

**ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ – ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ****Πρόγραμμα Μεταπτυχιακών Σπουδών****Προηγμένα Συστήματα Πληροφορικής – Ανάπτυξης Λογισμικού και Τεχνητής Νοημοσύνης****Μεταπτυχιακή Διατριβή**

Τίτλος Διατριβής	(Ελληνικά) Android app εξατομικευμένης, κοινωνικής & AI-powered διατροφικής καθοδήγησης και παρακολούθησης. (Αγγλικά) Android app of personalized, social & AI-powered nutritional guidance and tracking.
Ονοματεπώνυμο Φοιτητή	ΜΠΟΜΠΟΤΗΣ ΠΕΤΡΟΣ
Πατρώνυμο	ΓΕΩΡΓΙΟΣ
Αριθμός Μητρώου	ΜΠΣΠ24029
Επιβλέπων	ΕΥΘΥΜΙΟΣ ΑΛΕΠΗΣ, Καθηγητής

Ημερομηνία Παράδοσης **Ιούλιος 2026**

3-Member Examination Committee

Τριμελής Εξεταστική Επιτροπή

Efthimios Alepis Professor

Ευθύμιος Αλέπης Καθηγητής

Maria Virvou Professor

Μαρία Βίρβου Καθηγήτρια

Dionisios Sotiropoulos Associate Professor

Σωτηρόπουλος Διονύσιος
Αναπληρωτής 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.

Περιεχόμενα

Copyright ©	3
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
Αρχιτεκτονική και Υλοποίηση Συστήματος Front-End	24
Περιβάλλον Ανάπτυξης και Θεμελιώδεις Τεχνολογίες	24
Οργάνωση Κώδικα και Αρχιτεκτονική Δομή	25
Αρχιτεκτονικό Πρότυπο MVVM και Διαχείριση Κατάστασης.....	26
Πλοήγηση και UI Layout	27
Ασύγχρονος Προγραμματισμός και Επικοινωνία με το Back-End.....	28
Προηγμένα Στοιχεία Διεπαφής και Οπτική Εμπειρία	28
Εγχειρίδιο χρήσης.....	29
Εγγραφή και Σύνδεση	29
Πίνακας Ελέγχου και Διαχείριση Γευμάτων	32
Αναζήτηση και Διαχείριση Συνταγών.....	36
Στατιστικά, Πρόοδος και Σωματικές Μετρήσεις	39

Κοινότητα και Κοινωνική Αλληλεπίδραση.....	42
Διαχείριση Προσωπικού Προφίλ	45
Προσωπικός Τεχνητός Διαιτολόγος	48
Μελλοντικές επεκτάσεις της εφαρμογής	50
Αξιολόγηση Χρηστών	51
Μεθοδολογία και Σχεδιασμός Έρευνας.....	51
Δημογραφικά Στοιχεία και Εξοικείωση Δείγματος.....	51
Ανάλυση Αποτελεσμάτων Αξιολόγησης	52
Σχόλια Χρηστών και Μελλοντικές Κατευθύνσεις	56
Συμπεράσματα	57
Βιβλιογραφία.....	58

Περίληψη (Abstract)

English

This master thesis focuses on the design and development of an advanced Android application called NutriHub, which aims at personalized monitoring of nutrition and physical progress. Architecturally, the system consists of a powerful back-end, developed with the Spring Boot framework, leveraging technologies such as Spring Security -OAuth2/JWT- for secure authentication, Spring Data JPA and PostgreSQL database. The front-end of the application was implemented entirely with Kotlin's modern UI framework Jetpack Compose, based on the MVVM standard.

On a functional level, the app allows users to create their personal profile, accurately record their daily calorie and macronutrient intake, and visualize their progress. The system's most significant innovation lies in the integration of a Context-Aware AI Dietician. Utilising AI models, the system is dynamically fed with the user's biometrics and current nutritional balance, returning targeted meal recommendations in real-time.

The ultimate goal of the work is to create a user-friendly digital tool that goes beyond simple calorie tracking, providing active, intelligent and personalized guidance to improve health and fitness.

Ελληνικά

Η παρούσα μεταπτυχιακή διατριβή επικεντρώνεται στον σχεδιασμό και την ανάπτυξη μιας προηγμένης Android εφαρμογής με την ονομασία NutriHub, η οποία στοχεύει στην εξατομικευμένη παρακολούθηση της διατροφής και της σωματικής προόδου. Αρχιτεκτονικά, το σύστημα αποτελείται από ένα ισχυρό back-end, ανεπτυγμένο με το Spring Boot Framework, αξιοποιώντας τεχνολογίες όπως Spring Security -OAuth2/JWT- για την ασφαλή ταυτοποίηση, Spring Data JPA και βάση δεδομένων PostgreSQL. Το front-end της εφαρμογής υλοποιήθηκε εξ ολοκλήρου με το σύγχρονο UI framework Jetpack Compose της Kotlin, βασισμένο στο πρότυπο MVVM.

Σε λειτουργικό επίπεδο, η εφαρμογή επιτρέπει στους χρήστες να δημιουργούν το προσωπικό τους προφίλ, να καταγράφουν με ακρίβεια την ημερήσια πρόσληψη θερμίδων και μακροθρεπτικών συστατικών, και να οπτικοποιούν την πρόοδό τους. Η σημαντικότερη καινοτομία του συστήματος έγκειται στην ενσωμάτωση ενός "Έξυπνου Διαιτολόγου" -Context-Aware AI Dietician. Αξιοποιώντας μοντέλα Τεχνητής Νοημοσύνης, το σύστημα τροφοδοτείται δυναμικά με τα βιομετρικά στοιχεία του χρήστη και το τρέχον διατροφικό του ισοζύγιο, επιστρέφοντας στοχευμένες προτάσεις γευμάτων σε πραγματικό χρόνο.

Απώτερος σκοπός της εργασίας είναι η δημιουργία ενός φιλικού προς τον χρήστη ψηφιακού εργαλείου, το οποίο ξεφεύγει από την απλή καταγραφή θερμίδων, παρέχοντας ενεργή, έξυπνη και προσωποποιημένη καθοδήγηση για τη βελτίωση της υγείας και της φυσικής κατάστασης.

Εισαγωγή

Στον σύγχρονο, απαιτητικό τρόπο ζωής, η σωστή διατροφή αποτελεί τον ακρογωνιαίο λίθο για τη διατήρηση της καλής υγείας και της σωματικής ευεξίας. Ωστόσο, η παρακολούθηση της ημερήσιας πρόσληψης θερμίδων και μακροθρεπτικών συστατικών, καθώς και η επιλογή των κατάλληλων γευμάτων για την επίτευξη συγκεκριμένων στόχων, αποτελεί συχνά μια χρονοβόρα και περίπλοκη διαδικασία. Υπάρχει, συνεπώς, η ανάγκη για μια έξυπνη, ολοκληρωμένη λύση που θα επιτρέπει στους χρήστες να θέτουν εξατομικευμένους διατροφικούς στόχους, να καταγράφουν την καθημερινή τους πρόοδο και να λαμβάνουν άμεση καθοδήγηση.

Η παρούσα εφαρμογή δημιουργήθηκε με σκοπό να καλύψει αυτές ακριβώς τις ανάγκες, προσφέροντας μια υπερσύγχρονη πλατφόρμα διαχείρισης διατροφής. Επιτρέπει στους χρήστες να δημιουργούν ένα λεπτομερές προφίλ με βάση τα βιομετρικά τους στοιχεία και το επίπεδο δραστηριότητάς τους και υπολογίζει αυτόματα τις εξατομικευμένες ημερήσιες ανάγκες τους. Μέσα από ένα εύχρηστο περιβάλλον καταγραφής, οι χρήστες μπορούν να προσθέτουν τρόφιμα στα γεύματά τους, παρακολουθώντας ταυτόχρονα τη διακύμανση του βάρους τους. Το στοιχείο, ωστόσο, που κάνει την εφαρμογή να ξεχωρίζει, είναι η ενσωμάτωση ενός "Έξυπνου Τεχνητού Διαιτολόγου". Το ενσωματωμένο σύστημα Τεχνητής Νοημοσύνης γνωρίζει ανά πάσα στιγμή το προφίλ του χρήστη καθώς και το ακριβές υπόλοιπο των μακροθρεπτικών του στόχων για την τρέχουσα ημέρα, προσφέροντας σε πραγματικό χρόνο εξαιρετικά εξατομικευμένες συμβουλές, προτάσεις γευμάτων και συνταγές. Παράλληλα, η εφαρμογή υποστηρίζει την αλληλεπίδραση των χρηστών μέσω μιας ψηφιακής κοινότητας, ενθαρρύνοντας την ανταλλαγή υγιεινών συνταγών και διατροφικών ιδεών.

Η εφαρμογή έχει κατασκευαστεί χρησιμοποιώντας έναν ισχυρό συνδυασμό τεχνολογιών. Στην πλευρά του back-end, αξιοποιήθηκε η γλώσσα Java σε συνδυασμό με τα frameworks Spring Boot και Spring Data JPA για τη διαχείριση του business logic και των δεδομένων μας. Ως βάση δεδομένων, χρησιμοποιείται η PostgreSQL, εξασφαλίζοντας την αποτελεσματική και ασφαλή διαχείριση των πληροφοριών. Η ασφάλεια της εφαρμογής ενισχύεται μέσω του Spring Security και της χρήσης JSON Web Tokens για την αυθεντικοποίηση των χρηστών, ενώ έχει ενσωματωθεί και η δυνατότητα άμεσης και ασφαλούς ταυτοποίησης μέσω λογαριασμού Google - Google Authentication. Η ριζική τεχνολογική αναβάθμιση συναντάται στην πλευρά του front-end, το οποίο υλοποιήθηκε αποκλειστικά σε γλώσσα Kotlin, κάνοντας χρήση του σύγχρονου εργαλείου Jetpack Compose. Αυτή η αρχιτεκτονική επιτρέπει τη δημιουργία ενός εξαιρετικά δυναμικού, γρήγορου και σύγχρονου περιβάλλοντος αλληλεπίδρασης.

Κατά τον σχεδιασμό, δόθηκε ιδιαίτερη βαρύτητα στην εμπειρία χρήστη και την προσβασιμότητα. Το UI καθοδηγεί τον χρήστη ομαλά, από την αρχική διαδικασία δημιουργίας λογαριασμού μέχρι την καθημερινή καταγραφή των γευμάτων του. Μέσα από σύγχρονα, καθαρά γραφήματα και οπτικές ενδείξεις προόδου, ο χρήστης αποκτά άμεσα πλήρη εικόνα της ημερήσιας θρεπτικής του πρόσληψης και της μακροπρόθεσμης εξέλιξης των σωματικών του μετρήσεων. Η εφαρμογή είναι πλήρως βελτιστοποιημένη για σύγχρονες κινητές συσκευές, επιτρέποντας στους χρήστες να την ενσωματώσουν απρόσκοπτα στην καθημερινή τους ρουτίνα.

Σε αυτό το σημείο, αξίζει να αναφέρουμε τα πολλαπλά οφέλη που προκύπτουν για τον ίδιο τον χρήστη. Η πλατφόρμα γεφυρώνει το χάσμα μεταξύ της απλής καταγραφής θερμίδων και της ουσιαστικής, επιστημονικά καθοδηγούμενης διατροφής, χάρη στον AI βοηθό της. Συνδυάζοντας προηγμένα τεχνολογικά χαρακτηριστικά, εργαλεία κοινότητας και έναν σχεδιασμό με απόλυτο επίκεντρο τον άνθρωπο, η εφαρμογή δεν αποτελεί απλώς ένα ψηφιακό ημερολόγιο τροφίμων, αλλά έναν ολοκληρωμένο, έξυπνο συνοδοιπόρο για όσους επιθυμούν να βελτιώσουν θεαματικά τις διατροφικές τους συνήθειες και, κατ' επέκταση, την υγεία τους.

Επισκόπηση Αρχιτεκτονικής

Η αρχιτεκτονική που ακολουθήθηκε για την υλοποίηση της εφαρμογής είναι η N-Tier αρχιτεκτονική. Αποτελείται από διακριτά επίπεδα, τα οποία περιλαμβάνουν 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, τη σχεσιακή βάση δεδομένων PostgreSQL, καθώς και τη διασύνδεση με εξωτερικές υπηρεσίες Τεχνητής Νοημοσύνης.

Η εφαρμογή Android αποτελεί το Presentation Layer και είναι το τμήμα με το οποίο ο χρήστης αλληλεπιδρά άμεσα. Για την ανάπτυξή του αξιοποιήθηκε το σύγχρονο αρχιτεκτονικό πρότυπο MVVM (Model-View-ViewModel), το οποίο διασφαλίζει τον πλήρη διαχωρισμό του UI από τη λογική διαχείρισης δεδομένων. Το επίπεδο View, υλοποιημένο με τη γλώσσα Kotlin και το σύγχρονο εργαλείο Jetpack Compose, αναλαμβάνει αποκλειστικά τη δυναμική απεικόνιση των πληροφοριών στην οθόνη. Παράλληλα, τα ViewModels διαχειρίζονται την κατάσταση (state) της εφαρμογής και περιλαμβάνουν ολόκληρο το business logic, επικοινωνώντας με τα αντίστοιχα Repositories για την άντληση και αποθήκευση των δεδομένων. Αυτός ο σχεδιασμός επιτρέπει στην εφαρμογή να αποκρίνεται άμεσα στις ενέργειες του χρήστη, διατηρώντας τον κώδικα οργανωμένο και επεκτάσιμο.

Ο back-end server χρησιμεύει ως ενδιάμεσος κρίκος μεταξύ της εφαρμογής Android και της βάσης δεδομένων. Η επικοινωνία μεταξύ τους πραγματοποιείται μέσω αιτημάτων HTTP, χρησιμοποιώντας το πρότυπο REST. Από την πλευρά του Android αποστέλλονται αιτήματα στο back-end και λαμβάνονται απαντήσεις που περιέχουν τα ανάλογα δεδομένα σε μορφή JSON. Το back-end χειρίζεται τα εισερχόμενα αιτήματα, επεξεργάζεται τα δεδομένα και αλληλεπιδρά με τη βάση δεδομένων PostgreSQL. Περιλαμβάνει διαφορετικά επίπεδα ώστε, αντί να συμβαίνουν όλες αυτές οι ενέργειες σε ένα επίπεδο, να διαχωρίζονται αυστηρά οι διάφορες ευθύνες του συστήματος.

Το επίπεδο που αναχαιτίζει και διαχειρίζεται τα αιτήματα του client είναι το REST Layer. Σε αυτό το επίπεδο ορίζονται τα API endpoints, τα οποία υλοποιούν λειτουργίες που σχετίζονται με την αυθεντικοποίηση των χρηστών, την καταγραφή γευμάτων, τον υπολογισμό μακροθρεπτικών συστατικών, την αλληλεπίδραση με τον Τεχνητό Διαιτολόγο και άλλα.

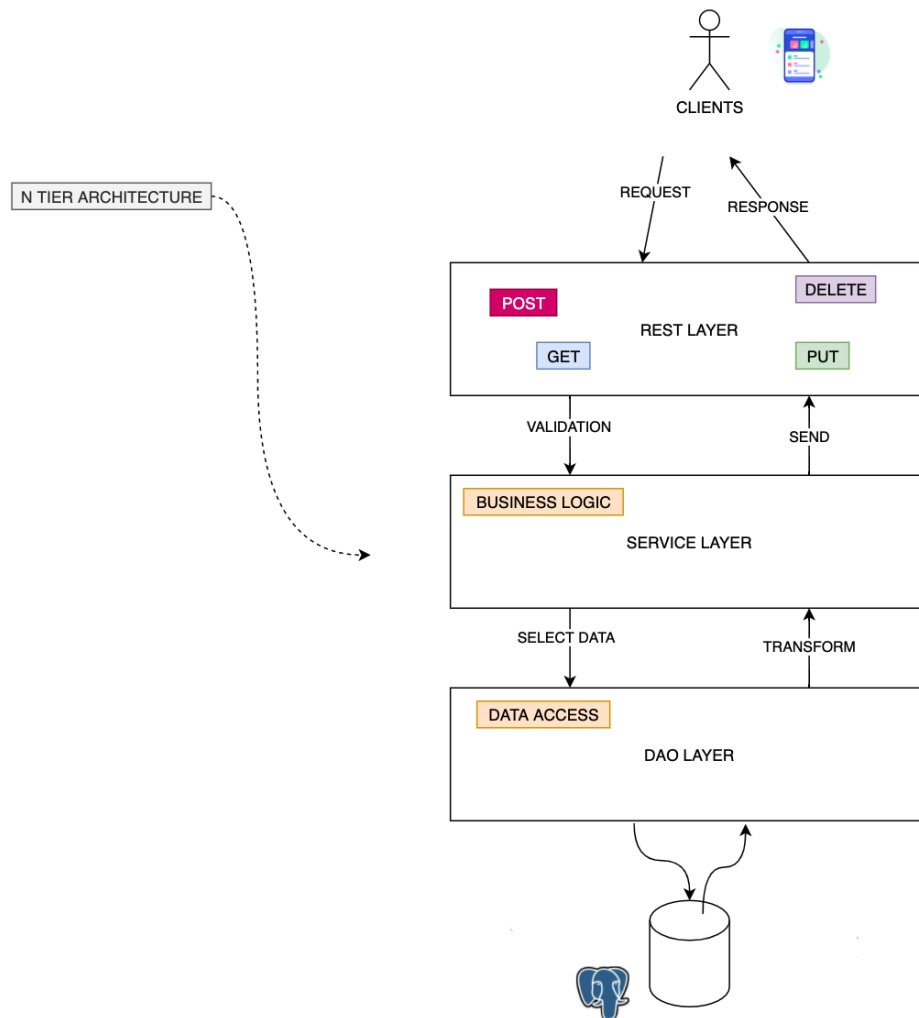
Στη συνέχεια, το REST Layer προωθεί τα αιτήματα στο Business Layer. Εδώ βρίσκεται η "καρδιά" του συστήματος και φιλοξενείται το business logic για ολόκληρο το back-end. Το επίπεδο αυτό περιέχει τη βασική λογική για τη δημιουργία και διαχείριση των ημερήσιων διατροφικών καταγραφών, τον υπολογισμό των θερμιδικών στόχων βάσει των βιομετρικών στοιχείων, τη δυναμική ενσωμάτωση του προφίλ του χρήστη στα αιτήματα προς τις εξωτερικές υπηρεσίες Τεχνητής Νοημοσύνης, αλλά και την ασφαλή διαχείριση exceptions.

Τέλος, το Business Layer επικοινωνεί με το DAO Layer, το οποίο είναι υπεύθυνο για την αλληλεπίδραση με τη βάση δεδομένων μας. Η βάση δεδομένων PostgreSQL αποθηκεύει όλα τα δεδομένα που σχετίζονται με τα προφίλ των χρηστών, τα καταγεγραμμένα τρόφιμα, το ιστορικό βάρους και το ιστορικό συνομιλιών με τον τεχνητό διαιτολόγο. Η αλληλεπίδραση με αυτήν πραγματοποιείται αποδοτικά μέσω του Spring Data JPA, το οποίο μεταφράζει τις λειτουργίες σε SQL queries.

Αξίζει να σημειωθεί πως καθοριστικό ρόλο στην αρχιτεκτονική παίζει το επίπεδο ασφαλείας, το οποίο υλοποιείται με τη χρήση του Spring Security και των JSON Web Tokens (JWT). Το σύστημα αυτό διασφαλίζει την προστατευμένη πρόσβαση στην εφαρμογή, επιβάλλοντας αυστηρή αυθεντικοποίηση χρηστών, ενώ έχει επεκταθεί για να υποστηρίξει ασφαλή ταυτοποίηση και μέσω λογαριασμών Google (OAuth2).

Αυτή η πολυεπίπεδη αρχιτεκτονική παρέχει σαφή διαχωρισμό των ευθυνών, εξασφαλίζει την ασφάλεια των δεδομένων και διευκολύνει την ταχύτατη και αποτελεσματική επικοινωνία μεταξύ του front-end, του back-end και της βάσης δεδομένων της εφαρμογής.

Το παρακάτω διάγραμμα παρέχει μια σαφή επισκόπηση του τρόπου με τον οποίο αλληλεπιδρούν μεταξύ τους τα διάφορα μέρη της εφαρμογής, δείχνοντας τη ροή των δεδομένων από το Android UI έως τον server και τη βάση.



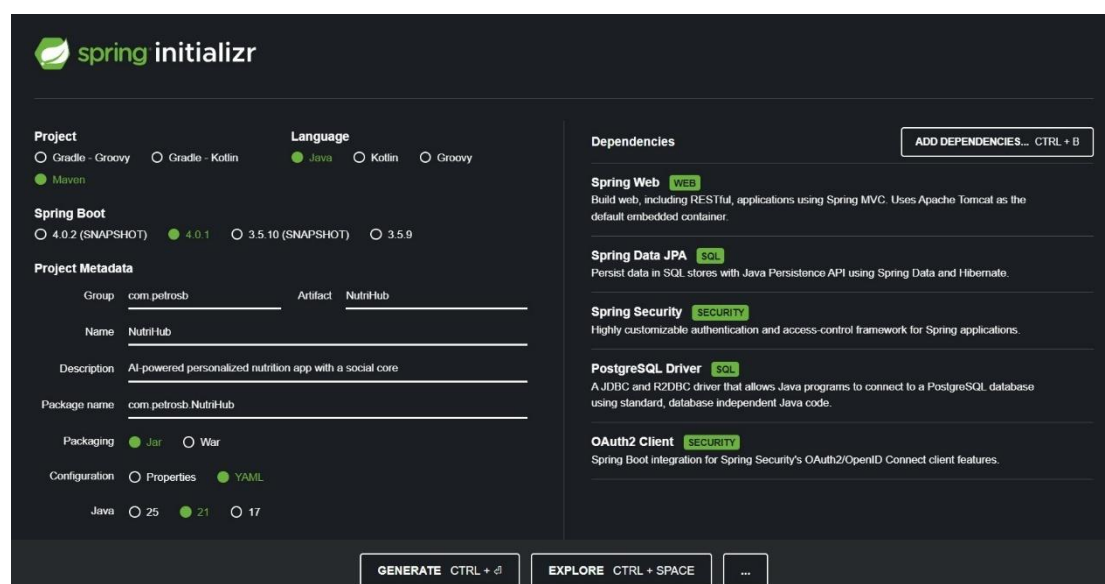
Διάγραμμα 1. Επισκόπηση αρχιτεκτονικής

Αρχικό στήσιμο εφαρμογής

Στήσιμο του back-end

Για την αρχικοποίηση της δομής του project, χρησιμοποιήθηκε το Spring Initializr. Το Spring Initializr 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 jwt για την υλοποίηση του JSON Web Token (JWT) καθώς και το java-faker για τη δημιουργία ψεύτικων δεδομένων για σκοπούς δοκιμών.



