



ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΕΠΙΚΟΙΝΩΝΙΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Τίτλος Πτυχιακής Εργασίας	(Ελληνικά) Android εφαρμογή διαχείρισης προγραμμάτων γυμναστικής και παρακολούθησης προόδου. (Αγγλικά) Android app for fitness program management and progress tracking
Όνοματεπώνυμο Φοιτητή	ΜΠΟΜΠΟΤΗΣ ΠΕΤΡΟΣ
Πατρώνυμο	ΓΕΩΡΓΙΟΣ
Αριθμός Μητρώου	Π20140
Επιβλέπων	ΕΥΘΥΜΙΟΣ ΑΛΕΠΗΣ, Καθηγητής

Copyright ©

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.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν αποκλειστικά τον συγγραφέα και δεν αντιπροσωπεύουν τις επίσημες θέσεις του Πανεπιστημίου Πειραιώς.

Ως συγγραφέας της παρούσας εργασίας δηλώνω πως η παρούσα εργασία δεν αποτελεί προϊόν λογοκλοπής και δεν περιέχει υλικό από μη αναφερόμενες πηγές.

2-Member Examination Committee

Διμελής Εξεταστική Επιτροπή

Efthimios Alepis Professor

Ευθύμιος Αλέπης Καθηγητής

Maria Virvou Professor

Μαρία Βίρβου Καθηγήτρια

Περιεχόμενα

Copyright ©	2
Περίληψη (Abstract).....	6
English.....	6
Ελληνικά	6
Εισαγωγή.....	7
Επισκόπηση Αρχιτεκτονικής	8
Αρχικό στήσιμο εφαρμογής	10
Στήσιμο του back-end	10
Στήσιμο της βάσης δεδομένων	10
Spring Boot.....	11
Application Context και Beans	11
Spring Data JPA	12
Υλοποίηση ασφάλειας	14
Κλάση Customer.....	14
Υλοποίηση ρόλων χρηστών.....	14
JSON Web Token	15
Κλάση SecurityFilterChainConfig.....	15
Κλάση SecurityConfig.....	16
Κλάση JWTAuthenticationFilter	17
Διαδικασία Αυθεντικοποίησης	18
Υλοποίηση του refresh token.....	18
Χειρισμός exception σχετικά με την ασφάλεια	19
Διάγραμμα της βάσης δεδομένων	20
Κλάσεις για την υλοποίηση της αρχιτεκτονικής	21
Annotations.....	22
Spring Web MVC Annotations	22
Spring Annotations	23
Στήσιμο front-end.....	24
Android UI Components	25
Bottom Navigation Bar και Fragments	25
RecyclerView	26
Tab layout και ViewPager	27
Εγχειρίδιο χρήσης (για το χρήστη).....	28
Εγγραφή και Σύνδεση	28
Ενότητα 1: Δημιουργία και διαχείριση του προγράμματός	30
Ενότητα 2: Εξερεύνηση και κοινή χρήση προγραμμάτων	35
Ενότητα 3: Παρακολούθηση των μετρήσεων του σώματος	37
Ενότητα 4: Διαχείριση του προφίλ.....	38
Εγχειρίδιο χρήσης (για το διαχειριστή).....	39

Μελλοντικές επεκτάσεις της εφαρμογής	43
Συμπεράσματα	44
Βιβλιογραφία.....	45

Περίληψη (Abstract)

English

This thesis focuses on the implementation of an Android application, which involves the creation of fitness programs, using the Spring Boot tool, in partnership with Spring Security and Spring Data JPA for the back-end, a PostgreSQL database and the Android UI components offered by the Android SDK for the frontend.

The app provides registered users with the ability to create personalized fitness programs, track their progress and publish their programs to the community section, making them available for download and use by other users.

The purpose of the app is to provide a comprehensive and easy-to-use platform that supports users in achieving their health and fitness goals through personalization, progress tracking and sharing knowledge and programs within the community.

Ελληνικά

Η παρούσα πτυχιακή επικεντρώνεται στην υλοποίηση μιας Android εφαρμογής, η οποία αφορά τη δημιουργία προγραμμάτων γυμναστικής, με τη χρήση του εργαλείου Spring Boot, σε συνεργασία με Spring Security και Spring Data JPA για το back-end, μιας PostgreSQL βάσης και των Android UI components που προσφέρονται από το Android SDK για το front-end.

Η εφαρμογή παρέχει στους εγγεγραμμένους χρήστες τη δυνατότητα να δημιουργήσουν εξατομικευμένα προγράμματα γυμναστικής, να παρακολουθούν της πρόοδό τους και να δημοσιεύουν τα προγράμματά τους στην ενότητα της κοινότητας, κάνοντάς τα διαθέσιμα για λήψη και χρήση από άλλους χρήστες.

Ο σκοπός της εφαρμογής είναι να προσφέρει μια ολοκληρωμένη και εύχρηστη πλατφόρμα που υποστηρίζει τους χρήστες στην επίτευξη των στόχων τους για υγεία και καλή φυσική κατάσταση, μέσα από την εξατομίκευση, την παρακολούθηση της προόδου και την ανταλλαγή γνώσεων και προγραμμάτων μέσα στην κοινότητα.

Εισαγωγή

Στον σημερινό ταχέως εξελισσόμενο κόσμο, η διατήρηση της καλής φυσικής κατάστασης είναι απαραίτητη για τη συνολική υγεία και ευεξία. Ωστόσο, η δημιουργία και η τήρηση ενός προγράμματος γυμναστικής μπορεί να αποτελέσει πρόκληση για πολλά άτομα. Υπάρχει ανάγκη για μια ολοκληρωμένη λύση που επιτρέπει στους χρήστες να σχεδιάζουν εξατομικευμένα προγράμματα γυμναστικής, να παρακολουθούν την πρόδοό τους και να συμμετέχουν σε μια κοινότητα ομοϊδεατών.

Η παρούσα εφαρμογή στοχεύει να καλύψει αυτές τις ανάγκες παρέχοντας μια ολοκληρωμένη πλατφόρμα για τους χρήστες. Η εφαρμογή επιτρέπει στους χρήστες να δημιουργούν εξατομικευμένα προγράμματα γυμναστικής προσαρμοσμένα στους συγκεκριμένους στόχους τους, με κάθε πρόγραμμα να περιλαμβάνει διάφορες ασκήσεις που έχουν προσαρμοσμένα σετ, όπου δύνανται να καταγράψουν επαναλήψεις, φορτία και σημειώσεις. Ακόμα, παρέχει στους χρήστες τη δυνατότητα να παρακολουθούν την πρόδοό τους καταγράφοντας το βάρος και το ποσοστό σωματικού λίπους, διατηρώντας ένα ιστορικό αυτών των μετρήσεων. Επιπλέον, οι χρήστες μπορούν να συμμετέχουν στην κοινότητα, διαμοιράζοντας τα προγράμματα γυμναστικής τους στην αντίστοιχη ενότητα, όπου άλλοι χρήστες μπορούν να τα δουν και να τα κατεβάσουν, καλλιεργώντας έτσι ένα υποστηρικτικό περιβάλλον και ενθαρρύνοντας την ανταλλαγή αποτελεσματικών στρατηγικών γυμναστικής.

Η εφαρμογή έχει κατασκευαστεί χρησιμοποιώντας έναν συνδυασμό τεχνολογιών. Στη πλευρά του back-end, χρησιμοποιήθηκε Java σε συνδυασμό με τα εργαλεία Spring Boot και Spring Data JPA για τη διαχείριση τόσο του business logic όσο και των δεδομένων μας. Η ασφάλεια της εφαρμογής ενισχύεται μέσω του Spring Security σε συνδυασμό με JSON Web Tokens (JWT), τα οποία παρέχουν αυθεντικοποίηση χρηστών με βάση τον/τους ρόλους τους. Αυτή η διάταξη προσφέρει ένα επιπλέον επίπεδο προστασίας, εξασφαλίζοντας την ακεραιότητα των δεδομένων των χρηστών και την ασφαλή πρόσβαση στις λειτουργίες του back-end. Ως βάση, χρησιμοποιούμε την PostgreSQL για την αποθήκευση δεδομένων χρηστών, εξασφαλίζοντας την αποτελεσματική διαχείριση των πληροφοριών. Η πλευρά του Android, υλοποιήθηκε με Java και τα Android UI components που προσφέρει το Android SDK. Η εφαρμογή συνδυάζει αυτές τις τεχνολογίες για να δημιουργήσει μια φιλική εμπειρία προς το χρήστη.

Ιδιαίτερη έμφαση δίνεται στην εμπειρία χρήστη και την προσβασιμότητα, διασφαλίζοντας ότι άτομα όλων των επιπέδων φυσικής κατάστασης μπορούν εύκολα να περιηγηθούν και να επωφεληθούν από τις λειτουργίες της. Η διεπαφή καθοδηγεί τους χρήστες κατά τη διαδικασία δημιουργίας και διαχείρισης των προγραμμάτων γυμναστικής τους, ενώ οι σαφείς απεικονίσεις και τα εργαλεία παρακολούθησης της προόδου παρέχουν πολύτιμες πληροφορίες για το ταξίδι της υγείας τους. Επιπλέον, η εφαρμογή έχει σχεδιαστεί για να είναι προσβάσιμη σε διάφορες συσκευές, συμπεριλαμβανομένων των smartphones και των tablet, επιτρέποντας στους χρήστες να την ενσωματώσουν απρόσκοπτα στην καθημερινή τους ρουτίνα.

Σε αυτό το σημείο, αξίζει να αναφέρουμε τα οφέλη αυτής της εφαρμογής που θα προκύψουν για τον ίδιο τον χρήστη. Πιο συγκεκριμένα, γεφυρώνει το χάσμα μεταξύ των ατομικών στόχων γυμναστικής και της συμμετοχής στην κοινότητα, προσφέροντας στους χρήστες μια μοναδική πλατφόρμα για την επίτευξη των στόχων τους. Μέσω των προηγμένων χαρακτηριστικών της και του σχεδιασμού της με επίκεντρο τον χρήστη, ξεχωρίζει ως ένα πολύτιμο εργαλείο για όσους επιθυμούν να ενισχύσουν το ταξίδι τους στη γυμναστική.

Επισκόπηση Αρχιτεκτονικής

Η αρχιτεκτονική που ακολουθήθηκε για την υλοποίηση της εφαρμογής είναι η N-Tier αρχιτεκτονική. Αποτελείται από διάφορα επίπεδα όπως: την εφαρμογή Android που χρησιμεύει ως front-end και συνεπώς αποτελεί το Presentation Layer, τον back-end server Spring Boot που περιέχει τα REST Layer, Business Layer και DAO Layer, καθώς και τη βάση δεδομένων PostgreSQL.

Η εφαρμογή Android, αποτελεί το Presentation Layer και είναι το τμήμα της εφαρμογής με το οποίο ο χρήστης αλληλεπιδρά άμεσα. Είναι το πρώτο και μοναδικό επίπεδο με το οποίο έρχεται σε άμεση επαφή ο χρήστης. Παρέχει τη διεπαφή μέσω της οποίας οι χρήστες δύνανται να περιηγηθούν στην εφαρμογή και να κάνουν χρήση αυτής.

Ο back-end server χρησιμεύει ως ενδιάμεσος κρίκος μεταξύ της εφαρμογής Android και της βάσης δεδομένων. Η επικοινωνία μεταξύ τους πραγματοποιείται μέσω αιτημάτων HTTP. Από την πλευρά του Android στέλνονται αιτήματα HTTP στο back-end και λαμβάνει απαντήσεις που περιέχουν τα ανάλογα δεδομένα. Το back-end χειρίζεται τα εισερχόμενα αιτήματα, επεξεργάζεται τα δεδομένα και αλληλεπιδρά με τη βάση δεδομένων PostgreSQL. Περιλαμβάνει διάφορα επίπεδα ώστε αντί να συμβαίνουν όλες αυτές οι ενέργειες σε ένα επίπεδο, να διαχωριστούν οι διάφορες «ευθύνες».

Το επίπεδο που αναχαιτίζει αυτά τα αιτήματα είναι το REST Layer. Σε αυτό το επίπεδο έχουμε τα API endpoints τα οποία υλοποιούν λειτουργίες που σχετίζονται με την αυθεντικοποίηση των χρηστών, τη δημιουργία προγραμμάτων γυμναστικής, την παρακολούθηση της προόδου, τα χαρακτηριστικά της κοινότητας και άλλα.

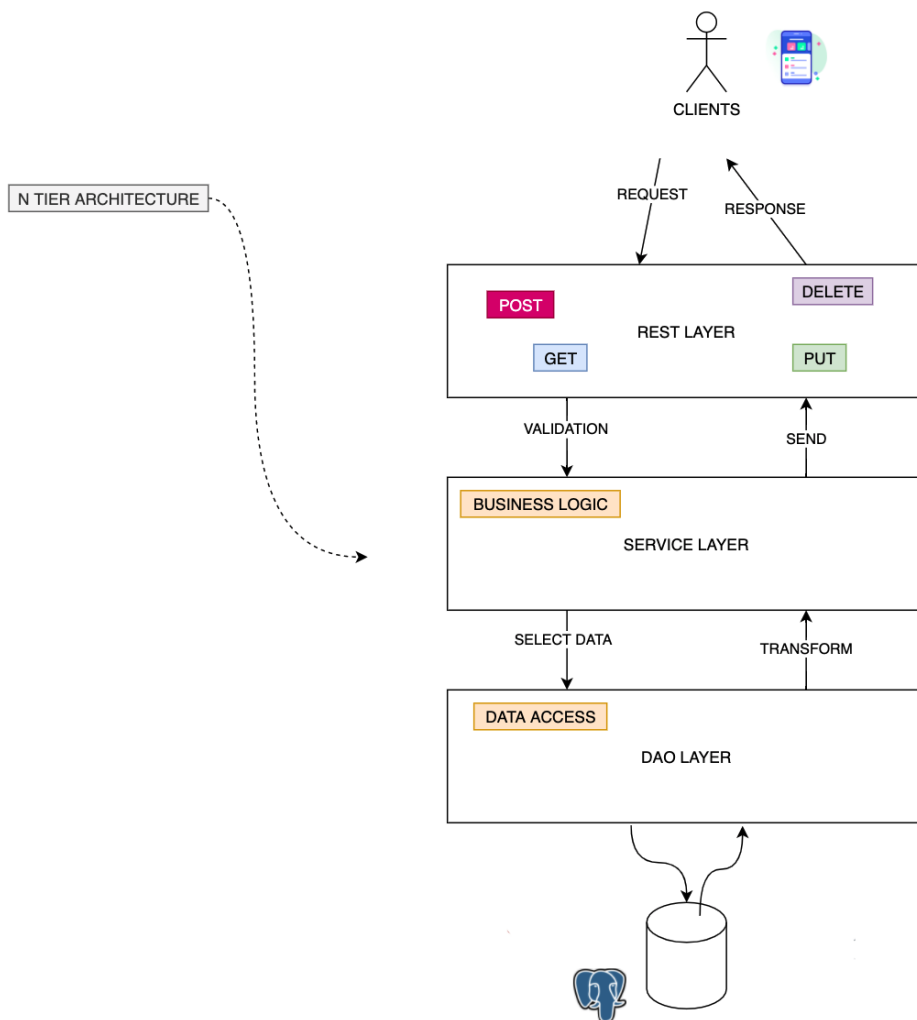
Στη συνέχεια, το REST Layer στέλνει τα αιτήματα στο Business Layer. Εδώ βρίσκεται το business logic για ολόκληρο το back-end. Το business logic περιέχει τη βασική λογική για τη δημιουργία και διαχείριση προγραμμάτων γυμναστικής, την παρακολούθηση της προόδου αλλά και πιο τεχνικών λειτουργιών όπως τα διάφορα exception.

Τέλος το Business Layer επικοινωνεί με το DAO Layer το οποίο είναι υπεύθυνο για την αλληλεπίδραση με την βάση δεδομένων μας. Η βάση δεδομένων PostgreSQL αποθηκεύει όλα τα δεδομένα που σχετίζονται με τους χρήστες, τα προγράμματα γυμναστικής, τις πληροφορίες προόδου και άλλα σχετικά δεδομένα. Η αλληλεπίδραση με αυτήν γίνεται μέσω ερωτημάτων SQL.

Αυτή η αρχιτεκτονική παρέχει σαφή διαχωρισμό των «ευθυνών» και διευκολύνει την αποτελεσματική επικοινωνία μεταξύ των front-end, back-end και βάσης δεδομένων της εφαρμογής.

Αξίζει να σημειωθεί πως υπάρχει και το επίπεδο ασφαλείας, που υλοποιείται με τη χρήση του Spring Security και του JWT και διασφαλίζει την ασφαλή πρόσβαση στην εφαρμογή, επιβάλλοντας αυθεντικοποίηση χρηστών.

Το παρακάτω διάγραμμα παρέχει μια σαφή επισκόπηση του τρόπου με τον οποίο αλληλεπιδρούν μεταξύ τους τα διάφορα μέρη της εφαρμογής. Δείχνει τη ροή των δεδομένων από την εφαρμογή Android στον back-end server και τελικά στη βάση δεδομένων PostgreSQL.



ΔΙΑΓΡΑΜΜΑ 1: Επισκόπηση αρχιτεκτονικής

Αρχικό στήσιμο εφαρμογής

Στήσιμο του back-end

Για την αρχικοποίηση της δομής του project, χρησιμοποιήθηκε το Spring Initializer. Το Spring Initializer είναι ένα διαδικτυακό εργαλείο που παρέχεται από την ομάδα της Spring για τη γρήγορη εκκίνηση του εκάστοτε Spring Boot project με τα απαιτούμενα dependencies και plugins.

