Специфікація програмного забезпечення

Мобільний застосунок для ведення калорій та контролю харчування

Software Requirements Specification

1.0

28.03.2025

Шуть Олександр Вікторович

Цимбал Мілена Русланівна

Іванов Ярослав В’ячеславович

Таранова Аліна Андріївна

**ІСТОРІЯ ЗМІН**

|  |  |  |  |
| --- | --- | --- | --- |
| **Дата** | **Опис** | **Автор** | **Коментарі** |
| 25.03.2025 | Створено пункти 1.1 - 1.4 | Шуть Олександр Вікторович |  |
| 26.03.2025 | Створено пункти 1.5, 2.1 - 2.3 | Цимбал Мілена Русланівна |  |
| 27.03.2025 | Створено пункти 2.4, 2.5, 3.1 - 3.2 | Іванов Ярослав В’ячеславович |  |
| 28.03.2025 | Створено пункти 3.3 - 3.5 | Таранова Аліна Андріївна |  |

**ЗАТВЕРДЖЕННЯ ДОКУМЕНТУ**

Наступну специфікацію вимог до програмного забезпечення було прийнято та схвалено:

|  |  |  |  |
| --- | --- | --- | --- |
| **Підпис** | **Друковане ім’я** | **Назва** | **Дата** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

1. ВСТУП

1.1 Огляд продукту

Мета продукту — розробити мобільний застосунок для контролю харчування й калорійності, з урахуванням індивідуальних фізіологічних параметрів та фітнес-цілей користувача. Додаток має стати зручним інструментом для формування здорових харчових звичок та досягнення фітнес-цілей.

Мобільний застосунок дозволяє користувачеві персоналізувати свій профіль, вказавши основні фізіологічні параметри, такі як вік, стать, зріст, вага та рівень фізичної активності, що впливають на автоматичний розрахунок добової норми калорій. Користувач також зазначає свою ціль (схуднення, підтримка ваги або набір маси). Основна функціональність додатку включає облік спожитої їжі за допомогою зручного пошуку продуктів у застосунку, додавання їх до денного раціону з автоматичним підрахунком калорій, білків, жирів і вуглеводів. Крім того, застосунок відображає графіки змін споживання калорій і дає змогу відстежувати прогрес у досягненні поставленої цілі. Додатково передбачено можливість внесення інформації про виконану протягом дня фізичну активність із підрахунком витрачених калорій, що впливає на загальний енергетичний баланс користувача.

Цільовою аудиторією мобільного застосунку є люди, які прагнуть контролювати своє харчування, слідкувати за калорійністю раціону та досягати фітнес або оздоровчих цілей, зокрема ті, хто хоче схуднути, набрати вагу або підтримувати здоровий спосіб життя.

1.2 Мета

Застосунок призначений для ведення персонального харчового щоденника, що дозволяє користувачеві здійснювати моніторинг спожитих калорій, контролювати кількість випитої води, а також відстежувати прогрес у досягненні особистих цілей, пов’язаних із масою тіла та фізіологічними показниками. Програмне забезпечення орієнтоване на широку аудиторію, зокрема осіб, які прагнуть контролювати власне харчування, підтримувати фізичну форму або досягти певного рівня фізіологічного стану.

Мобільний застосунок надає можливість користувачу персоналізувати взаємодію з інтерфейсом, враховуючи такі параметри, як вік, стать, ріст, вага, рівень фізичної активності та тип статури. У результаті, користувач отримує індивідуально розраховану добову норму калорій, що сприяє підвищенню ефективності контролю за харчуванням. Крім того, система дозволяє здійснювати пошук продуктів за назвами, додавати нові, відсутні у базі, надсилаючи запит модератору, що розширює можливості використання застосунку в реальному середовищі.

Основна мета продукту полягає в забезпеченні інструменту, що сприяє формуванню здорових звичок харчування, підвищенню усвідомленості щодо складу продуктів та кількісного споживання калорій, білків, жирів і вуглеводів. Застосунок також виконує роль аналітичного помічника, надаючи користувачу зворотний зв’язок щодо відповідності його споживання встановленим цілям, що стимулює користувача до самодисципліни та системного підходу до власного харчування.

Ключові переваги застосунку полягають у його інтуїтивному інтерфейсі, адаптивності до індивідуальних параметрів користувача, можливості відстеження прогресу у динаміці, а також доступності інструментів сортування та фільтрації продуктів за калорійністю, а також вмістом білків, вуглеводів та жирів. Додатково, реалізується функціональність для розподілу продуктів за прийомами їжі дозволяє здійснювати гнучкий облік щоденного раціону.

У процесі розробки мобільного застосунку передбачено дотримання сучасних стандартів безпеки, продуктивності та зручності користування. Інформація, що надається користувачем, обробляється з урахуванням конфіденційності та захищеності персональних даних. Оптимізація продуктивності та ефективність обчислень забезпечують стабільну роботу програмного продукту навіть за умов обмежених ресурсів мобільного пристрою, що робить його придатним для щоденного використання широким колом користувачів.

1.3 Межі

Програмний продукт реалізується у вигляді мобільного застосунку для Android, створеного з використанням мови Kotlin у поєднанні з мережею через Retrofit, та серверної частини на Nest.js з використанням TypeScript. Для аутентифікації та авторизації користувачів застосовується технологія JWT, що забезпечує захищене керування сесіями та контроль доступу.

Серверна частина обробляє запити, керує ролями (користувач, модератор, адміністратор), а також відповідає за валідацію, облік сесій, роботу з базою даних та логіку додавання і перегляду харчових продуктів. Для зберігання даних використовується MongoDB, що дозволяє ефективно оперувати з динамічною та структуровано гнучкою інформацією користувачів і продуктів.

Мобільний застосунок забезпечує облік калорій, води, ведення харчового щоденника, перегляд статистики, персоналізацію цілей та формування запитів на додавання нових продуктів. Межі системи обмежуються Android-платформою, централізованим сервером і клієнт-серверною архітектурою без реалізації веб або десктопної версії в межах цього проєкту.