Εικόνα 1. Spring Initializr

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

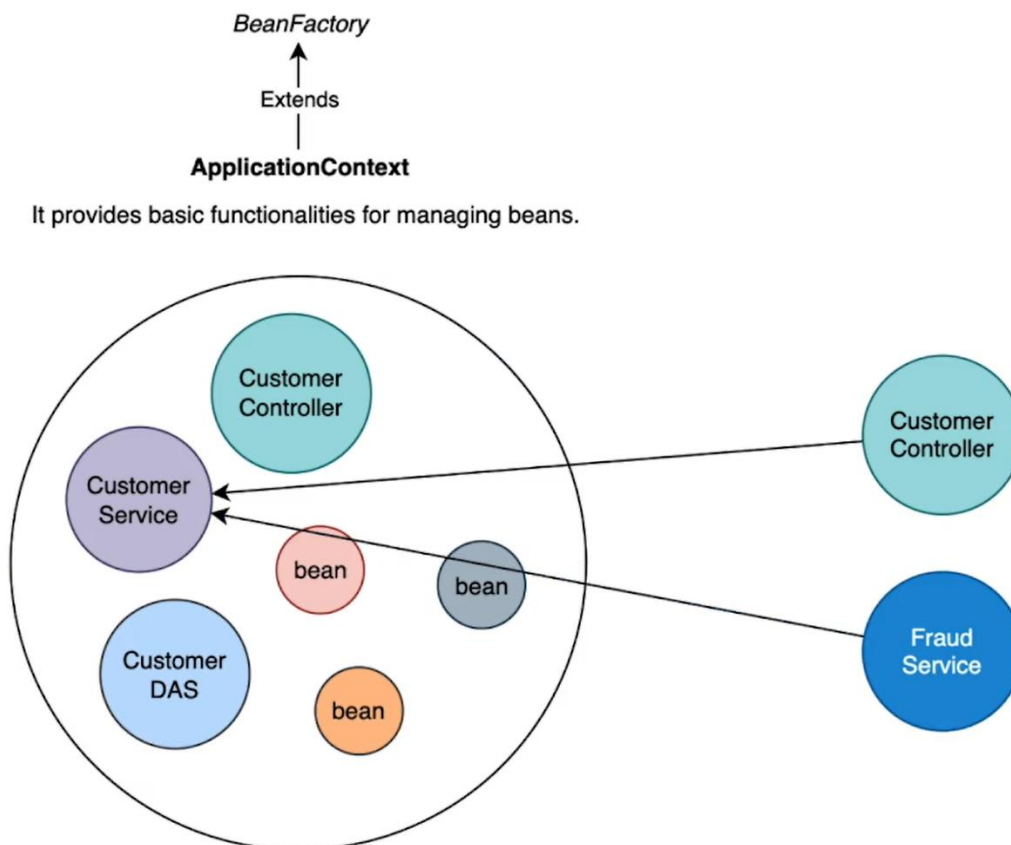
Το Spring Boot είναι ένα framework για την εύκολη κατασκευή αυτόνομων εφαρμογών. Είτε θέλει να δημιουργήσει κανείς back-end εφαρμογές είτε full-stack εφαρμογές χρησιμοποιώντας Java ή Kotlin, το Spring Boot είναι η κατάλληλη εφαρμογή. Το Spring Boot είναι μακράν το πιο δημοφιλές framework αυτή τη στιγμή. 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 περνούν 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.



Διάγραμμα 2. Application Context και Beans

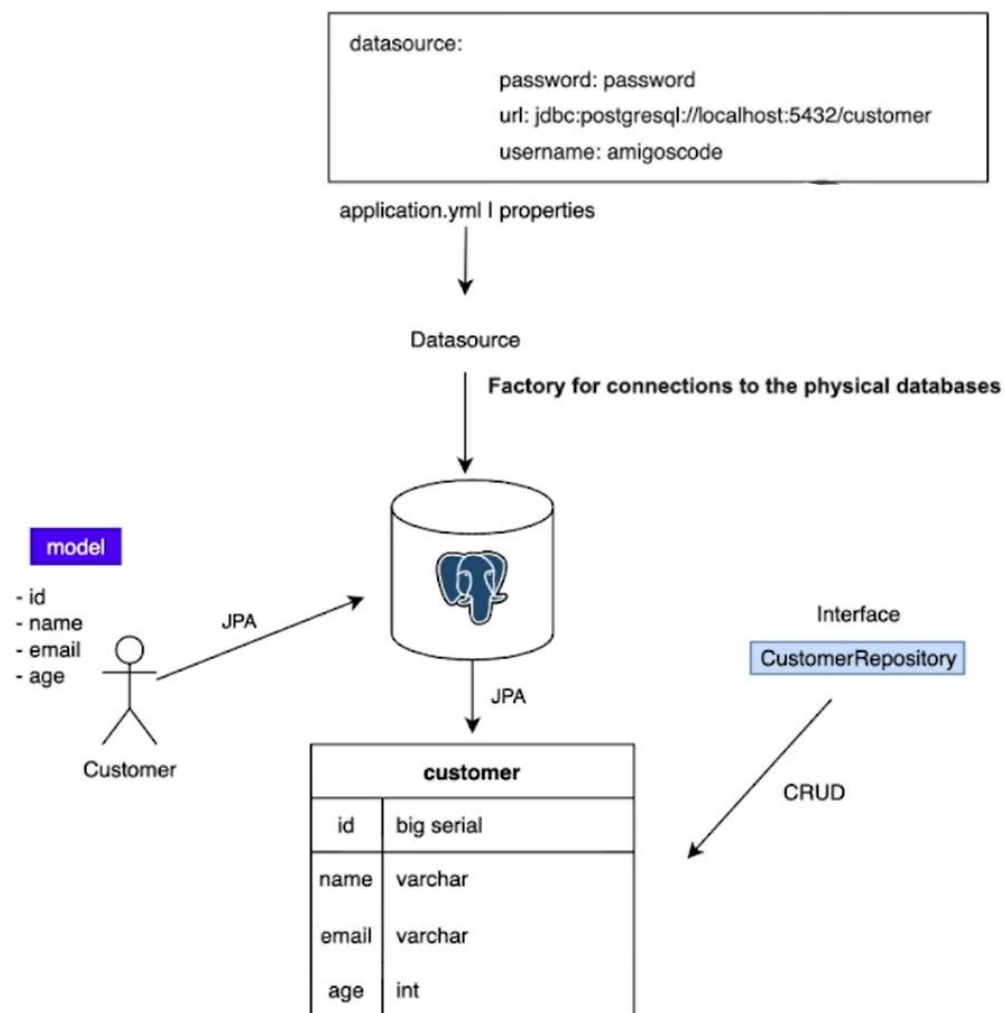
Spring Data JPA

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 queries 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.



Διάγραμμα 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 CustomerUserDetailsService, 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 τον πυρήνα της αυθεντικοποίησης βάσει ρόλων στην εφαρμογή. Η κλάση Role ορίζει ποιοι ρόλοι υπάρχουν και εξασφαλίζει την αναγνώρισή τους από το 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. 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 τα εισερχόμενα HTTP Requests. 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, 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 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.

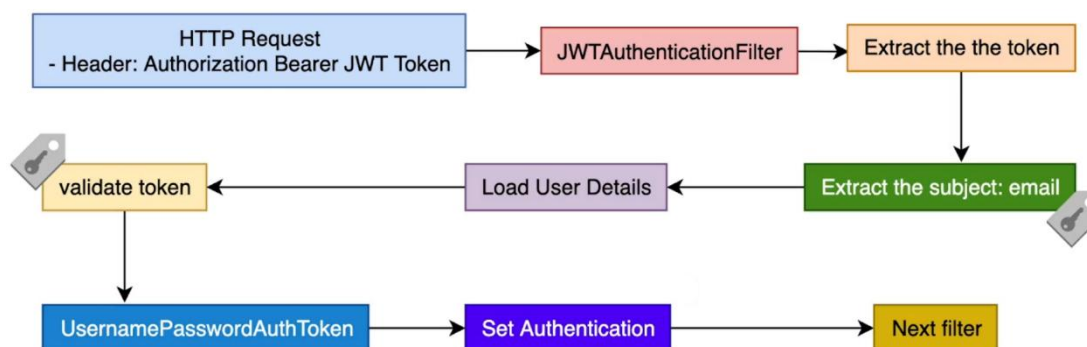
Όταν πραγματοποιείται ένα αίτημα, το φίλτρο ελέγχει πρώτα την παρουσία της επικεφαλίδας Authorization. Εάν η επικεφαλίδα είτε λείπει είτε δεν αρχίζει με «Bearer », το φίλτρο προχωράει περνώντας το αίτημα στο επόμενο φίλτρο χωρίς να εκτελεί καμία πιστοποίηση ταυτότητας.

Εάν βρεθεί έγκυρη επικεφαλίδα Authorization, το φίλτρο εξάγει το JWT από αυτήν αφαιρώντας το πρόθεμα «Bearer ». Στη συνέχεια, χρησιμοποιεί τη 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.

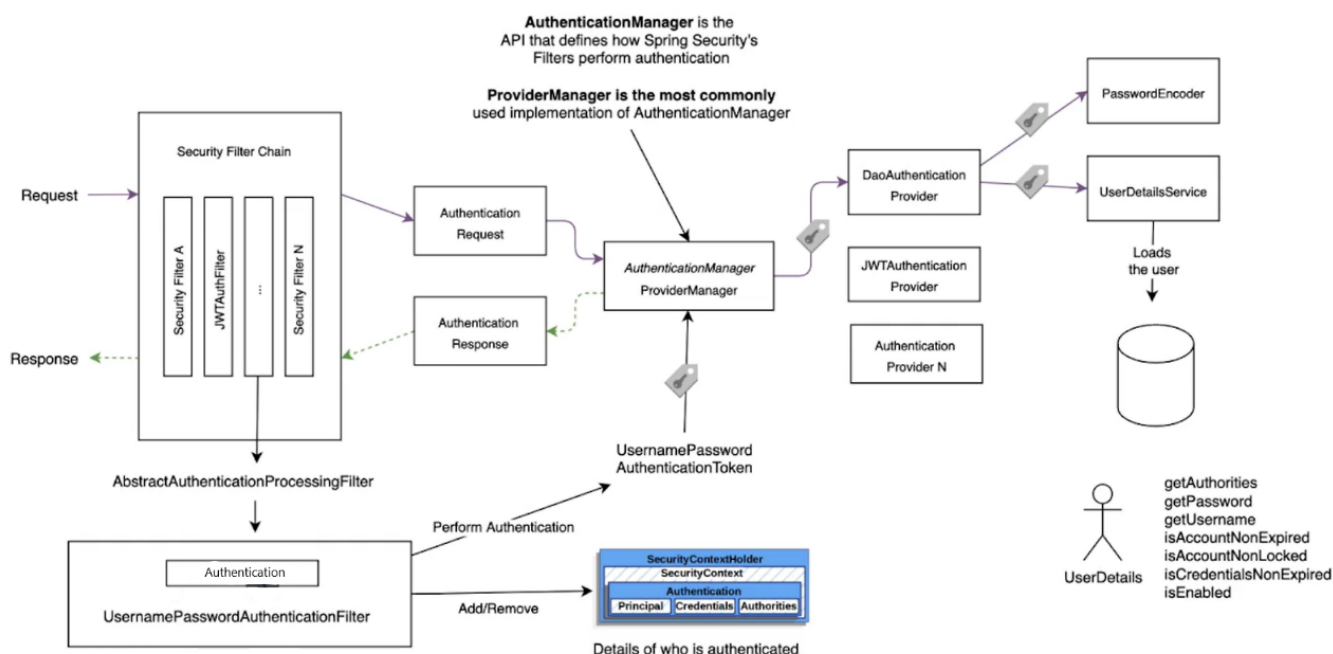
Το παρακάτω διάγραμμα παρέχει μια σαφή επισκόπηση του τρόπου με τον οποίο λειτουργεί το JWTAuthenticationFilter. Δείχνει τα διάφορα στάδια από τα οποία περνά το αίτημα HTTP.



Διάγραμμα 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.



Διάγραμμα 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 του 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

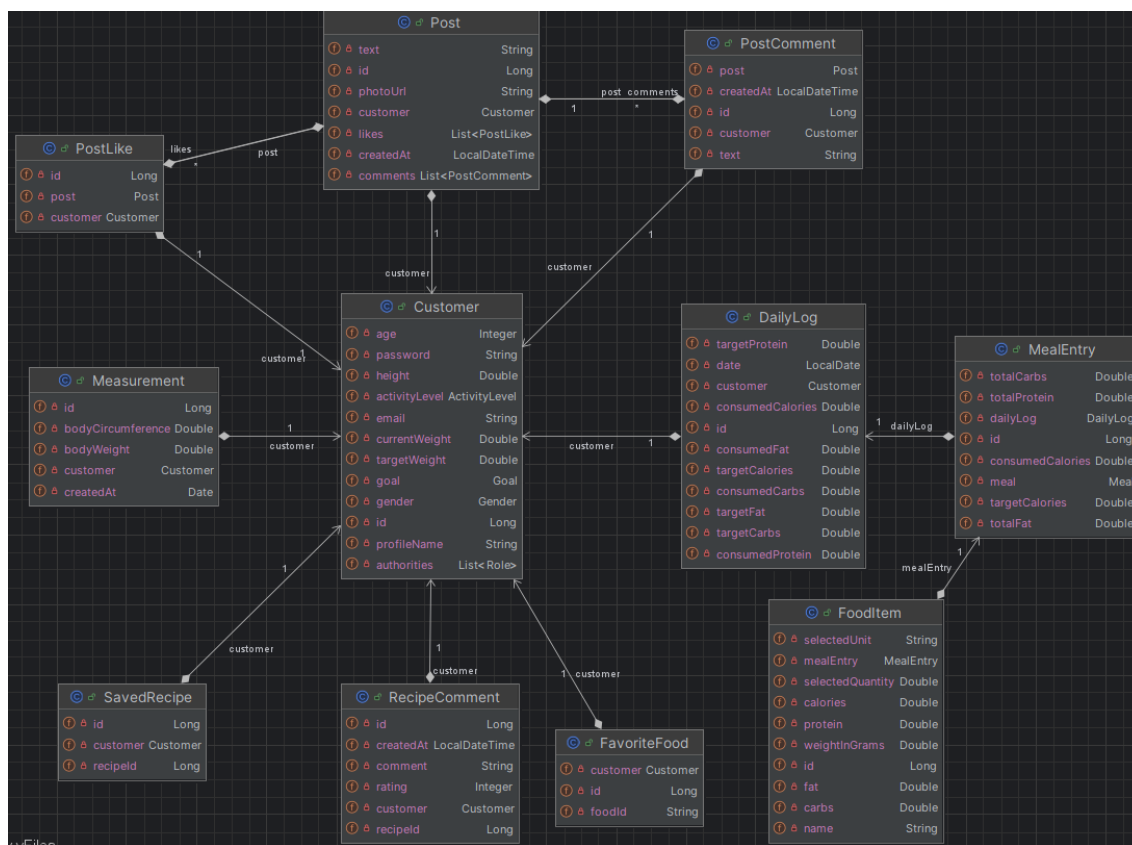
Για την απεικόνιση του διαγράμματος της βάσης δεδομένων επιλέχθηκε η χρήση ενός διαγράμματος κλάσεων UML. Το διάγραμμα κλάσεων UML παρέχει μια σαφή και δομημένη περιγραφή της δομής του συστήματος, καταγράφοντας τους τύπους των αντικειμένων και τις στατικές σχέσεις που υφίστανται μεταξύ τους.

Με τον όρο κλάση 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 κλάσεων 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 «  ») και operations public (που συμβολίζεται με «  »).

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.



Διάγραμμα 6. 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 annotations. 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.

Αρχιτεκτονική και Υλοποίηση Συστήματος Front-End

Περιβάλλον Ανάπτυξης και Θεμελιώδεις Τεχνολογίες

Για τη δημιουργία της Android πλευράς της εφαρμογής, χρησιμοποιήθηκε το Android Studio IDE, το οποίο αποτελεί το επίσημο και πλέον αξιόπιστο περιβάλλον ανάπτυξης της Google. Κατά την αρχικοποίηση του έργου, επιλέχθηκε το template «Empty Activity» με προεπιλεγμένη υποστήριξη για το Jetpack Compose. Μετά τη δημιουργία του project, το Android Studio δημιουργεί αυτόματα τα απαραίτητα αρχεία και φακέλους, όπως την κύρια κλάση MainActivity.kt, τον φάκελο res για τους πόρους, τον φάκελο theme και το θεμελιώδες αρχείο AndroidManifest.xml.

Στη συνέχεια, διασφαλίστηκε ότι το αρχείο AndroidManifest.xml έχει ρυθμιστεί σωστά με τα απαραίτητα δικαιώματα και τις ιδιότητες της εφαρμογής. Πιο συγκεκριμένα, δεδομένου ότι η εφαρμογή απαιτεί συνεχή πρόσβαση στο διαδίκτυο για την επικοινωνία με τον back-end server, περιλαμβάνει την άδεια `<uses-permission android:name="android.permission.INTERNET"/>`, διασφαλίζοντας ότι η εφαρμογή μπορεί να κάνει απρόσκοπτα HTTP requests. Παράλληλα, στο ίδιο αρχείο ρυθμίστηκε το προσαρμοσμένο θέμα εκκίνησης (Theme.App.Starting) που αξιοποιεί το Splash Screen API, εξασφαλίζοντας μια επαγγελματική και ομαλή οπτική μετάβαση κατά το άνοιγμα της εφαρμογής.

Το επόμενο βήμα ήταν η προσθήκη των απαιτούμενων dependencies στο αρχείο ρύθμισης build.gradle.kts, όπου συμπεριελήφθησαν βιβλιοθήκες όπως το Retrofit, το Dagger Hilt, το Coil, τα Kotlin Coroutines και τα Material Design 3 Components. Πιο συγκεκριμένα, το Retrofit επιλέχθηκε ως ο κύριος HTTP client, καθώς απλοποιεί τη διαδικασία υποβολής HTTP requests στο back-end API, χειρίζεται αποτελεσματικά τις HTTP responses και μετατρέπει αυτόματα τα δεδομένα JSON σε αντικείμενα Kotlin, DTOs. Η αρχιτεκτονική συνοχή ενισχύθηκε μέσω του Dagger Hilt, ενός εργαλείου Dependency Injection που αναλαμβάνει την αυτόματη δημιουργία και παροχή κλάσεων, όπως τα Repositories και τα API interfaces, στα ViewModels, μειώνοντας τον επαναλαμβανόμενο κώδικα. Αναφορικά με τον τομέα των πολυμέσων και της εμφάνισης, προστέθηκε το Coil, μια βιβλιοθήκη φόρτωσης και προσωρινής αποθήκευσης εικόνων κατασκευασμένη εξ ολοκλήρου σε Kotlin, η οποία επιτρέπει την αποτελεσματική και ταχύτατη φόρτωση εικόνων από το διαδίκτυο απευθείας μέσα στα στοιχεία του Jetpack Compose. Αντίστοιχα, τα Material 3 Components παρέχουν σύγχρονα στοιχεία και θέματα σχεδίασης, εξασφαλίζοντας μια συνεπή, προσαρμόσιμη και ελκυστική διεπαφή χρήστη. Τέλος, προστέθηκαν οι απαραίτητες βιβλιοθήκες του Android Credential Manager για την ενσωμάτωση της ασφαλούς ταυτοποίησης μέσω λογαριασμού Google.

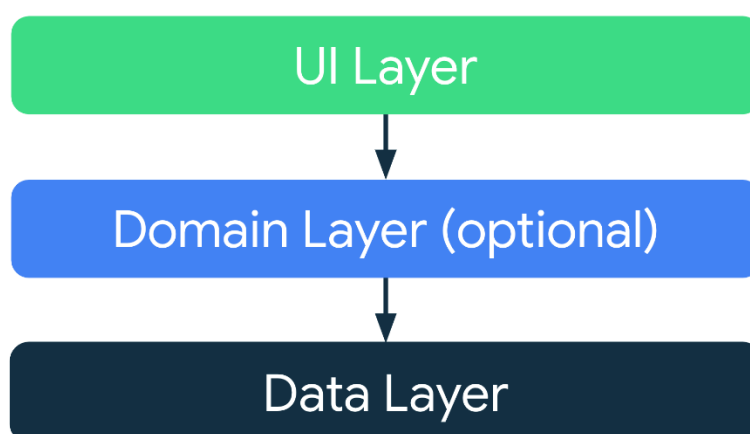
Ακολουθώντας αυτά τα βήματα και ενσωματώνοντας τα αυστηρά καθορισμένα dependencies, δημιουργείται ένα στιβαρό, ασφαλές και τεχνολογικά προηγμένο front-end περιβάλλον, χρησιμοποιώντας το Android Studio. Αυτή η ρύθμιση διασφαλίζει ότι η εφαρμογή είναι καλά εξοπλισμένη με τις πλέον βασικές και σύγχρονες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

Οργάνωση Κώδικα και Αρχιτεκτονική Δομή

Η δόμηση του πηγαίου κώδικα της εφαρμογής ακολουθεί πιστά τις αρχές Clean Architecture και του διαχωρισμού των αρμοδιοτήτων, Separation of Concerns. Ο κώδικας δεν είναι τοποθετημένος σε έναν ενιαίο χώρο, αλλά κατανέμεται ιεραρχικά σε διακριτούς φακέλους, εξασφαλίζοντας ότι κάθε τμήμα του συστήματος έχει μια σαφή και μοναδική ευθύνη. Αυτή η προσέγγιση καθιστά την εφαρμογή εξαιρετικά επεκτάσιμη και διευκολύνει την αποσφαλμάτωση και τη συντήρησή της.

Στο χαμηλότερο επίπεδο της δομής συναντάμε τον φάκελο data, ο οποίος είναι υπεύθυνος για την άντληση και τη διαχείριση της πληροφορίας. Ο φάκελος αυτός υποδιαιρείται περαιτέρω στο local τμήμα, όπου φιλοξενείται ο διαχειριστής των τοπικών δεδομένων, TokenManager, για την ασφαλή αποθήκευση των JWT tokens, και στο remote τμήμα. Στο remote ορίζεται το interface επικοινωνίας με το API (NutriHubApi), ο μηχανισμός αναχαίτισης αιτημάτων (AuthInterceptor) και ο υποφάκελος dto (Data Transfer Objects), ο οποίος περιέχει τις κλάσεις που αντικατοπτρίζουν ακριβώς τη δομή των JSON δεδομένων που ανταλλάσσονται με τον διακομιστή. Ακριβώς από πάνω βρίσκεται ο φάκελος domain, ο οποίος φιλοξενεί τα Repositories. Τα Repositories λειτουργούν ως γέφυρα, απομονώνοντας πλήρως την προέλευση των δεδομένων από το υπόλοιπο σύστημα, παρέχοντας καθαρές μεθόδους για την εκτέλεση λειτουργιών. Επιπρόσθετα, ένας ξεχωριστός φάκελος common φιλοξενεί βοηθητικές κλάσεις και σταθερές που χρησιμοποιούνται καθολικά στο έργο, όπως η κλάση Resource.

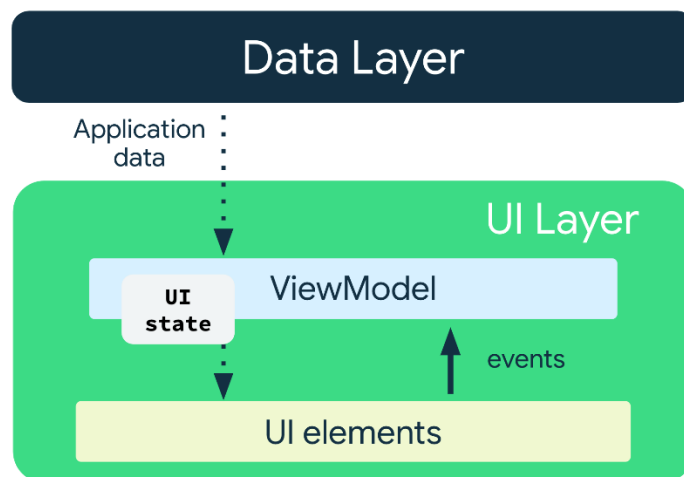
Το υψηλότερο επίπεδο ανήκει στον φάκελο presentation, όπου συγκεντρώνεται όλη η λογική του UI. Εδώ, η οργάνωση δεν γίνεται ανά τύπο αρχείου, αλλά ανά λειτουργικότητα, είναι δηλαδή feature-based. Φάκελοι όπως auth, dashboard, ai και search περιέχουν αποκλειστικά τον κώδικα που αφορά το συγκεκριμένο χαρακτηριστικό, ενθυλακώνοντας τόσο τα στοιχεία της οθόνης Composables, όσο και τα αντίστοιχα ViewModels. Τέλος, προκειμένου όλα αυτά τα ανεξάρτητα επίπεδα να μπορέσουν να συνεργαστούν αρμονικά κατά την εκτέλεση της εφαρμογής, αξιοποιείται ο φάκελος di (Dependency Injection). Εκεί βρίσκεται το AppModule, το οποίο διατηρεί τις οδηγίες που σχετίζονται με το Dependency Injection (όπως το πώς θα κατασκευαστεί ο HTTP client), αναλαμβάνοντας να "συνδέσει" τα Repositories με τα ViewModels αυτόματα, διατηρώντας ταυτόχρονα την αρχιτεκτονική με χαμηλό coupling.