Τα απαιτούμενα dependencies περιλαμβάνουν το Spring Web για τη δημιουργία RESTful APIs, το Spring Data JPA για την εύκολη αλληλεπίδραση με τη βάση δεδομένων χρησιμοποιώντας Object-Relational Mapping (ORM), το Spring Security για την υλοποίηση χαρακτηριστικών ασφαλείας όπως η αυθεντικοποίηση, το postgresql για την ενσωμάτωση της βάσης δεδομένων PostgreSQL, το jjwt για την υλοποίηση του JSON Web Token (JWT) καθώς και το java-faker για τη δημιουργία ψεύτικων δεδομένων για σκοπούς δοκιμών.

Εικόνα1: Spring Initializer

Στήσιμο της βάσης δεδομένων

Προχωρώντας στη ρύθμιση της βάσης δεδομένων, χρησιμοποιήθηκαν το Docker και η PostgreSQL για τη δημιουργία μιας containerized βάσης δεδομένων. Το Docker παρέχει έναν εύχρηστο τρόπο διαχείρισης και ανάπτυξης εφαρμογών, επιτρέποντας την εύκολη εγκατάσταση και τη φορητότητα σε διαφορετικά περιβάλλοντα.

Προκειμένου να υλοποιηθεί ένα PostgreSQL container, δημιουργήθηκε ένα αρχείο docker-compose.yml όπου ορίστηκαν τα configurations. Έπειτα με την εντολή docker compose up -d δημιουργείται το container.

Στο αρχείο application.yml του project, καθορίστηκαν οι λεπτομέρειες σύνδεσης με τη βάση δεδομένων. Αυτό περιλαμβάνει τη διεύθυνση URL της βάσης δεδομένων, το όνομα χρήστη, τον κωδικό πρόσβασης και άλλες πρόσθετες επιλογές διαμόρφωσης. Τα spring.datasource configurations ρυθμίστηκαν ώστε να «δείχνουν» τη βάση δεδομένων PostgreSQL που εκτελείται στο Docker container.

Με αυτόν τον τρόπο, η εφαρμογή είναι σε θέση να συνδεθεί με τη βάση δεδομένων, επιτρέποντας την απρόσκοπτη αλληλεπίδραση μεταξύ τους που κρίνεται απαραίτητη για την αποθήκευση, ανάκτηση και επεξεργασία δεδομένων όπως απαιτείται από τις λειτουργίες της εφαρμογής.

Spring Boot

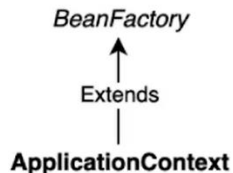
Το Spring Boot είναι ένα framework για την εύκολη κατασκευή αυτόνομων εφαρμογών. Είτε θέλει να δημιουργήσει κανείς back-end εφαρμογές είτε full-stack εφαρμογές χρησιμοποιώντας Java ή Kotlin, το Spring Boot είναι η κατάλληλη εφαρμογή. Το Spring Boot είναι μακράν το πιο δημοφιλές framework αυτή τη στιγμή. Αν χρειάζεται κανείς ασφάλεια, μπορεί να χρησιμοποιήσει τα διαθέσιμα module ασφαλείας. Προσφέρει εξαιρετική ευκολία αναφορικά με τη σύνδεση με βάσεις δεδομένων, είτε απαιτείται σύνδεση με MongoDB, PostgreSQL ή MySQL,. Περιλαμβάνει επίσης μετρήσεις για τον έλεγχο της συμπεριφοράς της εφαρμογής στην παραγωγή. Επιπλέον, έχει ενσωματωμένο dependency injection, μεγάλη κοινότητα και πολλά άλλα χαρακτηριστικά.

Application Context και Beans

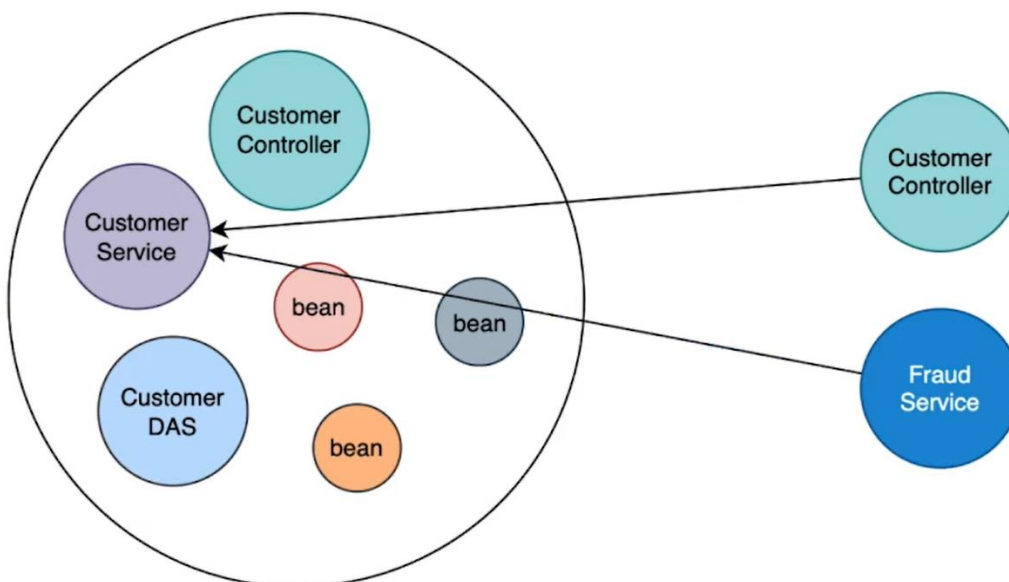
Όπως αναφέρθηκε το Spring Boot είναι ένα dependency injection framework. Ο συνιστάμενος τρόπος να γίνει dependency injection είναι να γίνουν instantiate και να «περαστούν» τα dependencies στον constructor. Αυτό σημαίνει πως πρέπει να δημιουργούνται instances αυτών για να περνιόνται ως ορίσματα. Σε ένα μεγάλο project όμως θα γέμιζε το heap με instances.

Την λύση στο πρόβλημα αυτό, την δίνουν τα Beans σε συνδυασμό με το application context. Τα Beans είναι Java objects τα οποία αντί να γίνονται instantiate με το new keyword, τα δημιουργεί το framework και από εκεί και έπειτα εκείνο είναι υπεύθυνο για την διαχείρισή τους, όπως παραδείγματος χάρη να τα κάνει inject σε άλλα beans.

Υπάρχει ένα αντικείμενο που λέγεται application context το οποίο παρέχει βασικές λειτουργίες σχετικές με την διαχείριση των Beans. Είναι το μέρος όπου αποθηκεύονται όλα τα Beans και κάθε φορά που πρέπει να κληθεί κάποιος για οποιαδήποτε ενέργεια, όπως inject, το διαχειρίζεται μόνο του. Έτσι, αν λόγω χάρη δύο κλάσεις θέλουν ως dependency το ίδιο Bean, αντί να γίνει instantiate δύο φορές, το framework φροντίζει να χρησιμοποιείται κάθε φορά το ίδιο instance.



It provides basic functionalities for managing beans.



ΔΙΑΓΡΑΜΜΑ 2: Application Context και Beans

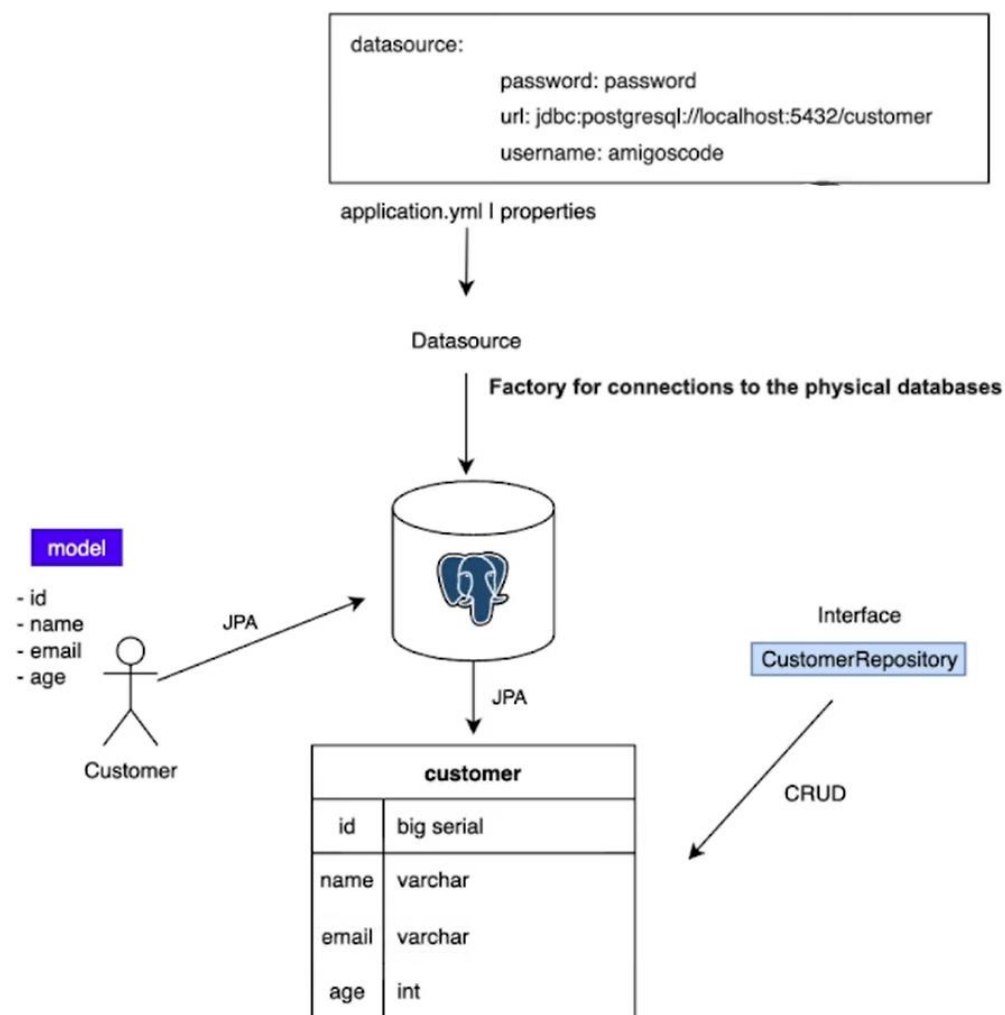
Spring Data JPA

Το Spring Data JPA (Java Persistence API) είναι μια ισχυρή επέκταση του Spring framework που βελτιώνει το επίπεδο πρόσβασης στα δεδομένα μιας εφαρμογής Spring. Παρέχει ένα επίπεδο αφαίρεσης και απλοποιεί τις αλληλεπιδράσεις με τη βάση δεδομένων. Το Spring Data JPA μειώνει τον boilerplate κώδικα που απαιτείται για τις λειτουργίες της βάσης δεδομένων, επιτρέποντας στους προγραμματιστές να επικεντρωθούν στη βασική λειτουργικότητα της εφαρμογής. Προσφέρει άνετη ενσωμάτωση με το Spring Boot, καθιστώντας εύκολη την εγκατάσταση και τη χρήση του σε σύγχρονες εφαρμογές Spring.

Ένας από τους πρωταρχικούς ρόλους του Spring Data JPA είναι το «entities mapping». Τα entities αντιπροσωπεύουν τους πίνακες σε μια σχεσιακή βάση δεδομένων και είναι κλάσεις στη Java. Το Spring Data JPA τις χρησιμοποιεί για να αντιστοιχίσει τα πεδία των κλάσεων σε στήλες της βάσης δεδομένων. Αυτή η διαδικασία αντιστοίχισης είναι ζωτικής σημασίας για το framework για την ακριβή διαχείριση και διατήρηση των entities. Μέσω annotation όπως @Entity, @Table, @Id και @Column, μπορεί κανείς να καθορίσει τον τρόπο με τον οποίο τα αντικείμενα Java αντιστοιχίζονται σε πίνακες και πεδία της βάσης δεδομένων. Επιπλέον, οι σχέσεις μεταξύ των entities, όπως οι σχέσεις ένα προς ένα, ένα προς πολλά και πολλά προς πολλά, διαχειρίζονται εύκολα μέσω annotation όπως @OneToOne, @OneToMany και @ManyToMany, διευκολύνοντας τα σύνθετα μοντέλα δεδομένων και διασφαλίζοντας την ακεραιότητα των δεδομένων.

Οι λειτουργίες CRUD (Δημιουργία, Ανάγνωση, Ενημέρωση, Διαγραφή) είναι θεμελιώδεις για κάθε αλληλεπίδραση με βάσεις δεδομένων και το Spring Data JPA τις απλοποιεί μέσω της διεπαφής JpaRepository. Με την επέκταση του JpaRepository, αποκτά κανείς πρόσβαση σε μια σειρά προκαθορισμένων μεθόδων για την εκτέλεση λειτουργιών CRUD χωρίς να χρειάζεται να γράψει boilerplate κώδικα. Μέθοδοι όπως save(), findById(), findAll(), deleteById() και deleteAll() παρέχουν την απαραίτητη λειτουργικότητα για τον χειρισμό βασικών λειτουργιών της βάσης δεδομένων. Επιπλέον, το Spring Data JPA υποστηρίζει προσαρμοσμένες μεθόδους ερωτημάτων που ορίζονται μέσω συμβάσεων ονοματοδοσίας μεθόδων ή του annotation @Query, επιτρέποντας πιο σύνθετα ερωτήματα και λειτουργίες προσαρμοσμένες σε συγκεκριμένες ανάγκες εφαρμογών. Αυτό καθιστά τη διαχείριση των δεδομένων απλή και αποτελεσματική, επιταχύνοντας σημαντικά την ανάπτυξη.

Το παρακάτω διάγραμμα παρέχει μια σαφή επισκόπηση του τρόπου με τον οποίο αλληλεπιδρούν μεταξύ τους τα παραπάνω components της εφαρμογής. Δείχνει τις απαιτούμενες ρυθμίσεις για την σύνδεση με τη βάση καθώς και πως γίνεται η αντιστοίχιση ενός entity στη βάση δεδομένων.



ΔΙΑΓΡΑΜΜΑ 3: Ρυθμίσεις Σύνδεσης και Αντιστοίχιση Entities σε Βάση Δεδομένων

Υλοποίηση ασφάλειας

Κλάση Customer

Προκειμένου το back-end να είναι ασφαλές, δημιουργήθηκε μία κλάση Customer για να αντιπροσωπεύει τις πληροφορίες των χρηστών στη βάση δεδομένων. Αυτή η κλάση περιλαμβάνει βασικά πεδία όπως το id, το όνομα χρήστη, τον κωδικό πρόσβασης και τους ρόλους, τα οποία αποθηκεύουν τις απαραίτητες πληροφορίες για την αυθεντικοποίηση του χρήστη. Επιπλέον, η κλάση Customer υλοποιεί τη διεπαφή UserDetails για να παρέχει τις απαιτούμενες μεθόδους που ανακτούν πεδία που σχετίζονται με την αυθεντικοποίηση.

Για το χειρισμό των αλληλεπιδράσεων της βάσης δεδομένων με τον Customer, υλοποιήθηκε ένα repository που επεκτείνει το JpaRepository. Με αυτό, κατέστη δυνατή η αποτελεσματική εκτέλεση CRUD λειτουργιών στην κλάση Customer χρησιμοποιώντας τις ενσωματωμένες μεθόδους της Spring Data JPA. Αυτή η προσέγγιση απλοποιεί τις λειτουργίες της βάσης δεδομένων και εξασφαλίζει αποτελεσματική διαχείριση δεδομένων.

Επιπλέον, υλοποιήθηκε η κλάση CustomerUserService, η οποία υλοποιεί τη διεπαφή UserDetailsService. Αυτή η διεπαφή περιέχει μία μόνο μέθοδο, την loadUserByUsername, η οποία έγινε override για να επιστραφεί ένα αντικείμενο UserDetails. Αυτό το αντικείμενο παρέχει πρόσβαση σε πεδία χρήστη που σχετίζονται με την αυθεντικοποίηση. Ως όνομα χρήστη επιλέχθηκε το πεδίο email που είναι «unique» και έτσι, με τη μέθοδο selectUserByEmail, μπορώ να ανακτήσω τα πεδία του χρήστη. Αυτή η ρύθμιση επιτρέπει μια βελτιωμένη και αποτελεσματική διαδικασία αυθεντικοποίησης στην εφαρμογή.

Υλοποίηση ρόλων χρηστών

Η κλάση Role αναπαριστά τον ρόλο του χρήστη μέσα στην εφαρμογή. Αυτή η κλάση είναι ζωτικής σημασίας για τον ορισμό των διαφορετικών ρόλων που μπορούν να ανατεθούν σε χρήστες, όπως ROLE_ADMIN ή ROLE_USER. Υλοποιεί τη διεπαφή GrantedAuthority που παρέχει η Spring Security. Η διεπαφή αυτή είναι απαραίτητη, διότι επιτρέπει στην Spring Security να αναγνωρίζει και να χρησιμοποιεί τους ρόλους που έχουν εκχωρηθεί στους χρήστες. Εάν δεν υλοποιούσε την GrantedAuthority, ο μηχανισμός αυθεντικοποίησης βάσει ρόλων δεν θα λειτουργούσε σωστά, καθώς η Spring Security βασίζεται σε αυτή τη διεπαφή για την ερμηνεία των ρόλων των χρηστών. Η κλάση Role περιλαμβάνει πεδία όπως το id, το οποίο χρησιμεύει ως πρωτεύον κλειδί για την κλάση role, και το authority, το οποίο αντιπροσωπεύει το όνομα του ρόλου.

