση στο παρελθόν, το πεδίο είναι “-“. Ο χρήστης μπορεί να προσθέσει ασκήσεις πατώντας το κουμπί “Add exercise”.



Εικόνα 10 - Δημιουργία νέας προπόνησης.



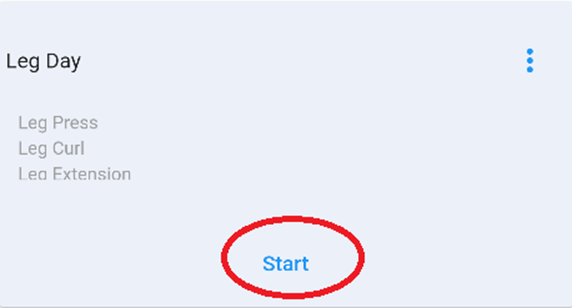
Εικόνα 11 - Προπόνηση με συμπληρωμένα πεδία.



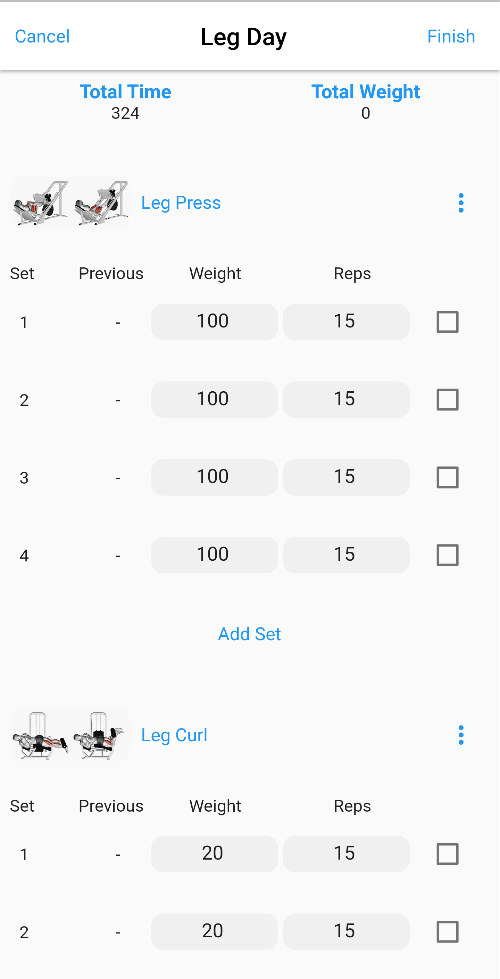
Εικόνα 12 - Μήνυμα σφάλματος.

### Πραγματοποίηση προπόνησης.

Μια προπόνηση μπορεί να πραγματοποιηθεί πατώντας την επιλογή “Start” στο προπονητικό πρόγραμμα της επιλογής του χρήστη.

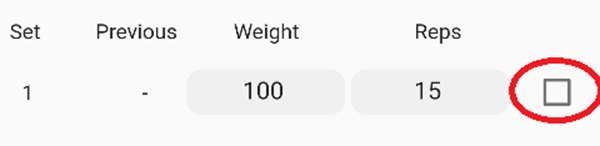


Εικόνα 13 - Έναρξη προπόνησης.



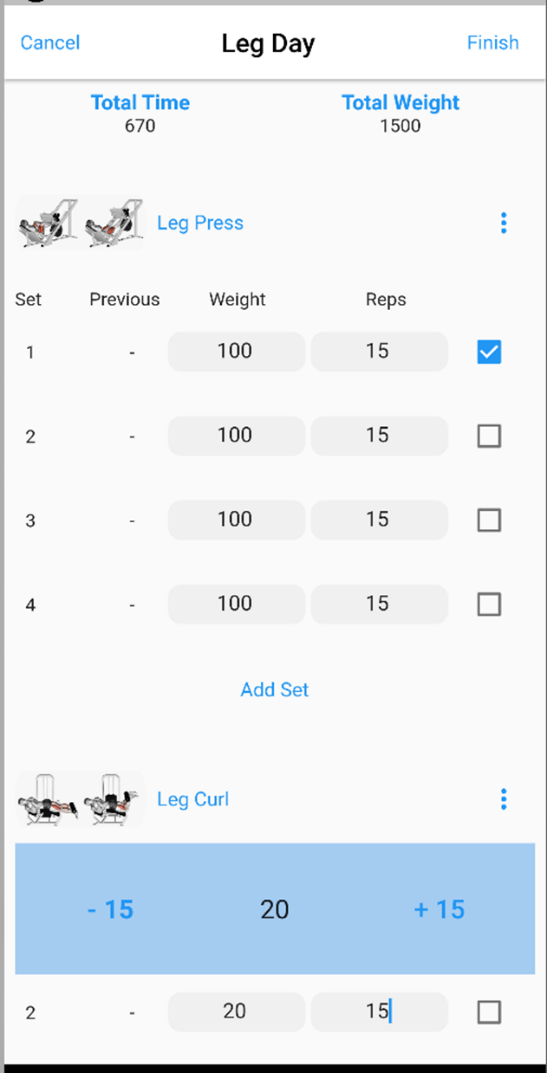
Εικόνα 14 - Σελίδα πραγματοποίησης προπόνησης.