Διάγραμμα 7. Διάγραμμα μιας Αρχιτεκτονικής της Εφαρμογής.

Αρχιτεκτονικό Πρότυπο MVVM και Διαχείριση Κατάστασης

Η εσωτερική λειτουργία της κάθε οθόνης της εφαρμογής είναι δομημένη αυστηρά γύρω από το αρχιτεκτονικό πρότυπο Model-View-ViewModel (MVVM). Ο στόχος αυτού του προτύπου είναι η καθιέρωση ενός Unidirectional Data Flow, διασφαλίζοντας ότι η οπτική απεικόνιση View παραμένει εντελώς ανεξάρτητη από την επιχειρηματική λογική και την ανάκτηση των δεδομένων.



Διάγραμμα 8. Τρόπος λειτουργίας του UDF

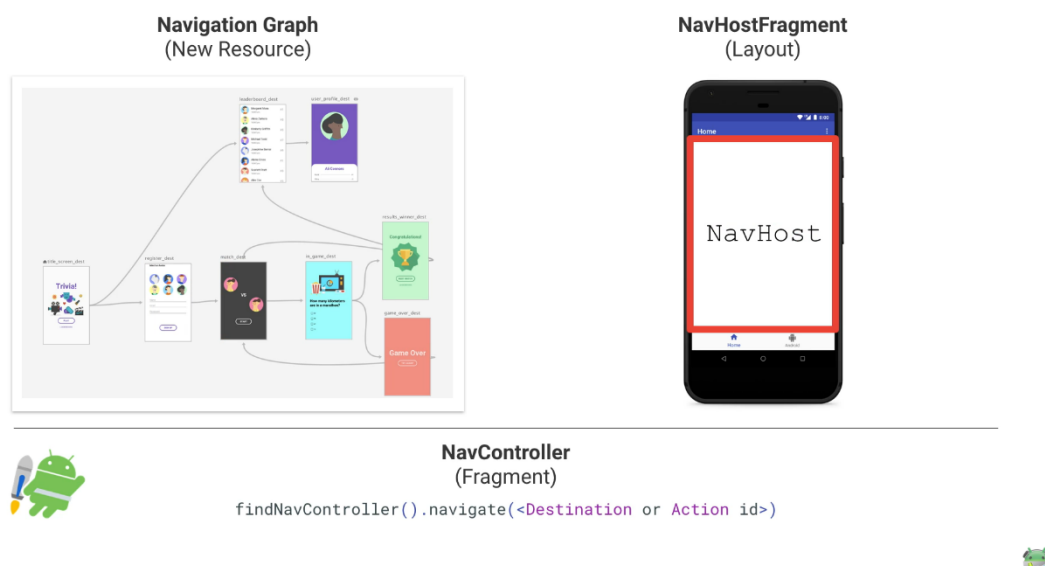
Στο επίπεδο του View, το οποίο υλοποιείται μέσω των συναρτήσεων του Jetpack Compose, η διεπαφή είναι απολύτως "παθητική". Δεν περιέχει λογική επεξεργασίας, παρά μόνο παρατηρεί τα δεδομένα και τα σχεδιάζει στην οθόνη. Κάθε τέτοια οθόνη, όπως το DashboardScreen ή το FoodDetailScreen, συνοδεύεται από το αντίστοιχο ViewModel π.χ. DashboardViewModel. Το ViewModel αποτελεί τον "εγκέφαλο" της οθόνης. Αναλαμβάνει να δεχτεί τις ενέργειες του χρήστη, όπως π.χ. το πάτημα ενός κουμπιού, να καλέσει ασύγχρονα τα αντίστοιχα Repositories μέσω των Kotlin Coroutines και στη συνέχεια να ενημερώσει το state της οθόνης.

Η διαχείριση αυτής της κατάστασης αποτελεί τον πυρήνα της διαδραστικότητας του front-end. Τα ViewModels διατηρούν το state χρησιμοποιώντας δομές όπως το MutableState για συνεχή δεδομένα, όπως κείμενα εισαγωγής ή λίστες και το MutableSharedFlow για γεγονότα μίας φοράς, όπως η πλοήγηση σε άλλη οθόνη. Χάρη στη φύση του Jetpack Compose, η διεπαφή παρακολουθεί συνεχώς αυτές τις μεταβλητές. Μόλις το ViewModel τροποποιήσει μια τιμή, το Compose ανιχνεύει την αλλαγή και ενεργοποιεί αυτόματα τη διαδικασία του recomposition, επανασχεδιάζοντας ακαριαία μόνο τα τμήματα της οθόνης που επηρεάστηκαν.

Για την απόλυτη τυποποίηση της επικοινωνίας μεταξύ του δικτύου και της διεπαφής, αξιοποιείται η βοηθητική κλάση Resource<T>. Αυτή η κλάση λειτουργεί ως ένα προστατευτικό περιβάλλον γύρω από τα δεδομένα και ενημερώνει το UI για την τρέχουσα φάση του αιτήματος. Διαθέτει τρία states: Loading (φόρτωση), Success (επιτυχία με τα δεδομένα) και Error (σφάλμα με το αντίστοιχο μήνυμα). Έτσι, όταν ένα ViewModel αιτείται δεδομένα από το API, το View ελέγχει απλώς το state του Resource. Αν είναι Loading, εμφανίζει αυτόματα έναν CircularProgressIndicator. Αν είναι Success, απεικονίζει τη λίστα με τα δεδομένα, και αν είναι Error, προβάλλει το αντίστοιχο μήνυμα σφάλματος. Αυτή η μεθοδολογία εξαλείφει την πολυπλοκότητα στη διεπαφή, προλαμβάνει απρόβλεπτα σφάλματα, crashes και εξασφαλίζει μια εξαιρετικά συνεπή και επαγγελματική εμπειρία χρήσης σε ολόκληρη την εφαρμογή.

Πλοήγηση και UI Layout

Η αρχιτεκτονική της πλοήγησης στο front-end έχει σχεδιαστεί με βάση τη σύγχρονη φιλοσοφία του Single Activity Architecture. Η κλάση MainActivity λειτουργεί ως το μοναδικό σημείο εισόδου και υποδοχής της εφαρμογής, καταργώντας την ανάγκη δημιουργίας πολλαπλών Activities, γεγονός που μειώνει δραματικά την κατανάλωση πόρων και τον χρόνο εναλλαγής μεταξύ των οθονών. Η διαχείριση της μετάβασης από τη μία οθόνη στην άλλη υλοποιείται εξ ολοκλήρου μέσω του Jetpack Compose Navigation. Το κεντρικό στοιχείο ελέγχου είναι το NavHost, το οποίο ορίζει το navigation graph και λειτουργεί ως δυναμικό container όπου εναλλάσσονται τα διάφορα Composables.



Εικόνα 2. Jetpack Compose Navigation

Για τη διασφάλιση της ορθότητας και της ασφάλειας κατά την πλοήγηση, τα routes δεν ορίζονται ως απλά strings διάσπαρτα στον κώδικα, αλλά έχουν συγκεντρωθεί σε μια sealed class με την ονομασία Screen. Η προσέγγιση αυτή εξαλείφει τα πληθώρα σφαλμάτων και επιτρέπει την ασφαλή και δομημένη μεταβίβαση παραμέτρων μεταξύ των οθονών.

Η κεντρική διάταξη της εφαρμογής θεμελιώνεται πάνω στο Scaffold, το οποίο παρέχει έτοιμες, τυποποιημένες «υποδοχές» για την τοποθέτηση βασικών στοιχείων του Material Design. Στο MainScreen, το Scaffold φιλοξενεί μια κάτω μπάρα πλοήγησης, BottomNavigationBar. Η μπάρα αυτή τροφοδοτείται από την κλάση BottomNavItem, επιτρέποντας στον χρήστη την άμεση και ταχύτατη εναλλαγή μεταξύ των τεσσάρων βασικών section της εφαρμογής: την Ημερήσια Καταγραφή (Tracking), την Αναζήτηση Τροφίμων και Συνταγών (Search), την Κοινότητα (Community) και το Προσωπικό Προφίλ (Me). Καθώς ο χρήστης αλληλεπιδρά με την κάτω μπάρα, ένας εσωτερικός NavController αναλαμβάνει να αντικαταστήσει το περιεχόμενο του κεντρικού πλαισίου με ρευστότητα, διατηρώντας το state και εξασφαλίζοντας μια απόλυτα συνεκτική εμπειρία χρήσης.

Ασύγχρονος Προγραμματισμός και Επικοινωνία με το Back-End

Η επικοινωνία του front-end με το API αποτελεί έναν από τους πιο κρίσιμους τομείς της εφαρμογής και έχει υλοποιηθεί με γνώμονα την ταχύτητα, την αξιοπιστία και την ασφάλεια. Δεδομένου ότι οι κλήσεις δικτύου απαιτούν χρόνο για να ολοκληρωθούν, η εφαρμογή αξιοποιεί πλήρως τις Kotlin Coroutines για την εκτέλεση ασύγχρονου προγραμματισμού. Κάθε αίτημα προς το API εκκινείται εντός του viewModelScope του εκάστοτε ViewModel. Με αυτόν τον τρόπο διασφαλίζεται απολύτως ότι το Main UI Thread παραμένει πάντα ελεύθερο για να ανταποκρίνεται άμεσα στις ενέργειες του χρήστη, αποτρέποντας φαινόμενα «παγώματος» της οθόνης (ANRs - Application Not Responding).

Ο HTTP client υλοποιείται μέσω της βιβλιοθήκης Retrofit. Στο αρχείο NutriHubApi, τα endpoints του REST API ορίζονται ξεκάθαρα μέσω δηλωτικών annotations, υποστηρίζοντας όλες τις μεθόδους (GET, POST, PUT, DELETE). Τα δεδομένα που αποστέλλονται και παραλαμβάνονται δεν είναι αδόμητα, αλλά μετατρέπονται αυτόματα (serialization/deserialization) σε αυστηρά ορισμένα αντικείμενα μεταφοράς δεδομένων, τα DTOs. Τα DTOs αυτά είναι οργανωμένα ανά θεματική ενότητα στον αντίστοιχο φάκελο, διασφαλίζοντας ότι η δομή της πληροφορίας στο Android ταυτίζεται απόλυτα με τις JSON αποκρίσεις της PostgreSQL βάσης του back-end.

Η ασφάλεια των επικοινωνιών έχει αυτοματοποιηθεί πλήρως μέσω του AuthInterceptor. Πρόκειται για έναν μηχανισμό που παρεμβάλλεται αθόρυβα σε κάθε εξερχόμενο HTTP αίτημα. Ο Interceptor επικοινωνεί με τον TokenManager για να ανακτήσει το αποθηκευμένο JSON Web Token του ενεργού χρήστη και το ενσωματώνει αυτόματα στο Authorization Header της κλήσης. Αυτή η αρχιτεκτονική απόφαση είναι ιδιαίτερα σημαντική, καθώς αφαιρεί την πολυπλοκότητα της διαχείρισης των tokens από τα μεμονωμένα Repositories, συγκεντρώνοντας τη λογική αυθεντικοποίησης σε ένα ενιαίο, ασφαλές σημείο ελέγχου.

Προηγμένα Στοιχεία Διεπαφής και Οπτική Εμπειρία

Η οπτική εμπειρία και η αλληλεπίδραση του χρήστη με την εφαρμογή βασίζονται στις αρχές του Material Design 3, προσφέροντας ένα σύγχρονο και καθαρό περιβάλλον. Για την αποτελεσματική και γρήγορη απεικόνιση λιστών με μεγάλο όγκο δεδομένων, όπως τα αποτελέσματα αναζήτησης χιλιάδων τροφίμων ή η ροή δημοσιεύσεων της κοινότητας, χρησιμοποιείται το δομικό στοιχείο LazyColumn του Jetpack Compose. Σε αντίθεση με τις παλαιότερες υλοποιήσεις, το LazyColumn δεν φορτώνει ολόκληρη τη λίστα στη μνήμη της συσκευής. Αντιθέτως, δημιουργεί και «σχεδιάζει» μόνο τα γραφικά συστατικά που είναι άμεσα ορατά στην οθόνη του χρήστη τη δεδομένη στιγμή, προσφέροντας απρόσκοπτη και απόλυτα ομαλό scrolling με ελάχιστο αποτύπωμα μνήμης.

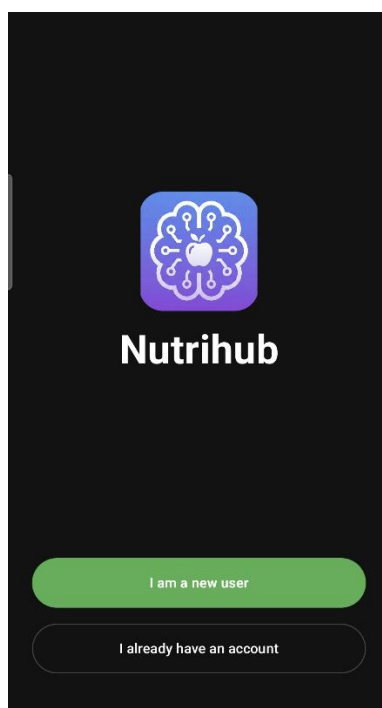
Ιδιαίτερη έμφαση έχει δοθεί στην οπτικοποίηση των δεδομένων. Στην οθόνη της ημερήσιας καταγραφής, οι ημερήσιοι στόχοι θερμίδων και μακροθρεπτικών συστατικών αποδίδονται δυναμικά μέσω γραφικών αναπαραστάσεων, όπως το CircularProgressIndicator και το LinearProgressIndicator. Τα στοιχεία αυτά υπολογίζουν σε πραγματικό χρόνο τα ποσοστά πλήρωσης των στόχων του χρήστη, προσφέροντας άμεση και κατανοητή ανατροφοδότηση. Επιπλέον, για την εκτέλεση σύνθετων ενεργειών χωρίς την απομάκρυνση του χρήστη από το τρέχον περιβάλλον του, αξιοποιούνται τα Modal Bottom Sheets. Για παράδειγμα, οι επιλογές σύνδεσης και εγγραφής, συμπεριλαμβανομένης της εγγενούς ενσωμάτωσης ταυτοποίησης μέσω Google, εμφανίζονται ως κομψά αναδυόμενα παράθυρα από το κάτω μέρος της οθόνης, διατηρώντας το οπτικό πλαίσιο της αρχικής σελίδας.

Τέλος, το feedback προς τον χρήστη για την επιτυχή ή ανεπιτυχή έκβαση των ενεργειών του διαχειρίζεται με εξαιρετικά διακριτικό τρόπο μέσω του SnackbarHost. Όταν το ViewModel εκπέμπει ένα μήνυμα σφάλματος ή επιβεβαίωσης, εμφανίζεται αυτόματα μια ενημερωτική ειδοποίηση στο κάτω μέρος της οθόνης για λίγα δευτερόλεπτα. Ολόκληρη η οπτική συνοχή της εφαρμογής ελέγχεται κεντρικά μέσω του φακέλου theme. Εκεί καθορίζονται με ακρίβεια οι χρωματικές παλέτες στο Color.kt, οι γραμματοσειρές στο Type.kt και η συνολική συμπεριφορά του θέματος στο Theme.kt, διασφαλίζοντας την απόλυτη ομοιομορφία της εφαρμογής.

Εγχειρίδιο χρήσης

Εγγραφή και Σύνδεση

Κατά την πρώτη εκκίνηση της εφαρμογής, θα σας υποδεχτεί η οθόνη φόρτωσης με το λογότυπό μας. Το σύστημα ελέγχει αθόρυβα στο παρασκήνιο εάν είστε ήδη συνδεδεμένοι. Εφόσον ανοίγετε την εφαρμογή για πρώτη φορά, θα οδηγηθείτε αυτόματα στην αρχική Οθόνη Καλωσορίσματος. Από αυτό το σημείο, έχετε δύο βασικές επιλογές: τη δημιουργία ενός νέου λογαριασμού ή τη σύνδεση σε έναν ήδη υπάρχοντα.



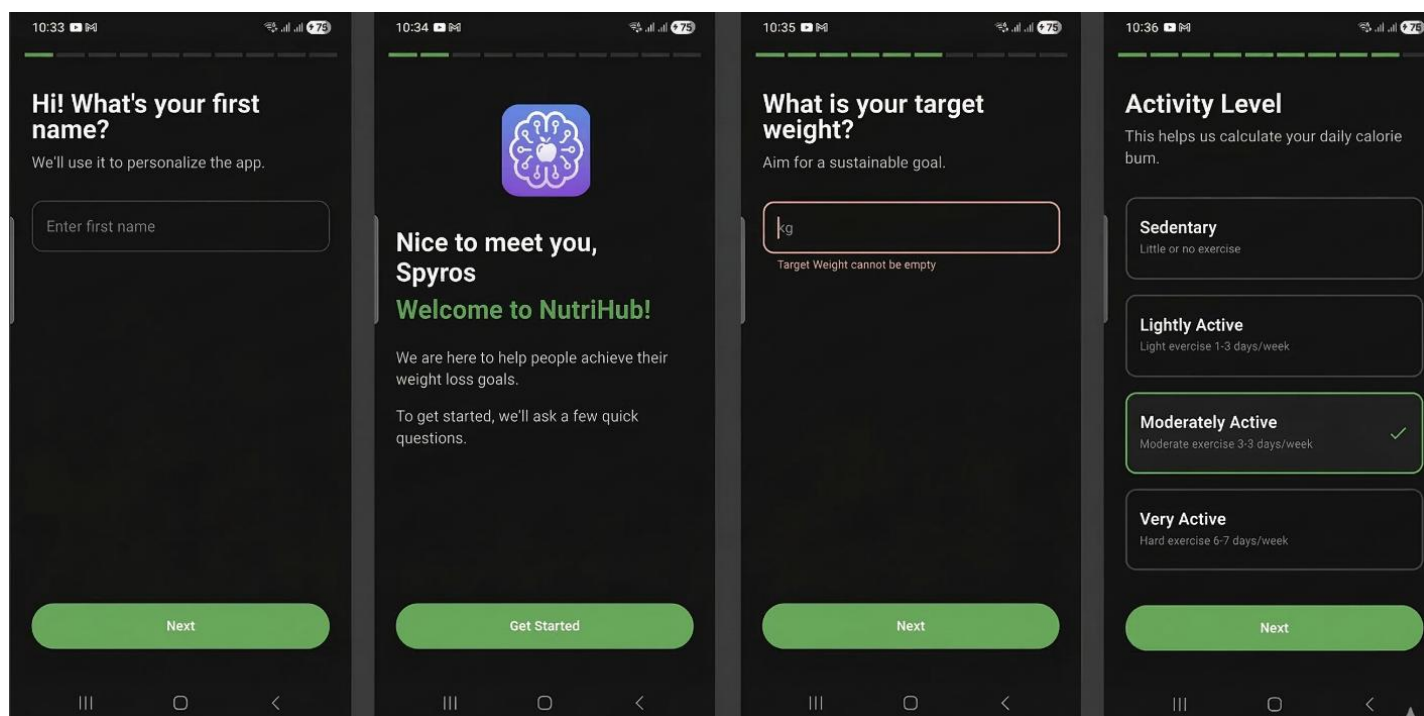
Εικόνα 2. Οθόνη Καλωσορίσματος

Διαδικασία Εγγραφής Νέου Χρήστη:

Εάν δεν διαθέτετε λογαριασμό, η διαδικασία ξεκινά πατώντας το πράσινο κουμπί με την ένδειξη «I am a new user» στην αρχική οθόνη. Αυτή η ενέργεια θα σας μεταφέρει σε μια διαδραστική ροή δημιουργίας προφίλ, η οποία έχει σχεδιαστεί για να συλλέξει τις απαραίτητες πληροφορίες ώστε η εφαρμογή και ο Έξυπνος Διαιτολόγος να προσαρμοστούν ακριβώς στις δικές σας ανάγκες. Στο πάνω μέρος της οθόνης θα παρατηρήσετε μια μπάρα προόδου, η οποία γεμίζει σταδιακά, δείχνοντάς σας σε ποιο βήμα βρίσκεστε.

Αρχικά, η εφαρμογή θα σας ζητήσει το μικρό σας όνομα για να κάνει την εμπειρία σας πιο προσωπική. Πατώντας το κουμπί «Next», θα περάσετε σε μια σειρά από σύντομες ερωτήσεις. Θα κληθείτε να επιλέξετε τον βασικό σας στόχο, δηλαδή εάν επιθυμείτε να χάσετε, να διατηρήσετε ή να αυξήσετε το βάρος σας. Στη συνέχεια, θα πρέπει να συμπληρώσετε το ύψος σας σε εκατοστά, το τρέχον βάρος σας σε κιλά, το βάρος-στόχο που επιθυμείτε να φτάσετε, καθώς και την ηλικία σας. Η εφαρμογή διαθέτει έξυπνους ελέγχους, οπότε σε περίπτωση που εισάγετε κάποια μη ρεαλιστική τιμή, θα εμφανιστεί ένα διακριτικό μήνυμα σφάλματος με κόκκινα γράμματα, ζητώντας σας να διορθώσετε την καταχώρηση πριν προχωρήσετε. Τέλος, θα επιλέξετε το βιολογικό σας φύλο και το επίπεδο της καθημερινής σας δραστηριότητας, από καθιστική ζωή έως πολύ δραστήρια, ώστε να υπολογιστούν με ακρίβεια οι ημερήσιες θερμιδικές σας ανάγκες.

Ενδεικτικά screenshots της διαδικασίας εγγραφής:

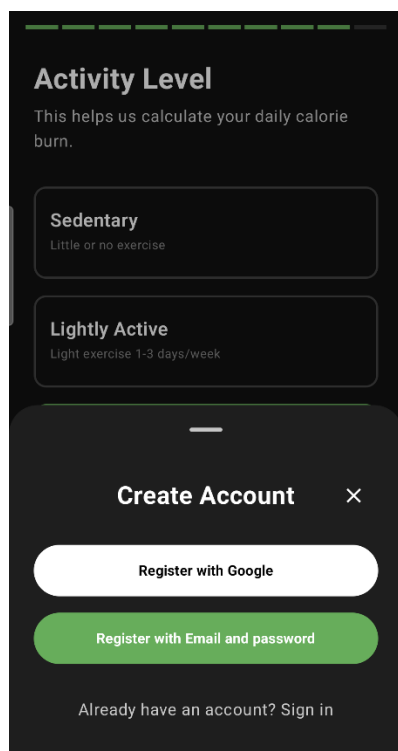


Εικόνα 3. Διαδικασία Εγγραφής

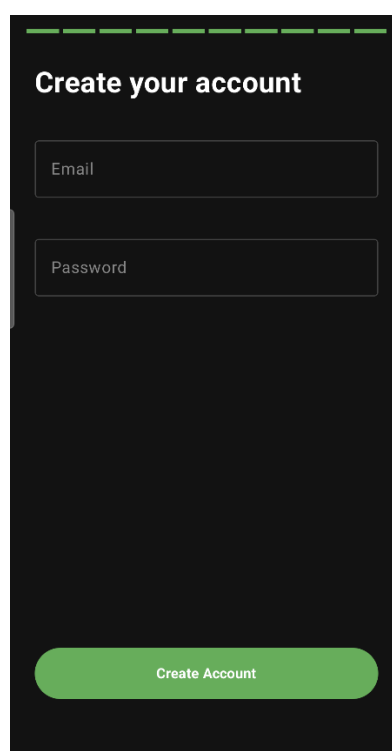
Μόλις ολοκληρώσετε την εισαγωγή των σωματικών σας στοιχείων, ένα αναδυόμενο παράθυρο θα εμφανιστεί από το κάτω μέρος της οθόνης, ζητώντας σας να επιλέξετε τον τρόπο με τον οποίο θέλετε να δημιουργήσετε τον λογαριασμό σας.