Η διεπαφή RoleRepository επεκτείνει το JpaRepository από το Spring Data JPA, επιτρέποντας λειτουργίες CRUD στην κλάση Role. Επεκτείνοντας το JpaRepository, κληρονομεί αρκετές μεθόδους για την αλληλεπίδραση με τη βάση δεδομένων, όπως η αποθήκευση, η διαγραφή και η εύρεση ρόλων. Επιπλέον, ορίζει μια μέθοδο, την findByAuthority(), η οποία επιτρέπει την εύρεση ενός ρόλου με βάση το όνομα της αρχής του. Αυτή η μέθοδος είναι ιδιαίτερα χρήσιμη όταν χρειάζεται να ανακτήσει κανείς συγκεκριμένους ρόλους κατά τη διάρκεια των διαδικασιών αυθεντικοποίησης.

Μαζί, αυτές οι κλάσεις αποτελούν τη ραχοκοκαλιά της αυθεντικοποίησης βάσει ρόλων στην εφαρμογή. Η κλάση Role ορίζει ποιοι ρόλοι υπάρχουν και εξασφαλίζει την αναγνώρισή τους από την Spring Security, ενώ η κλάση RoleRepository παρέχει τα μέσα για την αλληλεπίδραση με τη βάση δεδομένων και την αποτελεσματική διαχείριση αυτών των ρόλων. Αυτή η ρύθμιση διασφαλίζει ότι στους χρήστες μπορούν να ανατεθούν ρόλοι και ότι αυτοί οι ρόλοι ερμηνεύονται και επιβάλλονται σωστά από το Spring Security, επιτρέποντας τον ασφαλή και ευέλικτο έλεγχο πρόσβασης σε όλη την εφαρμογή.

JSON Web Token

Για να υλοποιήσω την αυθεντικοποίηση των χρηστών βάσει ρόλων, χρησιμοποίησα το Spring Security σε συνδυασμό με JSON Web Tokens (JWT). Τι είναι όμως και πως δημιουργώ ένα JWT;

Το JWT είναι ένα τυχαίο string που χωρίζεται σε 3 μέρη. Το πρώτο εξ αυτών είναι το header, όπου αναφέρεται τι είδους token είναι –στην περίπτωση αυτή jwt– και το signing algorithm που χρησιμοποιήθηκε.

Έπειτα υπάρχει το payload, το οποίο περιέχει τα claims, δηλαδή πληροφορίες για τον χρήστη και το token, όπως τον issuer -ποιος το έφτιαξε-, το subject -σε ποιον ανήκει και μπορεί να κάνει χρήση αυτού και άρα πρέπει να είναι κάποιο unique πεδίο-, την ημερομηνία λήξης του token, τότε δημιουργήθηκε και τα scopes με τα οποία ελέγχω σε ποια resources έχει πρόσβαση ο εκάστοτε χρήστης. Στην εφαρμογή, στα scopes αναφέρω τον/-ους ρόλο/-ους του εκάστοτε χρήστη και βάση αυτού/-ών έχει πρόσβαση στα αντίστοιχα endpoint. Όλες αυτές οι πληροφορίες είναι key-value ζευγάρια.

Το token ολοκληρώνεται με την προσθήκη του signature. Πρόκειται για έναν συνδυασμό από το header, το payload και το secret key, κωδικοποιημένα από το signing algorithm

Προκειμένου να δημιουργηθεί το token, υλοποιήθηκε η κλάση JWTUtil. Στη κλάση αυτή με την μέθοδο issueToken δημιουργούνται JWTs όπως ακριβώς περιεγράφηκαν αναλυτικά παραπάνω. Ουσιαστικά χρησιμοποιείται ένας JWT builder στον οποίο περνάω τα claims και το υπογράφο. Η υπογραφή γίνεται με το secret key, η μετατροπή του οποίου από key σε string γίνεται με την χρήση της μεθόδου getSigningKey, και το signing algorithm HS256. Επιπλέον, υλοποιήθηκαν η μέθοδος getClaims που επιστρέφει τα claims ενός token -payload-, η μέθοδος getSubject που με τη χρήση της getClaims επιστρέφει το subject καθώς και η μέθοδος isValid που σε συνεργασία με την μέθοδο isTokenExpired ελέγχουν την εγκυρότητα του token σε σχέση με το subject και την ημερομηνία λήξης του.

Κλάση SecurityFilterChainConfig

Η κλάση SecurityFilterChainConfig περιέχει τη μέθοδο securityFilterChain, η οποία διαμορφώνει τις ρυθμίσεις ασφαλείας για την εφαρμογή χρησιμοποιώντας το Spring Security.

Πρώτον, η μέθοδος απενεργοποιεί την προστασία CSRF (Cross-Site Request Forgery). Η προστασία CSRF είναι συνήθως σημαντική για τις εφαρμογές ιστού για την αποτροπή κακόβουλων ενεργειών εκ μέρους πιστοποιημένων χρηστών. Ωστόσο, στην συγκεκριμένη περίπτωση απενεργοποιείται διότι η εφαρμογή χρησιμοποιεί το back-end ως REST API και δεν πρόκειται να γίνεται πρόσβαση σε αυτό μέσω φυλλομετρητή, άρα η προστασία από CSRF επιθέσεις είναι περιττή.

Στη συνέχεια, ενεργοποιείται το CORS (Cross-Origin Resource Sharing) με τις προεπιλεγμένες ρυθμίσεις. Το CORS επιτρέπει στην εφαρμογή να αντλεί με ασφάλεια πόρους από διαφορετικό domain από αυτόν στον οποίο φιλοξενείται η εφαρμογή.

Στη συνέχεια ορίζονται οι κανόνες αυθεντικοποίησης για τις εισερχόμενες αιτήσεις HTTP. Στη μέθοδο requestMatchers() γράφω το url των endpoint και το είδος της αίτησης που διαχειρίζεται. Έπειτα ή με την χρήση του permitAll() δίνω πρόσβαση σε όλους τους χρήστες ανεξαρτήτως scopes ή με την χρήση του hasRole() δίνω πρόσβαση μόνο σε όσους έχουν το συγκεκριμένο scope. Τέλος, με το anyRequest().authenticated() γίνεται authenticated κάθε άλλο endpoint.

Το session management ρυθμίζεται ώστε να είναι STATELESS. Αυτό σημαίνει ότι ο διακομιστής δεν θα δημιουργεί ή δεν θα χρησιμοποιεί συνόδους HTTP για την αποθήκευση πληροφοριών χρήστη. Αντ' αυτού, θα βασίζεται σε tokens, όπως το JWT (JSON Web Token), για τη διατήρηση της κατάστασης αυθεντικοποίησης του χρήστη, τα οποία είναι κατάλληλα για RESTful APIs. Ο λόγος για τον οποίο επιλέχθηκε να είναι STATELESS είναι επειδή η εφαρμογή χρησιμοποιεί jwt

και άρα δεν χρειάζεται κάποια HTTP σύνοδος και ούτε να αποθηκεύεται κάποιο state μεταξύ των αιτημάτων.

Στη συνέχεια, η μέθοδος ορίζει έναν authentication Provider. Αυτός ο πάροχος είναι υπεύθυνος για την επαλήθευση των διαπιστευτηρίων του χρήστη και έχει ρυθμιστεί σε άλλο σημείο στην εφαρμογή για να ενσωματωθεί στη διαδικασία του authentication.

Ακόμα, τίθεται το jwtAuthenticationFilter στην αλυσίδα φίλτρων ασφαλείας. Αυτό το φίλτρο τοποθετείται πριν από το UsernamePasswordAuthenticationFilter και είναι υπεύθυνο για την εξαγωγή και την επικύρωση των JWT από τις επικεφαλίδες των αιτήσεων.

Τέλος, η μέθοδος καλεί την http.build() για να δημιουργήσει και να επιστρέψει την αλυσίδα φίλτρων ασφαλείας, καθιστώντας την έτοιμη για χρήση στην εφαρμογή.

Κλάση SecurityConfig

Η κλάση SecurityConfig είναι μια configuration κλάση για τη ρύθμιση του Spring Security. Ορίζει διάφορα Beans που είναι ζωτικής σημασίας για την αυθεντικοποίηση και την κωδικοποίηση του κωδικού πρόσβασης στην εφαρμογή.

Ένα από αυτά είναι ο PasswordEncoder. Ο PasswordEncoder χρησιμοποιεί τον BCryptPasswordEncoder για την κωδικοποίηση των κωδικών πρόσβασης. Ο πρωταρχικός σκοπός του είναι η κωδικοποίηση των κωδικών πρόσβασης πριν από την αποθήκευσή τους στη βάση δεδομένων και η επικύρωση των «raw» κωδικών πρόσβασης έναντι των κωδικοποιημένων κατά τη διάρκεια του ελέγχου ταυτότητας. Η BCrypt είναι μια ισχυρή συνάρτηση κατακερματισμού που παρέχει ασφάλεια έναντι των επιθέσεων, καθιστώντας την εξαιρετική επιλογή για την κωδικοποίηση κωδικών πρόσβασης.

Εξίσου σημαντικός είναι και ο AuthenticationManager. Αποτελεί ένα βασικό μέρος του Spring Security καθώς ορίζει πως τα φίλτρα πραγματοποιούν την αυθεντικοποίηση. Στην προκειμένη περίπτωση επιλέχθηκε ο ProviderManager το πιο συχνά χρησιμοποιούμενο implementation του. Σε εκείνον «περνάει» το UsernamePasswordAuthenticationToken που περιέχει το username και το password και με τη σειρά του αναθέτει τη διαδικασία της αυθεντικοποίησης σε διάφορους AuthenticationProviders

Ο Provider που επιλέχθηκε είναι ο DaoAuthenticationProvider. Αυτός ανακτά τα στοιχεία του χρήστη από το UserDetailsService και χρησιμοποιεί τον PasswordEncoder για την επικύρωση των κωδικών πρόσβασης. Πιο συγκεκριμένα, ο DaoAuthenticationProvider αντλεί τα στοιχεία του χρήστη, συμπεριλαμβανομένων του ονόματος χρήστη, του κωδικού πρόσβασης και των ρόλων, από το UserDetailsService, κωδικοποιώντας «raw» κωδικούς πρόσβασης και συγκρίνοντας τους με τους αποθηκευμένους κωδικοποιημένους κωδικούς πρόσβασης χρησιμοποιώντας τον PasswordEncoder.

Κλάση JWTAuthenticationFilter

Η κλάση JWTAuthenticationFilter έχει σχεδιαστεί για να αναχαίτζει τα εισερχόμενα αιτήματα HTTP και να επικυρώνει το JWT (JSON Web Token) που περιλαμβάνεται στις επικεφαλίδες των αιτήσεων. Ο πρωταρχικός του ρόλος είναι να διασφαλίζει ότι μόνο πιστοποιημένοι χρήστες μπορούν να έχουν πρόσβαση σε συγκεκριμένα endpoints εντός της εφαρμογής.

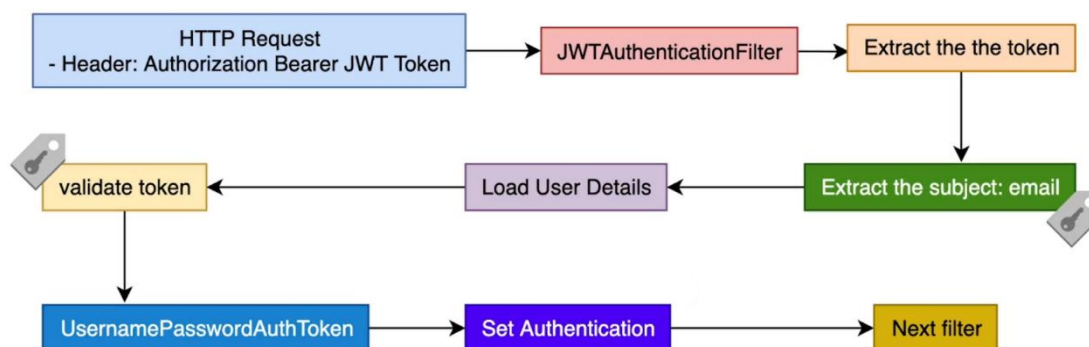
Όταν πραγματοποιείται ένα αίτημα, το φίλτρο ελέγχει πρώτα την παρουσία της επικεφαλίδας Authorization. Εάν η επικεφαλίδα είτε λείπει είτε δεν αρχίζει με «Bearer », το φίλτρο προχωράει περνώντας το αίτημα στο επόμενο φίλτρο χωρίς να εκτελεί καμία πιστοποίηση ταυτότητας.

Εάν βρεθεί έγκυρη επικεφαλίδα Authorization, το φίλτρο εξάγει το JWT από αυτήν αφαιρώντας το πρόθεμα «Bearer ». Στη συνέχεια, χρησιμοποιεί μια βοηθητική κλάση (JWTUtil) για να εξάγει το subject, το οποίο είναι το email χρήστη, από το token. Στη συνέχεια ελέγχει αν το subject δεν είναι null και αν ο χρήστης δεν έχει ήδη πιστοποιηθεί στο SecurityContextHolder.

Έπειτα, το φίλτρο χρησιμοποιεί την υπηρεσία UserDetailsService για να φορτώσει τα στοιχεία του χρήστη με βάση το εξαγόμενο subject. Επικυρώνει το JWT και εάν το token διαπιστωθεί ότι είναι έγκυρο, δημιουργείται ένα UsernamePasswordAuthenticationToken χρησιμοποιώντας τα στοιχεία του χρήστη -credentials και authorities.

Αυτό το authentication token δηλώνεται στη συνέχεια στο SecurityContextHolder, πράγμα που σημαίνει ότι ο χρήστης έχει πλέον πιστοποιηθεί για την τρέχουσα αίτηση. Στη συνέχεια, το φίλτρο προωθεί την αίτηση στα επόμενα φίλτρα, διασφαλίζοντας ότι όλα τα επόμενα φίλτρα αναγνωρίζουν τον χρήστη ως πιστοποιημένο.

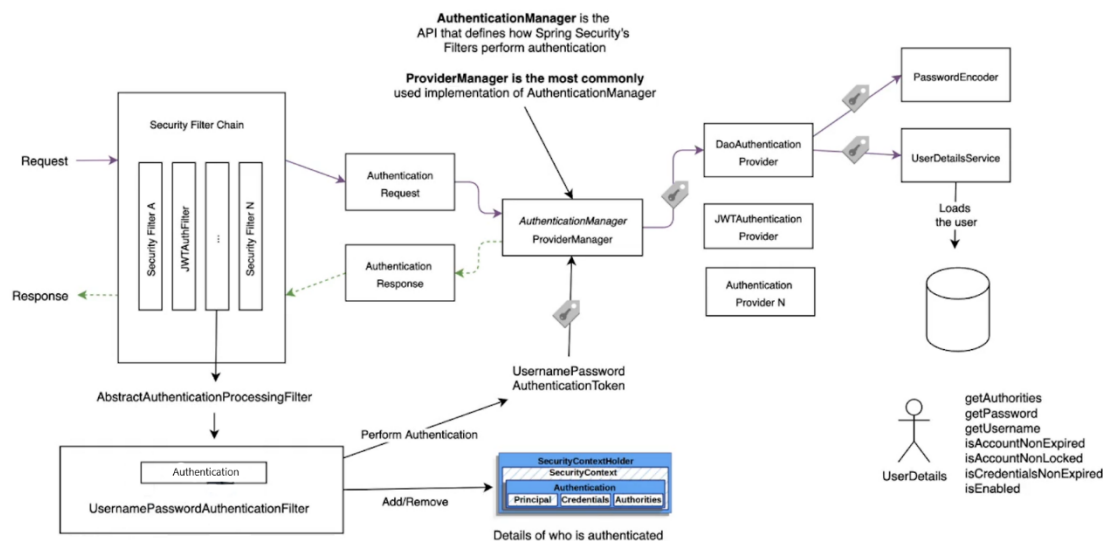
Το παρακάτω διάγραμμα παρέχει μια σαφή επισκόπηση του τρόπου με τον οποίο λειτουργεί το JWTAuthenticationFilter. Δείχνει τα διάφορα στάδια από τα οποία περνά το αίτημα HTTP.



ΔΙΑΓΡΑΜΜΑ 4: JWTAuthenticationFilter

Διαδικασία Αυθεντικοποίησης

Το παρακάτω διάγραμμα παρέχει μια σαφή επισκόπηση του τρόπου με τον οποίο αλληλεπιδρούν μεταξύ τους τα διάφορα components που συντελούν στην ασφάλεια της εφαρμογής. Δείχνει τη διαδρομή που κάθε αίτημα HTTP ακολουθεί.



ΔΙΑΓΡΑΜΜΑ 5: Διαδικασία Αυθεντικοποίησης

Υλοποίηση του refresh token

Το refresh token διασφαλίζει ότι οι χρήστες μπορούν να λαμβάνουν νέα access token όταν λήγουν τα τρέχοντα access token χωρίς να χρειάζεται να πιστοποιούνται εκ νέου με το όνομα χρήστη και τον κωδικό πρόσβασής τους. Αυτή η διαδικασία ξεκινά όταν ένας χρήστης συνδέεται, οπότε και δημιουργείται ένα refresh token. Η διαχείριση γίνεται στην κλάση AuthenticationService. Η μέθοδος authenticate επαληθεύει τα διαπιστευτήρια του χρήστη και μετά την επιτυχή πιστοποίηση, η μέθοδος issueToken δημιουργεί ένα access token. Παράλληλα, η μέθοδος createRefreshToken δημιουργεί ένα refresh token για τον πιστοποιημένο χρήστη και το επιστρέφει ως μέρος της AuthenticationResponse.

