

Міністерство освіти і науки України
Луцький національний технічний університет
Факультет комп'ютерних та інформаційних технологій
Кафедра інженерії програмного забезпечення

КВАЛІФІКАЦІЙНА РОБОТА
ЗА СТУПЕНЕМ ВИЩОЇ ОСВІТИ «БАКАЛАВР»

СТВОРЕННЯ ДОДАТКУ «MY FIT» НА PYTHON
CREATING THE «MY FIT» APPLICATION IN PYTHON

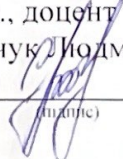
спеціальність 121 Інженерія програмного забезпечення
(шифр і назва спеціальності)

освітня програма «Інженерія програмного забезпечення»
(назва освітньої програми)

Виконав: здобувач вищої освіти
Групи ПІЗ-41
Куш Ілля Андрійович


(підпис)

Керівник:
к.т.н., доцент
Самчук Людмила Михайлівна


(підпис)

Кваліфікаційну роботу
допущено до захисту
«8» 06 2024 р.
к.т.н., доцент
Гарант освітньої програми:
Ліщина Наталія Миколаївна


(підпис)

Луцьк – 2024 року

ЛУЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

Факультет комп'ютерних та інформаційних технологій
Кафедра інженерії програмного забезпечення
Ступінь вищої освіти: бакалавр
Галузь знань: 12 Інформаційні технології
Спеціальність: 121 Інженерія програмного забезпечення
Освітня програма: «Інженерія програмного забезпечення»

ЗАТВЕРДЖУЮ

Завідувач кафедри

М. М. Мисирко
« 2 » 02 2024 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧУ ВИЩОЇ ОСВІТИ

Кушу Іллі Андрійовичу
(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи: Створення додатку «My fit» на Python

Керівник роботи: к.т.н., доцент, Самчук Людмила Михайлівна

затверджені наказом закладу вищої освіти від «30» грудня 2023 р. № 477/01-02

2. Строк подання здобувачем вищої освіти кваліфікаційної роботи «5» червня 2024 р.

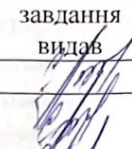
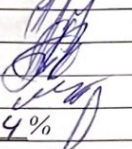
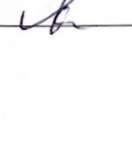
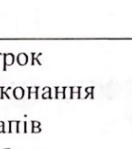
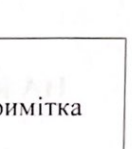
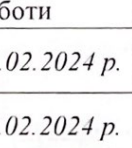
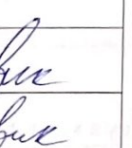
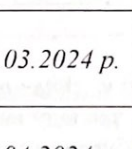
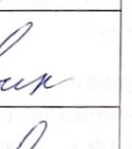
3. Вихідні дані до роботи: технічне та програмне забезпечення ЕОМ, вимоги до розробки програмного забезпечення, ергономічні вимоги до функціонування програмного засобу.

4. Зміст розрахунково-пояснювальної записки (перелік питань, що потрібно розробити):

Аналіз сучасного стану проблеми, існуючих методів і засобів її розв'язання, аналіз, визначення вимог до розроблюваного програмного забезпечення та проектування програмного забезпечення, опис функціонального наповнення об'єкта проектування, практична реалізація об'єкта проектування.

5. Перелік графічного матеріалу: 12 рисунків; 1 схема; 2 діаграми.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис	
		завдання видав	завдання прийняв
Аналіз предметної області	Самчук Л. М.		
Специфікації вимог до розробленої системи	Самчук Л. М.		
Розробка об'єкта проєктування	Самчук Л. М.		
Нормоконтроль	Повстяна Ю. С.		
Гарант ОП	Ліщина Н. М.		
Показник запозичень тексту		5,4%	
Академічна доброчесність	Яцук А. А.		

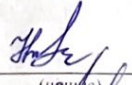
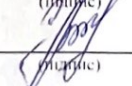
7. Дата видачі завдання «2» лютого 2024 р.