1.4 Посилання

1. Документація по Kotlin – Офіційна документація мови Kotlin для розробки мобільних застосунків. Доступно на сайті Kotlin Docs | Kotlin Documentation (<https://kotlinlang.org/docs/home.html>).
2. Документація по Nest js – Офіційна документація фреймворку Nest.js для розробки бекенду. Доступно на сайті Documentation | NestJS - A progressive Node.js framework (<https://docs.nestjs.com/>[l](https://kotlinlang.org/docs/home.html)).
3. Документація по Figma – Офіційна документація застосунку Figma для розробки дизайну застосунків. Доступно на сайті Figma Design – Figma Learn - Help Center (<https://help.figma.com/hc/en-us/categories/360002042553>).
4. Документація по Mongo DB – Інструкції з налаштування та використання бази даних Mongo DB. Доступно на сайті MongoDB Documentation (<https://www.mongodb.com/docs/>).
5. JSON Web Token (JWT) – Стандарт для створення токенів автентифікації. Деталі можна знайти на офіційному ресурсі Internet Engineering Task Force (IETF) (<https://datatracker.ietf.org/doc/html/rfc7519>).
6. Документація по Retrofit– Офіційна документація бібліотеки Retrofit для виконання HTTP-запитів до REST API. Доступно на сайті Retrofit (retrofit API) (<https://square.github.io/retrofit/2.x/retrofit/>).

1.5 Означення та абревіатури

Означення:

Харчовий продукт – елемент бази даних, що містить інформацію про калорійність, вміст білків, жирів і вуглеводів на 100 грамів.

Прийом їжі – класифікація споживання їжі за часом доби (сніданок, обід, вечеря, перекуси тощо), яка використовується для обліку спожитих калорій.

Ціль користувача – обраний напрям фізичної трансформації (схуднення, набір маси, підтримка форми), який визначає добову норму калорій.

Тип статури – категорія фізіологічних особливостей користувача (ектоморф, мезоморф, ендоморф), що враховується при персоналізації харчового плану.

Рівень активності – оцінка фізичної навантаженості користувача (сидячий, помірний, активний), що впливає на розрахунок енергетичних потреб.

Модератор – користувач з розширеними правами, який має доступ до обробки запитів на додавання нових продуктів.

Адміністратор – користувач з максимальними правами, який керує ролями інших користувачів та має доступ до адміністративної панелі.

Харчовий щоденник – персоніфікований розділ застосунку, в якому відображаються всі прийоми їжі, спожиті калорії та вода за день.

Абревіатури:

API (Application Programming Interface) – Програмний інтерфейс взаємодії додатків.

SRS (Software Requirements Specification) – Специфікація вимог до програмного забезпечення.

DB (Database) – База даних.

UI (User Interface) – Користувацький інтерфейс.

HTTPS (HyperText Transfer Protocol Secure) – Протокол безпечної передачі даних у мережі Інтернет.

REST (Representational State Transfer) – Архітектурний стиль для побудови веб-сервісів.

NoSQL (Not Only SQL) – Тип баз даних, орієнтований на гнучке зберігання даних, зокрема MongoDB.

BMI (Body Mass Index) – Індекс маси тіла, що використовується для оцінки фізіологічного стану людини.

JWT (JSON Web Token) – Формат токенів для аутентифікації.

CRUD (Create, Read, Update, Delete) – Набір основних операцій для роботи з даними у базі даних, що включає створення, читання, оновлення та видалення записів.

MB (Megabyte) – мегабайт, одиниця вимірювання обсягу цифрової інформації.

2. ЗАГАЛЬНИЙ ОПИС

2.1 Перспективи продукту

Мобільний додаток є системою для контролю харчування, активності та досягнення фітнес-цілей, який пропонує персоналізований підхід до здорового способу життя. Додаток забезпечує точний облік спожитих і витрачених калорій, зручну організацію прийомів їжі, моніторинг фізичної активності, а також можливість додавання нових продуктів у базу. Застосунок створено для тих, хто прагне контролювати свій раціон, покращити фізичну форму та вести більш усвідомлений і здоровий спосіб життя.

У застосунку реалізується двоетапна система реєстрації користувача. На першому етапі користувач створює обліковий запис, вводить унікальне ім’я користувача та пароль із підтвердженням. Після успішної реєстрації відбувається другий етап — персоналізація профілю. Користувач заповнює анкету, вказує стать, зріст, вагу, дату народження, рівень фізичної активності (сидячий, помірний або активний), а також обирає тип статури. Далі визначає свою ціль — схуднення, набір м’язової маси або підтримка форми — та зазначає бажану вагу. Зібрані дані використовуються для формування персоналізованих рекомендацій і розрахунку добової норми калорій.

Однією з ключових особливостей додатка є можливість додавання та відстеження спожитих продуктів із детальним урахуванням калорій, білків, жирів і вуглеводів. Користувачі можуть легко знаходити продукти через пошукову систему, фільтрувати їх за категоріями (наприклад, фрукти, овочі, м’ясо, напої) та вказувати кількість спожитої їжі. Це забезпечує точний облік раціону та дозволяє користувачам краще розуміти, як їхні харчові звички впливають на досягнення поставлених цілей. Функція сортування продуктів за калорійністю додатково спрощує вибір їжі відповідно до потреб.

Важливим елементом є автоматичний розрахунок добової норми калорій, який базується на персональних даних користувача, таких як зріст, вага, тип статури, рівень фізичної активності та цілі. Ця функція допомагає користувачам отримувати чіткі рекомендації щодо споживання калорій, що сприяє більш усвідомленому підходу до харчування. Крім того, додаток відстежує кількість випитої води та спалених калорій, надаючи цілісну картину щоденної активності.

Функціонал додавання невідомих продуктів до бази даних через запити до модератора розширює можливості додатка, дозволяючи користувачам вносити унікальні продукти та збагачувати базу для всіх учасників. Це створює динамічне середовище, де спільнота може брати участь у вдосконаленні платформи, додаючи нові продукти з їхніми поживними характеристиками.