Η μέθοδος createRefreshToken της κλάσης RefreshTokenService χειρίζεται την πραγματική δημιουργία του refresh token. Ανακτά τον χρήστη και ελέγχει αν ο χρήστης έχει refresh token. Εάν έχει, ενημερώνει την τιμή του token και την ημερομηνία λήξης. Εάν δεν έχει, δημιουργεί ένα νέο refresh token με νέα ημερομηνία λήξης και τιμή token. Με τον τρόπο αυτό διασφαλίζεται ότι κάθε χρήστης έχει ένα μοναδικό και έγκυρο refresh token που συνδέεται με το λογαριασμό του.

Όταν ένας χρήστης ζητά ένα access token χρησιμοποιώντας ένα refresh token, καλείται η μέθοδος refreshToken στην κλάση AuthenticationController. Αυτή η μέθοδος αναζητά το refresh token στη βάση δεδομένων χρησιμοποιώντας τη μέθοδο findByToken. Στη συνέχεια, επαληθεύει τη λήξη του token χρησιμοποιώντας τη μέθοδο verifyExpiration. Εάν το token έχει λήξει, διαγράφεται και εκπέμπεται ένα exception. Εάν το token είναι έγκυρο, ανακτάται ο σχετικός χρήστης.

Μετά την ανάκτηση του χρήστη, η μέθοδος ελέγχει το ρόλο του χρήστη και δημιουργεί ένα νέο access token με τον κατάλληλο ρόλο. Στη συνέχεια παράγει ένα νέο refresh token και επιστρέφει τόσο το νέο refresh token όσο και το νέο access token. Αυτό εξασφαλίζει ότι ο χρήστης μπορεί να συνεχίσει να έχει πρόσβαση στην εφαρμογή χωρίς να χρειάζεται να εισάγει εκ νέου τα διαπιστευτήριά του.

Η μέθοδος `verifyExpiration` της κλάσης `RefreshTokenService` ελέγχει αν ένα refresh token έχει λήξει συγκρίνοντας την ημερομηνία λήξης του token με την τρέχουσα ώρα. Εάν το token έχει λήξει, διαγράφει το token και πετάει ένα `TokenRefreshException`, προτρέποντας τον χρήστη να υποβάλει νέο αίτημα σύνδεσης.

Χειρισμός exception σχετικά με την ασφάλεια

Η `AuthEntryPoint` χρησιμοποιείται από το framework για να χειριστεί τα exceptions που σχετίζονται με την αυθεντικοποίηση. Η μέθοδος `commence` καλείται κάθε φορά που αυτά συμβαίνουν και για τον λόγο αυτό γίνεται override και ανατίθεται ο ρόλος αυτός στον `HandlerExceptionResolver`. Με τον τρόπο αυτό καταφέρνω και παίρνω τα σωστά status code για τα αντίστοιχα exception.

Δεν υπάρχει ωστόσο κάποιο μήνυμα λάθους να επεξηγεί τι το προκάλεσαι και για τον σκοπό αυτό δημιουργήθηκαν οι `ApiError` και `DefaultExceptionHandler`. Η `ApiError` είναι ένα record με πεδία τις απαραίτητες πληροφορίες που πρέπει να αναγράφονται σε ένα error. Ο `DefaultExceptionHandler` έχει μια μέθοδο για κάθε είδος exception και στέλνει το αντίστοιχο «`ApiError`» response.

Διάγραμμα της βάσης δεδομένων

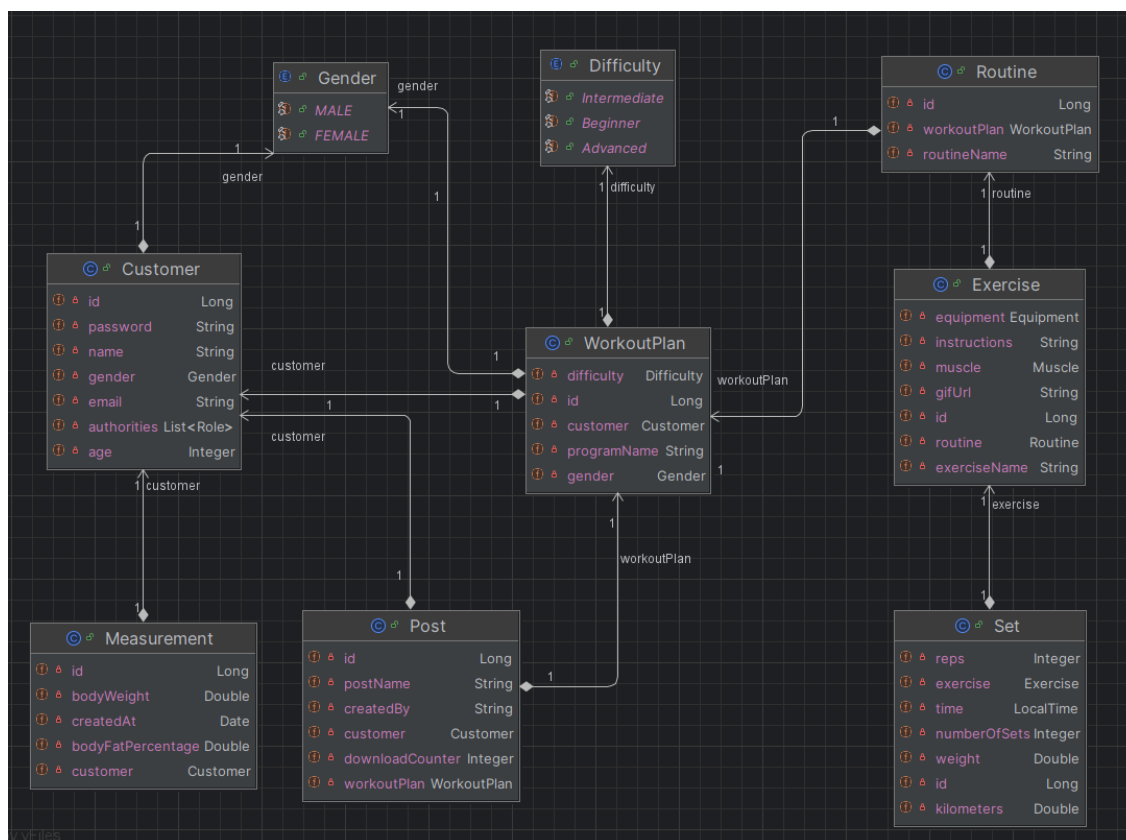
Για την απεικόνιση του διαγράμματος της βάσης δεδομένων επιλέχθηκε η χρήση ενός διαγράμματος τάξεων UML. Το διάγραμμα τάξεων UML παρέχει μια σαφή και δομημένη περιγραφή της δομής του συστήματος, καταγράφοντας τους τύπους των αντικειμένων και τις στατικές σχέσεις που υφίστανται μεταξύ τους.

Με τον όρο κλάση (τάξη) εννοούμε ένα λογικό πρότυπο (θα μπορούσαμε να πούμε blueprint) για τη δημιουργία αντικειμένων που μοιράζονται κοινές ιδιότητες, συμπεριφορές και μεθόδους και δημιουργούνται κατά την διάρκεια της εκτέλεσης του προγράμματος. Τα αντικείμενα με την σειρά τους αποτελούν στιγμιότυπα (instances) κλάσεων.

Ένα διάγραμμα τάξεων περιγράφει την δομή του συστήματος (γι'αυτό και ανήκει στην κατηγορία των structural diagrams) και περιλαμβάνει τους τύπους των αντικειμένων στο σύστημα και τα διάφορα είδη των στατικών σχέσεων που υπάρχουν μεταξύ τους. Απεικονίζει τις ιδιότητες και τις λειτουργίες των κλάσεων, τις συνδέσεις τους καθώς και τους περιορισμούς που εφαρμόζονται στους τρόπους με τους οποίους συνδέονται τα αντικείμενα.

Η UML χρησιμοποιεί τον όρο attribute για να αναφερθεί στις ιδιότητες (properties) της κλάσης και τον όρο operation για να αναφερθεί στις λειτουργίες (methods) της. Οι κλάσεις αναπαρίστανται με ορθογώνιο χωρισμένο σε τρία μέρη. Το πρώτο τμήμα (πάνω) περιέχει το όνομα της κλάσης, το δεύτερο (μέση) περιέχει τα attributes και το τρίτο (κάτω) περιέχει τα operations. Συνήθως, τα attributes των κλάσεων έχουν access modifier private (που συμβολίζεται με « **i** ») και operations public (που συμβολίζεται με « **i** »).

Το παρακάτω διάγραμμα δημιουργήθηκε με τη χρήση του IntelliJ IDEA, ενός ισχυρού και ευέλικτου εργαλείου ανάπτυξης λογισμικού. Το διάγραμμα απεικονίζει το σχήμα της βάσης δεδομένων, καταγράφοντας τις κλάσεις, τα attributes και τις operations τους, καθώς και τις σχέσεις που υπάρχουν μεταξύ των διαφόρων κλάσεων. Σημειώνεται ότι οι μέθοδοι getters, setters και οι constructors δεν αναφέρονται στο διάγραμμα αυτό.



ΔΙΑΓΡΑΜΜΑ 5: Database Schema

Κλάσεις για την υλοποίηση της αρχιτεκτονικής

Σε ένα Spring Boot project με αρχιτεκτονική n-tier, κάθε entity όπως αυτές στο παραπάνω διάγραμμα έχει πολλά συσχετιζόμενα components: ένα Controller, ένα Service, ένα DataAccessService, ένα JpaRepository και records για δημιουργία και ενημέρωση. Η κατανόηση των ρόλων και των αρμοδιοτήτων αυτών των components είναι ζωτικής σημασίας για την κατανόηση της συνολικής αρχιτεκτονικής.

Ο Controller χρησιμεύει ως σημείο εισόδου για τη διαχείριση των αιτήσεων ιστού. Είναι υπεύθυνος για την αντιστοίχιση των αιτήσεων HTTP σε συγκεκριμένες μεθόδους και την επιστροφή των απαντήσεων. Όταν λαμβάνεται ένα αίτημα, καθορίζει την κατάλληλη μέθοδο για κλήση, επεξεργάζεται το αίτημα και στέλνει πίσω μια απάντηση, με τη μορφή JSON. Αυτό το στοιχείο διασφαλίζει ότι οι αλληλεπιδράσεις των χρηστών διαχειρίζονται και δρομολογούνται κατάλληλα εντός της εφαρμογής.

Στη συνέχεια, το Service περιέχει την λογική της εφαρμογής. Επεξεργάζεται τα δεδομένα που λαμβάνονται από τον Controller και αλληλεπιδρά με το DataAccessService για την εκτέλεση λειτουργιών. Είναι ιδιαίτερα σημαντικό για τη διασφάλιση της επικύρωσης και της ακεραιότητας των δεδομένων και το συντονισμό εργασιών. Καλεί τις μεθόδους του DataAccessService για την ανάκτηση και τον χειρισμό δεδομένων, διασφαλίζοντας ότι η βασική λειτουργικότητα της εφαρμογής είναι ισχυρή και αξιόπιστη.

Το DataAccessService ενεργεί ως ενδιάμεσος μεταξύ του Service και της βάσης δεδομένων. Εκτελεί λειτουργίες πρόσβασης σε δεδομένα, όπως η δημιουργία, η ανάγνωση, η ενημέρωση και η διαγραφή (CRUD), μετασχηματίζοντας δεδομένα από και προς τα entities ανάλογα με τις ανάγκες.

Το JpaRepository είναι μια διεπαφή της Spring Data που παρέχει λειτουργίες CRUD και πρόσθετες μεθόδους για την αλληλεπίδραση με τη βάση δεδομένων. Χρησιμοποιώντας το Java Persistence API (JPA), το JpaRepository αφαιρεί τις πολυπλοκότητες των αλληλεπιδράσεων με τη βάση δεδομένων, προσφέροντας έναν απλό τρόπο διαχείρισης δεδομένων. Έτσι απλοποιεί την πρόσβαση στα δεδομένα, επιτρέποντας την επικέντρωση στην λογική και όχι στις περίπλοκες λειτουργίες της βάσης δεδομένων.

Annotations

Μια από τις προκλήσεις που έρχεται αντιμέτωπος κανείς όταν δουλεύει με Spring είναι ότι περιλαμβάνει πολλά configuration και επαναλαμβανόμενα κομμάτια κώδικα. Το Spring MVC κάνει πολλά πράγματα ευκολότερα, αλλά εξακολουθεί να περιλαμβάνει αρκετό configuration. Το Spring Boot κρύβει τις περισσότερες από τις βαρετές λεπτομέρειες, μέσω της χρήσης αυτοματισμού και έξυπνων προεπιλογών όπως τα annotation. Έτσι, το Spring Boot διευκολύνει πολύ τη δημιουργία αυτόνομων, παραγωγικών εφαρμογών βασισμένων στο Spring.

Παρακάτω παρατίθενται τα annotations που χρησιμοποιούνται στην παρούσα εφαρμογή, μαζί με μια σύντομη επεξήγηση της λειτουργίας τους.

@SpringBootApplication

Το @SpringBootApplication είναι τα annotation @Configuration, @EnableAutoConfiguration και @ComponentScan συνδυασμένα, διαμορφωμένα με τα προεπιλεγμένα χαρακτηριστικά τους. Προσθέτουμε αυτό το annotation μόνο μία φορά, στην κύρια κλάση της εφαρμογής μας.

Spring Web MVC Annotations

@Controller

Χαρακτηρίζει την κλάση ως web controller. Είναι μια εξειδίκευση του annotation @component, η οποία επιτρέπει στο Spring να εντοπίζει αυτόματα τα implementation classes/beans σαρώνοντας το classpath.

@RestController

Το annotation @RestController είναι μια σύνταξη ευκολίας για το @Controller και το @ResponseBody μαζί. Αυτό υποδεικνύει ότι η κλάση είναι controller και ότι όλες οι μέθοδοι της επιστημασμένης κλάσης θα επιστρέφουν μια απόκριση JSON.

@ResponseBody

Το @ResponseBody είναι ένα βοηθητικό annotation που λέει στην Spring να σειριοποιήσει αυτόματα την τιμή(ες) επιστροφής των μεθόδων αυτής της κλάσης σε απαντήσεις HTTP. Κατά τη δημιουργία ενός JSON endpoint, αυτός είναι ένας καταπληκτικός τρόπος για να μετατρέψει κανείς «μαγικά» τα αντικείμενα σε JSON. Αν χρησιμοποιήσουμε το annotation @RestController στην κλάση μας, δεν χρειαζόμαστε καθόλου αυτό το annotation, επειδή η @RestController κληρονομεί από αυτήν.

@RequestBody

Το annotation @RequestBody χρησιμοποιείται για το «binding» του σώματος της αίτησης HTTP σε ένα αντικείμενο Java. Το @RequestBody αποτελεί μέρος του framework spring και χρησιμοποιείται σε συνδυασμό με το spring MVC.

@RequestMapping(method = RequestMethod.GET, value = «/path»)

Το annotation @RequestMapping(method = RequestMethod.GET, value = «/path») καθορίζει μια μέθοδο στον controller που θα πρέπει να είναι υπεύθυνη για την εξυπηρέτηση της αίτησης HTTP στο δεδομένο endpoint. Η Spring χειρίζεται τις μηχανικές λεπτομέρειες για το πώς επιτυγχάνεται αυτό για εσάς. Απλά καθορίζετε τις παραμέτρους method και path στο annotation και η Spring θα δρομολογήσει τα αιτήματα στις σωστές μεθόδους. Εάν δεν καθορίσετε μια τιμή μεθόδου, η μέθοδος θα είναι προεπιλεγμένη σε GET.

@GetMapping(value = «/path»)

Μια συντομευμένη μορφή του @RequestMapping ειδικά για αιτήσεις HTTP GET, η οποία λαμβάνει μόνο ένα προαιρετικό όρισμα τιμής. Η ανάγνωση στο CRUD.

`@PostMapping(value = «/path»)`

Μια συντομευμένη μορφή του `@RequestMapping` ειδικά για αιτήσεις HTTP POST, η οποία δέχεται μόνο ένα προαιρετικό όρισμα τιμής. Η δημιουργία στο CRUD.

`@PutMapping(value = «/path»)`

Μια συντομευμένη μορφή του `@RequestMapping` ειδικά για αιτήσεις HTTP PUT, η οποία δέχεται μόνο ένα προαιρετικό όρισμα τιμής. Η ενημέρωση στην CRUD.

`@DeleteMapping(value = «/path»)`

Μια συντομευμένη μορφή του `@RequestMapping` ειδικά για αιτήσεις HTTP DELETE, η οποία δέχεται μόνο ένα προαιρετικό όρισμα τιμής. Η διαγραφή στο CRUD.

`@RequestParam(value=«name», defaultValue=«World»)`

Φυσικά, οι μέθοδοι που χειρίζονται τις αιτήσεις μπορεί να λαμβάνουν παραμέτρους. Για βοήθεια με το «binding» των παραμέτρων HTTP στα ορίσματα των μεθόδων, μπορεί να χρησιμοποιήσει κανείς το annotation `@RequestParam(value=«name», defaultValue=«World»)`. Η Spring θα αναλύσει τις παραμέτρους της αίτησης και θα τις τοποθετήσει κατάλληλα στα ορίσματα της μεθόδου σας.