КАЛЕНДАРНИЙ ПЛАН

№ З/п	Назва етапів кваліфікаційної роботи бакалавра	Строк виконання етапів роботи	Примітка
1	Огляд літературних джерел по темі кваліфікаційної роботи бакалавра	07.02.2024 р.	
2	Аналіз проблеми розробки та впровадження об'єкту проєктування	27.02.2024 р.	
3	Обґрунтування вибору шляхів, технологій (алгоритмів) і засобів вирішення поставленого завдання	12.03.2024 р.	
4	Розробка функціонально-структурної схеми роботи об'єкта проєктування та проєктування бази даних	17.04.2024 р.	
5	Практична реалізація об'єкта проєктування та розробка бази даних	18.05.2024 р.	
6	Тестування та налагодження об'єкта проєктування	01.06.2024 р.	
7	Здача чистового варіанту кваліфікаційної роботи бакалавра на кафедру	05.06.2024 р.	

Здобувач вищої освіти

Керівник кваліфікаційної роботи


(прізвище)

(прізвище)

Кузи І. А.
(прізвище, ініціали)

Самчук Л. М.
(прізвище, ініціали)

Міністерство освіти і науки України
Луцький національний технічний університет

**Рецензія
на кваліфікаційну роботу**

Здобувач вищої освіти: Куш Ілля Андрійович

Тема: Створення додатку «My Fit» на Python

Коротка характеристика кваліфікаційної роботи: Робота присвячена створенню додатку «My Fit» на мові програмування Python. Розроблений додаток є ключовим інструментом для підрахунку калорій, що допомагає користувачам контролювати своє харчування та досягати своїх фітнес-цілей. Розробка додатку була здійснена за допомогою технологій Python, Django, SQLite, HTML, CSS та JavaScript.

Самостійні розробки і пропозиції автора: У роботі детально описаний процес створення додатку «My Fit» та доведено ефективність обраних засобів та методів розробки. Автор самостійно розробив інтерфейс користувача та основні функціональні модулі додатку, зокрема систему підрахунку калорій та базу даних продуктів харчування.

Практичне значення роботи: Додаток «My Fit» був успішно реалізований і є корисним інструментом для людей, що прагнуть контролювати своє харчування та підтримувати здоровий спосіб життя. Додаток протестовано в умовах, наближених до реальних. Кваліфікаційна робота має значне практичне значення.

Недоліки: В роботі відсутні рекомендації щодо подальшого розвитку та масштабування додатку.

Загальний висновок: Робота виконана на високому науковому рівні, має логічну та послідовну структуру, відповідає встановленим вимогам, може бути представлена до захисту на державній екзаменаційній комісії та заслуговує позитивної оцінки.

Рецензент: к.т.н, доцент, Мороз Сергій Анатолійович

Рецензент кваліфікаційної роботи _____

(підпис)

«03» серпня 2024 р.

Міністерство освіти і науки України
Луцький національний технічний університет

Факультет комп'ютерних та інформаційних технологій
Кафедра інженерії програмного забезпечення

ВІДГУК
КЕРІВНИКА НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА
Здобувач вищої освіти Куш Ілля Андрійович
група ІІЗ-41

Тема кваліфікаційної роботи: Створення додатку «My Fit» на Python
Керівник к.т.н, доцент, Самчук Людмила Михайлівна

Актуальність теми: У сучасному світі здоровий спосіб життя та контроль за харчуванням стають все більш актуальними. З метою надання користувачам зручного інструменту для підрахунку калорій було прийнято рішення про створення додатку «My Fit» на Python. Цей додаток дозволяє користувачам ефективно контролювати своє харчування, досягати фітнес-цілей та підтримувати здоровий спосіб життя.

Об'єкт дослідження: Об'єктом дослідження є додаток «My Fit», який має на меті допомогти користувачам у підрахунку калорій та управлінні своїм харчуванням. Додаток забезпечує зручний інтерфейс для введення даних про їжу та розрахунку калорійності.