Організація прийомів їжі (сніданок, обід, вечеря, перекуси) дозволяє користувачам чітко структурувати свій раціон протягом дня. Кожен прийом їжі супроводжується детальною інформацією про калорії, що допомагає аналізувати та коригувати харчові звички.

Профіль користувача, де відображаються персональні дані, цілі та прогрес, створює зручний простір для моніторингу досягнень і внесення змін до налаштувань, таких як цільова вага чи рівень активності.

Для адміністраторів та модераторів додаток передбачає спеціальну панель управління, яка дозволяє обробляти запити на додавання продуктів, керувати ролями користувачів і забезпечувати належну роботу платформи. Це сприяє підтримці актуальності бази даних і високої якості користувацького досвіду.

Загалом, платформа створює зручне та інтерактивне середовище для управління харчуванням і досягнення фітнес-цілей. Поєднання персоналізованих рекомендацій, гнучкої системи пошуку та фільтрації продуктів, а також інструментів для моніторингу прогресу робить додаток універсальним рішенням для широкої аудиторії. Він не лише допомагає користувачам досягати своїх цілей, але й сприяє розвитку культури здорового харчування, мотивуючи до усвідомлених змін у способі життя.

2.2 Функції продукту

Мобільний додаток для контролю харчування та фітнесу пропонує персоналізований підхід до здорового способу життя. Він дозволяє користувачам вести облік спожитих і витрачених калорій, відстежувати фізичну активність, моніторити споживання води та зручно організовувати прийоми їжі.

Система реєстрації включає два етапи: створення облікового запису та заповнення анкети з особистими даними (зріст, вага, активність, цілі). На основі цієї інформації додаток розраховує добову норму калорій.

Користувачі можуть знаходити продукти через пошук, фільтрувати їх, вказувати кількість і бачити дані про калорії, білки, жири та вуглеводи. Додаток підтримує додавання нових продуктів через модерацію та дозволяє сортувати їжу за калорійністю.

Профіль відображає цілі, прогрес і дозволяє змінювати налаштування. Для адміністраторів передбачено панель керування запитами та базою даних. Модератори мають доступ до обробки запитів на додавання нових продуктів, перевірки їхніх харчових характеристик і оновлення інформації в базі, що сприяє її актуальності та достовірності.

Завдяки широкому функціоналу та зручному інтерфейсу, додаток допомагає користувачам досягати фітнес-цілей і підтримувати здоровий спосіб життя.

2.3 Характеристики користувачів

Мобільний додаток для контролю харчування, фізичної активності та реалізації фітнес-цілей чітко розподіляє ролі учасників, надаючи кожній із них унікальний набір функцій і прав доступу до компонентів платформи. Учасники поділяються на зареєстрованих користувачів, модераторів і адміністраторів.

Авторизовані користувачі отримують доступ до всіх основних функцій платформи після реєстрації та входу в систему. Вони можуть створювати власні записи про харчування, фізичну активність, пропонувати нові страви до бази даних.

Модератори мають розширені можливості для управління контентом та поведінкою користувачів. Вони можуть перевіряти та модерувати записи щодо отриманих запитів на нові страви від користувачів, а саме відхиляти або приймати страви, що будуть додані до бази даних платформи.

Адміністратори мають найвищі привілеї на платформі та доступ до адміністративної панелі. Вони здійснюють загальне управління платформою, можуть змінювати ролі користувачів (надаючи чи позбавляючи прав модераторів). Адміністратори відповідають за належне функціонування платформи та її розвиток.

Неавторизовані користувачі не матимуть доступу до функціональності платформи, оскільки їм необхідно пройти реєстрацію та авторизацію для того, щоб скористатися можливостями системи. Без авторизації вони не зможуть взаємодіяти з контентом чи переглядати персоналізовані рекомендації.

2.4 Загальні обмеження

Розробка мобільного застосунку для контролю харчування, фізичної активності та досягнення фітнес-цілей передбачає низку технічних обмежень, які впливають на вибір архітектури, функціональність та загальну продуктивність системи.

Одним із ключових аспектів є апаратні обмеження мобільних пристроїв, на яких працюватиме застосунок. Зокрема, необхідно враховувати версію Android, починаючи з якої додаток буде підтримуватись, а також відповідну версію SDK. Обмеження щодо обсягу оперативної пам’яті, частоти процесора або наявності енергозберігаючих режимів можуть впливати на стабільність роботи застосунку, особливо при виконанні фонових задач або синхронізації з сервером.

Асинхронна обробка даних є критично важливою для забезпечення безперебійного досвіду користувача. У межах застосунку одночасно можуть виконуватись різні завдання — запис спожитих калорій, оновлення статистики, фонове збереження даних, запити до серверу. Асинхронний підхід дозволяє уникнути блокування головного потоку, забезпечуючи плавність інтерфейсу та оперативність реакції на дії користувача.

Окрему увагу приділено збереженню історії записів споживання калорій. Ця функція дозволяє користувачам переглядати прогрес, аналізувати свої харчові звички та порівнювати дані за різні періоди. Для цього застосунок зберігає часові мітки, типи продуктів, кількість калорій та інші параметри, пов’язані з конкретними записами. Це створює додаткове навантаження на базу даних, тому важливо забезпечити ефективну структуру зберігання та швидкий доступ до історичних даних.

Питання безпеки та конфіденційності користувацьких даних є одним з пріоритетів. Застосунок оперує персональною інформацією — включаючи фізичні параметри, харчові вподобання та приватні дані аккаунта — тому впроваджено шифрування з’єднань (HTTPS), безпечне зберігання локальних даних, а також механізми аутентифікації та контролю доступу. Це дозволяє захистити інформацію від несанкціонованого доступу та гарантує користувачам безпечне середовище для ведення власного фітнес-щоденника.

2.5 Припущення й залежності

У процесі розробки мобільного застосунку зроблено низку припущень і враховано ключові залежності, які безпосередньо впливають на його функціональність, продуктивність та надійність.