`@PathVariable(«placeholderName»)`

Ένας άλλος συνηθισμένος τρόπος για παροχή πληροφοριών στο back-end είναι να τις κωδικοποιήσετε στο URL. Στη συνέχεια, μπορείτε να χρησιμοποιήσετε το annotation `@PathVariable(«placeholderName»)` για να φέρετε τις τιμές από το URL στα ορίσματα της μεθόδου.

Spring Annotations

`@Bean`

Αναλύθηκε παραπάνω (βλέπε Application Context και Beans).

`@Component`

Αυτο το annotation λέει στο framework ότι αυτή η κλάση είναι ένα component και θα πρέπει να διαχειρίζεται από το spring container. Εκτός από το annotation `@Component`, μπορεί επίσης κανείς να χρησιμοποιήσει τα `@Repository`, `@Service` και `@Controller`. Κάθε ένας από αυτά τα annotation έχει συγκεκριμένο σκοπό και η spring θα διαχειριστεί τα component ανάλογα.

`@Service`

Το `Service`, σε αντίθεση με τους άλλους τύπους component, δεν προσφέρει καμία ιδιαίτερη λειτουργικότητα σε σχέση με το `@Component` και χρησιμοποιείται απλώς για να δείξει περαιτέρω την πρόθεση της κλάσης.

`@Repository`

Χρησιμοποιείται σε κλάσεις που έχουν άμεση πρόσβαση στη βάση δεδομένων - έτσι, για παράδειγμα, μπορεί κανείς να συμπεράνει ότι αυτό είναι σημαντικό στο DAO Layer.

Στήσιμο front-end

Για τη δημιουργία της Android πλευρά της εφαρμογής, χρησιμοποιήθηκε το Android Studio IDE. Εκεί επιλέχθηκε το template «Empty Views Activity». Μετά τη δημιουργία του project, το Android Studio δημιουργεί αυτόματα τα απαραίτητα αρχεία και φακέλους, όπως το MainActivity.java, τον κατάλογο res για τους πόρους και το AndroidManifest.xml.

Στη συνέχεια, προστέθηκαν τα απαιτούμενα dependencies στο αρχείο build.gradle. Πιο συγκεκριμένα, συμπεριελήφθησαν dependencies όπως το Volley για αιτήσεις δικτύου, το Glide για φόρτωση και προσωρινή αποθήκευση εικόνων και το Material Components για σύγχρονο σχεδιασμό του UI.

Στη συνέχεια, διασφαλίστηκε ότι το αρχείο AndroidManifest.xml έχει ρυθμιστεί σωστά με τα απαραίτητα δικαιώματα και δραστηριότητες. Για παράδειγμα, δεδομένου ότι η εφαρμογή απαιτεί πρόσβαση στο διαδίκτυο, συμπεριελήφθηκε στο αρχείο η άδεια `<uses-permission android:name="android.permission.INTERNET"/>`. Αυτό διασφαλίζει ότι η εφαρμογή μπορεί να κάνει αιτήσεις δικτύου.

Με τα πάντα ρυθμισμένα, προχώρησα στη δημιουργία και εκτέλεση του project. Προκειμένου να το εκτελέσει κανείς, δύναται είτε να συνδέσει μια συσκευή Android είτε να χρησιμοποιήσει έναν εξομοιωτή που παρέχεται από το Android Studio. Κάνοντας κλικ στο κουμπί «Run», αναπτύσσεται και δοκιμάζεται η εφαρμογή, διασφαλίζοντας ότι λειτουργεί όπως αναμένεται.

Όπως αναφέρθηκε, η εφαρμογή που υλοποιήθηκε βασίζεται σε αρκετά dependencies για την ενίσχυση της λειτουργικότητας και του UI της. Το Volley απλοποιεί τη διαδικασία υποβολής αιτημάτων δικτύου στο back-end API, καθώς χειρίζεται αποτελεσματικά τα αιτήματα και τις απαντήσεις HTTP. Το Glide, μια βιβλιοθήκη φόρτωσης και προσωρινής αποθήκευσης εικόνων, επιτρέπει την αποτελεσματική φόρτωση και προσωρινή αποθήκευση εικόνων από το διαδίκτυο. Τα Material Components παρέχουν σύγχρονα στοιχεία και θέματα σχεδίασης, εξασφαλίζοντας μια συνεπή και ελκυστική διεπαφή χρήστη.

Ακολουθώντας αυτά τα βήματα και ενσωματώνοντας τα καθορισμένα dependencies, δημιουργείται ένα στιβαρό front-end περιβάλλον, χρησιμοποιώντας το Android Studio. Αυτή η ρύθμιση διασφαλίζει ότι η εφαρμογή είναι καλά εξοπλισμένη με βασικές βιβλιοθήκες για την επικοινωνία μέσω δικτύου, το χειρισμό εικόνων και τα σύγχρονα στοιχεία UI, διευκολύνοντας την ανάπτυξη μιας εφαρμογής υψηλής ποιότητας.

Android UI Components

Bottom Navigation Bar και Fragments

Στην εφαρμογή, το `BottomNavigationView` και τα `fragments` συνεργάζονται για να δημιουργήσουν μια δυναμική εμπειρία χρήστη μέσα στο `UserActivity`. Το `BottomNavigationView` χρησιμεύει ως το πρωταρχικό στοιχείο πλοήγησης στο κάτω μέρος της οθόνης, επιτρέποντας στους χρήστες να εναλλάσσονται μεταξύ των διαφόρων τμημάτων της εφαρμογής με ευκολία. Κάθε τμήμα αντιπροσωπεύεται από ένα `fragment`, το οποίο είναι ένα επαναχρησιμοποιήσιμο UI component που διαχειρίζεται ένα τμήμα της διεπαφής της εφαρμογής.

Η κλάση `UserActivity` είναι η κύρια δραστηριότητα που «φιλοξενεί» τα `fragments`. Κατά τη δημιουργία, η μέθοδος `onCreate` ορίζει την προβολή περιεχομένου, η οποία περιλαμβάνει το `BottomNavigationView` και ένα `FrameLayout` που λειτουργεί ως container για τα `fragments`. Το αρχικό `fragment` που εμφανίζεται είναι το `WorkoutFragment`, το οποίο ορίζεται με την κλήση της μεθόδου `replaceFragment`.

Το `BottomNavigationView` αρχικοποιείται με την εύρεση της προβολής του με βάση το ID και τον ορισμό ενός `OnItemSelectedListener`. Αυτός ο listener ανταποκρίνεται στις επιλογές του χρήστη στα «στοιχεία» του μένου που δύναται να προηγηθεί (`workout`, `community`, `measurements`, `profile`). Ανάλογα με το επιλεγμένο «στοιχείο», το αντίστοιχο `fragment` (`WorkoutFragment`, `CommunityFragment`, `MeasurementsFragment`, `ProfileFragment`) ενσωματώνεται και μεταβιβάζεται στη μέθοδο `replaceFragment` για να εμφανιστεί.

Η μέθοδος `replaceFragment` είναι ζωτικής σημασίας για αυτή τη ρύθμιση. Χρησιμοποιεί τον `FragmentManager` για να ξεκινήσει την μετάβαση, δηλαδή να αντικαταστήσει το τρέχον `fragment` εντός του `FrameLayout`. Αυτή η μέθοδος διασφαλίζει ότι εμφανίζεται μόνο ένα `fragment` κάθε φορά, παρέχοντας ομαλή μετάβαση μεταξύ διαφορετικών τμημάτων της εφαρμογής.

Συνοψίζοντας, ο συνδυασμός του `BottomNavigationView` και των `fragments` εντός του `UserActivity` δημιουργεί μια ισχυρή και φιλική προς το χρήστη δομή πλοήγησης. Το `BottomNavigationView` παρέχει ένα ομαλό τρόπο για τους χρήστες να μεταβαίνουν μεταξύ των διαφόρων τμημάτων της εφαρμογής, ενώ τα `fragments` διασφαλίζουν ότι κάθε τμήμα είναι ανεξάρτητο και αυτοτελές. Η μέθοδος `replaceFragment` διευκολύνει τις ομαλές μεταβάσεις μεταξύ των `fragments`, βελτιώνοντας τη συνολική εμπειρία του χρήστη.

RecyclerView

Διάφορα components του Android UI συνεργάζονται για να δημιουργήσουν μια διαδραστική εμπειρία χρήστη κατά τη χρήση της εφαρμογής. Ένα από τα βασικά, αποτελεί το RecyclerView, το οποίο εμφανίζει αποτελεσματικά ένα μεγάλο σύνολο δεδομένων σε μια λίστα με δυνατότητα κύλισης. Πρόκειται για σημαντική βελτίωση σε σχέση με παλαιότερα component όπως το ListView.

Ας πάρουμε για παράδειγμα το RecyclerView που χρησιμεύει για την εμφάνιση της λίστας ασκήσεων. Συνοδεύεται από το RecyclerViewAdapter, το οποίο λειτουργεί ως γέφυρα μεταξύ της πηγής δεδομένων και του RecyclerView. Ο Adapter είναι υπεύθυνος για τη δημιουργία των απαραίτητων προβολών για κάθε στοιχείο του συνόλου δεδομένων. Η κλάση ExerciseAdapter επεκτείνει τον RecyclerViewAdapter και δεσμεύει τα δεδομένα των ασκήσεων στις προβολές. Χρησιμοποιεί μια List<ExerciseResponse> για να κρατάει τα δεδομένα και παρέχει υλοποιήσεις για τις βασικές μεθόδους του Adapter: onCreateViewHolder, onBindViewHolder και getItemCount. Η μέθοδος onCreateViewHolder «γεμίζει» το layout για κάθε στοιχείο, δημιουργώντας ένα instance του ExerciseViewHolder, ενώ η μέθοδος onBindViewHolder δεσμεύει τα δεδομένα της άσκησης στις αντίστοιχες προβολές, όπως ο καθορισμός κειμένου για το όνομα της άσκησης και την ομάδα μυών και η φόρτωση μιας εικόνας χρησιμοποιώντας τη βιβλιοθήκη Glide για το GIF της άσκησης.

Η κλάση ExerciseViewHolder κρατάει references στις προβολές μέσα στη διάταξη για κάθε στοιχείο. Ορίζει onClick listeners για τα στοιχεία. Όταν γίνεται κλικ σε ένα στοιχείο, ανοίγει την ExerciseDetailActivity, μεταφέροντας τις σχετικές λεπτομέρειες της άσκησης μέσω ενός intent. Αυτό εξασφαλίζει μια απρόσκοπτη εμπειρία πλοήγησης για τους χρήστες, επιτρέποντάς τους να βλέπουν λεπτομερείς πληροφορίες για κάθε άσκηση.

Η διαχείριση του φιλτραρίσματος των εμφανιζόμενων ασκήσεων γίνεται μέσω ενός συνδυασμού ενός SearchView και δύο Spinner. Το SearchView επιτρέπει στους χρήστες να αναζητούν ασκήσεις με βάση το όνομα, ενημερώνοντας δυναμικά τη λίστα καθώς ο χρήστης πληκτρολογεί. Τα Spinner επιτρέπουν το φιλτράρισμα με βάση τον εξοπλισμό και την ομάδα μυών, παρέχοντας προκαθορισμένες λίστες επιλογών από τις οποίες οι χρήστες μπορούν να επιλέξουν. Κάθε spinner συμπληρώνεται με αυτές τις επιλογές και οι επιλεγμένες τιμές τους χρησιμοποιούνται για τη βελτίωση των εμφανιζόμενων ασκήσεων.

Η μέθοδος handleFiltering καλείται κάθε φορά που αλλάζει το query αναζήτησης ή οι επιλογές του spinner. Αυτή η μέθοδος διατρέχει την πλήρη λίστα ασκήσεων (exerciseList) και ελέγχει κάθε στοιχείο με βάση το τρέχον query αναζήτησης και τις επιλεγμένες τιμές φίλτρου. Μόνο τα στοιχεία που ανταποκρίνονται σε όλα τα κριτήρια προστίθενται στη filteredExerciseList. Η μέθοδος filterList του Adapter καλείται στη συνέχεια για να ενημερώσει την εμφανιζόμενη λίστα στο RecyclerView, διασφαλίζοντας ότι ο χρήστης βλέπει μόνο τις ασκήσεις που πληρούν τα κριτήριά του.

Συνοπτικά, το RecyclerView, ο Adapter και το ViewHolder συνεργάζονται για την αποτελεσματική εμφάνιση και διαχείριση της λίστας ασκήσεων. Τα στοιχεία SearchView και Spinner παρέχουν ισχυρές δυνατότητες φιλτραρίσματος, επιτρέποντας στους χρήστες να περιορίσουν τις εμφανιζόμενες ασκήσεις με βάση τις προτιμήσεις τους. Η αλληλεπίδραση μεταξύ αυτών των στοιχείων διασφαλίζει ότι οι χρήστες μπορούν εύκολα να βρискουν και να πλοηγούνται στις ασκήσεις, βελτιώνοντας τη συνολική εμπειρία του χρήστη.

Με ανάλογο τρόπο συνεργάζονται:

- Τα SetAdapter, SetResponse και SetViewHolder στο ExerciseDetailActivity για την εμφάνιση της λίστας των σετ της άσκησης,
- Τα UserAdapter, UserResponse και UserViewHolder στο ManageUsersActivity για την εμφάνιση της λίστας των χρηστών,
- Τα PostAdapter, PostResponse και PostViewHolder στο CommunityFragment για την εμφάνιση της λίστας των αναρτήσεων με τα κοινοποιημένα προγράμματα,

- Τα MeasurementAdapter, MeasurementResponse και MeasurementViewHolder στο MeasurementsFragment για την εμφάνιση του ιστορικού των μετρήσεων

Tab layout και ViewPager

Το TabLayout μαζί με το ViewPager είναι ένας ισχυρός συνδυασμός που χρησιμοποιείται για την υλοποίηση διεπαφών με καρτέλες, επιτρέποντας στους χρήστες να μετακινούνται απρόσκοπτα μεταξύ τους. Ας πάρουμε για παράδειγμα το PostDetailActivity. Στο πλαίσιο της PostDetailActivity, αυτά τα components χρησιμοποιούνται για την παρουσίαση των προγραμμάτων προπόνησης που σχετίζονται με μια συγκεκριμένη ανάρτηση.

Στην κλάση PostDetailActivity, η RoutinePagerAdapterPost και η RoutineFragmentPost συνεργάζονται για να εμφανίσουν τις ρουτίνες που σχετίζονται με ένα πρόγραμμα γυμναστικής. Όταν δημιουργείται η PostDetailActivity, αρχικοποιεί τα απαραίτητα στοιχεία, μεταξύ των οποίων είναι τα, routineNamesList, routineIdsList, tabLayout, viewPager και κουμπιά για τις ενέργειες λήψης, διαγραφής και επεξεργασίας. Ορίζει επίσης onClick listeners για αυτά τα κουμπιά ώστε να χειρίζονται τις αντίστοιχες ενέργειες.

Μία από τις βασικές λειτουργίες είναι η μέθοδος loadRoutines. Αυτή η μέθοδος στέλνει ένα αίτημα GET στον διακομιστή για να αντλήσει ρουτίνες που σχετίζονται με ένα πρόγραμμα γυμναστικής χρησιμοποιώντας το workoutPlanId. Η απάντηση του διακομιστή περιέχει ID και ονόματα ρουτινών, τα οποία στη συνέχεια προστίθενται στη λίστα routineIdsList και routineNamesList. Μόλις ανακτηθούν τα δεδομένα, δημιουργείται ένα νέο instance του RoutinePagerAdapterPost και οι λίστες αυτές μεταβιβάζονται σε αυτό. Στη συνέχεια, ο Adapter ορίζεται στο viewPager και το tabLayout συνδέεται με το viewPager. Αυτή η ρύθμιση επιτρέπει στο viewPager να εμφανίζει ρουτίνες σε layout με καρτέλες, με κάθε καρτέλα να αντιπροσωπεύει μια ρουτίνα.

Η κλάση RoutinePagerAdapterPost επεκτείνει τον FragmentStatePagerAdapter και είναι υπεύθυνη για την παροχή fragments στον viewPager. Διατηρεί δύο λίστες: routineIdsList και routineNamesList. Η μέθοδος getItem του Adapter επιστρέφει ένα νέο instance της RoutineFragmentPost για κάθε ρουτίνα, περνώντας το ID της ρουτίνας στο fragment. Η μέθοδος getCount επιστρέφει τον αριθμό των ρουτινών και η getPageTitle επιστρέφει το όνομα της ρουτίνας για κάθε καρτέλα.

Η κλάση RoutineFragmentPost είναι ένα fragment που εμφανίζει ασκήσεις που σχετίζονται με μια ρουτίνα. Η μέθοδος onCreateView «γεμίζει» το layout και ρυθμίζει το RecyclerView με έναν LinearLayoutManager. Η μέθοδος loadExercises στέλνει ένα αίτημα GET στο διακομιστή για να αντλήσει ασκήσεις για το δεδομένο ID ρουτίνας. Η απόκριση του API περιέχει τις λεπτομέρειες των ασκήσεων, όπως ID, όνομα, μουσική ομάδα, εξοπλισμό, διεύθυνση URL GIF και οδηγίες, οι οποίες προστίθενται στην exerciseList. Στη συνέχεια δημιουργείται ένας ExerciseAdapter με αυτή τη λίστα και ορίζεται στο RecyclerView για την εμφάνιση των ασκήσεων.