Κατά την διάρκεια της προπόνησης, ο χρήστης μπορεί να επεξεργαστεί τα βάρη και τις επαναλήψεις των σετ, μπορεί να προσθέσει και να αφαιρέσει ασκήσεις. Επίσης, μια διαφορά από την σελίδα της δημιουργίας/επεξεργασίας είναι ότι ο χρήστης, μπορεί να δηλώσει μια άσκηση ως ολοκληρωμένη.



Εικόνα 15 - Δήλωση άσκησης ως ολοκληρωμένη.

Αυτό έχει ως αποτέλεσμα να εμφανιστεί ένας μετρητής στο κάτω μέρος, ο οποίος λειτουργεί ως χρονόμετρο για αποθεραπεία μέχρι το επόμενο σετ.



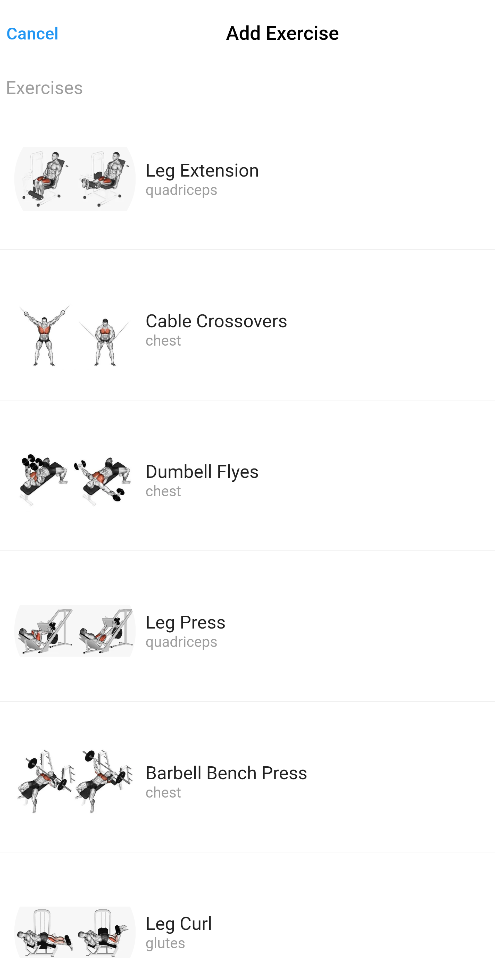
Εικόνα 16 - Χρονόμετρο αποθεραπείας.

Τέλος, στο πάνω μέρος της σελίδας ο χρήστης μπορεί να δει τον συνολικό χρόνο της προπόνησης, και το σύνολο κιλών των ασκήσεων (κιλά \* επαναλήψεις).

Πατώντας το κουμπί “Finish”, η προπόνηση αποθηκεύεται στην Google Firebase βάση δεδομένων και δημιουργείται μία εγγραφή της προπόνησης στο ιστορικό καταγραφής.