Першочергово передбачено, що застосунок працюватиме виключно на мобільних пристроях під керуванням операційної системи Android. Це забезпечує сумісність із більшістю сучасних смартфонів, дозволяє використовувати актуальні можливості платформи Kotlin, зберігаючи при цьому стабільність роботи та адаптивність інтерфейсу на різних типах екранів. Припускається, що користувач має стабільне інтернет-з’єднання, оскільки взаємодія з серверною частиною здійснюється за допомогою REST API через бібліотеку Retrofit.

Функціональність пошуку та фільтрації продуктів реалізовано на стороні сервера. Застосовується алгоритм прямого пошуку за назвою з підтримкою часткового співпадіння, що забезпечує швидку відповідь на запити користувача. Крім того, передбачено сортування результатів за калорійністю та вмістом поживних речовин (білків, жирів, вуглеводів), що дозволяє формувати індивідуальні фільтри. Для прискорення пошуку використовується індексація даних у базі MongoDB.

Алгоритм обчислення добової калорійності ґрунтується на базовій формулі BMR, яка залежить від таких параметрів, як стать, зріст, вага, вік, рівень фізичної активності та тип статури. Отримане значення коригується відповідно до обраної цілі користувача (схуднення, набір маси або підтримка форми), формуючи рекомендовану добову норму калорій. Додатково враховуються калорії, спалені під час фізичної активності, які користувач може фіксувати вручну.

Програма створена з урахуванням можливого зростання кількості користувачів, тому важливим припущенням є забезпечення масштабованості. Серверна архітектура, побудована на Nest.js, дозволяє горизонтально масштабувати систему, розподіляти навантаження та оперативно реагувати на запити великої кількості клієнтів. MongoDB як база даних вибрана завдяки гнучкості структури зберігання та високій швидкодії при обробці великих обсягів динамічної інформації.

Однією з критичних залежностей є безпека та конфіденційність персональних даних. Вся передача інформації між клієнтом і сервером захищена протоколом HTTPS. Для авторизації та збереження сесій застосовується технологія JWT, що дозволяє ефективно керувати правами доступу. Дані зберігаються у відповідності до принципів мінімізації доступу та ізоляції користувацької інформації.

Застосунок передбачає механізм модерації контенту, що базується на ієрархії ролей користувачів. Роль модератора дає змогу обробляти запити на додавання нових продуктів, перевіряти їхню харчову цінність та затверджувати оновлення бази даних. Адміністратор, у свою чергу, має повний доступ до керування системою, зокрема до призначення ролей.

Система також розроблена з урахуванням мінімальної залежності від сторонніх сервісів. Основна функціональність, включно з обробкою даних, автентифікацією, модерацією та розрахунками, реалізована власними засобами без прив’язки до зовнішніх API, що гарантує автономність і стабільність роботи навіть за умов обмеженого доступу до інтернету або відмови сторонніх сервісів.

3. КОНКРЕТНІ ВИМОГИ

3.1 Вимоги до зовнішніх інтерфейсів

У цьому розділі описано, як мобільний додаток взаємодіятиме з користувачами, іншими системами та пристроями, а також як відбуватиметься обмін даними.

3.1.1 Інтерфейс користувача

Інтерфейс мобільного додатка розроблено з акцентом на інтуїтивність і адаптивність для різних розмірів екранів смартфонів і планшетів. Основні елементи інтерфейсу включають головне меню, віджет з добовою нормою калорій, поля для пошуку та додавання продуктів, фільтри для сортування продуктів за категоріями та поживними речовинами, а також розділи для відстеження прийомів їжі (сніданок, обід, вечеря, перекуси). Користувачі можуть створювати, редагувати та видаляти записи про спожиту їжу, переглядати та оновлювати профіль із персональними даними (зріст, вага, цілі), а також подавати запити на додавання нових продуктів до бази даних.

3.1.2 Апаратний інтерфейс

Додаток сумісний із мобільними пристроями, що працюють на операційній системі Android версії 10.0 або новішій. Для стабільної роботи сам додаток потребує щонайменше 300 MB оперативної пам’яті та процесор із тактовою частотою не менше 1.9 ГГц від двох ядер, а також стабільним інтернет-з’єднанням зі швидкістю від 10 Мбіт/с.

3.1.3 Програмний інтерфейс

Платформа використовує RESTful API для взаємодії між клієнтською частиною (мобільним додатком) і сервером. Серверна частина реалізована з використанням фреймворку Nest.js, що підтримує CRUD-операції для управління даними про продукти, профілі користувачів, прийоми їжі та запити до модераторів. Аутентифікація та авторизація здійснюються за допомогою JWT. Для зберігання даних застосовується нереляційна база даних MongoDB, яка містить колекції для користувачів, продуктів і записів про харчування. Клієнтська частина розроблена на Kotlin з використанням Android Studio, що забезпечує високу продуктивність і нативну інтеграцію з платформою Android.

3.1.4 Комунікаційний протокол

Мобільний застосунок використовує клієнт-серверну архітектуру з комунікацією через HTTPS-протокол у стилі REST API. Обмін даними відбувається у форматі JSON за допомогою HTTP-запитів.

Для захисту даних застосовується JWT-аутентифікація: після входу користувач отримує токен, який надсилається в заголовку кожного запиту. На стороні Android-клієнта використовується бібліотека Retrofit для обміну з сервером у фоновому режимі.

3.1.5 Обмеження пам’яті

Мобільний застосунок орієнтований на пристрої з обмеженими ресурсами, тому споживання пам’яті оптимізовано. Основна частина даних зберігається на сервері, а на клієнтському пристрої кешуються лише тимчасові дані.

Очікуване навантаження на оперативну пам’ять застосунку — не більше 300 MB під час активного використання. Усі фонові обчислення (розрахунок калорій, обробка статистики) виконуються з мінімальним використанням ресурсів, що забезпечує стабільну роботу навіть на бюджетних Android-пристроях.