Αυτή η ρύθμιση επιτρέπει στο PostDetailActivity να εμφανίζει δυναμικά τις ρουτίνες και τις σχετικές ασκήσεις τους σε layout με καρτέλες. Το RoutinePagerAdapterPost χειρίζεται τη δημιουργία fragments για κάθε ρουτίνα και το RoutineFragmentPost συμπληρώνει αυτά τα fragments με δεδομένα ασκήσεων που ανακτώνται από το διακομιστή. Το tabLayout και το viewPager παρέχουν μια ομαλή και φιλική προς το χρήστη διεπαφή για την πλοήγηση στις διάφορες ρουτίνες και την προβολή των ασκήσεών τους.

Με ανάλογο τρόπο συνεργάζονται τα RoutinePagerAdapter και RoutineFragment στο WorkoutFragment για την εμφάνιση του ιστορικού των ασκήσεων της εκάστοτε ρουτίνας.

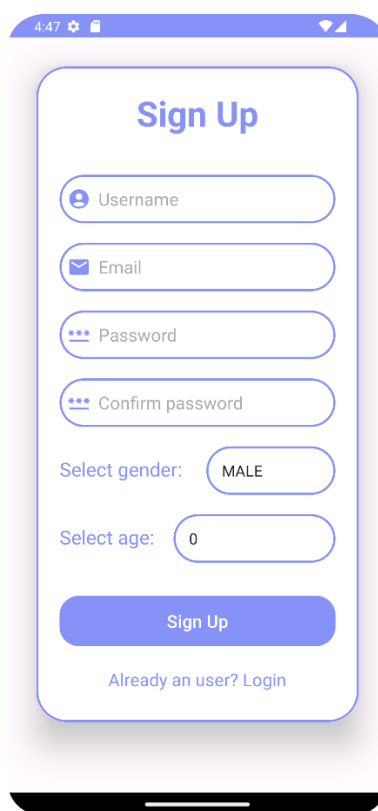
Εγχειρίδιο χρήσης (για το χρήστη)

Εγγραφή και Σύνδεση

Αυτή η ενότητα θα σας καθοδηγήσει στα αρχικά βήματα χρήσης της εφαρμογής: εγγραφή και σύνδεση.

Όταν εκκινήσετε για πρώτη φορά την εφαρμογή, θα συναντήσετε μια φόρμα εγγραφής. Εδώ, πρέπει να συμπληρώσετε τα στοιχεία σας για να δημιουργήσετε έναν λογαριασμό. Ξεκινήστε επιλέγοντας ένα όνομα χρήστη και εισάγοντας μια έγκυρη διεύθυνση ηλεκτρονικού ταχυδρομείου. Στη συνέχεια, επιλέξτε το φύλο σας και την ηλικία σας. Έπειτα, δημιουργήστε έναν ισχυρό κωδικό πρόσβασης που να πληροί τις απαιτήσεις ασφαλείας, όπως να έχει τουλάχιστον 6 χαρακτήρες και να περιλαμβάνει τόσο γράμματα όσο και αριθμούς. Μόλις συμπληρώσετε όλες τις απαιτούμενες πληροφορίες, πατήστε το κουμπί «Sign Up» για να υποβάλετε τη φόρμα.

Εάν έχετε ήδη λογαριασμό και δεν χρειάζεται να εγγραφείτε, μπορείτε να μεταβείτε στη φόρμα σύνδεσης απευθείας από τη φόρμα εγγραφής. Απλά πατήστε το κουμπί «Already an user? Login» Αυτό θα σας μεταφέρει στη σελίδα σύνδεσης χωρίς να χρειάζεται να ολοκληρώσετε τη διαδικασία εγγραφής.

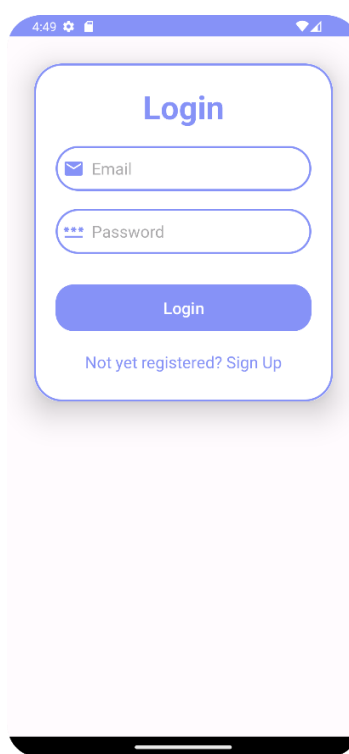


The image shows a mobile application interface for a sign-up form. At the top, the status bar displays the time 4:47, a star icon, and signal strength. The form itself is titled "Sign Up" in a bold, dark blue font. Below the title are four input fields: "Username" with a person icon, "Email" with an envelope icon, "Password" with three asterisks, and "Confirm password" with three asterisks. Below these are two dropdown menus: "Select gender:" with "MALE" selected, and "Select age:" with "0" selected. At the bottom of the form is a large blue button labeled "Sign Up" and a link that says "Already an user? Login". The entire form is set against a light blue background with rounded corners and a subtle shadow.

Εικόνα 2: Φόρμα Εγγραφής

Μετά την επιτυχή εγγραφή σας, θα μεταφερθείτε στη φόρμα σύνδεσης. Εδώ, θα πρέπει να εισάγετε το email που χρησιμοποιήσατε κατά την εγγραφή σας, μαζί με τον κωδικό πρόσβασής σας. Μόλις εισαγάγετε τα διαπιστευτήριά σας, πατήστε το κουμπί «Login» για να αποκτήσετε πρόσβαση στο λογαριασμό σας. Εάν οι πληροφορίες είναι σωστές, θα συνδεθείτε και θα ανακατευθυνθείτε στην κύρια σελίδα της εφαρμογής, όπου μπορείτε να ξεκινήσετε τη δημιουργία του προγράμματος γυμναστικής σας.

Εάν βρίσκεστε στη φόρμα σύνδεσης και συνειδητοποιήσετε ότι δεν έχετε ακόμη λογαριασμό, μπορείτε εύκολα να μεταβείτε στη φόρμα εγγραφής πατώντας το κουμπί «Not yet registered? Sign Up». Αυτό θα σας μεταφέρει πίσω στη σελίδα εγγραφής όπου μπορείτε να δημιουργήσετε έναν νέο λογαριασμό.

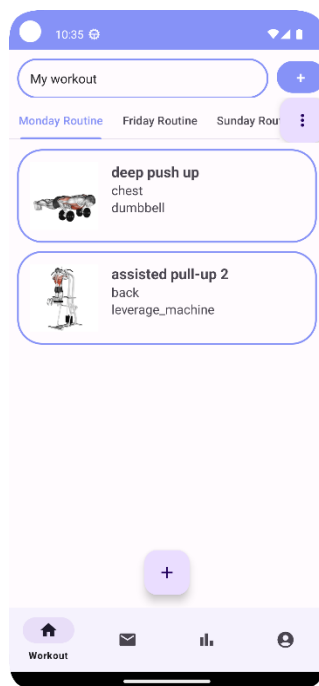


Εικόνα 3: Φόρμα Σύνδεσης

Όταν μεταφερθείτε στην κύρια σελίδα της εφαρμογής μετά τη σύνδεση, θα βρείτε ένα μενού πλοήγησης στο κάτω μέρος με τέσσερις ενότητες. Ας τις εξερευνήσουμε λεπτομερώς

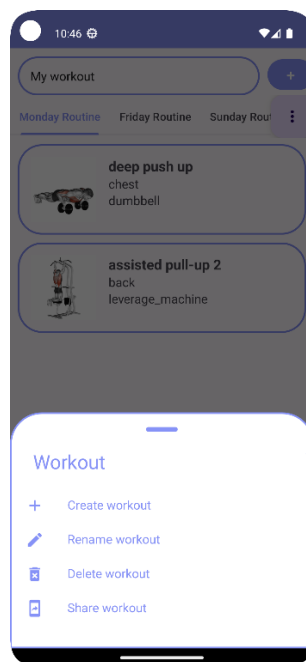
Ενότητα 1: Δημιουργία και διαχείριση του προγράμματός

Στην πρώτη ενότητα, έχετε τα εργαλεία για να δημιουργήσετε και να διαχειριστείτε το πρόγραμμα γυμναστικής σας. Εάν αυτή είναι η πρώτη φορά που συνδέεστε, η εφαρμογή θα δημιουργήσει αυτόματα ένα πρόγραμμα με ρουτίνα για εσάς. Εάν έχετε συνδεθεί ξανά στο παρελθόν, η εφαρμογή θα φορτώσει τυχόν προγράμματα που έχετε δημιουργήσει προηγουμένως.



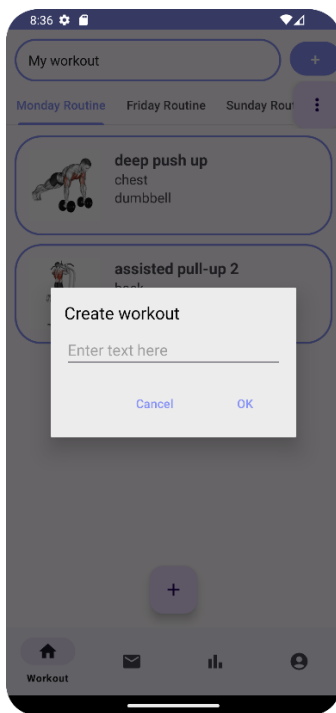
Εικόνα 4: Ενότητα 1

Στα δεξιά του ονόματος του προγράμματός σας, θα δείτε ένα κουμπί «+». Πατώντας αυτό το κουμπί, θα εμφανιστεί ένα μενού επιλογών στο κάτω μέρος της οθόνης το οποίο θα σας δίνει επιλογές για να μετονομάσετε, να διαγράψετε, να κοινοποιήσετε το τρέχον επιλεγμένο πρόγραμμά σας ή να δημιουργήσετε ένα νέο.



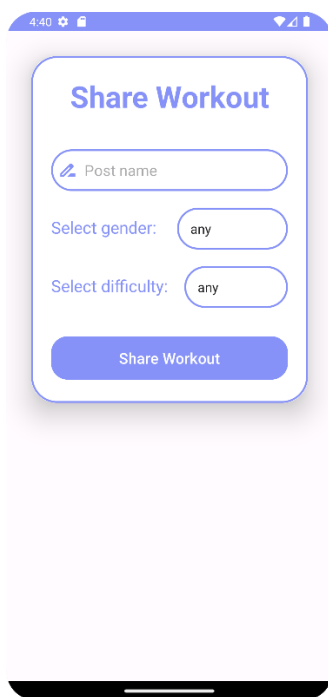
Εικόνα 5: Μενού Επιλογών για Πρόγραμμα Γυμναστικής

Εάν επιλέξετε να μετονομάσετε ένα υπάρχον πρόγραμμα ή να δημιουργήσετε ένα νέο, θα εμφανιστεί ένα αναδυόμενο παράθυρο όπου μπορείτε να εισάγετε το όνομα. Αφού πληκτρολογήσετε το επιθυμητό όνομα, κάντε κλικ στο κουμπί «OK» για να αποθηκεύσετε τις αλλαγές σας. Αν αλλάξετε γνώμη μπορείτε να κλικάρετε στο κουμπί «Cancel».



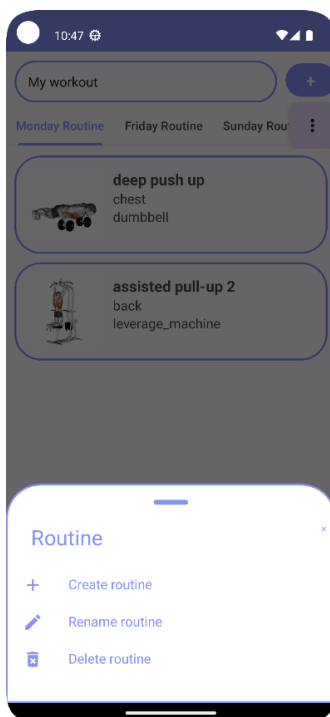
Εικόνα 6: Αναδυόμενο Παράθυρο Δημιουργίας Προγράμματος Γυμναστικής

Αν επιλέξετε να μοιραστείτε το πρόγραμμά σας, θα μεταφερθείτε σε μια φόρμα όπου μπορείτε να δώσετε πρόσθετες λεπτομέρειες για το πρόγραμμά σας, όπως το όνομά του, το επίπεδο δυσκολίας (αρχάριος, μέσος, προχωρημένος) και το φύλο-στόχο. Αφού συμπληρώσετε αυτές τις πληροφορίες, κάντε κλικ στο κουμπί «Share Workout» για να καταστήσετε το πρόγραμμά σας διαθέσιμο στο τμήμα της κοινότητας για να το δουν και να το κατεβάσουν άλλοι χρήστες.



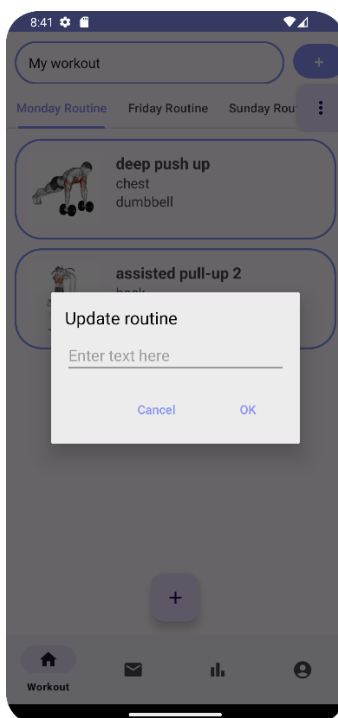
Εικόνα 7: Φόρμα Κοινοποίησης Προγράμματος

Ομοίως, δίπλα από το όνομα του προγράμματός σας, υπάρχει ένα άλλο κουμπί «:». Πατώντας αυτό το κουμπί θα εμφανιστεί ένα παράθυρο διαλόγου στο κάτω μέρος της οθόνης όπου μπορείτε να μετονομάσετε, να διαγράψετε την τρέχουσα επιλεγμένη ρουτίνα σας ή να δημιουργήσετε μια νέα.



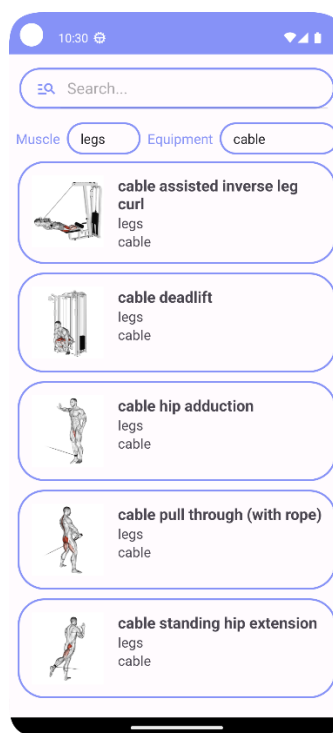
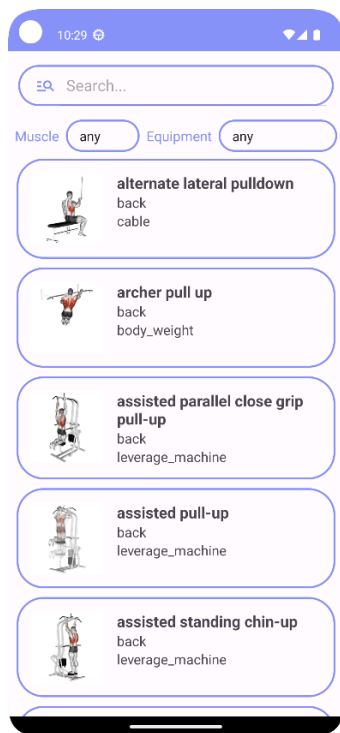
Εικόνα 8: Μενού Επιλογών για Ρουτίνα

Εάν επιλέξετε την επιλογή για τη μετονομασία μιας υπάρχουσας ρουτίνας ή τη δημιουργία μιας νέας, θα εμφανιστεί ένα αναδυόμενο παράθυρο για την εισαγωγή του ονόματος. Κάντε κλικ στο κουμπί «OK» όταν τελειώσετε για να αποθηκεύσετε τις αλλαγές σας. Αν αλλάξετε γνώμη μπορείτε να κλικάρετε στο κουμπί «Cancel».



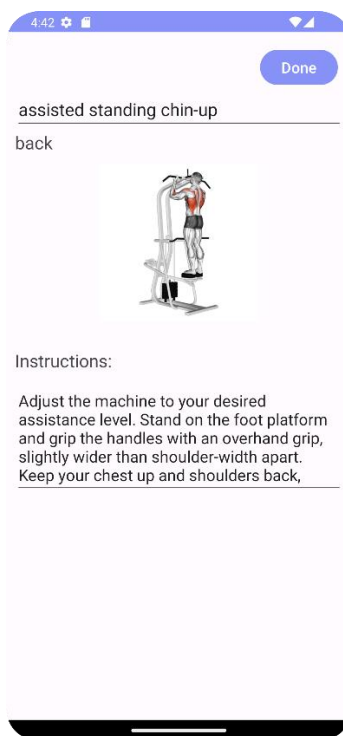
Εικόνα 9: Αναδυόμενο Παράθυρο Μετονομασίας Ρουτίνας

Στο κάτω μέρος της οθόνης στο κέντρο, υπάρχει ένα κουμπί «+» που σας ανακατευθύνει σε άλλη σελίδα για την προσθήκη ασκήσεων στην τρέχουσα επιλεγμένη ρουτίνα σας. Εδώ, μπορείτε να επιλέξετε από πάνω από 1300 ασκήσεις, κάνοντας αναζήτηση χρησιμοποιώντας τη μπάρα αναζήτησης ή φιλτράροντάς τες ανά μυϊκή ομάδα ή/και τύπο εξοπλισμού.