### Λίστα ασκήσεων

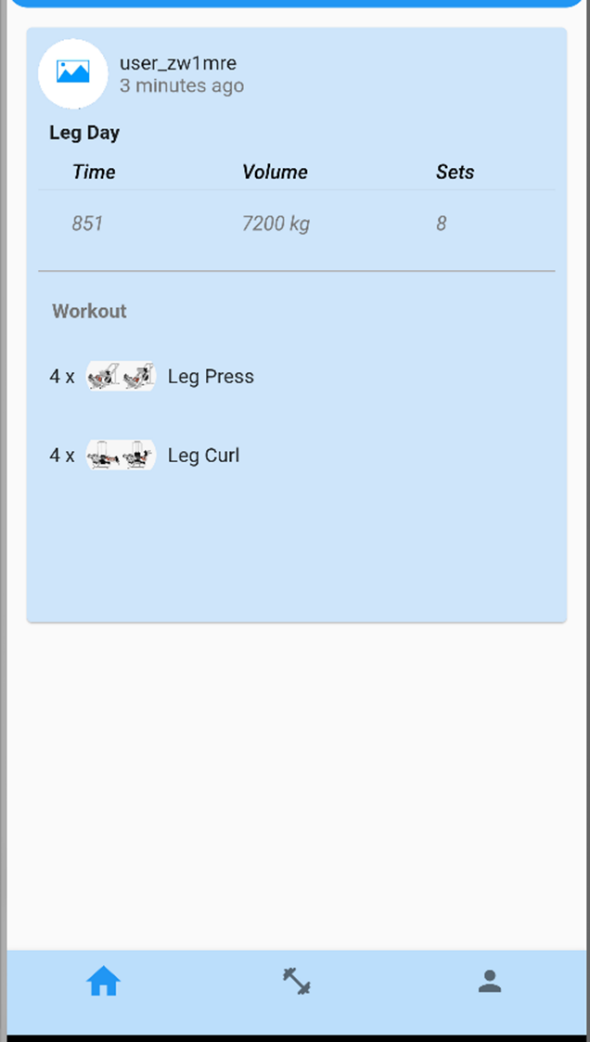
Ο χρήστης, στην οθόνη της προπόνησης (προσθήκη, επεξεργασία, πραγματοποίηση) έχει την επιλογή να προσθέσει ασκήσεις. Πατώντας το κουμπί “Add exercise” εμφανίζεται μια λίστα με ασκήσεις που δημιουργήθηκαν κατά την πρώτη σύνδεση του χρήστη και πατώντας πάνω στην άσκηση, αυτή προστίθεται στην προπόνηση.



Εικόνα 17 - Λίστα ασκήσεων.

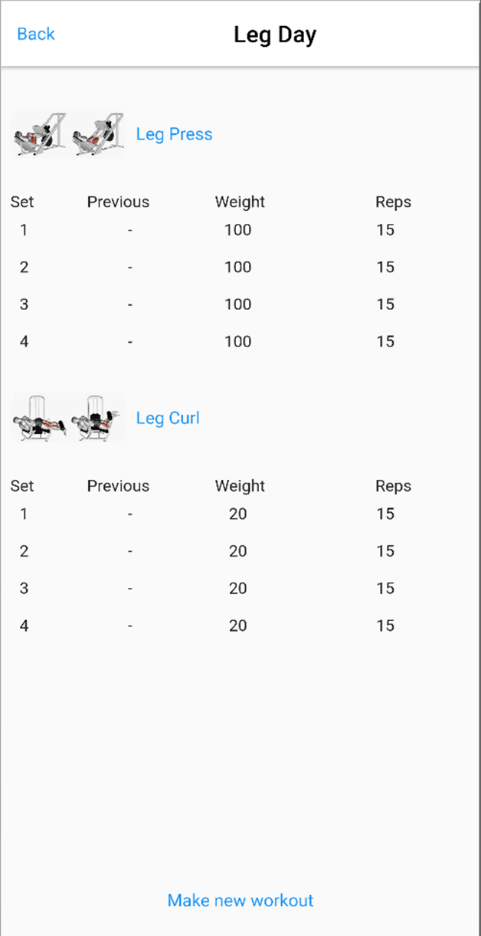
### Διαχείριση του Ιστορικού

Οι χρήστες έχουν πρόσβαση στο ιστορικό των προπονητικών προγραμμάτων που έχουν ακολουθήσει στο παρελθόν.



Εικόνα 18 - Ιστορικό προπονήσεων με εγγραφές.