3.1.6 Операції

Мобільний додаток для контролю харчування пропонує користувачам набір операцій для управління раціоном, профілем і взаємодії з базою продуктів. Користувачі можуть додавати, редагувати та видаляти записи про спожиті продукти, вказуючи їхню кількість і прив’язку до певного прийому їжі (сніданок, обід, вечеря, перекус). Передбачено можливість створення запитів до модератора для додавання нових продуктів до бази даних із зазначенням назви, калорійності, вмісту білків, жирів, вуглеводів і категорії. Користувачі можуть переглядати та оновлювати свій профіль, включаючи дані про зріст, вагу, тип статури, рівень активності, цілі та цільову вагу. Для зручності навігації реалізовано пошук продуктів за назвою, фільтрацію за категоріями і сортування за калоріями, білками, жирами чи вуглеводами. Користувачі можуть додавати вправи, обираючи їх зі списку та вказуючи тривалість у хвилинах. Система автоматично розраховує спалені калорії, а записи зберігаються в профілі. Модератори можуть схвалювати або відхиляти запити на додавання продуктів, а адміністратори — керувати ролями користувачів через панель адміністрування.

3.1.7 Функції продукту

Мобільний додаток забезпечує реєстрацію нових користувачів через унікальне ім’я і пароль із підтвердженням, а також авторизацію за цими даними. Основний функціонал включає автоматичний розрахунок добової норми калорій на основі персональних даних (стать, зріст, вага, тип статури, рівень активності, цілі) та відображення її у віджеті на головній сторінці. Користувачі можуть додавати продукти до прийомів їжі через пошукову форму з випадаючим списком, відстежувати спожиті калорії, білки, жири, вуглеводи та кількість випитої води. Система дозволяє створювати запити на додавання нових продуктів до бази, які обробляються модераторами. Профіль користувача містить персональні дані, аватарку та статистику прогресу (поточна вага, цільова вага). Функція редагування профілю дає змогу оновлювати інформацію, а панель адміністрування дозволяє модераторам і адміністраторам керувати запитами та ролями. Додаток також відображає аналітику споживання макронутрієнтів і підтримує фільтрацію продуктів за категоріями та сортування за поживними характеристиками для полегшення планування раціону. Додаток підтримує облік фізичної активності з автоматичним підрахунком спалених калорій. Обрані вправи та їх тривалість враховуються в щоденній статистиці.

3.1.8 Припущення й залежності

Мобільний застосунок розрахований на роботу виключно на Android-пристроях з версією ОС не нижче 10.0. Передбачається постійна наявність інтернет-з’єднання для обміну даними з сервером через REST API. Функціональність клієнта залежить від стабільної роботи серверної частини, реалізованої на Nest.js із базою даних MongoDB. Облік фізичної активності здійснюється вручну, без інтеграції з фітнес-трекерами. Продуктивність застосунку та швидкість пошуку залежить від ефективної індексації даних на сервері. Масштабованість і надійність системи передбачено на рівні архітектури, однак стабільність роботи залежить від хостингового середовища та оновлень сторонніх бібліотек.

3.2 Властивості програмного продукту

3.2.1 Додавання нових продуктів до бази даних

3.2.1.1 Вступ

Функціонал додавання нових продуктів дозволяє користувачам подавати запити на включення невідомих продуктів до бази даних мобільного додатка для контролю харчування. Це дає змогу розширювати базу продуктів, адаптувати її до потреб користувачів і забезпечувати точне відстеження калорій та макронутрієнтів для різноманітних раціонів.

#### 3.2.1.2 Вхідні дані

Користувачі вводять дані про новий продукт, включаючи назву, калорійність на 100 грамів, вміст білків, жирів і вуглеводів на 100 грамів, а також категорію продукту.

#### 3.2.1.3 Обробка

Під час подання запиту система перевіряє введені дані на відповідність вимогам: усі обов’язкові поля (назва, калорійність, макронутрієнти, категорія) мають бути заповнені, а числові значення (калорії, білки, жири, вуглеводи) — бути невід’ємними та реалістичними. Якщо дані коректні, запит надсилається до панелі модератора для розгляду. У разі невідповідності вимогам (наприклад, порожні поля або нереальні значення калорій) користувач отримує повідомлення про помилку. Після схвалення модератором продукт додається до бази даних MongoDB, асоціюючись із категорією та стаючи доступним для всіх користувачів.

#### 3.2.1.4 Вихідні дані

Після успішного подання запиту користувач отримує сповіщення про те, що запит надіслано на розгляд модератору. У разі схвалення продукт стає доступним для пошуку та додавання до прийомів їжі.

#### 3.2.1.5 Обробка помилок

У разі помилок під час подання запиту (наприклад, некоректний формат даних, порожні обов’язкові поля або невідповідність категорії) система відображає повідомлення з описом проблеми та рекомендаціями щодо виправлення.

3.2.2 Додавання прийому їжі  
 3.2.2.1 Вступ

Функціонал додавання прийому їжі дозволяє користувачам фіксувати спожиті продукти протягом дня, групуючи їх за окремими прийомами — сніданком, обідом, вечерею або перекусами. Це дає змогу точно відстежувати добове споживання калорій та інших поживних речовин, формувати харчові звички й отримувати аналітику для досягнення поставлених цілей — наприклад, схуднення, набору ваги або підтримання здорового способу життя.

3.2.2.2 Вхідні дані

Користувач обирає страву або продукт та вводить інформацію про прийом їжі: її вагу та категорію прийому (сніданок, обід, вечеря, перекус). Їжа обирається з внутрішньої бази продуктів.

3.2.2.3 Обробка

Після введення інформації, система розраховує калорійність, а також вміст білків, жирів та вуглеводів відповідно до бази даних харчових продуктів. Дані автоматично зберігаються та прив’язуються до конкретної дати й часу. Користувач може редагувати або видаляти записи про прийоми їжі. Система синхронізує зміни у профілі користувача, оновлюючи загальні щоденні показники.

3.2.2.4 Вихідні дані

Після успішного додавання прийому їжі користувач отримує оновлену інформацію про загальну кількість спожитих калорій за день, а також детальну статистику за макронутрієнтами. Вся історія прийомів їжі доступна в щоденнику харчування з можливістю перегляду за днями.