Характеристика теоретичного рівня, наявності самостійних розробок і практичної значимості роботи: Студентом був розроблений додаток «My Fit» на мові програмування Python з використанням фреймворку Django та бази даних SQLite. Додаток також включає фронтенд, реалізований за допомогою HTML, CSS та JavaScript. Розробка включає систему підрахунку калорій, базу даних продуктів харчування, зручний інтерфейс користувача та інші функціональні модулі. Додаток відповідає всім поставленим вимогам, включаючи високу продуктивність, зручність користування, безпеку даних та адаптивний дизайн. Впроваджені заходи забезпечують надійну та стабільну роботу додатку. Додаток був ретельно протестований в умовах, наближених до реальних, що підтвердило його готовність до використання та високу практичну значимість.

Зауваження та недоліки: Істотних недоліків не виявлено.

Загальний висновок: Робота виконана на високому рівні та заслуговує позитивної оцінки за умови успішного захисту.

Керівник

(підпис)

Самчук Людмила Михайлівна
(прізвище, ім'я, по батькові)

«07» 06 2024 р.

АНОТАЦІЯ

Куш І. А. Створення додатку «My fit» на Python. Рукопис.

Кваліфікаційна робота бакалавра ОП «Інженерія програмного забезпечення». Луцький національний технічний університет. Луцьк, 2024.

Кваліфікаційна робота бакалавра складається зі вступу, трьох розділів, висновків і пропозицій, списку використаних джерел та додатків.

У першому розділі здійснено аналіз сучасного стану проблеми розробки додатку і сформульовано завдання. В другому розділі визначено вимоги до розроблюваної системи, а також засоби, методи і алгоритми розробки. У третьому розділі розроблено структуру та прототип додатку. У висновках узагальнено інформацію, відображену у попередніх частинах. Головними результатами дослідження є розроблене програмне забезпечення з інтуїтивно зрозумілим інтерфейсом, яке дозволяє користувачам ефективно планувати та відстежувати свої тренування.

Ключові слова: мобільний додаток, Android, метод проектування, інструменти розробки, тестування, мова програмування, фітнес, Python, тренування, здоров'я, спорт, інтерфейс, мотивація.

ABSTRACT

Kush I. Creation of the "My fit" application on Python. Manuscript.

Bachelor's qualifying thesis of the OP "Software Engineering". Lutsk National Technical University, 2024.

The bachelor's qualification work consists of an introduction, three sections, conclusions and proposals, a list of used sources and appendices.

In the first chapter, an analysis of the current state of the problem of developing addition was carried out and the task was formulated. The second section defines the requirements for the developed system, as well as means, methods and development algorithms. In the third section, the structure and prototype of the application are developed. The conclusions summarize the information presented in the previous parts. The main results of the research are developed software with an intuitive interface that allows users to effectively plan and track their training.

Keywords: mobile application, Android, design method, development tools, testing, programming language, fitness, Python, training, health, sports, interface, motivation.

ЗМІСТ

ВСТУП	7
РОЗДІЛ 1 АНАЛІЗ РОЗРОБКИ І ПОСТАНОВКА ЗАВДАНЬ	9
1.1 Аналіз сучасного стану проблеми	9
1.2 Постановка завдання на кваліфікаційну роботу бакалавра	12
Висновки до розділу 1	15
РОЗДІЛ 2 СПЕЦИФІКАЦІЯ ВИМОГ ДО РОЗРОБЛЮВАЛЬНОЇ СИСТЕМИ ..	17
2.1 Аналіз визначення вимог до розроблюваного програмного забезпечення та проектування програмного забезпечення	17
2.2 Вибір засобів, методів і алгоритмів вирішення поставленого завдання ..	19
Висновки до розділу 2	25
РОЗДІЛ 3 РОЗРОБКА ДОДАТКУ MY FIT НА PYTHON	27
3.1 Практична реалізація об'єкта проектування	27
3.2 Тестування та налагодження інформаційно-комп'ютерної системи	30
3.3 Розробка бази даних	34
3.4 Розробка інсталяційного пакета	39
Висновок до розділу 3	41
ВИСНОВКИ	42
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	44
ДОДАТКИ	46

ВСТУП

Актуальність теми. У сучасному світі зростає попит на здоровий спосіб життя та фізичну активність. Одним із ключових аспектів здорового життя є підтримка оптимальної ваги та контроль над нею. У зв'язку з цим, розробка ефективних та зручних засобів для контролю ваги та схуднення, набуває все більшої актуальності.