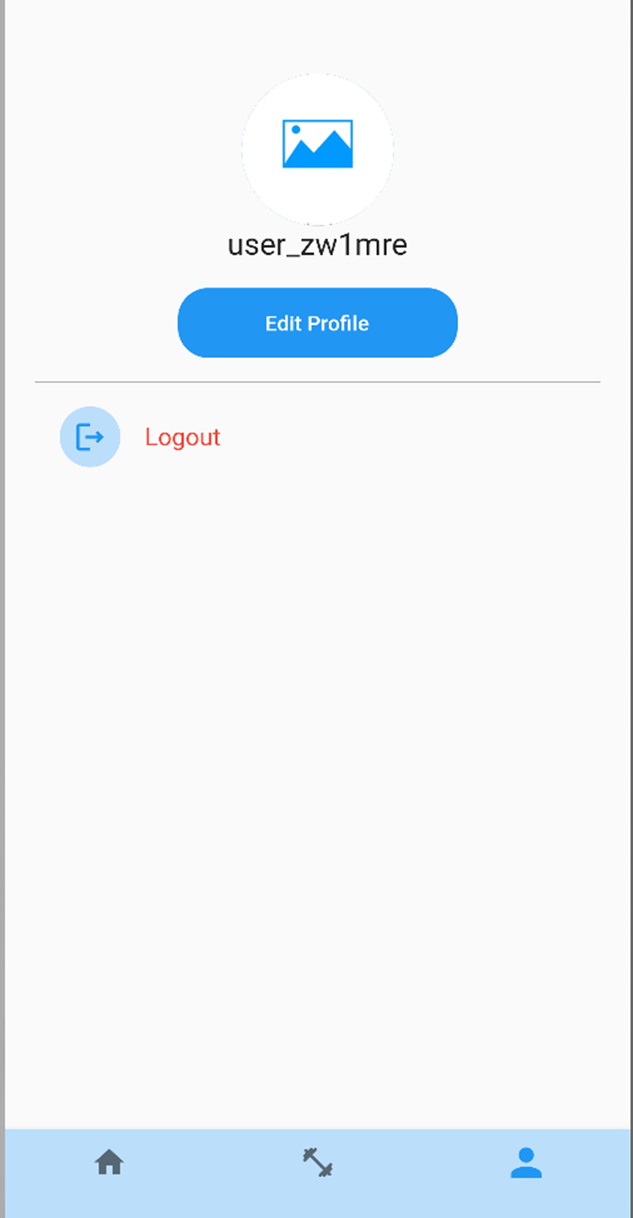
Ένας χρήστης εκτός από την απλή παρακολούθηση του ιστορικού, έχει και την επιλογή να δει περισσότερες πληροφορίες για την προπόνηση πατώντας πάνω στο τετράγωνο της επιλογής του, όπως επίσης να δημιουργήσει νέο προπονητικό πρόγραμμα βασισμένο σε αυτό πατώντας το “Make new workout”.



Εικόνα 19 - Επιπλέον πληροφορίες από το ιστορικό προπονήσεων.

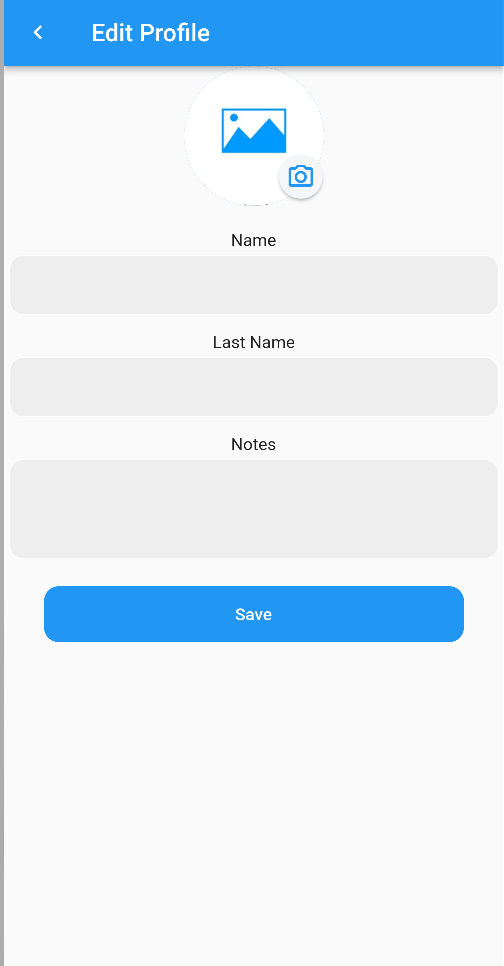
### Επεξεργασία Προσωπικού Προφίλ

Πατώντας στο τρίτο εικονίδιο, ο χρήστης έρχεται σε επαφή με την οθόνη που περιέχει πληροφορίες για αυτόν καθώς και την επιλογή να αποσυνδεθεί.