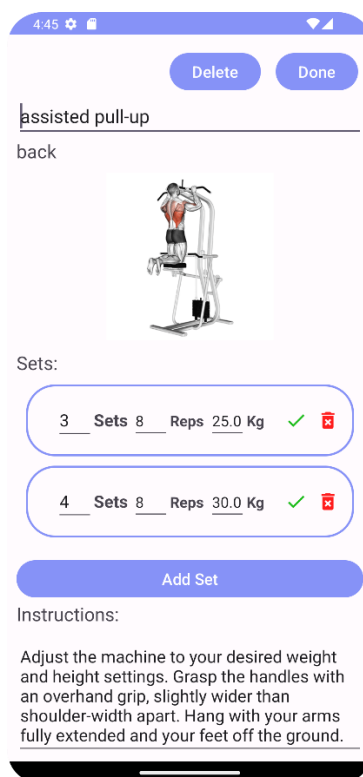
Εικόνα 10: Αφιλτράριστη αναζήτηση ασκήσεων **Εικόνα 11: Φιλτραρισμένη αναζήτηση ασκήσεων**

Όταν επιλέξετε μια άσκηση κάνοντας κλικ πάνω της, θα μεταφερθείτε για να δείτε τις λεπτομέρειές της. Για να ολοκληρώσετε την προσθήκη της άσκησης στη ρουτίνα σας, απλά κάντε κλικ στο κουμπί «Done» στην επάνω δεξιά γωνία της οθόνης.



Εικόνα 12: Λεπτομέρειες Άσκησης

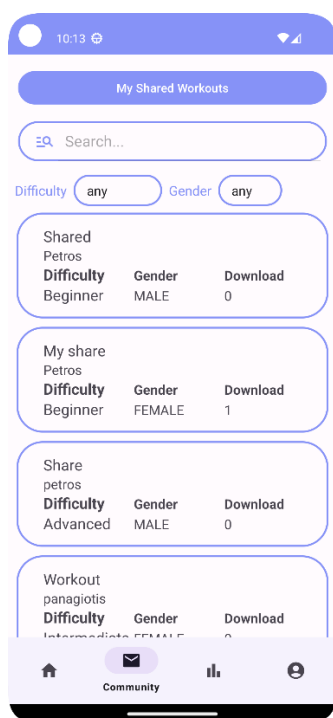
Αφού προσθέσετε μια άσκηση, θα τη δείτε να εμφανίζεται στην αντιστοιχη ρουτίνα του προγράμματός σας. Κάνοντας κλικ στην άσκηση, θα μεταφερθείτε στη σελίδα με τις πληροφορίες της, όπου μπορείτε να τη μετονομάσετε, να προσθέσετε σετ, επαναλήψεις, βάρη και να δείτε τις οδηγίες για την εκτέλεσή της. Εάν χρειαστεί, μπορείτε να επεξεργαστείτε αυτές τις οδηγίες ώστε να ταιριάζουν στις προτιμήσεις σας. Στην επάνω δεξιά γωνία της οθόνης, υπάρχουν τα κουμπιά «Delete» και «Done». Χρησιμοποιήστε το κουμπί «Διαγραφή» για να αφαιρέσετε την άσκηση από το πρόγραμμά σας και το κουμπί «Done» για να αποθηκεύσετε τυχόν αλλαγές που κάνατε.



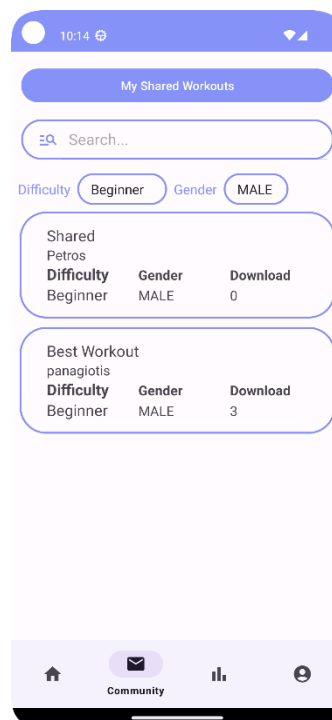
Εικόνα 13: Επεξεργασία Άσκησης

Ενότητα 2: Εξερεύνηση και κοινή χρήση προγραμμάτων

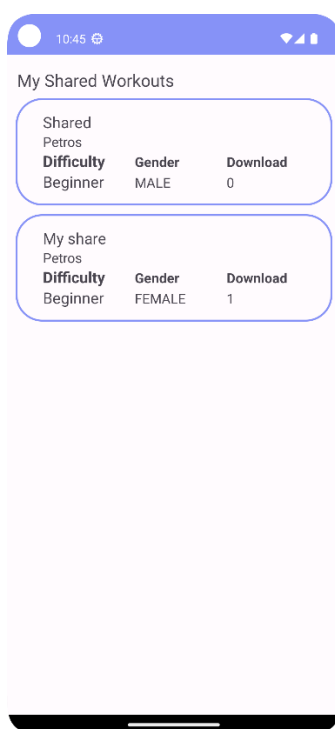
Η δεύτερη ενότητα επικεντρώνεται στο διαμοιρασμό των προγραμμάτων. Στην κορυφή αυτής της σελίδας, υπάρχει ένα κουμπί που σας επιτρέπει να δείτε τα κοινοποιημένα από εσάς προγράμματα. Μπορείτε να περιηγηθείτε σε αυτά ή να χρησιμοποιήσετε τη μπάρα αναζήτησης μαζί με τα φίλτρα φύλου και ηλικίας για να βρείτε τα κοινοποιημένα προγράμματα άλλων χρηστών.



Εικόνα 14: Αφιλτράριστη αναζήτηση

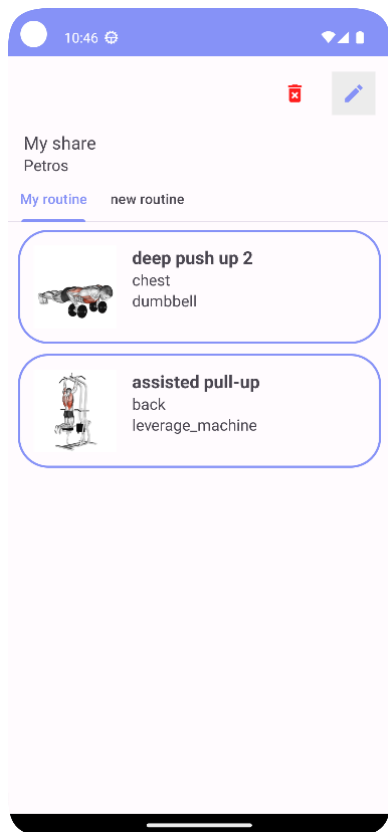


Εικόνα 15: Φιλτραρισμένη αναζήτηση

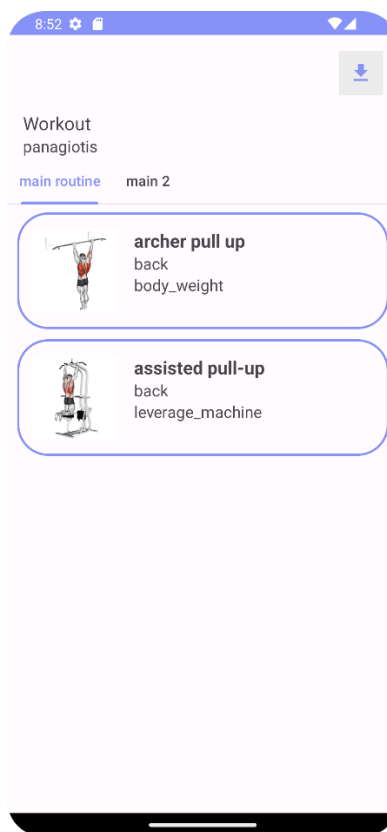


Εικόνα 16: Κοινοποιημένα Προγράμματα Χρήστη

Όταν επιλέγετε ένα πρόγραμμα κάνοντας κλικ σε αυτό, θα ανακατευθυνθείτε για να δείτε τις πληροφορίες του. Επίσης σας δίνεται η δυνατότητα να το κατεβάσετε εάν δεν είναι δικό σας, για να το προσθέσετε στη συλλογή σας. Αν από την άλλη ανήκει σε εσάς τότε μπορείτε να το μετονομάσετε ή ακόμα και να το διαγράψετε. Αυτή η λειτουργία όχι μόνο σας βοηθά να ανακαλύψετε νέα προγράμματα προπόνησης, αλλά σας επιτρέπει επίσης να επωφεληθείτε από τη συλλογική γνώση και εμπειρία της κοινότητας.



Εικόνα 17: Κοινοποιημένο Πρόγραμμα Χρήστη

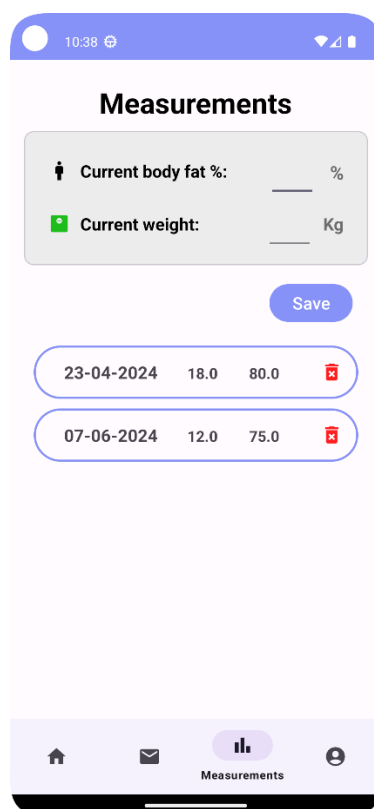


Εικόνα 18: Κοινοποιημένο Πρόγραμμα Άλλου

Ενότητα 3: Παρακολούθηση των μετρήσεων του σώματος

Στην τρίτη ενότητα, μπορείτε να εισάγετε και να παρακολουθείτε τις μετρήσεις του σώματός σας, όπως το ποσοστό σωματικού λίπους και το βάρος σας. Αυτή η ενότητα είναι ζωτικής σημασίας για την παρακολούθηση της προόδου της φυσικής σας κατάστασης με την πάροδο του χρόνου. Για να εισαγάγετε τις μετρήσεις σας, απλώς εισαγάγετε το τρέχον ποσοστό σωματικού λίπους και το βάρος σας και, στη συνέχεια, κάντε κλικ στο κουμπί «Save» για να αποθηκεύσετε αυτές τις καταχωρήσεις. Η εφαρμογή θα διατηρεί ένα ιστορικό όλων των προηγούμενων μετρήσεών σας, εμφανίζοντάς τις στην ίδια σελίδα για εύκολη αναφορά σε αυτές. Αυτά τα δεδομένα σας επιτρέπουν να βλέπετε την πρόοδό σας με την πάροδο του χρόνου και να κάνετε προσαρμογές στο πρόγραμμα γυμναστικής σας, ανάλογα με τις ανάγκες.

Αν χρειαστεί να αφαιρέσετε τυχόν ξεπερασμένες ή λανθασμένες καταχωρήσεις, μπορείτε να διαγράψετε μεμονωμένες μετρήσεις από αυτή τη σελίδα. Αυτό συμβάλλει στη διατήρηση της ακρίβειας και της επικαιρότητας των δεδομένων προόδου σας. Η τακτική παρακολούθηση αυτών των μετρήσεων θα σας προσφέρει πολύτιμες πληροφορίες σχετικά με το πώς το πρόγραμμα γυμναστικής σας επηρεάζει τα γενικά επίπεδα υγείας και φυσικής κατάστασης.

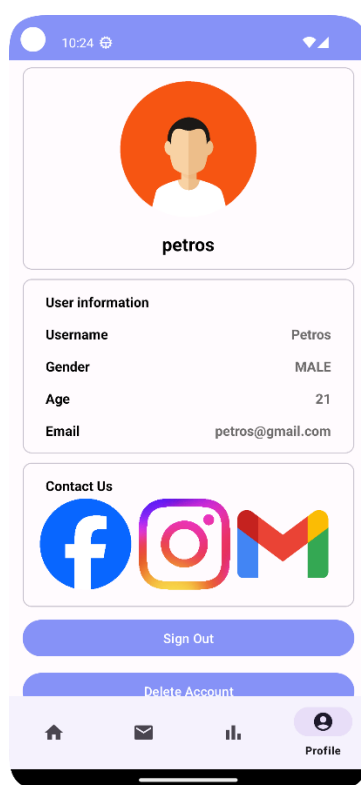


Εικόνα 19: Ενότητα Μετρήσεων

Ενότητα 4: Διαχείριση του προφίλ

Η τέταρτη ενότητα παρέχει αναλυτικές πληροφορίες σχετικά με το προφίλ σας. Εδώ, μπορείτε να δείτε τα προσωπικά σας στοιχεία, όπως τη διεύθυνση ηλεκτρονικού ταχυδρομείου σας, την ηλικία σας, το όνομα χρήστη και άλλες σχετικές πληροφορίες. Η διατήρηση αυτών των πληροφοριών σε ισχύ διασφαλίζει ότι το προφίλ σας αντικατοπτρίζει με ακρίβεια τα προσωπικά σας στοιχεία, γεγονός που μπορεί να βελτιώσει την εμπειρία σας με την εφαρμογή.

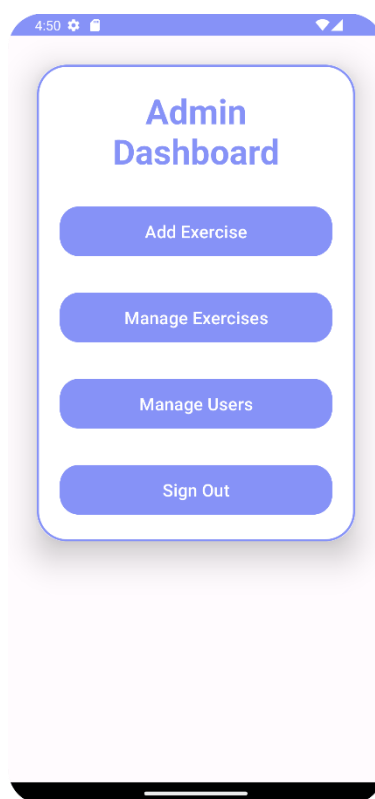
Επιπλέον, αυτή η ενότητα περιλαμβάνει στοιχεία επικοινωνίας για την υποστήριξη πελατών, διευκολύνοντας τη λήψη βοήθειας σε περίπτωση που αντιμετωπίσετε οποιοδήποτε πρόβλημα ή έχετε ερωτήσεις σχετικά με την εφαρμογή. Στο κάτω μέρος αυτής της ενότητας, θα βρείτε δύο σημαντικά κουμπιά: ένα για την έξοδο από το λογαριασμό σας και ένα για την πλήρη διαγραφή του λογαριασμού σας. Χρησιμοποιήστε το κουμπί «Sign Out» αν θέλετε να αποσυνδεθείτε προσωρινά από την εφαρμογή. Αν αποφασίσετε να σταματήσετε να χρησιμοποιείτε την εφαρμογή συνολικά, μπορείτε να χρησιμοποιήσετε το κουμπί «Delete Account» για να διαγράψετε οριστικά τον λογαριασμό σας και όλα τα σχετικά δεδομένα.



Εικόνα 20: Ενότητα για Διαχείριση του Προφίλ

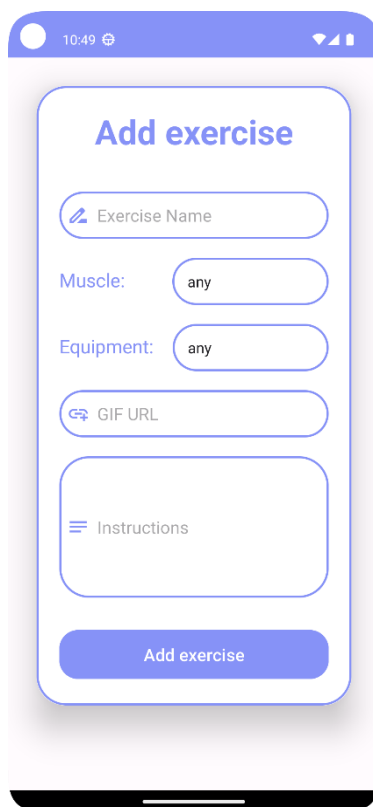
Εγχειρίδιο χρήσης (για το διαχειριστή)

Ως διαχειριστής της εφαρμογής, έχετε πρόσβαση σε ένα εξειδικευμένο περιβάλλον με εργαλεία για τη διαχείριση ασκήσεων και χρηστών. Όταν συνδεθείτε με τα διαπιστευτήριά σας ως διαχειριστής, θα δείτε μια σελίδα με τέσσερα κουμπιά, καθένα από τα οποία παρέχει διαφορετικές λειτουργίες.