Оцінка сучасного стану проблеми показує, що існують різноманітні фітнес-додатки, проте багато з них не відповідають потребам користувачів у плануванні тренувань, моніторингу харчування та відстеження прогресу. Практично розв'язані задачі полягають у створенні додатків, які надають користувачам зручний інструмент для досягнення їх фітнес-цілей.

Світові тенденції вирішення цих задач вказують на зростання популярності фітнес-додатків, що надають більше інтерактивних та персоналізованих функцій для користувачів. Тенденція використання штучного інтелекту та аналізу даних у фітнес-індустрії дозволяє створювати більш ефективні та інноваційні рішення.

Актуальність цієї роботи полягає в необхідності розробки нового, високоякісного фітнес-додатку, який би задовольняв потреби користувачів у контролі ваги та схудненні. Метою даної роботи є створення зручного та ефективного інструменту для планування тренувань, моніторингу харчування та відстеження прогресу у здоров'ї.

Ця робота спирається на попередні дослідження у галузі розробки фітнес-додатків та програмування, а також враховує взаємозв'язок з іншими суміжними роботами, спрямованими на покращення фізичного здоров'я та спортивної активності.

Сфера здорового харчування та фітнесу постійно розвивається. Вона включає в себе різні компоненти: додатки для підрахунку калорій, фітнес-трекери, онлайн-консультації з дієтологами, різноманітні рецепти здорових страв. Враховуючи глобальну інформатизацію сучасного суспільства та масове

використання сучасних гаджетів, доречною є розробка додатку, який сприяв би контролю за харчуванням та досягненню здорового способу життя.

Метою виконання даної роботи є розробка та реалізація Android-програми, яка допоможе користувачу у процесі контролю за споживаними калоріями та плануванням здорового харчування. Для досягнення цієї мети було розроблено та сформульовано низку завдань:

- відстеження калорійності споживаних;
- складання індивідуального плану харчування;
- аналіз харчових звичок і рекомендації щодо їх покращення;
- провести аналіз існуючих додатків для підрахунку калорій;
- дослідити та виявити технології, які можуть бути корисні в процесі реалізації задачі;
- оволодіти методами і методиками досліджень, які допоможуть у виконанні вище поставленої роботи.

На основі проведеного аналізу вимог та технологій скласти чіткі вимоги до власного додатку:

- аналізувати складені вимоги;
- спроектувати систему відповідно до вимог;
- реалізувати спроектовану систему;
- провести тестування реалізованої системи.

Об'єктом розробки є процес контролю за харчуванням та підрахунок калорій.

Предметом розробки є Android-програма, яка створюється для допомоги користувачеві в процесі контролю споживаних калорій та планування здорового харчування.

РОЗДІЛ 1

АНАЛІЗ РОЗРОБКИ І ПОСТАНОВКА ЗАВДАНЬ

1.1 Аналіз сучасного стану проблеми

Завдання цієї роботи розробити мобільний додаток «My fit» на Python для зручного та ефективного планування тренувань, харчування та прогресу здорового способу життя. Мобільні додатки користуються великим попитом, оскільки багато користувачів користуються телефоном і виконують усі функції. Це швидший спосіб, і він завжди поруч [1].

Необхідно визначити основні і допоміжні функції. Оригінальна версія створює функцію, яку приємно мати і її завжди можна додати пізніше як оновлення. Таким чином, ми можемо оцінити початкові витрати на розробку і, якщо можливо, зменшити їх.