Η πρώτη και πιο γρήγορη μέθοδος είναι η Εγγραφή μέσω Google (Register with Google). Επιλέγοντας αυτό το λευκό κουμπί, η συσκευή σας θα εμφανίσει τους διαθέσιμους λογαριασμούς Google που έχετε στο κινητό σας. Με ένα μόνο πάτημα στον λογαριασμό της επιλογής σας, η εγγραφή ολοκληρώνεται αυτόματα και μεταφέρεστε απευθείας στην κεντρική οθόνη της εφαρμογής, χωρίς να χρειαστεί να απομνημονεύσετε νέους κωδικούς πρόσβασης.

Η δεύτερη μέθοδος είναι η κλασική Εγγραφή με Email και Κωδικό (Register with Email and password). Εάν επιλέξετε αυτό το πράσινο κουμπί, το αναδυόμενο παράθυρο θα κλείσει και θα οδηγηθείτε στο τελικό βήμα της οθόνης. Εκεί θα πρέπει να πληκτρολογήσετε μια έγκυρη διεύθυνση email και έναν ασφαλή κωδικό πρόσβασης. Πατώντας «Create Account», τα δεδομένα σας αποστέλλονται με ασφάλεια. Μόλις ο λογαριασμός δημιουργηθεί επιτυχώς, θα εμφανιστεί ένα ενημερωτικό μήνυμα στο κάτω μέρος της οθόνης και το σύστημα θα σας μεταφέρει αυτόματα στην οθόνη Σύνδεσης για να εισέλθετε για πρώτη φορά.



Εικόνα 4. Αναδυόμενο Παράθυρο



Εικόνα 5. Φόρμα Εγγραφής

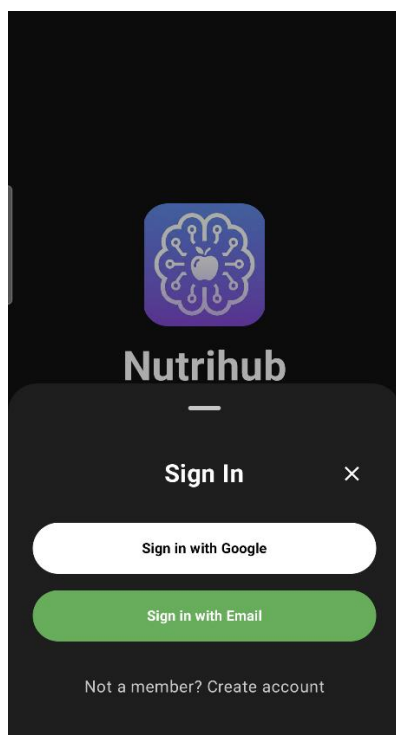
Διαδικασία Σύνδεσης Υπάρχοντος Χρήστη:

Εάν έχετε ήδη δημιουργήσει λογαριασμό, δεν χρειάζεται να περάσετε ξανά από τη ροή ερωτήσεων. Στην αρχική Οθόνη Καλωσορίσματος, αρκεί να πατήσετε το κουμπί «I already have an account». Αμέσως θα αναδυθεί ένα μενού επιλογών από το κάτω μέρος της οθόνης.

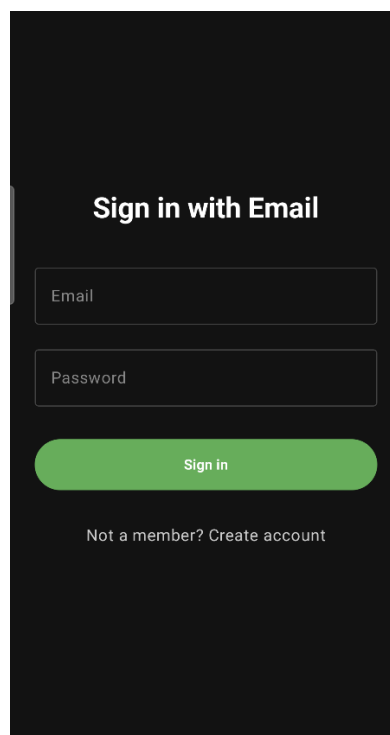
Όπως και στην εγγραφή, προσφέρονται δύο μέθοδοι σύνδεσης. Η πρώτη μέθοδος είναι η Σύνδεση μέσω Google (Sign in with Google). Εάν είχατε δημιουργήσει τον λογαριασμό σας με αυτόν τον τρόπο, πατήστε το αντίστοιχο κουμπί. Η εφαρμογή θα αναγνωρίσει με ασφάλεια τον λογαριασμό σας μέσω της συσκευής σας και θα σας μεταφέρει ακαριαία στην οθόνη ημερήσιας καταγραφής της εφαρμογής.

Η δεύτερη μέθοδος είναι η Σύνδεση με Email (Sign in with Email). Επιλέγοντας το αντίστοιχο κουμπί, το αναδυόμενο παράθυρο κλείνει και μεταφέρεστε στην οθόνη σύνδεσης. Εδώ, καλείστε να συμπληρώσετε το email και τον κωδικό πρόσβασης που δηλώσατε κατά την εγγραφή σας. Κατά την πληκτρολόγηση, αν έχετε κάνει κάποιο λάθος, τα πλαίσια κειμένου θα αλλάξουν χρώμα και θα γίνουν κόκκινα, εμφανίζοντας την αιτία του σφάλματος, π.χ. λάθος κωδικός. Πατώντας το κουμπί «Sign in», θα εμφανιστεί ένας δείκτης φόρτωσης εντός του κουμπιού, υποδεικνύοντας ότι το σύστημα ελέγχει τα στοιχεία σας. Εφόσον τα στοιχεία είναι σωστά, η πρόσβαση εγκρίνεται και μεταβαίνετε στο εσωτερικό της εφαρμογής.

Αξίζει να σημειωθεί πως η εφαρμογή είναι σχεδιασμένη ώστε να συγχωρεί τα λάθη. Εάν ξεκινήσατε κατά λάθος τη διαδικασία Σύνδεσης ενώ θέλατε να κάνετε Εγγραφή ή το αντίστροφο, σε κάθε αναδυόμενο παράθυρο και οθόνη σύνδεσης υπάρχει στο κάτω μέρος ένας διακριτικός σύνδεσμος-κείμενο, ο οποίος σας επιτρέπει να αλλάξετε άμεσα ροή, επιστρέφοντας σας στο σωστό σημείο χωρίς καμία ταλαιπωρία.



Εικόνα 6. Αναδυόμενο Μενού Επιλογών

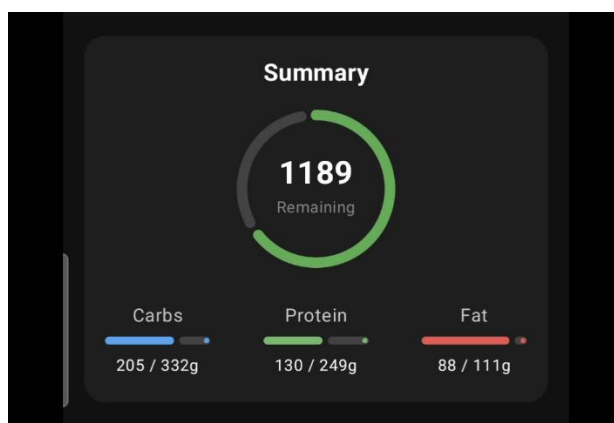


Εικόνα 7. Φόρμα Σύνδεσης

Πίνακας Ελέγχου και Διαχείριση Γευμάτων

Η κεντρική οθόνη της εφαρμογής, γνωστή ως οθόνη ημερήσιας καταγραφής, αποτελεί το βασικό κέντρο λειτουργίας και παρακολούθησης της καθημερινής διατροφής του χρήστη. Με την είσοδο σε αυτή την οθόνη, παρέχεται άμεσα μια πλήρης και οπτικοποιημένη εικόνα της διατροφικής προόδου για την τρέχουσα ημέρα.

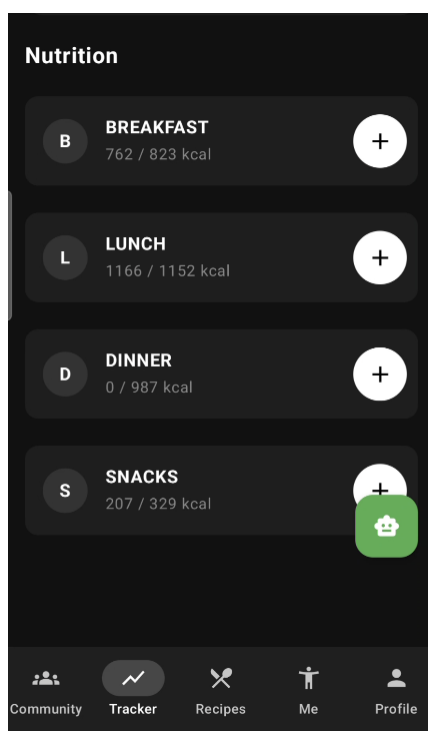
Στο πάνω μέρος της οθόνης βρίσκεται ο πίνακας της ημερήσιας σύνοψης. Στο κέντρο του βρίσκεται ένα δυναμικό κυκλικό γράφημα, το οποίο απεικονίζει τις συνολικές θερμίδες που έχουν καταναλωθεί σε σχέση με τον ημερήσιο στόχο, αναγράφοντας καθαρά τον αριθμό των θερμίδων που απομένουν για την επίτευξή του. Ακριβώς από κάτω, η πληροφορία αναλύεται περαιτέρω μέσω τριών οριζόντιων γραμμικών δεικτών, οι οποίοι αντιστοιχούν στα τρία βασικά μακροθρεπτικά συστατικά, δηλαδή τους υδατάνθρακες, τις πρωτεΐνες και τα λιπαρά. Για κάθε συστατικό, αναγράφονται τα ακριβή γραμμάρια που έχουν καταναλωθεί συγκριτικά με τον εξατομικευμένο στόχο, επιτρέποντας τον άμεσο και λεπτομερή έλεγχο της ποιότητας της διατροφής.



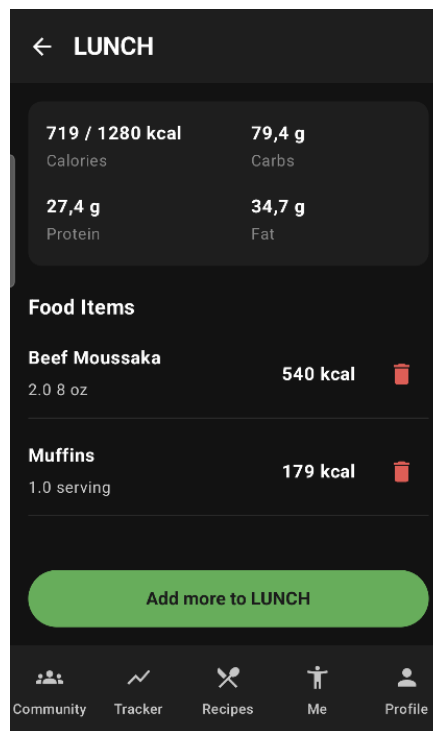
Εικόνα 8. Πίνακας Ημερίσιας Σύνοψης

Κάτω από τη σύνοψη των μακροθρεπτικών, στην οθόνη υπάρχουν οι κατηγορίες των ημερήσιων γευμάτων, οι οποίες κατανέμονται σε Πρωινό, Μεσημεριανό, Βραδινό και Σνακ. Για κάθε γεύμα, το σύστημα εμφανίζει με μια ματιά το σύνολο των θερμίδων που έχουν ήδη καταγραφεί.

Η διαδικασία προσθήκης ενός τροφίμου σε κάποιο γεύμα είναι σχεδιασμένη ώστε να προσφέρει μέγιστη ευελιξία, παρέχοντας πολλαπλούς τρόπους καταγραφής. Ο πρώτος και πιο άμεσος τρόπος είναι μέσα από την ίδια την οθόνη Ημερήσιας Καταγραφής, πατώντας το ευδιάκριτο κουμπί προσθήκης με το σύμβολο του συν που βρίσκεται δίπλα από κάθε κατηγορία γεύματος. Εναλλακτικά, ο χρήστης μπορεί να πατήσει πάνω στην κάρτα ενός συγκεκριμένου γεύματος για να μεταβεί στην αναλυτική προβολή του. Μέσα από αυτή τη λεπτομερή οθόνη, παρέχεται επίσης ένα κεντρικό κουμπί προσθήκης, επιτρέποντας την απρόσκοπτη συνέχιση της καταγραφής χωρίς να απαιτείται επιστροφή στην αρχική οθόνη.



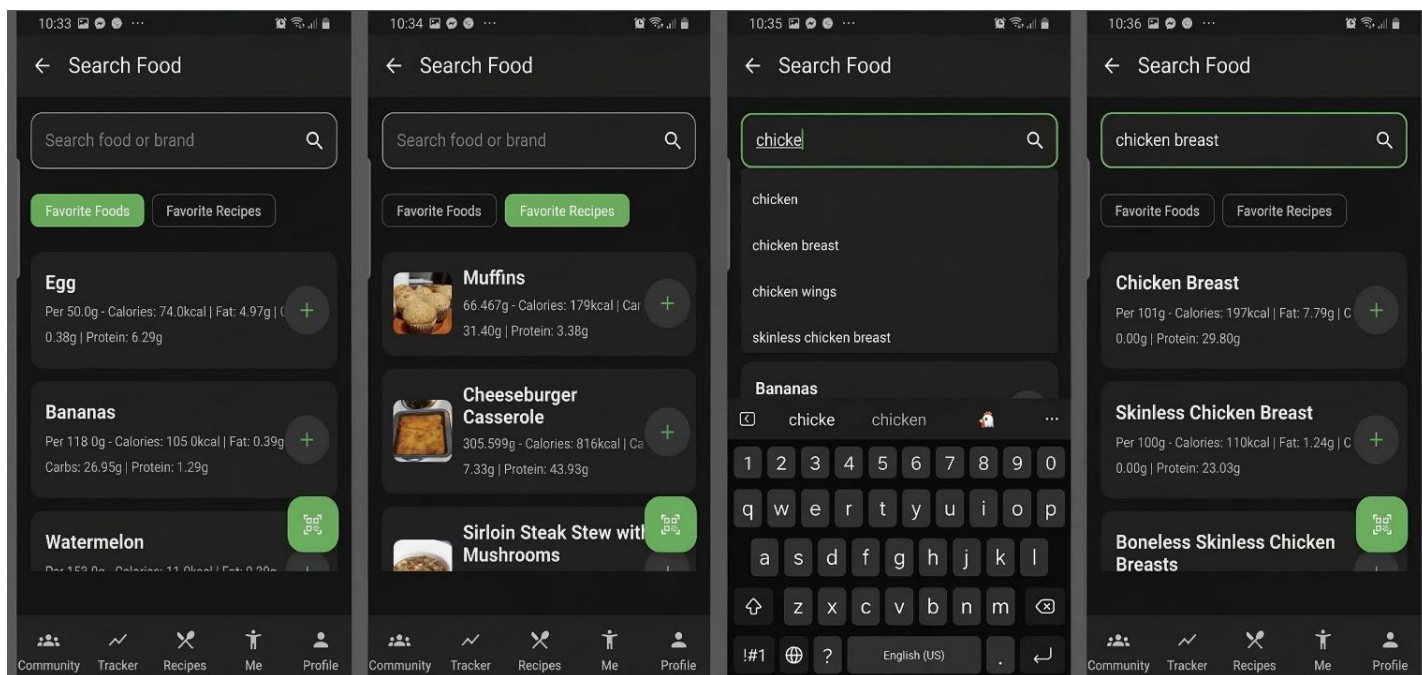
Εικόνα 9. Καρτέλες Γευμάτων



Εικόνα 10. Λεπτομέρειες Γεύματος

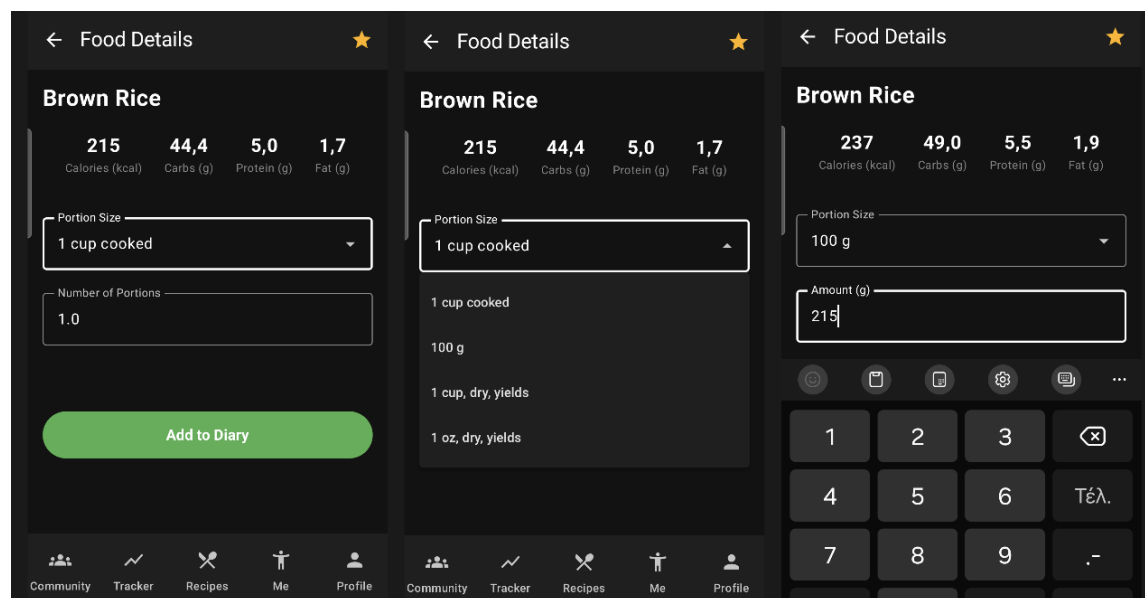
Ανεξάρτητα από τον τρόπο που θα επιλέξει να ξεκινήσει τη διαδικασία προσθήκης, ο χρήστης οδηγείται στην οθόνη αναζήτησης τροφίμων. Εδώ, προσφέρονται τρεις διαφορετικές και πανίσχυρες μέθοδοι εντοπισμού του επιθυμητού είδους. Η πρώτη μέθοδος είναι η κλασική αναζήτηση μέσω της μπάρας κειμένου, όπου ο χρήστης πληκτρολογεί το όνομα του τροφίμου για να αναζητήσει στη βάση δεδομένων. Για διευκόλυνση, η αναζήτηση παρέχει στον χρήστη αυτόματη συμπλήρωση αποτελεσμάτων. Η δεύτερη μέθοδος αφορά την καρτέλα των Αγαπημένων. Σε αυτή την ενότητα είναι συγκεντρωμένα όλα τα μεμονωμένα τρόφιμα, αλλά και οι συνταγές, που ο χρήστης έχει επισημάνει παλαιότερα με το εικονίδιο του αστεριού, επιτρέποντας την ταχύτατη καταγραφή ειδών που καταναλώνονται σε καθημερινή βάση. Και οι δύο αυτοί τρόποι εμφανίζουν μια λίστα αποτελεσμάτων αποτελούμενη από τρόφιμα ή συνταγές αντίστοιχα. Κάθε αποτέλεσμα έχει ευδιάκριτο κουμπί προσθήκης με το σύμβολο του συν που βρίσκεται στα δεξιά του με το οποίο μπορεί να το προσθέσει στο γεύμα του σύμφωνα με τις ποσότητες που αναγράφονται στην καρτέλα.

Ακολουθούν ενδεικτικά screenshots:



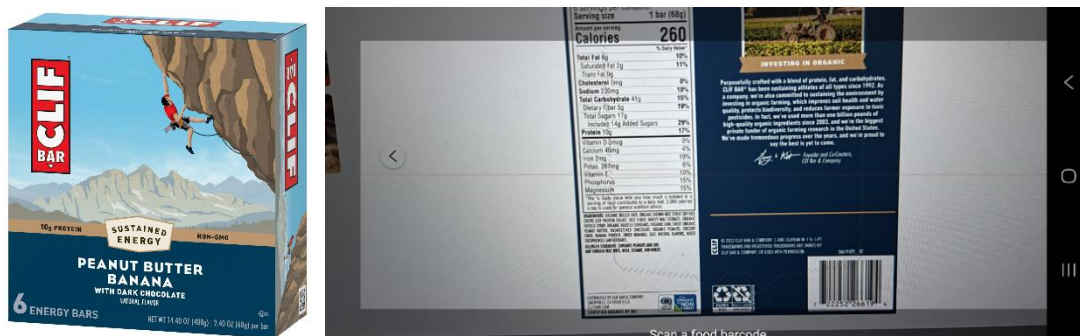
Εικόνα 11. Τρόποι Εύρεσης Τροφίμου

Εναλλακτικά, πατώντας πάνω στο επιθυμητό αποτέλεσμα από την αναζήτηση ή τα αγαπημένα, ανοίγει η οθόνη λεπτομερειών του τροφίμου, η οποία παρουσιάζει την πλήρη διατροφική του ανάλυση. Σε αυτή την οθόνη ενσωματώνεται μια ιδιαίτερα προηγμένη λειτουργία διαχείρισης μερίδων. Ο χρήστης μπορεί να επιλέξει το μέγεθος της μερίδας μέσα από ένα αναδυόμενο μενού. Εάν επιλέξει μια τυπική μερίδα (π.χ. φλιτζάνι ή φέτα), ορίζει απλώς τον αριθμό των μερίδων. Εάν, ωστόσο, το τρόφιμο υποστηρίζει μέτρηση σε γραμμάρια, η εφαρμογή αναγνωρίζει αυτόματα τη συγκεκριμένη μονάδα μέτρησης. Το πεδίο εισαγωγής μεταβάλλεται δυναμικά, ζητώντας από τον χρήστη να εισάγει την ακριβή ποσότητα σε γραμμάρια. Καθώς πληκτρολογεί οποιονδήποτε αριθμό, το σύστημα επανυπολογίζει ακαριαία, σε πραγματικό χρόνο, όλες τις θερμίδες και τα μακροθρεπτικά συστατικά. Επιπλέον, από αυτή την οθόνη μπορεί να προσθέσει ή να αφαιρέσει το τρόφιμο από τα αγαπημένα του. Μόλις οριστεί η ακριβής ποσότητα, η διαδικασία ολοκληρώνεται πατώντας το αντίστοιχο κουμπί εισαγωγής.