3.2.2.5 Обробка помилок

У разі, якщо користувач не вказав обов’язкові дані (наприклад, вагу продукту або прийом їжі), система повідомляє про помилку з відповідним підказуванням. У випадку втрати зв’язку з сервером або збоїв у збереженні даних виводиться відповідне повідомлення із рекомендаціями повторити дію пізніше.

3.2.3 Перегляд та оновлення профілю

3.2.3.1 Вступ

Функціонал перегляду та оновлення профілю дозволяє користувачу контролювати особисту інформацію, а також змінювати налаштування, пов’язані з цілями здоров’я та харчування. Це забезпечує персоналізований досвід використання додатку, враховуючи індивідуальні параметри, такі як вік, стать, рівень фізичної активності та ціль.

#### 3.2.3.2 Вхідні дані

Для перегляду профілю користувач відкриває відповідну сторінку, де відображаються збережені дані: ім’я користувача, стать, зріст, вага, тип статури, рівень фізичної активності, дата народження, ціль, цільова вага та аватарка. Для оновлення профілю користувач вводить або змінює ці дані, включаючи можливість завантаження нової аватарки (зображення у форматах JPEG, PNG або JPG).

#### 3.2.3.3 Обробка

Після внесення змін система оновлює профіль користувача в базі даних та перераховує рекомендовану норму споживання калорій та макронутрієнтів на день, виходячи з нових параметрів. Дані використовуються для генерації персоналізованої цілі. Усі зміни синхронізуються з сервером та зберігаються в обліковому записі.

#### 3.2.3.4 Вихідні дані

Користувач отримує оновлений вигляд свого профілю, включаючи нові параметри та змінені цілі. Ці дані впливають на подальші розрахунки в додатку (наприклад, щоденну норму калорій), які відображаються в головному інтерфейсі.

#### 3.2.3.5 Обробка помилок

У разі введення некоректних або неповних даних (наприклад, нечислове значення у полі «вік» або відсутність обов’язкових полів) система виводить відповідне повідомлення з поясненням. Якщо не вдалося зберегти зміни через проблеми з мережею чи сервером, користувачу пропонується повторити дію пізніше або перевірити підключення.

3.2.4 Пошук, сортування та фільтрація продуктів

3.2.4.1 Вступ

Функціонал пошуку, сортування та фільтрації продуктів у мобільному додатку для контролю харчування дозволяє користувачам швидко знаходити продукти в базі даних, щоб додавати їх до прийомів їжі. Ця функція забезпечує зручність у виборі продуктів шляхом введення назви, сортування за поживними характеристиками (калорії, білки, жири, вуглеводи) та фільтрації за категоріями, що сприяє ефективному плануванню раціону.

#### 3.2.4.2 Вхідні дані

Користувач вводить пошуковий запит у текстове поле (наприклад, частину або повну назву продукту). Для сортування обирається один із параметрів: калорії, білки, жири або вуглеводи (за зростанням або спаданням). Для фільтрації користувач вибирає одну або кілька категорій продуктів із переліку. Усі введені дані надсилаються до системи для обробки.

#### 3.2.4.3 Обробка

Система обробляє пошуковий запит, виконуючи пошук у базі даних MongoDB за назвою продукту. Якщо застосовується фільтрація, система обмежує результати до продуктів із вибраних категорій. Сортування виконується за обраним параметром, використовуючи відповідні поля в базі даних. Запит оптимізовано за допомогою індексів у MongoDB для швидкого доступу до даних. Якщо пошуковий запит порожній, система повертає всі продукти з урахуванням фільтрів і сортування. У разі некоректних даних користувач отримує повідомлення про помилку.

#### 3.2.4.4 Вихідні дані

Після обробки запиту користувач отримує список продуктів, що відповідають критеріям пошуку, фільтрації та сортування. Кожен продукт відображається з назвою, калорійністю, вмістом білків, жирів, вуглеводів і категорією. Список оновлюється у разі підтвердження пошуку у разі натискання на кнопку пошуку, фільтрів або параметрів сортування. Якщо результати відсутні, відображається повідомлення "Продукти не знайдено".

#### 3.2.4.5 Обробка помилок

Якщо під час пошуку виникають проблеми, користувач отримує повідомлення про тимчасову помилку з пропозицією повторити запит. У разі введення некоректних даних система відображає повідомлення з описом проблеми. Якщо пошуковий запит занадто широкий і повертає надмірну кількість результатів, система може обмежити їх кількість.

3.2.5 Облік фізичної активності

3.2.5.1 Вступ

Функція обліку фізичної активності надає користувачу можливість фіксувати виконані вправи протягом дня. Вона дозволяє додати запис про активність шляхом вибору вправи зі списку та зазначення тривалості її виконання у хвилинах. Ці дані використовуються для автоматичного розрахунку кількості спалених калорій, що впливають на загальний енергетичний баланс користувача.

#### 3.2.5.2 Вхідні дані

Користувач взаємодіє з формою додавання активності, в якій обирає назву вправи зі списку та вводить тривалість вправи у хвилинах. Дані вводяться вручну, поле часу обов’язкове для заповнення й очікує ціле додатне число.

#### 

#### 3.2.5.3 Обробка

Після відправлення форми клієнтська частина надсилає введені дані на сервер через API-запит. Серверна логіка виконує перевірку коректності введеної тривалості та на основі коефіцієнтів, встановлених для кожної вправи, обчислює кількість спалених калорій. Результат додається до загального добового енергетичного балансу користувача, оновлюється статистика за день і створюється відповідний запис у базі даних.

#### 3.2.5.4 Вихідні дані

Користувач отримує на екрані підтвердження з інформацією про кількість спалених калорій за обрану активність. У профілі оновлюється добовий показник енергетичного балансу, а також з’являється запис про виконану вправу з її параметрами (назва, тривалість, витрачені калорії).