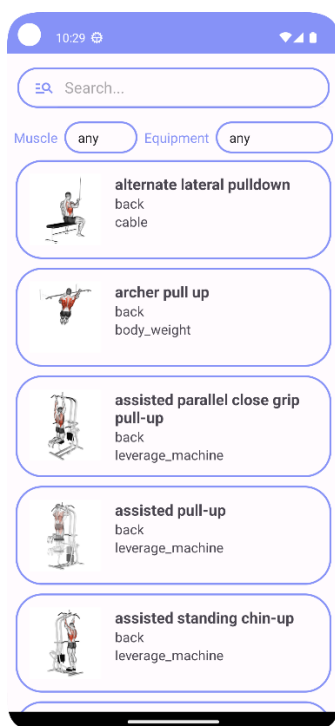
Εικόνα 21: Ταμπλό Διαχειριστή

Για να προσθέσετε μια νέα άσκηση, κάντε κλικ στο πρώτο κουμπί με την ένδειξη «Add Exercise». Αυτό θα σας ανακατευθύνει σε μια φόρμα όπου μπορείτε να εισάγετε τα στοιχεία της νέας άσκησης. Πρέπει να συμπληρώσετε το όνομα της άσκησης, να καθορίσετε τη μουσική ομάδα-στόχο, να υποδείξετε τον απαιτούμενο εξοπλισμό, να δώσετε μια διεύθυνση URL για το GIF της άσκησης (gifUrl) ώστε να υπάρξει οπτική επίδειξη της άσκησης και να γράψετε λεπτομερείς οδηγίες για τον τρόπο εκτέλεσης. Αφού συμπληρώσετε τη φόρμα, κάντε κλικ στο κουμπί «Add Exercise» για να αποθηκεύσετε τη νέα άσκηση στη βάση δεδομένων.



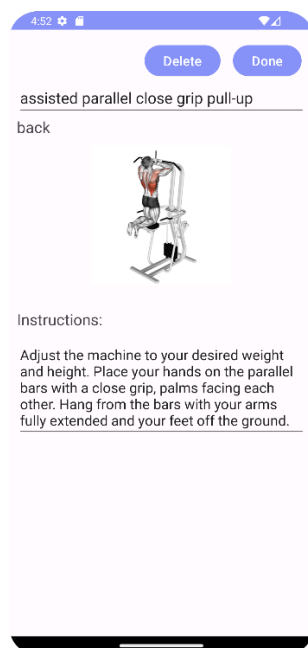
Εικόνα 22: Φόρμα Προσθήκης Άσκησης

Για τη διαχείριση των υφιστάμενων ασκήσεων, κάντε κλικ στο δεύτερο κουμπί με την ένδειξη «Manage Exercises». Αυτό θα σας μεταφέρει σε μια σελίδα όπου μπορείτε να αναζητήσετε ασκήσεις. Χρησιμοποιήστε τη γραμμή αναζήτησης για να βρείτε ασκήσεις με βάση το όνομά τους και εφαρμόστε φίλτρα μυών και εξοπλισμού για να περιορίσετε την αναζήτησή σας.



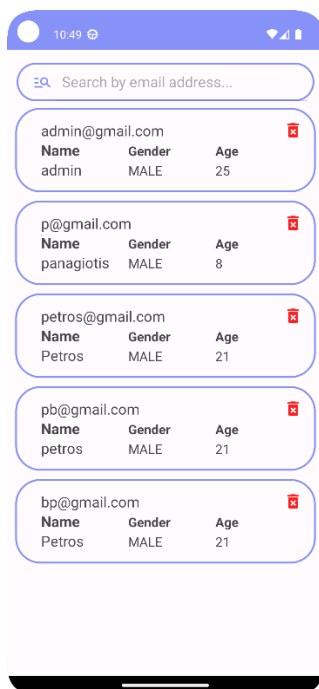
Εικόνα 23: Αναζήτηση Ασκήσεων

Όταν επιλέξετε μια άσκηση από τα αποτελέσματα της αναζήτησης, θα μεταφερθείτε στη σελίδα με τις λεπτομέρειές της. Εδώ, μπορείτε να μετονομάσετε την άσκηση και να τροποποιήσετε τις οδηγίες της. Για να αποθηκεύσετε τις αλλαγές σας, κάντε κλικ στο κουμπί «Done». Εάν πρέπει να διαγράψετε μια άσκηση, κάντε κλικ στο κουμπί «Delete» στην επάνω δεξιά γωνία της σελίδας λεπτομερειών της άσκησης.



Εικόνα 24: Επεξεργασία Άσκησης

Για να διαχειριστείτε τους εγγεγραμμένους χρήστες, κάντε κλικ στο τρίτο κουμπί με την ένδειξη «Manage Users». Αυτό θα σας οδηγήσει σε μια σελίδα που εμφανίζει όλους τους εγγεγραμμένους χρήστες. Εδώ, μπορείτε να προβάλετε τα στοιχεία των χρηστών και να διαγράψετε χρήστες, εάν είναι απαραίτητο. Για να διαγράψετε έναν χρήστη, απλά κάντε κλικ στο εικονίδιο του κάδου απορριμμάτων δίπλα στα στοιχεία του χρήστη. Αυτή η ενέργεια θα διαγράψει οριστικά τον χρήστη από το σύστημα.



Εικόνα 25: Σελίδα Διαχείρισης Χρηστών

Τέλος, για να αποσυνδεθείτε, κάντε κλικ στο τέταρτο κουμπί με την ένδειξη «Αποσύνδεση». Με αυτόν τον τρόπο θα αποσυνδεθείτε από τον λογαριασμό διαχειριστή σας, διασφαλίζοντας την ασφάλεια των διαπιστευτηρίων σας και αποτρέποντας τη μη εξουσιοδοτημένη πρόσβαση.

Ακολουθώντας αυτές τις οδηγίες, μπορείτε να διαχειρίζεστε αποτελεσματικά τις ασκήσεις και τους λογαριασμούς χρηστών, εξασφαλίζοντας την ομαλή λειτουργία της εφαρμογής. Εάν αντιμετωπίσετε οποιοδήποτε πρόβλημα ή χρειάζεστε περαιτέρω βοήθεια, παρακαλούμε επικοινωνήστε με την ομάδα υποστήριξής μας για βοήθεια.

Μελλοντικές επεκτάσεις της εφαρμογής

Η τρέχουσα έκδοση της εφαρμογής παρέχει ισχυρή λειτουργικότητα για τη δημιουργία και τη διαχείριση εξατομικευμένων προγραμμάτων γυμναστικής, την παρακολούθηση της προόδου και την αλληλεπίδραση με μια κοινότητα χρηστών. Ωστόσο, έχουν προγραμματιστεί αρκετές συναρπαστικές επεκτάσεις για το μέλλον, ώστε να βελτιωθεί η εμπειρία του χρήστη και να διευρυνθούν οι δυνατότητες της εφαρμογής.

Ένας από τους πρωταρχικούς τομείς επέκτασης θα είναι η προσθήκη προηγμένων αναλύσεων και πληροφοριών. Οι χρήστες θα μπορούν να λαμβάνουν πιο λεπτομερή ανάλυση της προόδου τους, συμπεριλαμβανομένων των συγκριτικών στατιστικών των προηγούμενων μετρήσεων, τα δεδομένα προπόνησης και τις μετρήσεις του σώματός τους. Αυτή η λειτουργία θα αξιοποιεί αλγόριθμους μηχανικής μάθησης για να προσφέρει εξατομικευμένες συστάσεις και να επισημαίνει περιοχές για βελτίωση.

Για να παρέχει πιο ολοκληρωμένη παρακολούθηση, η εφαρμογή θα ενσωματωθεί με δημοφιλείς φορητές συσκευές, όπως έξυπνα ρολόγια. Αυτή η ενσωμάτωση θα επιτρέπει στους χρήστες να συγχρονίζουν τα δεδομένα προπόνησης, τους καρδιακούς παλμούς, τις συνήθειες ύπνου και άλλες μετρήσεις υγείας απευθείας με την εφαρμογή. Αυτή η απρόσκοπτη ενσωμάτωση δεδομένων θα βοηθήσει τους χρήστες να αποκτήσουν μια πιο ολιστική εικόνα του ταξιδιού τους σε θέματα υγείας και φυσικής κατάστασης.

Η βιβλιοθήκη ασκήσεων θα ενημερώνεται και θα επεκτείνεται συνεχώς, ώστε να περιλαμβάνει μεγαλύτερη ποικιλία ασκήσεων και προπονήσεων. Αυτό θα περιλαμβάνει ασκήσεις για διαφορετικά επίπεδα φυσικής κατάστασης, νέες τάσεις προπόνησης και ρουτίνες προσαρμοσμένες για συγκεκριμένους στόχους, όπως προπόνηση δύναμης, ευλυγισία ή καρδιαγγειακή υγεία. Επιπλέον, οι ασκήσεις θα κατηγοριοποιούνται με διάφορα κριτήρια, όπως το μέρος του σώματος, ο απαιτούμενος εξοπλισμός και το επίπεδο δυσκολίας.

Αναγνωρίζοντας τη σημασία της διατροφής στη γυμναστική, η εφαρμογή θα εισαγάγει μια λειτουργία παρακολούθησης διατροφής και σχεδιασμού γευμάτων. Οι χρήστες θα μπορούν να καταγράφουν την ημερήσια πρόσληψη θερμίδων, να λαμβάνουν διατροφική ανάλυση και να λαμβάνουν εξατομικευμένα σχέδια γευμάτων με βάση τους στόχους γυμναστικής και τις διατροφικές προτιμήσεις. Αυτή η λειτουργία θα παρέχει μια ολοκληρωμένη προσέγγιση της υγείας και της φυσικής κατάστασης συνδυάζοντας την άσκηση με τη σωστή διατροφή.

Η διεπαφή χρήστη θα υποβάλλεται σε συνεχείς βελτιώσεις για τη βελτίωση της χρηστικότητας και της αισθητικής. Οι μελλοντικές ενημερώσεις θα επικεντρωθούν στην παροχή μιας πιο προσαρμοσμένης προς το χρήστη εμπειρίας, επιτρέποντας στους χρήστες να εξατομικεύουν την εμφάνιση και τη λειτουργικότητα της εφαρμογής ώστε να ταιριάζει καλύτερα στις ανάγκες τους.

Για να γίνει η εφαρμογή προσιτή σε ένα ευρύτερο κοινό, θα εισαχθεί η υποστήριξη πολλαπλών γλωσσών. Αυτό θα επιτρέψει στους χρήστες από διαφορετικές περιοχές να χρησιμοποιούν την εφαρμογή στη γλώσσα της προτίμησής τους, επεκτείνοντας έτσι την παγκόσμια εμβέλεια και συμμετοχικότητα της εφαρμογής.

Με την ενσωμάτωση αυτών των επεκτάσεων, η εφαρμογή θα συνεχίσει να εξελίσσεται, προσφέροντας στους χρήστες μεγαλύτερη αξία και μια πλουσιότερη εμπειρία γυμναστικής. Αυτές οι βελτιώσεις όχι μόνο θα καλύψουν τις ποικίλες ανάγκες της τρέχουσας βάσης χρηστών, αλλά και θα προσελκύσουν νέους χρήστες που αναζητούν μια ολοκληρωμένη λύση γυμναστικής.

Συμπεράσματα

Η εφαρμογή αποτελεί μια ολοκληρωμένη λύση για τους λάτρεις της γυμναστικής που επιθυμούν να δημιουργήσουν εξατομικευμένα προγράμματα προπόνησης, να παρακολουθούν την πρόοδό τους και να συμμετέχουν σε μια υποστηρικτική κοινότητα. Με τη φιλική προς το χρήστη διεπαφή, τις προσαρμόσιμες επιλογές και τις ισχυρές δυνατότητες παρακολούθησης, η εφαρμογή δίνει στους χρήστες τη δυνατότητα να αναλάβουν τον έλεγχο του ταξιδιού τους στη γυμναστική.

Για τους χρήστες, η εφαρμογή προσφέρει τη δυνατότητα να προσαρμόζουν τις προπονήσεις στις ατομικές ανάγκες, να παρακολουθούν τις μετρήσεις του σώματος με την πάροδο του χρόνου και να έχουν πρόσβαση σε μια πληθώρα προγραμμάτων που μοιράζεται η κοινότητα. Η ενσωμάτωση προηγμένων λειτουργιών, όπως η λεπτομερής διαχείριση ασκήσεων, η παρακολούθηση της προόδου και η δυνατότητα κοινής χρήσης και λήψης προγραμμάτων, διασφαλίζει ότι οι χρήστες έχουν όλα τα εργαλεία που χρειάζονται για να παραμένουν παρακινημένοι και να επιτυγχάνουν τους στόχους γυμναστικής τους.

Για τους διαχειριστές, η εφαρμογή παρέχει μια βελτιωμένη διεπαφή για την αποτελεσματική διαχείριση των ασκήσεων και των χρηστών. Με εύχρηστες φόρμες για την προσθήκη νέων ασκήσεων και ισχυρές λειτουργίες αναζήτησης και επεξεργασίας, οι διαχειριστές μπορούν να διασφαλίσουν ότι η βιβλιοθήκη ασκήσεων παραμένει επίκαιρη και πολύτιμη για τους χρήστες. Επιπλέον, τα εργαλεία διαχείρισης χρηστών βοηθούν στη διατήρηση μιας ασφαλούς και ενεργής κοινότητας.

Κοιτάζοντας μπροστά, οι προγραμματισμένες επεκτάσεις υπόσχονται να οδηγήσουν την εφαρμογή σε νέα ύψη. Η ενσωμάτωση προηγμένων αναλύσεων, φορητών συσκευών και εικονικής προπόνησης θα παρέχει στους χρήστες βαθύτερες πληροφορίες και πιο εξατομικευμένη καθοδήγηση. Η βελτίωση της βιβλιοθήκης ασκήσεων, η προσθήκη παρακολούθησης διατροφής και η επέκταση των κοινωνικών λειτουργιών θα καλύψουν ένα ευρύτερο φάσμα στόχων και προτιμήσεων γυμναστικής. Η υποστήριξη πολλαπλών γλωσσών και η ενσωμάτωση με άλλες εφαρμογές υγείας θα καταστήσουν την εφαρμογή πιο περιεκτική και διασυνδεδεμένη.

Εν κατακλείδι, η εφαρμογή είναι μια δυναμική και εξελισσόμενη πλατφόρμα που έχει σχεδιαστεί για να ανταποκρίνεται στις ποικίλες ανάγκες της κοινότητας της γυμναστικής. Τα τρέχοντα χαρακτηριστικά της προσφέρουν μια ισχυρή βάση για την διαχείριση της φυσικής κατάστασης, ενώ οι μελλοντικές επεκτάσεις υπόσχονται να βελτιώσουν ακόμη περισσότερο την αξία και τη χρησιμότητά της. Με τη συνεχή προσαρμογή στις ανάγκες των χρηστών της και την ενσωμάτωση των τελευταίων εξελίξεων στην τεχνολογία γυμναστικής, η εφαρμογή είναι έτοιμη να παραμείνει ένα κορυφαίο εργαλείο στον κλάδο της υγείας και της γυμναστικής.

Βιβλιογραφία

- 1) Politou, Eugenia, Efthimios Alepis, Maria Virvou, and Constantinos Patsakis. 2022. "Privacy and Data Protection Challenges in the Distributed Era." Springer.
- 2) Sigala, Effrosyni, Efthimios Alepis, and Constantinos Patsakis. 2020. "Measuring the Quality of Street Surfaces in Smart Cities Through Smartphone Crowdsensing." In 2020 11th International Conference on Information, Intelligence, Systems and Applications (IISA), 1–8. IEEE.
- 3) Alonistioti, Nancy, Evangelia Aikaterini Tsihrintzi, Konstantina Chrysafiadi, and Efthimios Alepis. 2023. "Requirements for Fuzzy Logic in Personalisation of Fire Emergency Alerts." In 2023 14th International Conference on Information, Intelligence, Systems & Applications (IISA), 1–8. IEEE.
- 4) Douladiris, Anargyros, and Efthimios Alepis. 2023. "Covid-19 New Cases Correlation Analysis: Weather Conditions, Citizen Traffic and Vaccination Statistics Impact in NARX Estimated Regressions in Attica, Greece." In 2023 14th International Conference on Information, Intelligence, Systems & Applications (IISA), 1–7. IEEE.
- 5) Virvou, Maria, Efthimios Alepis, George A Tsihrintzis, and Lakhmi C Jain. 2020. "Machine Learning Paradigms: Advances in Learning Analytics." Machine Learning Paradigms: Advances in Learning Analytics, 1–5.
- 6) Michail, Tselepatiotis, and Efthimios Alepis. 2023. "Design of Real-Time Multiplayer Word Game for the Android Platform Using Firebase and Fuzzy Logic." In 2023 14th International Conference on Information, Intelligence, Systems & Applications (IISA), 1–8. IEEE.
- 7) [Java Master Class – by Mama Samba Braima Nelson](#)
- 8) [Relational Database and SQL Essentials - by Mama Samba Braima Nelson](#)
- 9) [Spring Boot Master Class - by Mama Samba Braima Nelson](#)
- 10) [Spring Data JPA - by Mama Samba Braima Nelson](#)
- 11) [Spring Security - by Mama Samba Braima Nelson](#)
- 12) [Intellij IDEA Developer Guide – by Mama Samba Braima Nelson](#)
- 13) [Spring Boot Refresh Token with JWT - by bezkoder](#)
- 14) [Bottom Navigation Bar - Android Studio | Fragments | Java -by Foxandroid](#)
- 15) [RecyclerView with Detailed Activity in Android Studio using Java - by Android Knowledge](#)
- 16) [Fragments in Android Studio 2022 | Tab Layout - by Android Knowledge](#)
- 17) Google: "Volley". Available at: <https://google.github.io/volley/>
- 18) bumptech/glide: An image loading and caching library for Android focused on smooth scrolling. Available at: <https://github.com/bumptech/glide>
- 19) Head First Java, 2nd Edition by Kathy Sierra and Bert Bates
- 20) Spring Boot in Action - by Craig Walls
- 21) Head First Android Development: A Brain-Friendly Guide" - by Dawn Griffiths and David Griffiths