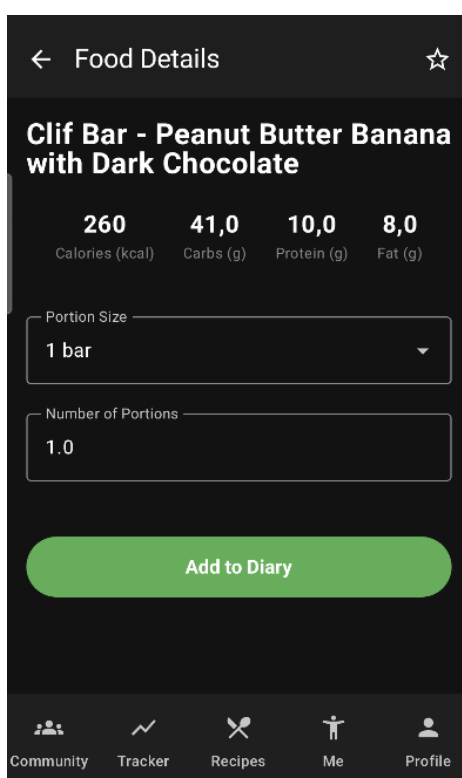


Εικόνα 12. Οθόνη Λεπτομερειών Τροφίμου

Η τρίτη, και εξαιρετικά πρακτική μέθοδος, είναι η χρήση του ενσωματωμένου Barcode Scanner. Σκανάροντας απλώς τη συσκευασία ενός προϊόντος με την κάμερα της κινητής συσκευής, η εφαρμογή αναγνωρίζει και ανακτά αυτόματα τις διατροφικές του πληροφορίες.



Εικόνα 14. Προϊόν - Σκανάρισμα από App



Εικόνα 15. Αποτέλεσμα Σκαναρίσματος

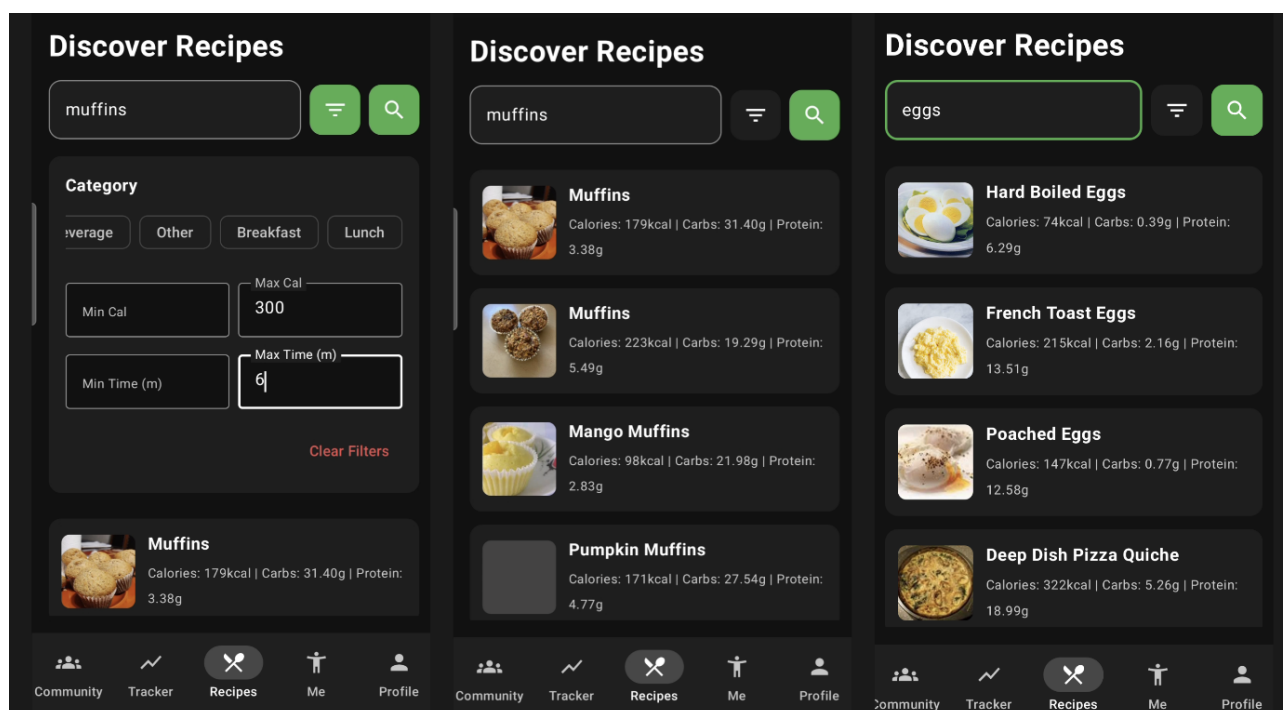
Τέλος, η εφαρμογή παρέχει τη δυνατότητα πλήρους ελέγχου και επεξεργασίας των καταχωρήσεων. Μπαίνοντας στην αναλυτική προβολή του εκάστοτε γεύματος, ο χρήστης βλέπει αναλυτικά, σε μορφή λίστας, όλα τα επιμέρους τρόφιμα που έχει καταναλώσει στο συγκεκριμένο γεύμα. Σε περίπτωση λανθασμένης καταχώρησης ή αλλαγής διατροφικού πλάνου, μπορεί πανεύκολα να αφαιρέσει οποιοδήποτε είδος - Βλέπε εικόνα 9. Αμέσως μετά τη διαγραφή, το σύστημα αναλαμβάνει αυτόματα να αφαιρέσει τις αντίστοιχες θερμίδες και τα γραμμάρια από το ημερήσιο σύνολο, ανανεώνοντας άμεσα τα γραφήματα και τους δείκτες στον Πίνακα Ελέγχου, ώστε να αντικατοπτρίζουν με απόλυτη ακρίβεια την τρέχουσα διατροφική εικόνα.

Αναζήτηση και Διαχείριση Συνταγών

Η ενότητα των συνταγών αποτελεί έναν πλούσιο και δυναμικό κατάλογο γαστρονομικής έμπνευσης, σχεδιασμένο για να βοηθά τους χρήστες να ανακαλύπτουν, να οργανώνουν και να αξιολογούν υγιεινές γευστικές επιλογές. Η πρόσβαση στο συγκεκριμένο περιβάλλον πραγματοποιείται εύκολα, επιλέγοντας το αντίστοιχο εικονίδιο από την μπάρα πλοήγησης στο κάτω μέρος της εφαρμογής.

Κατά την είσοδο στην οθόνη των συνταγών, στο επάνω μέρος της οθόνης βρίσκεται το πεδίο αναζήτησης κειμένου, το οποίο συνοδεύεται από ένα ολοκληρωμένο σύστημα δυναμικών φίλτρων. Ο χρήστης μπορεί να αναζητήσει το ιδανικό πιάτο πληκτρολογώντας απλώς βασικά συστατικά ή το όνομα μιας συγκεκριμένης συνταγής. Ωστόσο, η αναζήτηση δεν περιορίζεται αποκλειστικά στην εισαγωγή ελεύθερου κειμένου. Μέσω των διαθέσιμων φίλτρων, παρέχεται η δυνατότητα εξειδίκευσης των αποτελεσμάτων με βάση συγκεκριμένα διατροφικά κριτήρια και προτιμήσεις. Για παράδειγμα, επιλέγοντας ένα συγκεκριμένο φίλτρο, η λίστα των συνταγών ανανεώνεται αυτόματα, ανασύροντας και παρουσιάζοντας όλες τις διαθέσιμες επιλογές που ταυτίζονται με το συγκεκριμένο κριτήριο. Αυτή η ευελιξία επιτρέπει την άμεση και ξεκούραστη εξερεύνηση νέων γευμάτων που εναρμονίζονται πλήρως με το διατροφικό πλάνο του χρήστη.

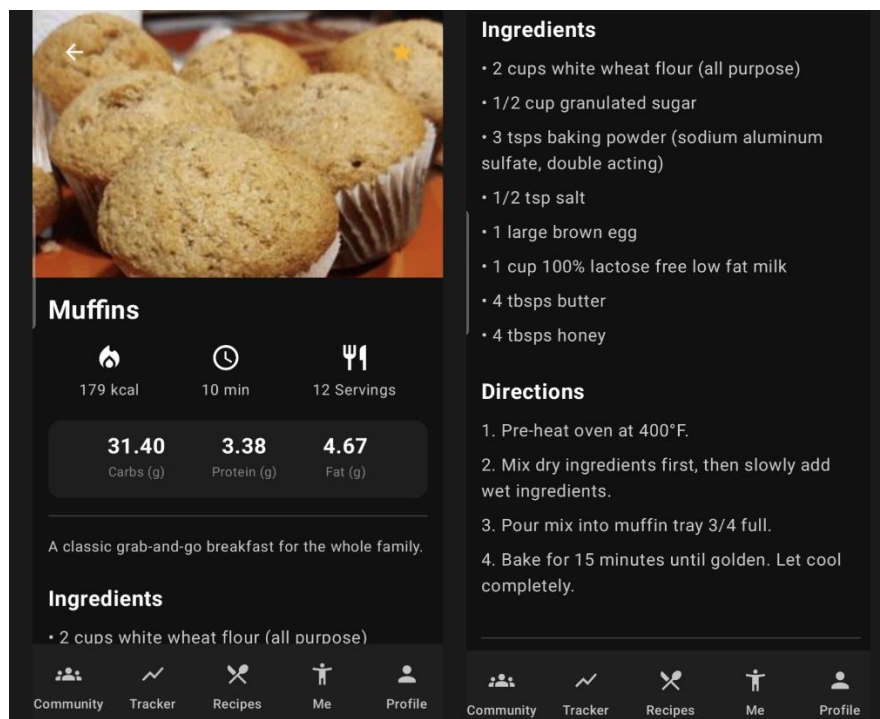
Τα αποτελέσματα της αναζήτησης εμφανίζονται από κάτω σε μια λίστα συνταγών που πληρούν τα κριτήρια αναζήτησης. Κάθε συνταγή παρουσιάζεται σε μια αυτόνομη κάρτα, η οποία περιλαμβάνει μια ελκυστική φωτογραφία του πιάτου, τον τίτλο του και μια γρήγορη σύνοψη των θερμίδων, προσφέροντας την απαραίτητη πληροφορία με μια ματιά.



Εικόνα 16. Αναζήτηση Συνταγών

Επιλέγοντας μια συνταγή από τη λίστα, ανοίγει η οθόνη των αναλυτικών πληροφοριών. Στην κορυφή της οθόνης παρουσιάζεται η ακριβής διατροφική ανάλυση του πιάτου ανά μερίδα, με ξεκάθαρους δείκτες για τις θερμίδες, τους υδατάνθρακες, τις πρωτεΐνες και τα λιπαρά. Επιπλέον, αναγράφεται ο απαιτούμενος χρόνος προετοιμασίας και πόσες μερίδες δύναται να φτιάξει κανείς. Αμέσως μετά, παρατίθεται η πλήρης λίστα των απαιτούμενων συστατικών, ακολουθούμενη από τις αναλυτικές οδηγίες παρασκευής, οι οποίες καθοδηγούν τον χρήστη βήμα προς βήμα στην εκτέλεση της συνταγής.

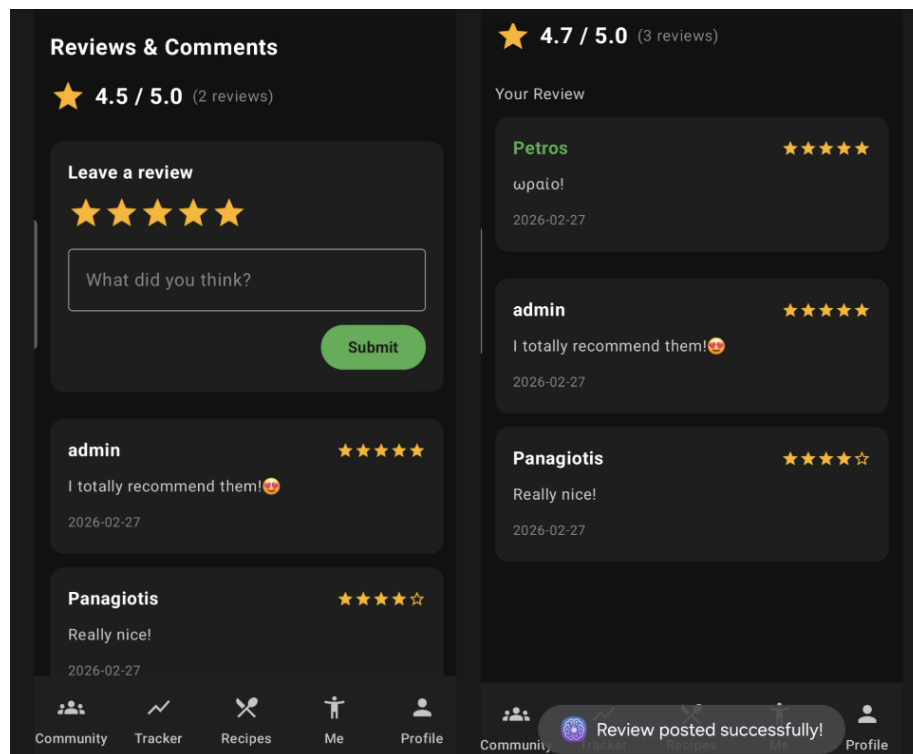
Μια επιπλέον εξαιρετικά χρηστική λειτουργία είναι η διαχείριση των αγαπημένων γευμάτων. Στο πάνω δεξιά μέρος της οθόνης, υπάρχει το ευδιάκριτο εικονίδιο ενός αστεριού. Πατώντας το, το αστέρι γεμίζει με έντονο χρώμα, υποδεικνύοντας ότι η συγκεκριμένη συνταγή έχει αποθηκευτεί επιτυχώς στην προσωπική λίστα αγαπημένων. Αυτή η ενέργεια επιτρέπει την ταχύτατη μελλοντική ανεύρεση των συνταγών που προτιμά ο χρήστης, χωρίς να απαιτείται η επανάληψη της διαδικασίας αναζήτησης, κάνοντας τον προγραμματισμό των γευμάτων πολύ πιο αποδοτικό.



Εικόνα 17. Οθόνη Λεπτομερειών Συνταγής

Η εφαρμογή, βέβαια, δεν περιορίζεται στην απλή παρουσίαση πληροφοριών, αλλά ενσωματώνει έντονα στοιχεία κοινωνικής αλληλεπίδρασης μέσω ενός ολοκληρωμένου συστήματος αξιολογήσεων και σχολίων. Στο κάτω μέρος της αναλυτικής οθόνης, υπάρχει η ενότητα των κριτικών. Εκεί, ο χρήστης μπορεί να δει τη συνολική βαθμολογία του πιάτου, η οποία προκύπτει από τον μέσο όρο των αξιολογήσεων, και να διαβάσει τα σχόλια που έχουν αφήσει άλλα μέλη της κοινότητας.

Επιπλέον, όλοι οι χρήστες έχουν τη δυνατότητα να συνεισφέρουν ενεργά στην πλατφόρμα. Μέσα από ένα ειδικά διαμορφωμένο πεδίο εισαγωγής, ο χρήστης μπορεί να γράψει τη δική του εμπειρία, να προτείνει τυχόν γευστικές παραλλαγές που δοκίμασε και να βαθμολογήσει τη συνταγή, επιλέγοντας από ένα έως πέντε αστέρια. Με το πάτημα του αντίστοιχου κουμπιού υποβολής, η κριτική καταχωρείται άμεσα στο σύστημα και το σχόλιο εμφανίζεται στη δημόσια ροή της συνταγής. Με αυτόν τον τρόπο, δημιουργείται ένα ζωντανό και υποστηρικτικό περιβάλλον γαστρονομικής ανταλλαγής, όπου η κοινότητα αλληλοβοηθείται στην επιλογή των καλύτερων και πιο νόστιμων υγιεινών γευμάτων.



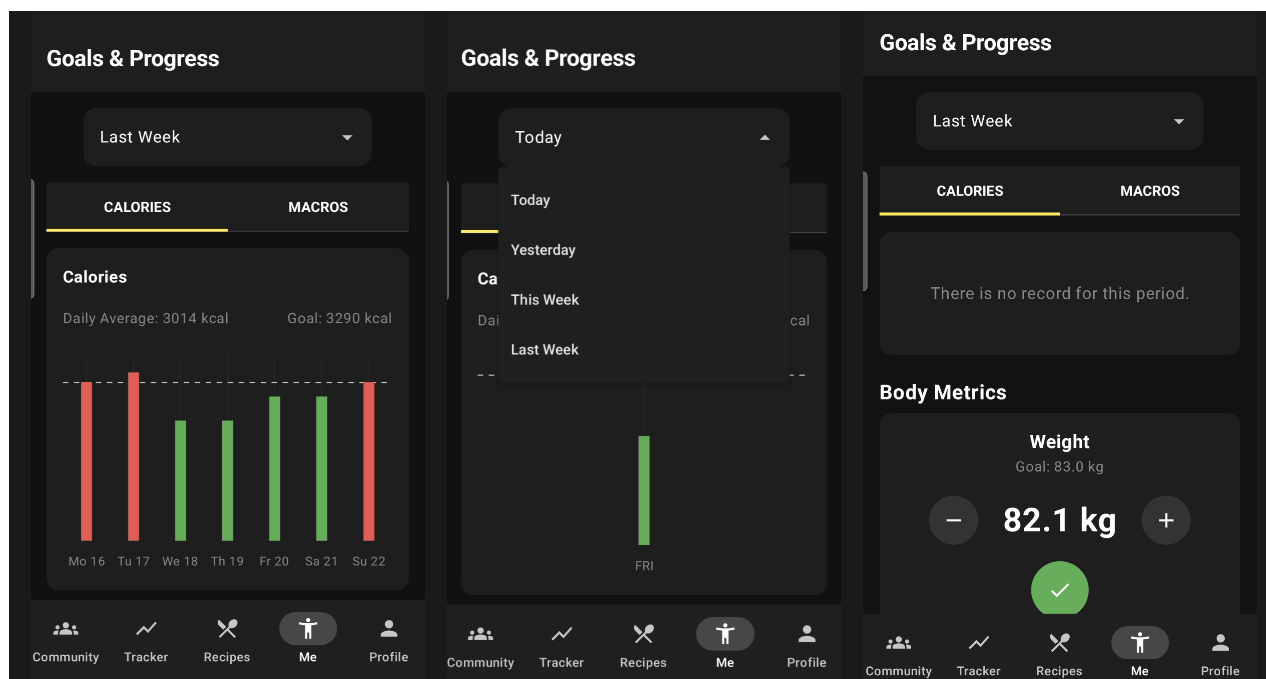
Εικόνα 18. Κριτικές – Υποβολή Κριτικής

Στατιστικά, Πρόοδος και Σωματικές Μετρήσεις

Η ενότητα "Me", η οποία φέρει τον τίτλο "Goals & Progress", αποτελεί το προσωπικό κέντρο ανάλυσης δεδομένων του χρήστη. Η πρόσβαση σε αυτήν επιτυγχάνεται μέσω του αντίστοιχου εικονιδίου στην κάτω μπάρα πλοήγησης. Σε αυτό το περιβάλλον, ο χρήστης αποκτά μια εις βάθος οπτική αναπαράσταση των διατροφικών του συνηθειών και της σωματικής του εξέλιξης σε βάθος χρόνου, έχοντας παράλληλα τη δυνατότητα να καταγράφει το τρέχον βάρος του.

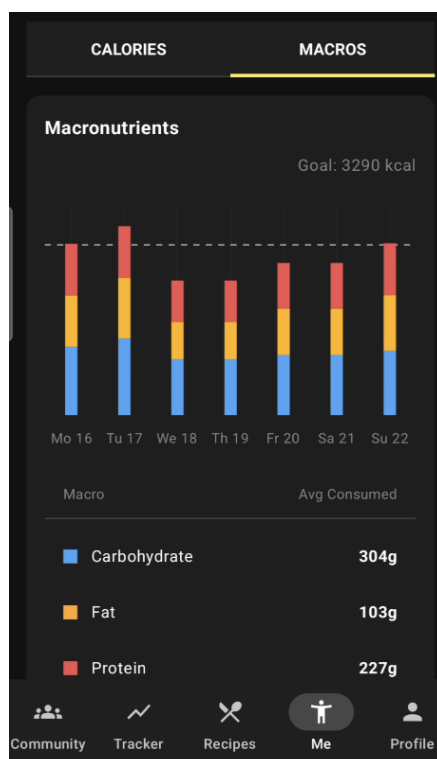
Στην κορυφή της οθόνης, ο χρήστης έχει στη διάθεσή του ένα αναδυόμενο μενού για την επιλογή του χρονικού πλαισίου που επιθυμεί να αναλύσει. Οι διαθέσιμες επιλογές περιλαμβάνουν τη "Σημερινή ημέρα"-Today, τη "Χθεσινή"-Yesterday, την "Τρέχουσα Εβδομάδα"-This Week και την "Προηγούμενη Εβδομάδα"-Last Week. Καθώς ο χρήστης επιλέγει διαφορετική χρονική περίοδο, όλα τα στατιστικά στοιχεία και τα γραφήματα ανανεώνονται δυναμικά. Σε περίπτωση που δεν υπάρχουν επαρκή δεδομένα για το επιλεγμένο διάστημα, η εφαρμογή ενημερώνει διακριτικά τον χρήστη με το μήνυμα "There is no record for this period".

Ακριβώς κάτω από το φίλτρο χρόνου, η ανάλυση της διατροφής χωρίζεται σε δύο κεντρικές καρτέλες: "CALORIES"-Θερμίδες και "MACROS"-Μακροθρεπτικά. Επιλέγοντας την καρτέλα των Θερμίδων, η εφαρμογή υπολογίζει και προβάλλει τον ημερήσιο μέσο όρο κατανάλωσης συγκριτικά με τον ημερήσιο στόχο του χρήστη. Η πληροφορία αυτή οπτικοποιείται μέσω ενός δυναμικού ραβδογράμματος. Στο γράφημα αυτό, μια οριζόντια διακεκομμένη γραμμή υποδεικνύει το όριο του θερμιδικού στόχου. Κάθε ράβδος του διαγράμματος αντιπροσωπεύει μια ημέρα και διαθέτει χρωματική κωδικοποίηση: χρωματίζεται πράσινη εάν οι καταναλωθείσες θερμίδες βρίσκονται κάτω ή ίσες με τον στόχο, και κόκκινη εάν ο χρήστης έχει υπερβεί το όριο.



Εικόνα 19. Γραφήματα Θερμίδων

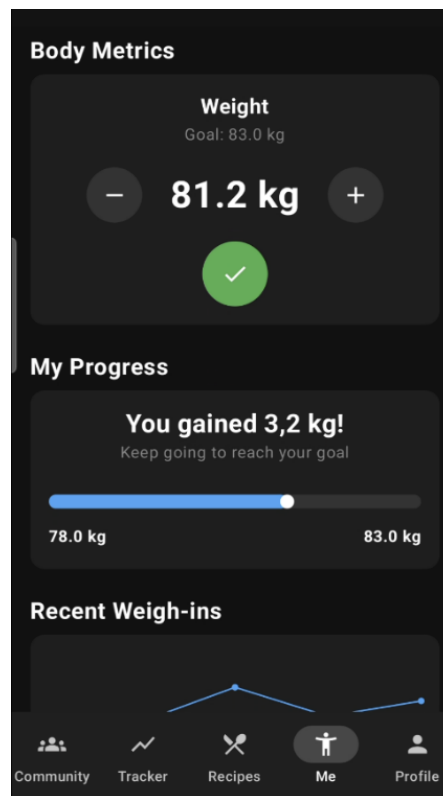
Μεταβαίνοντας στην καρτέλα των Μακροθρεπτικών, η εστίαση περνάει στην ποιότητα της διατροφής. Εδώ, η οθόνη εμφανίζει ένα στοιβαγμένο ραβδόγραμμα. Κάθε ράβδος της επιλεγμένης χρονικής περιόδου διαιρείται οπτικά με διαφορετικά χρώματα, δείχνοντας την ακριβή αναλογία των θερμίδων που προήλθαν από υδατάνθρακες (με μπλε χρώμα), από λιπαρά (με πορτοκαλί χρώμα) και από πρωτεΐνες (με κόκκινο χρώμα). Κάτω από το συγκεκριμένο γράφημα, παρατίθεται ένα αναλυτικό υπόμνημα που υπολογίζει και εμφανίζει στον χρήστη τον ακριβή μέσο όρο γραμμαρίων που καταναλώθηκε για κάθε ένα από τα τρία συστατικά.