#### 3.2.5.5 Обробка помилок

Якщо користувач не ввів тривалість, система виводить повідомлення про помилку з інструкцією заповнення. У разі введення некоректного формату форма блокується до виправлення. Якщо під час обробки сталася помилка з’єднання із сервером, користувач бачить повідомлення про відсутність підключення або технічну помилку із пропозицією повторити дію пізніше.

3.2.6 Реєстрація та вхід

3.2.6.1 Вступ

Функціонал реєстрації та входу в мобільний додаток для контролю харчування забезпечує створення облікового запису та безпечний доступ до персоналізованих функцій, таких як відстеження раціону, профіль користувача та історія харчування. Ця функція дозволяє користувачам налаштувати свій профіль і використовувати додаток для досягнення фітнес-цілей.

#### 3.2.6.2 Вхідні дані

#### Для реєстрації користувач вводить унікальне ім’я користувача, пароль і підтвердження пароля. Для входу необхідно вказати ім’я користувача та пароль. Усі дані надсилаються до системи для обробки.

#### 3.2.6.3 Обробка

Під час реєстрації система перевіряє унікальність імені користувача в базі даних MongoDB і відповідність пароля підтвердженню. Пароль хешується перед збереженням у колекції користувачів. Після успішної реєстрації створюється обліковий запис, і користувач перенаправляється до другого етапу персоналізації. При вході система порівнює введені дані з хешованим паролем у базі даних і, у разі збігу, видає JWT для аутентифікації подальших запитів. Некоректні дані призводять до відображення повідомлення про помилку.

#### 3.2.6.4 Вихідні дані

Після успішної реєстрації користувач отримує сповіщення про створення облікового запису та перенаправлення до етапу персоналізації. Після входу користувач переходить на головну сторінку додатка, де відображаються персоналізовані дані, такі як добова норма калорій. JWT зберігається на пристрої для підтримки активної сесії.

#### 3.2.6.5 Обробка помилок

Якщо під час реєстрації ім’я користувача вже зайняте, паролі не збігаються або поля залишені порожніми, система відображає повідомлення з описом проблеми. При вході некоректне ім’я користувача чи пароль призводить до повідомлення "Невірне ім’я користувача або пароль". У разі помилки сервера користувач отримує сповіщення про тимчасову проблему з пропозицією спробувати ще раз.

3.2.7 Модерація контенту

3.2.7.1 Вступ

Функціонал модерації контенту в мобільному додатку для контролю харчування дозволяє модераторам обробляти запити користувачів на додавання нових продуктів до бази даних. Це забезпечує підтримку актуальності та якості бази продуктів, дозволяючи додавати нові позиції з точними даними про калорійність, макронутрієнти та категорію, що сприяє зручності використання додатка для всіх користувачів.

#### 3.2.7.2 Вхідні дані

#### Модератор отримує запит від користувача через панель адміністрування, який містить дані про новий продукт: назву, калорійність на 100 грамів, вміст білків, жирів, вуглеводів на 100 грамів, категорію продукту. Модератор також має доступ до опцій схвалення або відхилення запиту.

#### 

#### 3.2.7.3 Обробка

Система відображає запити на додавання продуктів у панелі модератора в базі даних MongoDB, де вони зберігаються в окремій колекції до обробки. Модератор переглядає дані запиту, перевіряючи їх на коректність і відсутність дублювання з існуючими продуктами. У разі схвалення продукт додається до основної колекції продуктів, стаючи доступним для всіх користувачів. При відхиленні запит видаляється.

#### 3.2.7.4 Вихідні дані

Після схвалення запиту продукт з’являється в базі даних і стає доступним для пошуку та додавання до прийомів їжі всіма користувачами. Модератор бачить оновлений список запитів у панелі адміністрування після обробки.

#### 3.2.7.5 Обробка помилок

Якщо під час обробки запиту виникають технічні проблеми, модератор отримує повідомлення про помилку з пропозицією повторити дію. Якщо модератор випадково схвалює дубльований продукт, система попереджає про збіг на основі пошуку за назвою. Сповіщення про помилки також надсилаються користувачу, якщо його запит не може бути оброблений через системний збій.

3.2.8 Адміністрування системи

3.2.8.1 Вступ

Функція адміністрування системи забезпечує керування правами доступу користувачів, обробку запитів. Вона доступна лише користувачам з роллю адміністратора через окрему панель керування.

#### 3.2.8.2 Вхідні дані

#### Адміністратор взаємодіє з інтерфейсом адміністрування, де може обирати користувачів зі списку, змінювати їхні ролі, переглядати запити на додавання продуктів.

#### 3.2.8.3 Обробка

Після вибору дії інтерфейс надсилає відповідний запит на сервер. Серверна частина перевіряє права доступу адміністратора, обробляє запит (наприклад, зміну ролі користувача, схвалення запиту) та вносить зміни до бази даних.

#### 3.2.8.4 Вихідні дані

Користувач з роллю адміністратора отримує оновлену інформацію в інтерфейсі — змінений статус користувача, підтвердження виконаної дії, актуальні дані про запити та системну інформацію щодо роботи платформи.

#### 3.2.8.5 Обробка помилок

#### 

#### У разі недостатніх прав доступу система блокує дію та відображає повідомлення про помилку. При спробі взаємодії з неіснуючим користувачем, некоректним запитом або збоях у базі даних адміністратор отримує відповідне повідомлення. У випадку втрати з’єднання з сервером інтерфейс повідомляє про технічну помилку та пропонує повторити дію пізніше.

3.3 Атрибути програмного продукту

3.3.1 Надійність

Мобільний додаток для контролю харчування забезпечує стабільну роботу з мінімальними ризиками збоїв. Очікуваний рівень безвідмовної роботи становить щонайменше 99% часу на рік, не враховуючи технічні роботи. База даних MongoDB підтримує регулярне резервне копіювання для швидкого відновлення даних у разі втрати чи пошкодження.

3.3.2 Доступність

Додаток доступний для користувачів 99% часу на рік, за винятком запланованих оновлень або технічних робіт. Серверна інфраструктура спроєктована для забезпечення безперебійного доступу, використовуючи балансування навантаження та автоматичне масштабування для обробки пікових навантажень, наприклад, під час масового додавання записів про харчування.