При розробці майбутніх додатків потрібно приділяти велику увагу дизайну. Можна сказати, що дизайн – це спосіб взаємодії користувачів із додатком. Існують різні думки щодо дизайну, деякі стверджують, що простий дизайн необхідний лише для того, щоб зосередитись на розробці програми, тоді як інші стверджують, що він починає розвиватися разом із дизайном, щоб привернути увагу користувача. Дизайн – це не тільки зовнішня оболонка програми, але й спосіб зробити технологію більш зручною та доступною [2].

Щоб розмістити додаток, потрібно створити обліковий запис розробника.

Для цього потрібно зареєструвати обліковий запис розробника, щоб ви могли розповсюджувати додаток.

Набір програм, що забезпечують управління компонентами комп'ютерної системи, такими як процесор, оперативна пам'ять, пристрій введення – виведення, мережеве обладнання, покращує логіку програми, беручи до уваги всі можливі сценарії, одночасно звертаючи увагу на обладнання, деталі і знаходячи рішення при виявленні проблем.

Android – це унікальна операційна система, і слід враховувати наступні аспекти [3]:

- при установці програми він займає більше місця (в два або чотири рази), ніж початковий розмір програми;
- при роботі з файлами на вбудованій флеш-карті швидкість зменшується в 10 разів за рахунок зменшення вільного місця;
- кожен процес може використовувати до 16 МБ (або, в деяких випадках, 24 МБ) оперативної пам'яті через додаток Linux. Між додатком і ядром знаходиться рівень API і рівень дека, заснований на власному коді;
- запускати багато програм на Android. Однак першим з них є основним і займає весь екран;
- переключитися з поточної програми на попередню або запустити нову програму. Це схоже на сканер з історією перегляду.

Коли Android декомунізує екземпляр activity, виникає проміжний стан, а останній – коли екземпляр знищується системою, а ресурси звільнюються. Діяльність має три стани:

- увімкнено або запущено – користувач може взаємодіяти з активним вікном;
- пауза – дія знаходиться вгорі і частково збігається з цією дією, перервана активність може бути знищена системою в критичних ситуаціях, коли недостатньо пам'яті;
- зупинено, якщо ця дія була повністю закрыта іншою дією. Це більше не буде видно користувачеві і може бути відкинуто системою, якщо для більш важливого процесу потрібна пам'ять.

Якщо потрібно знову побачити на екрані активність, знищену системою, необхідно повністю перезавантажитися і відновити її до колишнього стану. Якщо потрібно виконати будь-яку дію, декомунізувати, наприклад, здійснити дзвінок, надіслати електронний лист або відобразити вікно, викликається ціль.

Служба Android працює як фоновий процес, представлений Android – Докладання – Клас обслуговування. Вони не мають інтерфейсу користувача (необхідний, якщо втручання користувача не потрібно). Служба працює у фоновому режимі, виконує мережеві запити на веб-сервер, обробляє

інформацію та ініціює повідомлення. Послугу можна запустити, і вона буде продовжувати працювати, поки хтось не зупинить її або вона сама не зупиниться. Послуга, на відміну від діяльності, розрахована на довгострокове існування. Вони можуть запускати його, завжди перезавантажувати, виконувати фіксовані завдання або виконувати трудомісткі завдання. Постачальник контенту використовується при передачі даних між додатками. Декомунізація контенту використовується при передачі даних між додатками.

Під час розробки фітнес-додатку, потрібно створити робочу програму, яка взаємодіє з сервером, з якого інформація надсилається до програми.

Аналіз сучасного стану проблеми в контексті фітнес-додатків на мові програмування Python для схуднення включає огляд літературних джерел, патентний пошук та розгляд існуючих аналогів проектованої системи.

Перш за все, дослідження літературних джерел та наукових публікацій дозволяє отримати уявлення про основні принципи та методи, що застосовуються в сучасних фітнес-додатках для схуднення. Важливо розглянути такі аспекти, як алгоритми планування тренувань, системи моніторингу харчування, а також використання методів мотивації та підтримки користувача.