Εικόνα 20. Γράφημα Μακροθρεπτικών

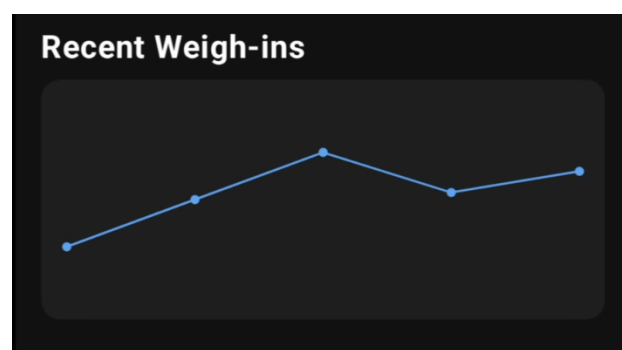
Προχωρώντας χαμηλότερα, συναντάμε την ενότητα "Body Metrics", η οποία είναι αφιερωμένη στην καταγραφή του βάρους. Η εφαρμογή υπενθυμίζει τον τελικό στόχο του χρήστη και παρέχει έναν εξαιρετικά απλό μηχανισμό καταχώρησης. Αντί για πληκτρολόγηση, ο χρήστης βλέπει το τρέχον βάρος του και μπορεί να το αυξήσει ή να το μειώσει ανά 0,1 κιλά, χρησιμοποιώντας τα κουμπιά "Συν" (+) και "Πλην" (-). Με το πάτημα του κεντρικού πράσινου κουμπιού επιβεβαίωσης, το νέο βάρος αποθηκεύεται άμεσα στο σύστημα και συγχρονίζεται με το συνολικό προφίλ.

Η αξιολόγηση αυτής της αλλαγής αποτυπώνεται ακριβώς από κάτω, στην ενότητα "My Progress". Το σύστημα αναλύει το αρχικό βάρος σε σχέση με την τελευταία μέτρηση και τον στόχο, εμφανίζοντας αυτόματα έναν ενθαρρυντικό τίτλο, όπως "You lost X kg!"-Έχασες X κιλά, "You gained X kg!"-Πήρες X κιλά ή "You are maintaining your weight!"-Διατηρείς το βάρος σου. Η απόσταση από την εκκίνηση έως την εκπλήρωση του στόχου απεικονίζεται με μια οριζόντια μπάρα προόδου, η οποία γεμίζει σταδιακά με μπλε χρώμα και φέρει μια λευκή κουκκίδα που υποδεικνύει το ακριβές τρέχον σημείο του χρήστη σε αυτό το ταξίδι.



Εικόνα 21. Μετρήσεις Βάρους

Τέλος, η οθόνη ολοκληρώνεται με την ενότητα "Recent Weigh-ins"-Πρόσφατες Ζυγίσεις. Εδώ, οι τελευταίες πέντε καταγραφές βάρους του χρήστη προβάλλονται σε ένα γράφημα γραμμής. Για να σχεδιαστεί η καμπύλη αυτού του διαγράμματος, απαιτούνται τουλάχιστον δύο καταχωρήσεις. Μέσω αυτής της οπτικοποίησης, ο χρήστης μπορεί να παρακολουθήσει με ευκολία τις αυξομειώσεις του βάρους του στον χρόνο, εντοπίζοντας άμεσα τη γενικότερη τάση της σωματικής του αλλαγής.



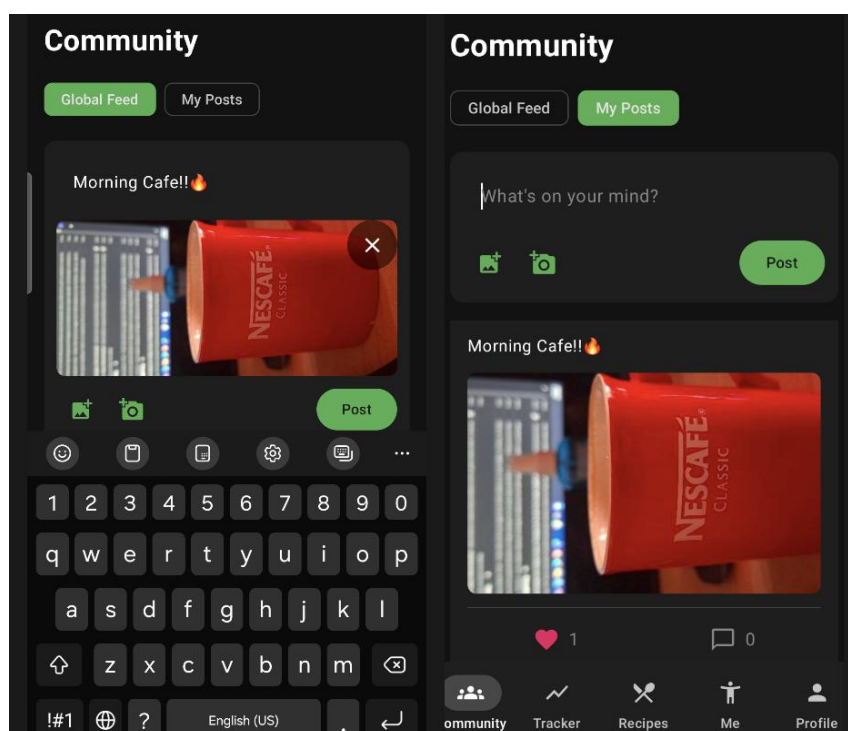
Εικόνα 22. Πρόσφατες Ζυγίσεις

Κοινότητα και Κοινωνική Αλληλεπίδραση

Η ενότητα "Community"-Κοινότητα αποτελεί τον κοινωνικό πυρήνα της εφαρμογής, προσφέροντας έναν ψηφιακό χώρο έκφρασης, αλληλοϋποστήριξης και έμπνευσης για όλους τους χρήστες. Η μετάβαση σε αυτό το περιβάλλον γίνεται εύκολα μέσω της αντίστοιχης επιλογής στην κεντρική κάτω μπάρα πλοήγησης. Κατά την είσοδό του, ο χρήστης αντικρίζει μια δυναμική ροή δημοσιεύσεων, η οποία έχει σχεδιαστεί στα πρότυπα των σύγχρονων μέσων κοινωνικής δικτύωσης.

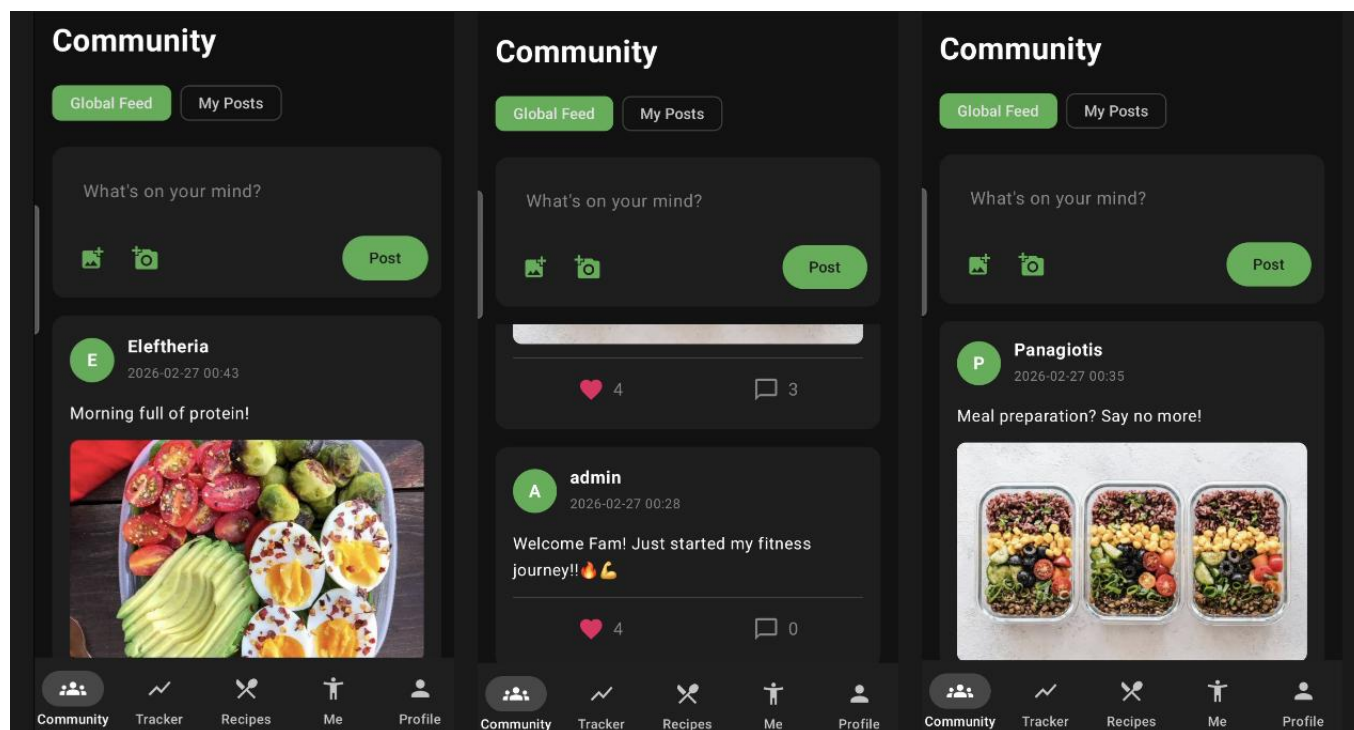
Στην κορυφή της οθόνης, παρέχονται δύο tabs για την οργάνωση του περιεχομένου. Το προεπιλεγμένο tab είναι το "Global Feed"-Παγκόσμια Ροή, όπου εμφανίζονται οι αναρτήσεις από όλα τα μέλη της κοινότητας. Ακριβώς δίπλα, υπάρχει η επιλογή "My Posts"-Οι Αναρτήσεις Μου, η οποία απομονώνει και προβάλλει αποκλειστικά τις δημοσιεύσεις που έχει δημιουργήσει ο ίδιος ο χρήστης, διευκολύνοντας την επισκόπηση του προσωπικού του ιστορικού.

Κάτω από τα tabs, βρίσκεται το πεδίο δημιουργίας νέας ανάρτησης. Ο χρήστης μπορεί να μοιραστεί τις σκέψεις του, την πρόοδό του ή κάποια διατροφική συμβουλή, πληκτρολογώντας κείμενο στο ειδικό πεδίο με την ένδειξη "What's on your mind?". Πέρα από το κείμενο, δίνεται η δυνατότητα εμπλουτισμού της δημοσίευσης με οπτικό υλικό. Μέσω των αντίστοιχων εικονιδίων, ο χρήστης μπορεί είτε να επιλέξει μια υπάρχουσα φωτογραφία από τη συλλογή της συσκευής του, είτε να ενεργοποιήσει την κάμερα για να τραβήξει μια νέα φωτογραφία εκείνη τη στιγμή. Η εφαρμογή ελέγχει και διαχειρίζεται αυτόματα τα απαραίτητα δικαιώματα πρόσβασης στην κάμερα. Εφόσον επιλεγεί κάποια εικόνα, εμφανίζεται άμεσα μια προεπισκόπηση της μέσα στο πεδίο δημιουργίας της ανάρτησης. Εάν ο χρήστης αλλάξει γνώμη, μπορεί να αφαιρέσει την εικόνα πατώντας το εικονίδιο διαγραφής που βρίσκεται πάνω στη μικρογραφία. Με το πάτημα του κουμπιού "Post", τα δεδομένα αποστέλλονται στον διακομιστή και η νέα ανάρτηση εμφανίζεται άμεσα στη ροή.



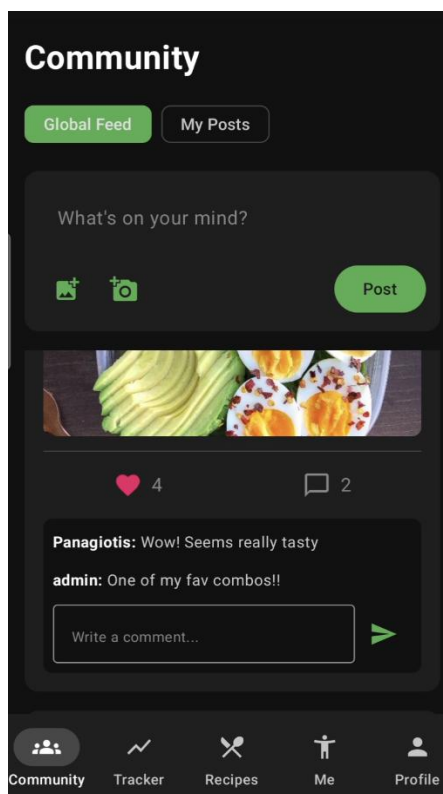
Εικόνα 23. Δημιουργία Ανάρτησης

Κάθε δημοσίευση στη ροή παρουσιάζεται με τη μορφή μιας αυτόνομης κάρτας, η οποία περιλαμβάνει το αρχικό γράμμα του ονόματος του δημιουργού σε ένα χρωματιστό κυκλικό πλαίσιο, το πλήρες όνομά του, την ημερομηνία και ώρα δημιουργίας, το κείμενο και, εφόσον υπάρχει, την επισυναπτόμενη φωτογραφία. Κάτω από το περιεχόμενο της ανάρτησης, παρέχονται ισχυρά εργαλεία αλληλεπίδρασης. Αρχικά, ο χρήστης μπορεί να δηλώσει ότι του αρέσει το περιεχόμενο μιας ανάρτησης, πατώντας την περιοχή με το εικονίδιο της καρδιάς. Η ενέργεια αυτή ενημερώνει τον μετρητή των "Likes" και το εικονίδιο χρωματίζεται ροζ, προσφέροντας άμεση οπτική επιβεβαίωση. Εάν πατηθεί ξανά, το "Like" αναιρείται.



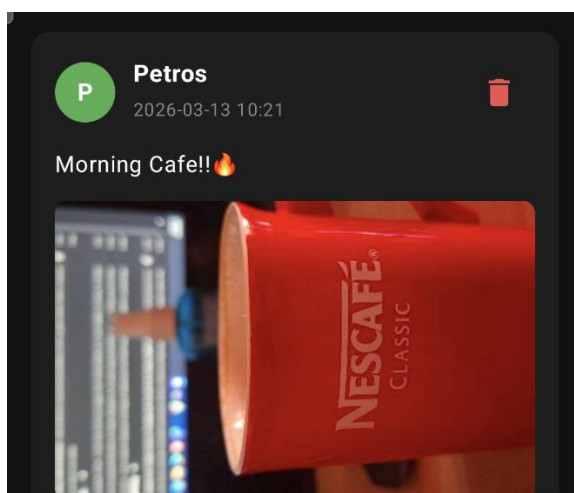
Εικόνα 24. Αναρτήσεις στο feed

Επιπλέον, υποστηρίζεται η ανάπτυξη συζητήσεων κάτω από κάθε ανάρτηση. Δίπλα από το εικονίδιο της καρδιάς, υπάρχει το εικονίδιο των σχολίων μαζί με τον αριθμό των υπάρχοντων απαντήσεων. Πατώντας σε αυτή την περιοχή, αναπτύσσεται το τμήμα των σχολίων της συγκεκριμένης δημοσίευσης. Εκεί, ο χρήστης μπορεί να διαβάσει τι έχουν γράψει τα υπόλοιπα μέλη, βλέποντας το όνομα του κάθε σχολιαστή με έντονα γράμματα και δίπλα το μήνυμά του. Στο κάτω μέρος αυτής της αναπτυσσόμενης ενότητας, υπάρχει ένα πεδίο κειμένου όπου ο χρήστης μπορεί να συντάξει τη δική του απάντηση και να την υποβάλει πατώντας το πράσινο εικονίδιο αποστολής.



Εικόνα 25. Σχόλια σε Post

Τέλος, η εφαρμογή διασφαλίζει τον πλήρη έλεγχο του χρήστη επί του δικού του περιεχομένου. Όταν ο χρήστης περιηγείται στη ροή και εντοπίζει μια δική του δημοσίευση, παρατηρεί ότι στην επάνω δεξιά γωνία της κάρτας εμφανίζεται ένα εικονίδιο κάδου απορριμμάτων. Αυτή η επιλογή, η οποία είναι αυστηρά ορατή μόνο στον δημιουργό της εκάστοτε ανάρτησης, επιτρέπει την οριστική διαγραφή της δημοσίευσης από την πλατφόρμα. Καθ' όλη τη διάρκεια αυτών των ενεργειών, το σύστημα ενημερώνει τον χρήστη για την έκβασή τους μέσω σύντομων αναδυόμενων μηνυμάτων στο κάτω μέρος της οθόνης.

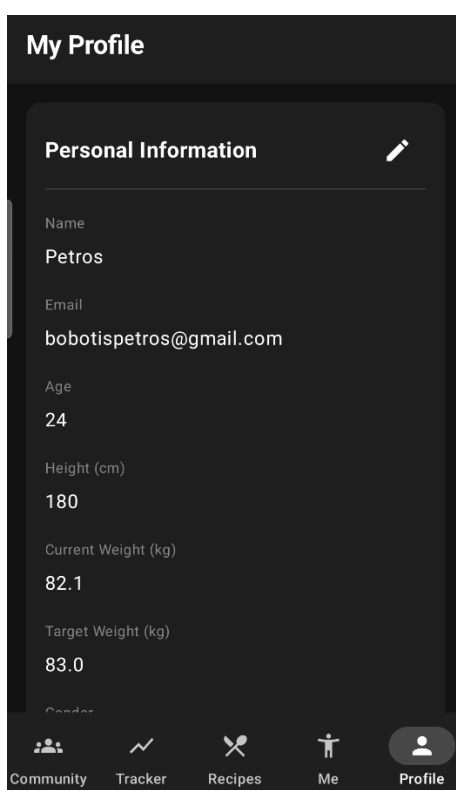


Εικόνα 26. Δυνατότητα Διαγραφής Post

Διαχείριση Προσωπικού Προφίλ

Η ενότητα του Προφίλ-Profile αποτελεί τον προσωπικό χώρο διαχείρισης του λογαριασμού και των βιομετρικών στοιχείων του χρήστη. Η πρόσβαση στην οθόνη αυτή πραγματοποιείται επιλέγοντας το τελευταίο εικονίδιο στην κεντρική κάτω μπάρα πλοήγησης. Μέσω αυτού του περιβάλλοντος, ο χρήστης έχει τον πλήρη έλεγχο των δεδομένων που καθορίζουν την εμπειρία του και τους υπολογισμούς των διατροφικών του στόχων στο NutriHub.

Μπαίνοντας στην οθόνη, ο χρήστης αντικρίζει μια κεντρική κάρτα με τον τίτλο "Personal Information"-Προσωπικές Πληροφορίες. Σε αυτή την προβολή, παρουσιάζονται οργανωμένα και με σαφήνεια όλα τα στοιχεία που είχε δηλώσει κατά την αρχική του εγγραφή ή στις μετέπειτα ενημερώσεις του. Πιο συγκεκριμένα, αναγράφονται το όνομα, η διεύθυνση ηλεκτρονικού ταχυδρομείου, η ηλικία, το ύψος σε εκατοστά, το τρέχον βάρος και το βάρος-στόχος σε κιλά, καθώς και το φύλο, ο βασικός διατροφικός στόχος και το επίπεδο της καθημερινής φυσικής δραστηριότητας. Στην αρχική της κατάσταση, η οθόνη είναι μόνο για ανάγνωση, προστατεύοντας τον χρήστη από τυχαίες αλλαγές των δεδομένων του.



Εικόνα 27. Προβολή Πληροφοριών Προφίλ

Εάν ο χρήστης επιθυμεί να τροποποιήσει οποιοδήποτε από τα παραπάνω στοιχεία, μπορεί να ενεργοποιήσει τη λειτουργία επεξεργασίας πατώντας το εικονίδιο με το μολύβι που βρίσκεται στην επάνω δεξιά γωνία της κάρτας. Αμέσως, η οθόνη μετασχηματίζεται και τα απλά κείμενα δίνουν τη θέση τους σε διαδραστικά πεδία εισαγωγής κειμένου. Για τα αριθμητικά και τα γενικά στοιχεία, όπως το όνομα, η ηλικία και τα κιλά, ο χρήστης μπορεί να πληκτρολογήσει ελεύθερα τις νέες τιμές. Για δεδομένα όμως που απαιτούν συγκεκριμένες επιλογές, η εφαρμογή προσφέρει αναδυόμενα μενού-drodowns. Συγκεκριμένα, για το φύλο, τον στόχο και το επίπεδο δραστηριότητας, ο χρήστης καλείται να επιλέξει από μια προκαθορισμένη λίστα ασφαλών επιλογών, αποτρέποντας έτσι τα τυπογραφικά λάθη.

Personal Information ✓

Name: Petros

Email: bobotispetros@gmail.com

Age: 24

Height (cm):

Target Weight (kg): 83.0

Gender: MALE

Goal: GAIN WEIGHT

Activity Level: VERY ACTIVE

SEDENTARY

LIGHTLY ACTIVE

MODERATELY ACTIVE

VERY ACTIVE

EXTRA ACTIVE

ommunity tracker recipes me rprofile

Εικόνα 28. Επεξεργασία Προφίλ

Όταν ο χρήστης ολοκληρώσει τις αλλαγές του, το εικονίδιο πάνω δεξιά έχει πλέον αλλάξει μορφή και απεικονίζει ένα σημάδι επιβεβαίωσης-Check. Πατώντας το, το σύστημα δεν αποθηκεύει τα δεδομένα τυφλά, αλλά πραγματοποιεί αυστηρούς ελέγχους εγκυρότητας στο παρασκήνιο. Εάν ο χρήστης έχει εισάγει κάποια μη έγκυρη τιμή, η διαδικασία αποθήκευσης διακόπτεται και εμφανίζεται ένα ενημερωτικό μήνυμα στο κάτω μέρος της οθόνης, υποδεικνύοντας το ακριβές σφάλμα. Με την επιτυχή ολοκλήρωση της ενημέρωσης, η λειτουργία επεξεργασίας απενεργοποιείται αυτόματα, τα πεδία κλειδώνουν ξανά και ο χρήστης ενημερώνεται με ένα μήνυμα επιτυχούς αποθήκευσης.

My Profile

Personal Information ✓

Name: Petros

Email: bobotispetros@gmail.com

Age: 150

Height (cm): 180

Current Weight (kg):

Target Weight (kg): 83.0

Please enter a realistic age (5 - 120).