Εικόνα 20 - Σελίδα χρήστη.

Πατώντας το κουμπί “Edit profile”, να επεξεργαστεί ορισμένες πληροφορίες όπως το όνομα, το επίθετο του, σημειώσεις καθώς και μπορεί να προσθέσει κάποια εικόνα.



Εικόνα 21 - Σελίδα επεξεργασίας προσωποποιημένων δεδομένων χρήστη.

# **Συμπεράσματα – Μελλοντικές κατευθύνσεις / Conclusions – Future Directions**

Στο πλαίσιο αυτής της μελέτης, διερευνήθηκε η δημιουργία και η λειτουργία της εφαρμογής "Workout Tracker" σε περιβάλλον κινητής τεχνολογίας. Η εφαρμογή αυτή αναπτύχθηκε με στόχο να παρέχει ένα αξιόπιστο και εύχρηστο εργαλείο για την καταγραφή, την παρακολούθηση και τη βελτίωση της φυσικής κατάστασης των χρηστών. Η εφαρμογή αυτή μας βοήθησε να κατανοήσουμε καλύτερα την λειτουργία του Flutter και είδαμε πόσο εύχρηστο εργαλείο είναι.

Η Google αλλά και η open source κοινότητα προγραμματιστών της έχουν κάνει μια υπέροχη προσπάθεια και το αποτέλεσμα είναι φανταστικό. Το Flutter είναι ένα καλό εργαλείο για την δημιουργία εφαρμογών και η διασύνδεση του με το Firebase κάνει την δημιουργία εφαρμογών εύκολη και Ευχάριστη.

Μελλοντικά, η εφαρμογή μπορεί να βελτιωθεί ακόμα περισσότερο με τους εξής τρόπους:

* Ο χρήστης να μπορεί να προσθέσει και επεξεργαστεί δικές του ασκήσεις.
* Ο χρήστης να μπορεί να καταγράφει περισσότερα στοιχεία για τον εαυτό του.
* Ο χρήστης να μπορεί να κρατάει ιστορικό για περισσότερα στοιχεία για τον εαυτό του (πχ κιλά)
* Να δημιουργηθεί ένα δίκτυο χρηστών οι οποίοι θα μπορούν να μοιράζονται προπονήσεις μεταξύ τους και κατακτήσεις. (πχ. Νέα ρεκόρ)
* Να γίνει ενσωμάτωση με κάποια πλατφόρμα όπως το MyFitnessPal (<https://www.myfitnesspal.com>) με την οποία θα μπορεί να καταγράφει δεδομένα όπως θερμίδες που καταναλώνει.

Όλα αυτά είναι αρκετά εύκολα να γίνουν λόγω των δυνατοτήτων που αναφέρθηκαν.

# **Βιβλιογραφία / References**

## Βιβλία

Marko, G. (2011). Learning android. *Editorial O’Reilly Media*.

|  |
| --- |
| Miola, A. (2020). Flutter Complete Reference: Create beautiful, fast and native apps for any device. *(No Title)*. |
| Eisenman, B. (2015). *Learning react native: Building native mobile apps with JavaScript*. " O'Reilly Media, Inc.".  Boduch, A. (2017). *React and react native*. Packt Publishing Ltd. |

## Διαδίκτυο

<https://skia.org/>

<https://docs.flutter.dev/>

<https://flutter.dev/>

<https://firebase.google.com/>

<https://firebase.flutter.dev/>

<https://dart.dev/>

<https://docs.fluter.dev/>

<https://reactnative.dev/>

<https://en.wikipedia.org/wiki/React_Native>

https://en.wikipedia.org/wiki/IOS

https://appinventiv.com/blog/cross-platform-app-frameworks/

# **Παραρτήματα / Appendixes**

## Τοποθεσία κώδικα

https://github.com/DVeneris/training\_logger.git