3.3.3 Безпека

Усі дані, що передаються між мобільним додатком і сервером, шифруються за допомогою протоколу HTTPS. Паролі користувачів хешуються за алгоритмом bcrypt перед збереженням у базі даних MongoDB. Аутентифікація реалізується через JWT, що забезпечує безпечний доступ до профілів і записів. Система захищена від атак типу NoSQL Injection, XSS і CSRF.

3.3.4 Супроводжуваність

Архітектура додатка є модульною, що полегшує оновлення та підтримку окремих компонентів, таких як пошук продуктів чи панель модерації. Код, написаний на Kotlin (клієнтська частина) і серверному фреймворку Nest.js, добре документований, що спрощує внесення змін. Система підтримує безперервне розгортання (CI/CD) для швидкого впровадження оновлень без зупинки роботи додатка.

3.3.5 Переносимість

Мобільний додаток розроблено на Kotlin в Android Studio, що забезпечує сумісність із пристроями на Android (версія 10.0 і вище). Серверна частина сумісна з ОС Linux і може розгортатися в контейнерах Docker, що полегшує її перенесення між різними хмарними платформами. Додаток адаптовано для різних розмірів екранів смартфонів і планшетів, забезпечуючи однакову функціональність на всіх підтримуваних пристроях.

3.3.6 Продуктивність

Застосунок забезпечує швидку реакцію на дії користувача: пошук продуктів, оновлення профілю або додавання їжі виконуються майже миттєво — зазвичай менш ніж за 1 секунду. Для прискорення обробки запитів реалізовано індексування полів у MongoDB, що дає змогу зменшити затримки при зверненні до часто використовуваних даних.

3.4 Вимоги бази даних

База даних є критичним компонентом системи мобільного додатку для контролю харчування, оскільки забезпечує централізоване, узгоджене та високопродуктивне зберігання всіх користувацьких і сервісних даних. Вона повинна підтримувати постійну доступність, масштабованість і швидкодію, враховуючи очікуване зростання кількості користувачів, запитів і обсягу інформації. Застосунок реалізує клієнт-серверну архітектуру, тому база даних має обробляти всі запити, пов’язані з автентифікацією, профілями, продуктами, прийомами їжі, фізичною активністю та модерацією в режимі реального часу.

Інформація про користувача включає набір обов’язкових параметрів: унікальне ім’я, стать, зріст, вагу, рівень фізичної активності, тип статури, дату народження, обрану ціль і цільову вагу. Ці дані потрібні не лише для візуалізації профілю, а й для точного розрахунку базової добової норми калорій, яка коригується під потреби користувача. При оновленні будь-якого з цих параметрів база даних повинна забезпечити миттєве оновлення пов’язаних значень, зокрема денного енергетичного плану. Продукти зберігаються зі всіма харчовими показниками — калорійність, вміст білків, жирів, вуглеводів на 100 грамів, а також мають категорійну класифікацію, що використовується для фільтрації і сортування. Усі ці поля мають бути оптимізовані індексами, оскільки саме вони найчастіше використовуються в пошукових запитах і побудові списків у клієнтському інтерфейсі.

Для обліку харчування зберігається зв’язок між користувачем, продуктом, прийомом їжі, часом додавання і фактичною кількістю продукту, що дозволяє проводити точний підрахунок добової спожитої калорійності. Аналогічно реалізовано збереження даних про фізичну активність: зберігається обрана вправа, тривалість у хвилинах, кількість витрачених калорій та ідентифікатор користувача. Ці дані інтегруються у статистику користувача та впливають на розрахунок залишку добової норми.

Додатково база повинна зберігати інформацію про запити користувачів на додавання нових продуктів. Кожен запит містить запропоновані характеристики продукту, дату створення, автора, статус (очікує, схвалено, відхилено). Така структура дає змогу реалізувати прозорий процес модерації, дозволяючи швидко відслідковувати історію рішень та реагувати на нові внесення до бази.

MongoDB як нереляційна база забезпечує гнучке структурування документів та добре працює з динамічними й різнорідними даними, що змінюються в залежності від активності користувача. Вона підтримує масштабування через шардування, що дозволяє ефективно розподіляти навантаження між вузлами та обслуговувати тисячі запитів щодня. Усі критичні поля мають бути проіндексовані для забезпечення швидкого доступу, зокрема під час фільтрації за категоріями, назвою продукту або харчовими показниками. Також повинні бути реалізовані механізми валідації даних на стороні бази, щоб уникнути некоректних або неповних записів, які можуть вплинути на розрахунки або відображення інформації в інтерфейсі.

3.5 Інші вимоги

3.5.1 Вимоги до захисту персональних даних

Система повинна забезпечувати надійний захист персональних даних користувачів, зокрема такої інформації, як вік, вага та індивідуальні цілі щодо здоров’я, шляхом їх зберігання у зашифрованому вигляді. Користувачі мають можливість отримати копію своїх даних або повністю видалити обліковий запис за власним запитом. Політика конфіденційності має чітко пояснювати, які дані збираються, з якою метою, а також передбачати отримання згоди користувача перед початком роботи з додатком.

3.5.2 Вимоги до прав інтелектуальної власності

Оскільки застосунок може містити інформаційні матеріали, зображення або поради щодо харчування й здоров’я, важливо дотримуватися авторських прав на контент, що використовується у системі. Адміністрація застосунку повинна гарантувати, що весь контент, включно з графікою, текстами й алгоритмами розрахунку, є ліцензованим або створеним спеціально для додатку. У разі використання сторонніх матеріалів слід вказати відповідні посилання та авторство.

3.5.3 Вимоги до екологічної ефективності

Інфраструктура застосунку повинна бути оптимізованою для мінімального споживання обчислювальних ресурсів. Серверна частина має використовувати ефективні алгоритми обробки запитів, кешування даних користувачів та оптимізовані схеми збереження інформації. Це сприятиме зменшенню енергоспоживання та забезпечить екологічно відповідальний підхід до розробки й підтримки системи.