Community tracker recipes me Profile

My Profile

Personal Information ✓

Name: Petros

Email: bobotispetros@gmail.com

Age: 24

Height (cm): 180

Current Weight (kg): 82.1

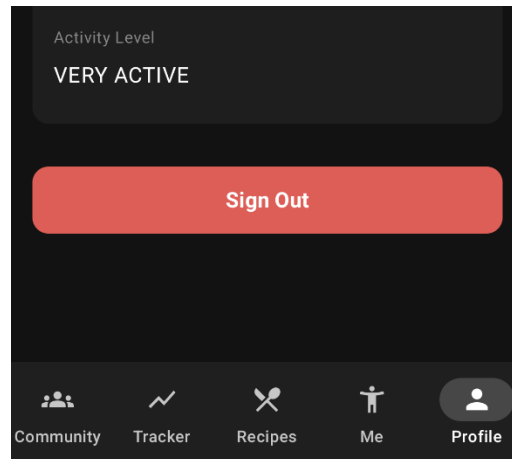
Target Weight (kg): 83.0

Profile updated successfully!

Community tracker recipes me Profile

Εικόνα 29. Ενημερωτικά Μηνύματα

Τέλος, στο κάτω μέρος της οθόνης, ακριβώς κάτω από την κάρτα των προσωπικών πληροφοριών, δεσπόζει ένα ευδιάκριτο κόκκινο κουμπί με την ένδειξη "Sign Out"-Αποσύνδεση. Πατώντας αυτό το κουμπί, η εφαρμογή διαγράφει με ασφάλεια τα διαπιστευτήρια σύνδεσης της τρέχουσας συνεδρίας από τη μνήμη της συσκευής και μεταφέρει αυτόματα τον χρήστη πίσω στην αρχική Οθόνη Καλωσορίσματος, προστατεύοντας έτσι τον λογαριασμό του από μη εξουσιοδοτημένη πρόσβαση.

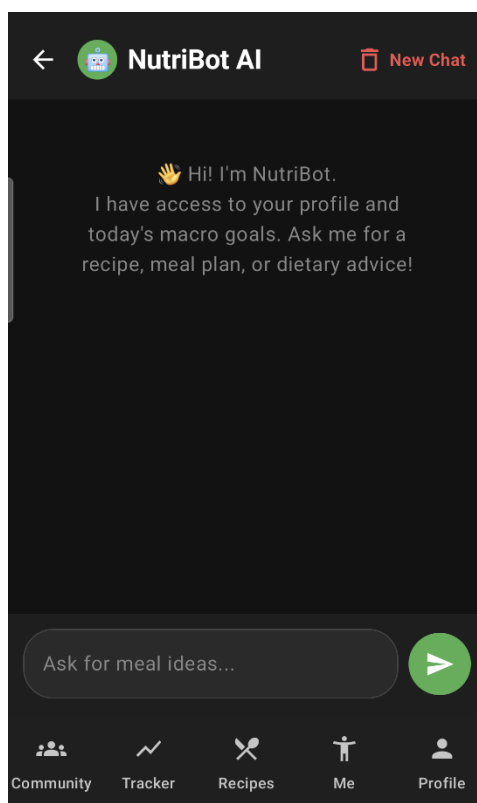


Εικόνα 30. Κουμπί Αποσύνδεσης

Προσωπικός Τεχνητός Διαιτολόγος

Η εφαρμογή προσφέρει μια πρωτοποριακή εμπειρία καθοδήγησης μέσω του ενσωματωμένου Τεχνητού Διαιτολόγου, ο οποίος φέρει την ονομασία "NutriBot AI". Η πρόσβαση σε αυτόν τον έξυπνο βοηθό είναι άμεση και γίνεται μέσα από την κεντρική οθόνη ημερήσιας καταγραφής, πατώντας το χαρακτηριστικό πράσινο αιωρούμενο κουμπί στο κάτω δεξί μέρος της οθόνης, το οποίο φέρει το εικονίδιο ενός ρομπότ.

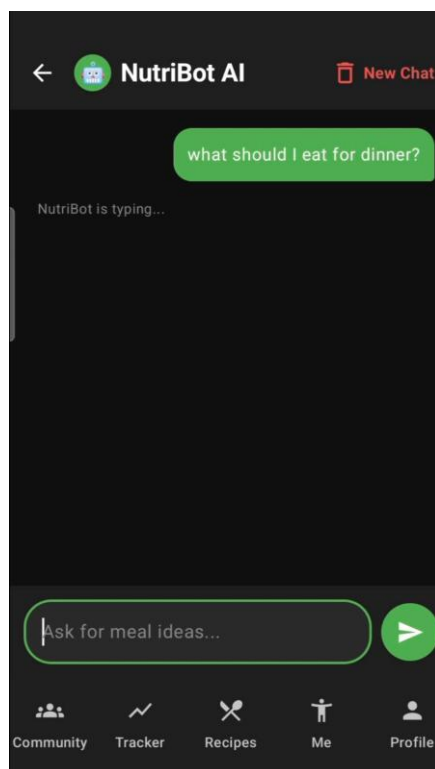
Με την είσοδο στην οθόνη συνομιλίας, εάν ο χρήστης δεν έχει προηγούμενο ιστορικό μηνυμάτων, υποδέχεται ένα κεντρικό εισαγωγικό μήνυμα. Το μήνυμα αυτό ενημερώνει τον χρήστη για τις δυνατότητες του NutriBot, τονίζοντας το σημαντικότερο πλεονέκτημά του, πως το σύστημα έχει πλήρη και ζωντανή πρόσβαση στο προσωπικό προφίλ του χρήστη, καθώς και στους ημερήσιους στόχους των μακροθρεπτικών του συστατικών.



Εικόνα 31. Άνοιγμα Chat με Nutribot AI

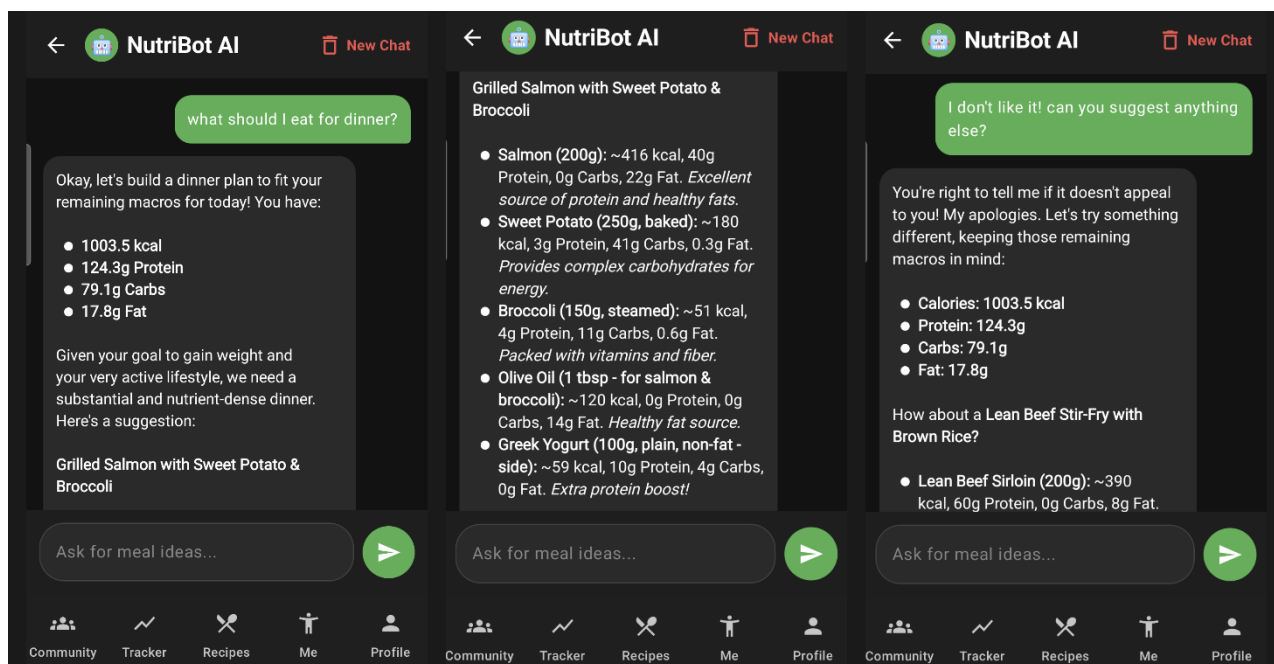
Η αλληλεπίδραση πραγματοποιείται μέσα από ένα σύγχρονο και οικείο περιβάλλον ανταλλαγής μηνυμάτων. Στο κάτω μέρος της οθόνης, υπάρχει ένα πεδίο εισαγωγής κειμένου όπου ο χρήστης μπορεί να πληκτρολογήσει το ερώτημά του. Δίπλα ακριβώς, βρίσκεται το κουμπί αποστολής. Για την αποφυγή λανθασμένων ή πολλαπλών ταυτόχρονων αποστολών, το σύστημα έχει ρυθμιστεί ώστε το κουμπί αποστολής να παραμένει ανενεργό εάν το πεδίο κειμένου είναι κενό ή εάν ο Τεχνητός Διαιτολόγος βρίσκεται ήδη στη διαδικασία παραγωγής μιας απάντησης.

Μόλις ο χρήστης αποστείλει το μήνυμά του, αυτό εμφανίζεται άμεσα στη ροή της συζήτησης σε ένα πράσινο πλαίσιο στα δεξιά της οθόνης. Ταυτόχρονα, εμφανίζεται μια διακριτική ένδειξη "NutriBot is typing..." (Ο NutriBot πληκτρολογεί...), η οποία πληροφορεί τον χρήστη ότι το αίτημά του επεξεργάζεται εκείνη τη στιγμή. Όταν η απάντηση του αϊ Διαιτολόγου είναι έτοιμη, προβάλλεται σε ένα σκούρο γκρι πλαίσιο στα αριστερά της οθόνης. Παράλληλα, η λίστα των μηνυμάτων κυλάει αυτόματα προς τα κάτω με κάθε νέο μήνυμα, διασφαλίζοντας ότι ο χρήστης βλέπει πάντα την πιο πρόσφατη πληροφορία χωρίς να χρειάζεται να μετακινήσει την οθόνη χειροκίνητα.



Εικόνα 32. Ο Ψηφιακός Διαιτολόγος "σκέφτεται"

Τέλος, η εφαρμογή προσφέρει πλήρη διαχείριση του ιστορικού των συνομιλιών. Κάθε φορά που ο χρήστης αποχωρεί και επιστρέφει στην οθόνη του AI, το ιστορικό των προηγούμενων μηνυμάτων του φορτώνεται αυτόματα, επιτρέποντάς του να ανατρέξει σε παλαιότερες συμβουλές που έλαβε. Εάν, ωστόσο, επιθυμεί να ξεκινήσει μια εντελώς νέα συζήτηση, αλλάζοντας ενδεχομένως θέμα, έχει στη διάθεσή του το κουμπί "New Chat"-Νέα Συνομιλία με το εικονίδιο του κάδου απορριμμάτων, το οποίο βρίσκεται στην επάνω δεξιά γωνία της οθόνης. Πατώντας το, όλο το ιστορικό της συνομιλίας διαγράφεται οριστικά, προσφέροντας έναν καθαρό chat για νέες διατροφικές αναζητήσεις.



Εικόνα 33. Ο Ψηφιακός Διαιτολόγος "σκέφτεται"

ΜΕΛΛΟΝΤΙΚΕΣ ΕΠΕΚΤΑΣΕΙΣ ΤΗΣ ΕΦΑΡΜΟΓΗΣ

Η τρέχουσα έκδοση της εφαρμογής παρέχει μια εξαιρετικά ισχυρή και ολοκληρωμένη λειτουργικότητα για την παρακολούθηση της διατροφής, τη διαχείριση εξατομικευμένων μακροθρεπτικών στόχων, την αλληλεπίδραση με τον Τεχνητό Διαιτολόγο και την ενεργή συμμετοχή σε μια κοινότητα υγιεινής διαβίωσης. Ωστόσο, καθώς ο τομέας της ψηφιακής υγείας εξελίσσεται ραγδαία, έχουν προγραμματιστεί αρκετές καινοτόμες επεκτάσεις για το μέλλον, με στόχο τη συνεχή βελτίωση της εμπειρίας του χρήστη και τη διεύρυνση των τεχνολογικών δυνατοτήτων του συστήματος.

Ένας από τους πρωταρχικούς τομείς μελλοντικής ανάπτυξης είναι η αναβάθμιση του ενσωματωμένου Τεχνητού Διαιτολόγου με δυνατότητες Μηχανικής Όρασης. Οι χρήστες θα μπορούν να βγάλουν φωτογραφία το γεύμα τους και η Τεχνητή Νοημοσύνη θα αναγνωρίζει αυτόματα τα τρόφιμα, εκτιμώντας το μέγεθος της μερίδας και υπολογίζοντας τις θερμίδες και τα μακροθρεπτικά συστατικά, καταχωρώντας τα απευθείας στο ημερολόγιο. Αντίστοιχα, φωτογραφίζοντας το εσωτερικό του ψυγείου τους, το σύστημα θα μπορεί να προτείνει υγιεινές συνταγές αξιοποιώντας αποκλειστικά τα διαθέσιμα υλικά, ελαχιστοποιώντας έτσι και τη σπατάλη τροφίμων.

Προκειμένου να παρέχεται μια πραγματικά ολιστική εικόνα της υγείας, η εφαρμογή θα διασυνδεθεί με δημοφιλή οικοσυστήματα υγείας, όπως το Google Health Connect, και με φορητές συσκευές. Αυτή η απρόσκοπτη ενσωμάτωση θα επιτρέπει στο σύστημα να αντλεί δεδομένα για τα ημερήσια βήματα, τις προπονήσεις και τις ενεργές θερμίδες που καίει ο χρήστης. Βάσει αυτών των δεδομένων, η εφαρμογή θα μπορεί να αναπροσαρμόζει δυναμικά και σε πραγματικό χρόνο το ημερήσιο όριο θερμίδων και υδατανθράκων, προσφέροντας απόλυτη εξατομίκευση σύμφωνα με το ενεργειακό ισοζύγιο της εκάστοτε ημέρας.

Αναγνωρίζοντας ότι η σωστή διατροφή δεν περιορίζεται μόνο στις θερμίδες, στους υδατάνθρακες, στις πρωτεΐνες και στα λιπαρά, μια μελλοντική επέκταση θα περιλαμβάνει την αυστηρή παρακολούθηση των μικροθρεπτικών συστατικών. Οι χρήστες θα μπορούν να ελέγχουν την πρόσληψη βιταμινών, μετάλλων, ιχνοστοιχείων, φυτικών ινών και νατρίου, διασφαλίζοντας την ποιότητα της διατροφής τους. Παράλληλα, θα ενσωματωθεί ένας ειδικός μηχανισμός παρακολούθησης της ημερήσιας ενυδάτωσης, καθώς η κατανάλωση νερού αποτελεί θεμελιώδη παράγοντα τόσο για την απώλεια βάρους όσο και για τη γενικότερη ευεξία.

Για την αύξηση της δέσμευσης των χρηστών και τη διατήρηση του κινήτρου τους σε υψηλά επίπεδα, η πλατφόρμα θα ενσωματώσει στοιχεία gamification. Η εισαγωγή ημερήσιων σερί καταγραφής, η απονομή ψηφιακών εμβλημάτων επιβράβευσης για την επίτευξη εβδομαδιαίων στόχων, καθώς και η δυνατότητα δημιουργίας διατροφικών προκλήσεων μέσα στην Κοινότητα, θα μετατρέψουν τη διαδικασία της διατροφικής παρακολούθησης σε μια πιο διασκεδαστική και διαδραστική εμπειρία.

Με την ενσωμάτωση αυτών των στρατηγικών επεκτάσεων, η εφαρμογή θα συνεχίσει να εξελίσσεται από ένα απλό εργαλείο καταγραφής σε έναν ολοκληρωμένο, έξυπνο ψηφιακό σύμβουλο υγείας. Αυτές οι βελτιώσεις όχι μόνο θα καλύψουν τις ολοένα αυξανόμενες απαιτήσεις της τρέχουσας βάσης χρηστών, αλλά θα καταστήσουν την εφαρμογή κορυφαία επιλογή στην ανταγωνιστική αγορά των εφαρμογών υγείας και ευεξίας.

Αξιολόγηση Χρηστών

Μεθοδολογία και Σχεδιασμός Έρευνας

Προκειμένου να αποτιμηθεί η χρηστικότητα, η λειτουργικότητα και η γενικότερη αξία της εφαρμογής **NutriHub**, σχεδιάστηκε και πραγματοποιήθηκε μια ποσοτική και ποιοτική πειραματική αξιολόγηση με τη συμμετοχή πραγματικών χρηστών. Η διανομή της εφαρμογής έγινε μέσω της δημιουργίας κατάλληλου APK αρχείου και της χρήσης της πλατφόρμας Ngrok, επιτρέποντας στους απομακρυσμένους χρήστες να αλληλοεπιδράσουν σε πραγματικό χρόνο με την τοπική σχεσιακή βάση δεδομένων PostgreSQL.

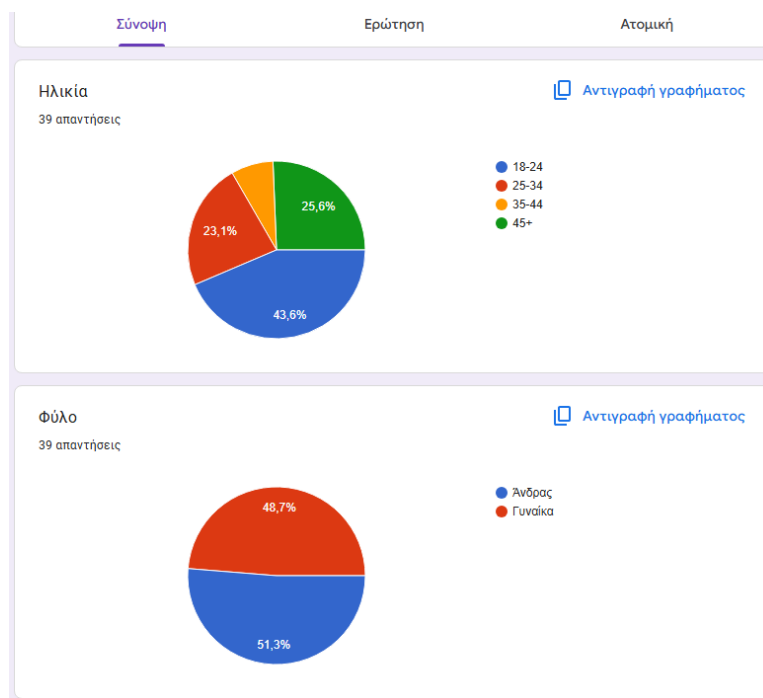
Η συλλογή των δεδομένων της αξιολόγησης πραγματοποιήθηκε μέσω της πλατφόρμας Google Forms. Το ερωτηματολόγιο δομήθηκε με γνώμονα τα διεθνή πρότυπα αποτίμησης συστημάτων λογισμικού, χρησιμοποιώντας κυρίως την πενταβάθμια κλίμακα Likert (όπου 1 = Διαφωνώ Απόλυτα και 5 = Συμφωνώ Απόλυτα). Οι ερωτήσεις κατανεμήθηκαν σε διακριτές θεματικές ενότητες με σκοπό την αξιολόγηση:

- Του interface και της εμπειρίας χρήστη (UI/UX),
- Των βασικών μηχανισμών καταγραφής και ανάκτησης δεδομένων,
- Της ευφυσής καθοδήγησης και των NLP αποκρίσεων του προσωπικού βοηθού,
- Της κοινωνικής αλληλεπίδρασης.

Η συμμετοχή στην έρευνα ήταν εθελοντική και απολύτως ανώνυμη, διασφαλίζοντας την εγκυρότητα και την αντικειμενικότητα των αποκρίσεων.

Δημογραφικά Στοιχεία και Εξοικείωση Δείγματος

Το δείγμα της έρευνας αποτέλεσαν χρήστες διαφόρων ηλικιακών ομάδων, με σκοπό την αποτύπωση της συμπεριφοράς τόσο των τεχνολογικά εξοικειωμένων ατόμων όσο και χρηστών με λιγότερες ψηφιακές δεξιότητες. Παράλληλα, εξετάστηκε το υπόβαθρο των συμμετεχόντων αναφορικά με τη χρήση παρόμοιων εμπορικών εφαρμογών όπως MyFitnessPal, Yazio κ.λπ., ώστε να αξιολογηθεί η καμπύλη εκμάθησης του NutriHub συγκριτικά με τον ανταγωνισμό.



Εικόνα 134. Ηλικιακή κατανομή του δείγματος των χρηστών.



Εικόνα 35. Ποσοστό εξοικείωσης των συμμετεχόντων με παρόμοιες εφαρμογές διατροφής.

Όπως προκύπτει από την ανάλυση των δημογραφικών στοιχείων, η πλειονότητα των χρηστών ανήκει σε παραγωγικές και νεανικές ηλικιακές ομάδες, οι οποίες αποτελούν και το κύριο target group της εφαρμογής, ενώ ένα σημαντικό ποσοστό του δείγματος είχε ήδη πρότερη εμπειρία με ψηφιακά ημερολόγια διατροφής, γεγονός που καθιστά τις κρίσεις τους για το NutriHub ιδιαίτερα βαρύνουσες.

Ανάλυση Αποτελεσμάτων Αξιολόγησης

1. Χρησιμότητα και Σχεδιασμός Διεπαφής (UI/UX)

Η υιοθέτηση του μοντέλου σχεδίασης **Jetpack Compose** και του **Material Design 3** αξιολογήθηκε με εξαιρετικά θετικά σχόλια. Οι χρήστες κλήθηκαν να βαθμολογήσουν την ευκολία πλοήγησης, την οπτική άνεση, καθώς και τη ροή της αρχικής οθόνης.

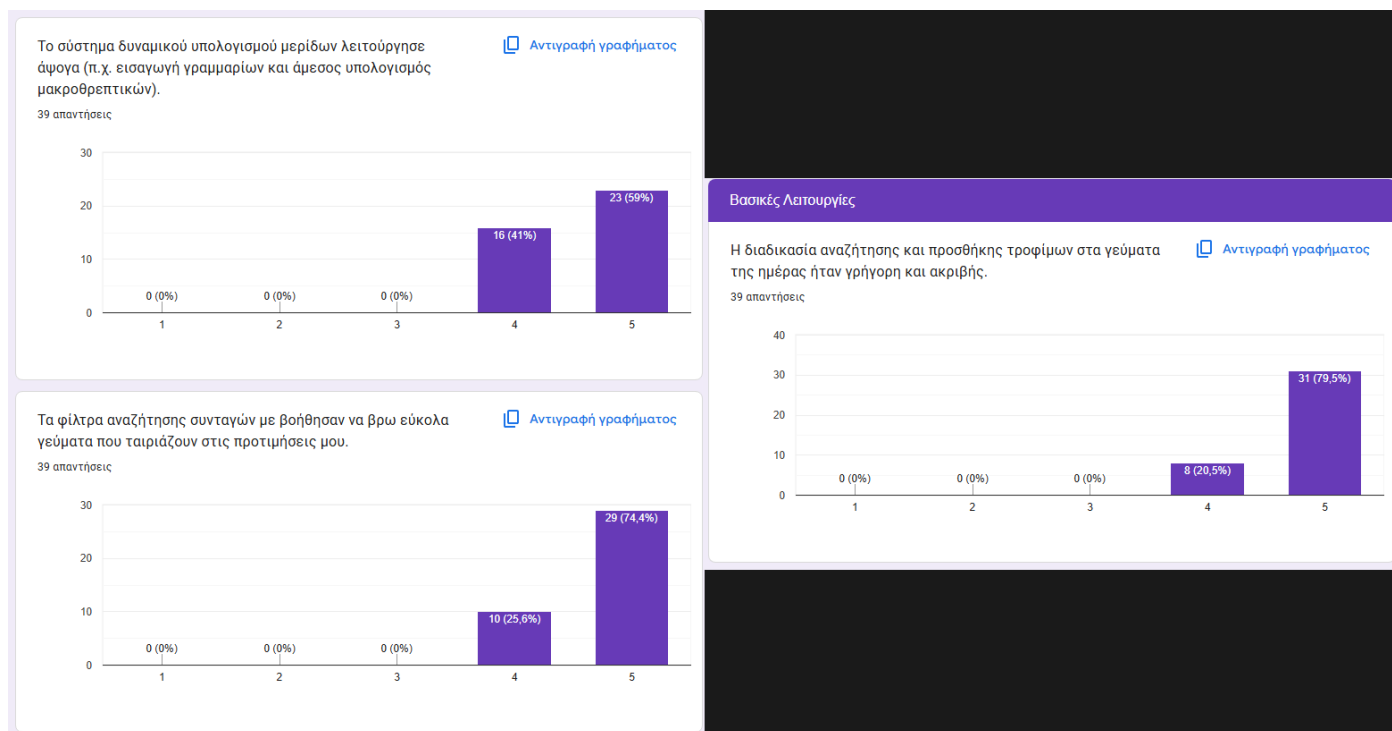


Εικόνα 36. Αποτελέσματα αξιολόγησης σχετικά με τη χρησιμότητα του UI/UX.

Οι αποκρίσεις κατέδειξαν υψηλά επίπεδα ικανοποίησης, με τη συντριπτική πλειονότητα των ερωτηθέντων να επιλέγει τις βαθμίδες 4 (Συμφωνώ) και 5 (Συμφωνώ Απόλυτα). Ιδιαίτερη αναφορά έγινε στην ομαλότητα του interface και στην άμεση οπτική ανατροφοδότηση των κυκλικών και γραμμικών δεικτών προόδου κατά την κατανάλωση των μακροθρεπτικών συστατικών.

2. Λειτουργικότητα Καταγραφής και Διαχείρισης Γευμάτων

Στον τομέα της λειτουργικότητας, η αξιολόγηση εστιάστηκε στην ταχύτητα αναζήτησης τροφίμων και στον δυναμικό υπολογισμό των θερμίδων ανάλογα με τα γραμμάρια της μερίδας. Τα αποτελέσματα επιβεβαίωσαν τη σταθερότητα του back-end και την ορθή επικοινωνία με το **FatSecret API**.

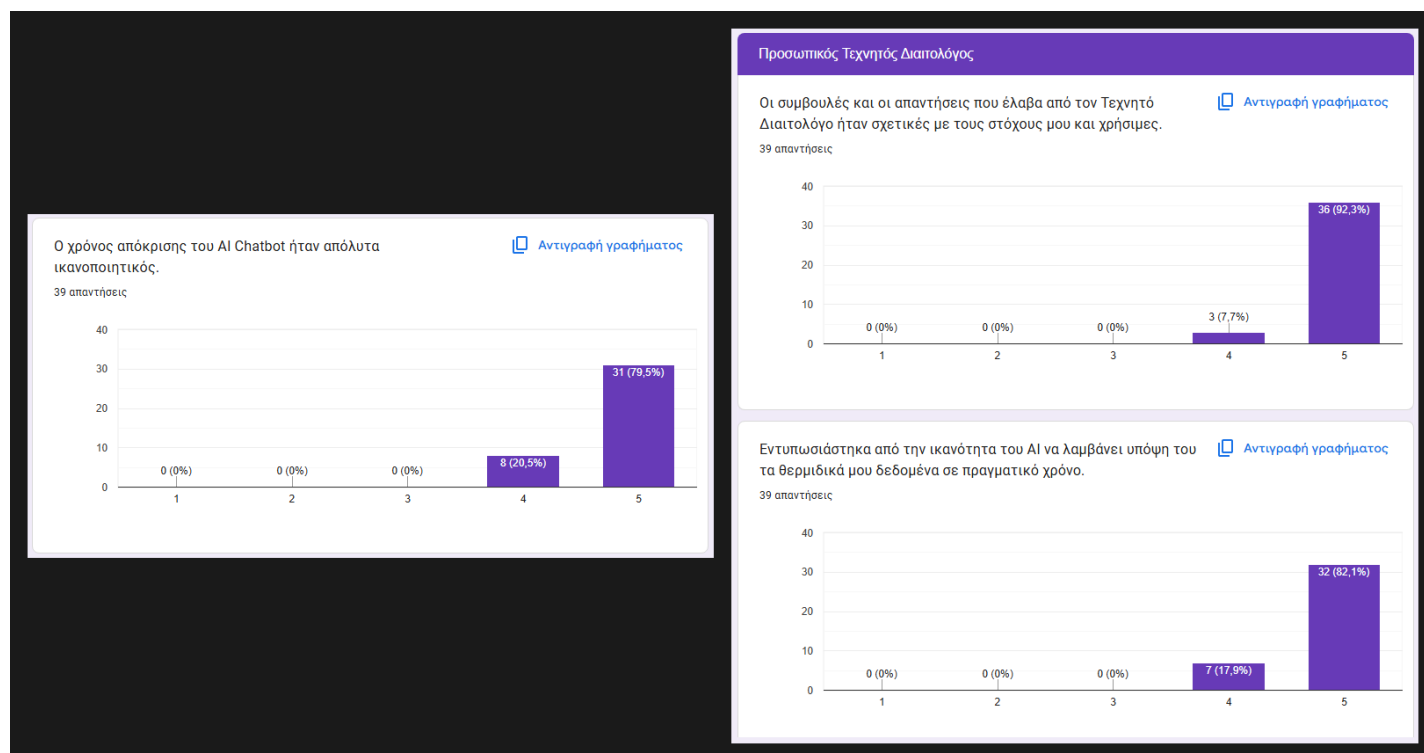


Εικόνα 37. Βαθμολογία ικανοποίησης από τη διαδικασία αναζήτησης και καταγραφής γευμάτων.

Οι χρήστες ανέφεραν ότι η διαδικασία προσθήκης γευμάτων είναι εξαιρετικά γρήγορη και δεν κουράζει, στοιχείο κρίσιμο για τη διατήρηση της καθημερινής δέσμευσης σε μια εφαρμογή fitness.

3. Αξιολόγηση του Προσωπικού AI Διαιτολόγου

Το πλέον καινοτόμο στοιχείο της διατριβής, η ενσωμάτωση του **Gemini API** για τη δημιουργία του **NutriBot AI**, αποτέλεσε το κεντρικό σημείο ενδιαφέροντος της έρευνας. Οι χρήστες αξιολόγησαν την ικανότητα του μοντέλου να παράγει εξατομικευμένες συνταγές, τη φυσικότητα της γλώσσας στην οθόνη της συνομιλίας, καθώς και το γεγονός ότι το AI γνωρίζει σε πραγματικό χρόνο το θερμιδικό υπόλοιπο του χρήστη.



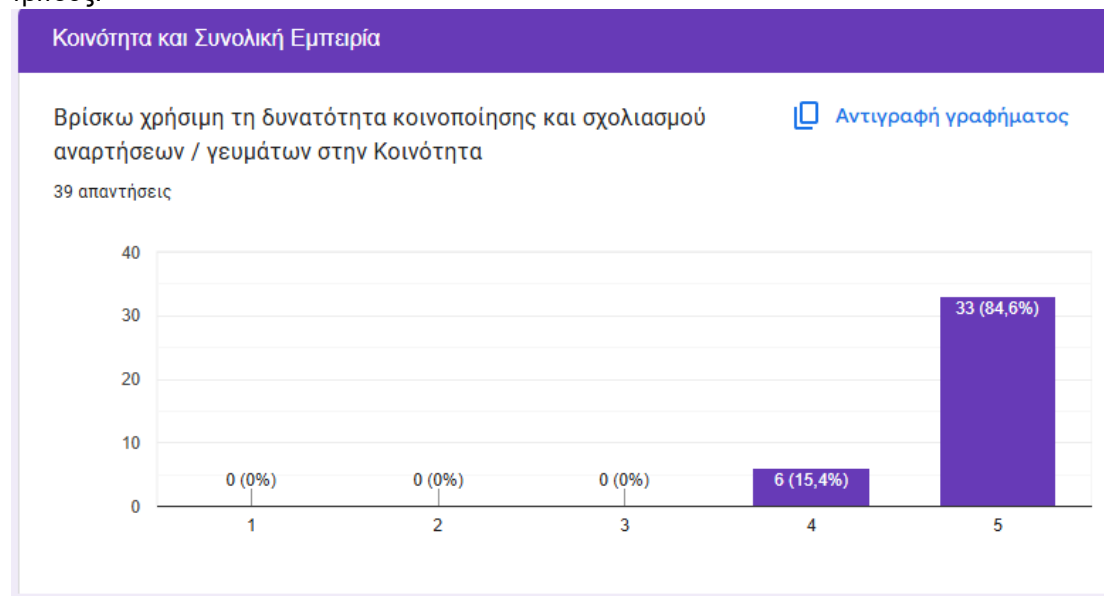
Εικόνα 37. Βαθμολογία ικανοποίησης από τη διαδικασία αναζήτησης και καταγραφής γευμάτων.

Τα διαγράμματα συλλογής των απαντήσεων δείχνουν σαφή συγκέντρωση των ψήφων στις επιλογές 4 και 5. Οι συμμετέχοντες αξιολόγησαν θετικά την ικανότητα του NutriBot να συνθέτει άμεσα συνταγές με βάση τα εναπομείναντα υλικά στο ψυγείο τους, καθώς και τον γρήγορο χρόνο απόκρισης.

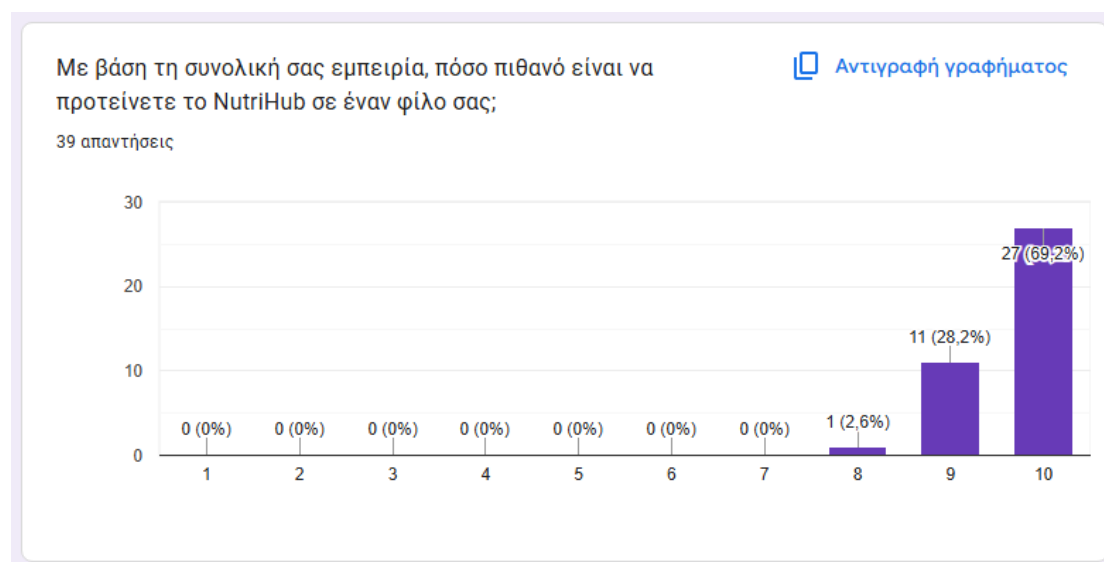
4. Κοινωνική Δικτύωση και Συνολική Αποτίμηση

Τέλος, αξιολογήθηκε η ενότητα της Κοινότητας, όπου οι χρήστες εξέφρασαν θετική άποψη για τη δυνατότητα διαμοιρασμού των γευμάτων τους και της αλληλεπίδρασης μέσω σχολίων και likes.

Η συνολική αποδοχή του συστήματος μετρήθηκε με βάση τον διεθνή δείκτη **Net Promoter Score** σε κλίμακα από 1 έως 10, αναφορικά με την πιθανότητα σύστασης της εφαρμογής σε τρίτους.



Εικόνα 38. Βαθμός χρησιμότητας των λειτουργιών κοινωνικής δικτύωσης



Εικόνα 39. Δείκτης Net Promoter Score για τη συνολική εμπειρία χρήσης.

Ο υψηλός μέσος όρος βαθμολογίας στον δείκτη NPS επιβεβαιώνει ότι το NutriHub αποτελεί μια άρτια, ανταγωνιστική και ακαδημαϊκά τεκμηριωμένη πλατφόρμα, η οποία καταφέρνει να συνδυάσει επιτυχώς την καταγραφή διατροφής με τις τεχνολογίες αιχμής της Τεχνητής Νοημοσύνης.

Σχόλια Χρηστών και Μελλοντικές Κατευθύνσεις

Μέσω των ερωτήσεων ελεύθερου κειμένου, συλλέχθηκαν πολύτιμα σχόλια που αντικατοπτρίζουν την εμπειρία των χρηστών. Στην ερώτηση «*Τι σας άρεσε περισσότερο στην εφαρμογή;*», οι απαντήσεις επικεντρώθηκαν στην αμεσότητα του NutriBot AI και στην απλότητα του σχεδιασμού.

Αντίθετα, στην ερώτηση «*Τι θα θέλατε να προστεθεί σε μια μελλοντική έκδοση;*», οι χρήστες πρότειναν λειτουργίες όπως η ενσωμάτωση Water Tracker, καθώς και η διασύνδεση με φορητές συσκευές, smartwatches. Τα συγκεκριμένα ποιοτικά δεδομένα λήφθηκαν σοβαρά υπόψη και αποτέλεσαν τη βάση για τη διαμόρφωση των Μελλοντικών Επεκτάσεων της διατριβής.

Τι σας άρεσε περισσότερο στην εφαρμογή;

33 απαντήσεις

Τα χρώματα και τα γράμματα. Είναι πολύ "καθαρό" στο μάτι

Μου αρέσει που το AI θυμάται ότι θέλω να χάσω κιλά και μου προτείνει τα σωστά πράγματα.

Απαντάει στο δευτερόλεπτο, ούτε που περιμένεις

Το ότι του γράφω τι έχω στο ψυγείο και μου βγάζει έτοιμο φαγητό.

Που το AI ξέρει τι εφάγα σήμερα και μου λέει "Σου λείπουν υδατάνθρακες, φάε μια μπανάνα "

Η ευκολία στη χρήση.

Νιώθω ότι έχω δικό μου διαιτολόγο όλη μέρα στο κινητό μου.

Ο AI διατροφολόγος. Σου γράφει τις συνταγές με βημάκια, πολύ ξεκάθαρα.

Η γρήγορη ανταπόκριση της εφαρμογής προστις επιθυμίες μου

Τι θα θέλατε να βελτιωθεί ή να προστεθεί σε μια μελλοντική έκδοση;

34 απαντήσεις

Να μου χτυπάει μια ειδοποίηση το μεσημέρι αν έχω ξεχάσει να περάσω το φαγητό μου.

Να μπορείς να κάνεις αναφορά σε κάποιο σχόλιο στην κοινότητα αν κάποιος γράψει κάτι προσβλητικό..

Σύνδεση με το Google Fit.

Θα ήθελα το ρομπότ να μου φτιάχνει αυτόματα κ τη λίστα για το σουπερμάρκετ της εβδομάδας

Να μπορώ να μοιράζομαι τη συνταγή που βρήκα στο Instagram μου.

Να έχω διαφορετικό στόχο θερμίδων τη μέρα που πάω γυμναστήριο και την ημέρα που κάθομαι σπίτι.

Δεν έχω να προτείνω κάτι.

Ίσως ένα widget για την αρχική οθόνη του κινητού για να βλέπω τις θερμίδες μου χωρίς να ανοίγω την εφαρμογή.

Συντανές γωρίς Ξηρούς καρπούς λόγω αλλεργίας

Εικόνα 40,. Απαντήσεις χρηστών

Συμπεράσματα

Η εφαρμογή αποτελεί μια ολοκληρωμένη και σύγχρονη λύση για άτομα που επιθυμούν να βελτιώσουν τις διατροφικές τους συνήθειες, να διαχειριστούν το βάρος τους και να υιοθετήσουν έναν πιο υγιεινό τρόπο ζωής. Μέσα από μια διαισθητική και φιλική προς τον χρήστη διεπαφή, δυναμικούς υπολογισμούς μακροθρεπτικών συστατικών και ισχυρά εργαλεία παρακολούθησης, η εφαρμογή δίνει στους χρήστες τη δύναμη να αναλάβουν τον πλήρη έλεγχο της καθημερινής τους διατροφής.

Για τους χρήστες, η πλατφόρμα προσφέρει τη δυνατότητα να θέτουν εξατομικευμένους στόχους με βάση τα δικά τους βιομετρικά χαρακτηριστικά, να καταγράφουν με ακρίβεια τα γεύματά τους και να παρακολουθούν τη σωματική τους πρόοδο μέσα από αναλυτικά γραφήματα. Η ενσωμάτωση προηγμένων λειτουργιών, όπως η σάρωση barcode, η δυναμική προσαρμογή μερίδων, η αναζήτηση συνταγών και η δυνατότητα ανταλλαγής ιδεών μέσω της ψηφιακής κοινότητας, διασφαλίζει ότι οι χρήστες διαθέτουν όλα τα απαραίτητα εφόδια για να παραμείνουν παρακινημένοι και να επιτυγχάνουν τους διατροφικούς τους στόχους.

Το στοιχείο που διαφοροποιεί ριζικά τη συγκεκριμένη εφαρμογή από τα συμβατικά ψηφιακά ημερολόγια είναι η άρτια ενσωμάτωση της Τεχνητής Νοημοσύνης. Ο προσωπικός Τεχνητός Διαιτολόγος NutriBot AI, ο οποίος γνωρίζει σε πραγματικό χρόνο το προφίλ και το ημερήσιο διατροφικό υπόλοιπο του εκάστοτε χρήστη, προσφέρει άμεση, εξατομικευμένη και στοχευμένη καθοδήγηση. Αυτή η καινοτομία αναβαθμίζει το σύστημα, μετατρέποντας την απλή καταγραφή δεδομένων σε μια διαδραστική εμπειρία συνεχούς εκπαίδευσης και υποστήριξης.

Κοιτάζοντας μπροστά, οι προγραμματισμένες επεκτάσεις υπόσχονται να οδηγήσουν την εφαρμογή σε νέα ύψη. Η εισαγωγή τεχνολογιών μηχανικής όρασης για την αυτοματοποιημένη αναγνώριση γευμάτων και η διασύνδεση με φορητές συσκευές -wearables- για τον δυναμικό υπολογισμό του ενεργειακού ισοζυγίου, θα παρέχουν στους χρήστες μια απόλυτα ολιστική εικόνα της υγείας τους. Επιπλέον, η παρακολούθηση μικροθρεπτικών συστατικών και η ενσωμάτωση στοιχείων gamification θα καλύψουν ένα ευρύτερο φάσμα αναγκών, καθιστώντας την πλατφόρμα πιο περιεκτική και παγκόσμια.

Εν κατακλείδι, η εφαρμογή είναι μια δυναμική και εξελισσόμενη πλατφόρμα, σχεδιασμένη να ανταποκρίνεται στις υψηλές απαιτήσεις του σύγχρονου τομέα της ψηφιακής υγείας. Χτισμένη πάνω σε τεχνολογίες αιχμής, τα τρέχοντα χαρακτηριστικά της προσφέρουν μια εξαιρετικά ισχυρή βάση για τη διαχείριση της διατροφής, ενώ οι μελλοντικές προσθήκες υπόσχονται να βελτιώσουν ακόμη περισσότερο την αξία της. Με τη συνεχή προσαρμογή στις ανάγκες των χρηστών της και την ενσωμάτωση των τελευταίων εξελίξεων στην Τεχνητή Νοημοσύνη, η εφαρμογή είναι έτοιμη να εδραιωθεί ως ένα κορυφαίο εργαλείο στον κλάδο της ευεξίας.

Βιβλιογραφία

- 1) [Sergio Di Meglio , Luigi Libero Lucio Starace and Sergio Di Martino. Starting a New REST API project? A Performance Benchmark of Frameworks and Execution Environments. CEUR Workshop Proceedings.](#)
- 2) [K. Guntupally, R. Devarakonda and K. Kehoe, "Spring Boot based REST API to Improve Data Quality Report Generation for Big Scientific Data: ARM Data Center Example," 2018 IEEE International Conference on Big Data.](#)
- 3) [Sourabh Sharma, *Modern API Development with Spring and Spring Boot: Design highly scalable and maintainable APIs with REST, gRPC, GraphQL, and the reactive paradigm* , Packt Publishing, 2021.](#)
- 4) [M. F. ELHejazi and W. H. A. Muragaa, "Improving the Security and Reliability of SDN Controller REST APIs Using JSON Web Token \(JWT\) with OpenID and auth2.0," 2024 IEEE 4th International Maghreb Meeting of the Conference on Sciences and Techniques of Automatic Control and Computer Engineering \(MI-STA\).](#)
- 5) [Giannikis, A., Alepis, E., & Virvou, M. \(2021\). Crowdsourcing Recognized Image Objects in Mobile Devices Through Machine Learning. 2021 IEEE 33rd International Conference on Tools with Artificial Intelligence \(ICTAI\), 560-567.](#)
- 6) [Ayush Vijaywargi, Uchinta Kumar Boddapati, Architectural Patterns in Android Development: Comparing MVP, MVVM, and MVI. International Journal for Research in Applied Science & Engineering Technology \(IJRASET\).](#)
- 7) [Aditya Undirwadkar. \(2025\). Understanding Jetpack Compose: Building Superior Android Apps. International Journal of Advanced Research in Science, Communication and Technology \(IJARSCT\).](#)
- 8) [Madona S. Wambua, *Modern Android 13 Development Cookbook: Over 70 recipes to solve Android development issues and create better apps with Kotlin and Jetpack Compose* , Packt Publishing, 2023.](#)
- 9) [T. Sasikala, J. J. D. Raj, B. Swathi, D. Didagur and G. S. Shetty, "Smart AI Chatbots for Tailored Nutrition and Fitness Guidance," 2025 International Conference on Knowledge Engineering and Communication Systems \(ICKECS\), 1-6.](#)
- 10) [Martin, R. C. \(2017\). *Clean Architecture: A Craftsman's Guide to Software Structure and Design*. Prentice Hall Press.](#)
- 11) [Politou, E., Alepis, E., Virvou, M., & Patsakis, C. \(2022\). Privacy and Data Protection Challenges in the Distributed Era. *Springer*.](#)
- 12) [Virvou, M., Alepis, E., Tsihrantzis, G. A., & Jain, L. C. \(2020\). Machine Learning Paradigms: Advances in Learning Analytics. *Springer*.](#)
- 13) [Google. \(2023\). *Android Developers: Jetpack Compose*. Google Developers.](#)
- 14) [Google. \(2023\). *Material Design 3 Guidelines*. Material Design.](#)
- 15) [Google Cloud. \(2024\). *Gemini API Documentation & Generative AI Models*. Google Cloud Platform.](#)
- 16) [FatSecret. \(2023\). *FatSecret Platform API Documentation*. FatSecret Developer.](#)

- 17) [Java Master Class – by Mama Samba Braima Nelson](#)
- 18) [Relational Database and SQL Essentials - by Mama Samba Braima Nelson](#)
- 19) [Spring Boot Master Class - by Mama Samba Braima Nelson](#)
- 20) [Spring Data JPA - by Mama Samba Braima Nelson](#)
- 21) [Spring Security - by Mama Samba Braima Nelson](#)
- 22) [IntelliJ IDEA Developer Guide – by Mama Samba Braima Nelson](#)
- 23) [Spring Boot Refresh Token with JWT - by bezkoder](#)