Патентний пошук дозволяє виявити існуючі технологічні рішення та патентовані інновації в галузі фітнес-додатків. Це допомагає з'ясувати, які технології вже використовуються та які можливості можуть бути використані в новій системі.

Під час аналізу існуючих аналогів проектованої системи варто звернути увагу на їхні переваги та недоліки. Це дозволить зрозуміти, що вже існує на ринку та які можливості можна покращити або доповнити у новій системі [4].

Порівняння існуючих рішень з потребами та очікуваннями цільової аудиторії дозволить обґрунтувати доцільність створення нової системи.

Отже, аналіз сучасного стану проблеми у контексті фітнес-додатків для схуднення на мові програмування Python допомагає зрозуміти потреби користувачів, виявити наявні технологічні рішення та їхні обмеження, а також

обґрунтувати необхідність розробки нової системи, яка відповідатиме потребам користувачів та використовуватиме передові технології для досягнення цілей схуднення та поліпшення здоров'я.

З цього виходять такі завдання на розробку додатку:

1) розробити функціональний фітнес-додаток на мові програмування Python, який буде спрямований на допомогу користувачам у схудненні та поліпшенні їхнього фізичного здоров'я;

2) забезпечити можливість планування та відстеження тренувань, включаючи різноманітні типи фізичних вправ та режими тренувань;

3) реалізувати систему моніторингу харчування, яка дозволяє користувачам вести щоденний журнал харчування та отримувати рекомендації щодо збалансованого харчування;

4) забезпечити можливість встановлення та відстеження особистих цілей схуднення та фізичної активності для кожного користувача;

5) реалізувати систему статистики та аналітики, яка дозволить користувачам переглядати свій прогрес у схудненні, покращенні фізичної форми та досягненні поставлених цілей;

6) забезпечити інтерактивний та інтуїтивно зрозумілий інтерфейс додатку, що забезпечить зручність та залучення користувачів до використання програми;

7) забезпечити безпеку та захист конфіденційності особистої інформації користувачів, включаючи дані щодо здоров'я, фізичної активності та харчування;

8) здійснити тестування та валідацію розробленого додатку для забезпечення його стабільності, ефективності та відповідності вимогам.

1.2 Постановка завдання на кваліфікаційну роботу бакалавра

Для виконання розробки Android – програми для контролю за споживаними калоріями та плануванням здорового харчування. розроблено та сформульовано низку завдань:

- відстеження калорійності споживаних;
- складання індивідуального плану харчування;
- аналіз харчових звичок і рекомендації щодо їх покращення;
- провести аналіз існуючих додатків для підрахунку калорій;
- дослідити та виявити технології, які можуть бути корисні в процесі реалізації задачі;
- оволодіти методами і методиками досліджень, які допоможуть у виконанні вище поставленої роботи.

Мобільний додаток можна в основному розділити на 3 типи:

- мобільні сайти, веб-додатки;
- гібридні додатки;
- нативні додатки.

Мобільні веб-сайти, веб-додатки – найпоширеніші типи додатків для мобільних пристроїв. Сучасні інструменти мають можливість переглядати класичні веб-сайти завдяки підтримці HTML5. Щоб почати бізнес, саме вони дозволяють добитися відмінних результатів за короткий проміжок часу за невеликі гроші. У той же час переваги та недоліки є кросплатформними.

Гібридні програми – це програми, які можуть підтримувати всі переваги API операційної системи. Завдяки push-повідомленням додаток сильно розростається, але основний вміст все ще є незалежною від платформи сторінкою з розміщенням на сервері макетом. Це дозволяє вносити деякі зміни у свій продукт без випуску нової версії. Досить завантажити зміни на сервер.

Нативні програми – це програми на основі ресурсів, які можуть виконувати найбільшу кількість функцій для кожної операційної системи. Власні програми мають переваги, як з точки зору функціональності, так і швидкості інших типів мобільних додатків. Популярні сайти використовують власні елементи керування (перемикачі, вкладки тощо), працює з додатком, який об'єднує та веб-сторінки як контент (рис. 1.1).

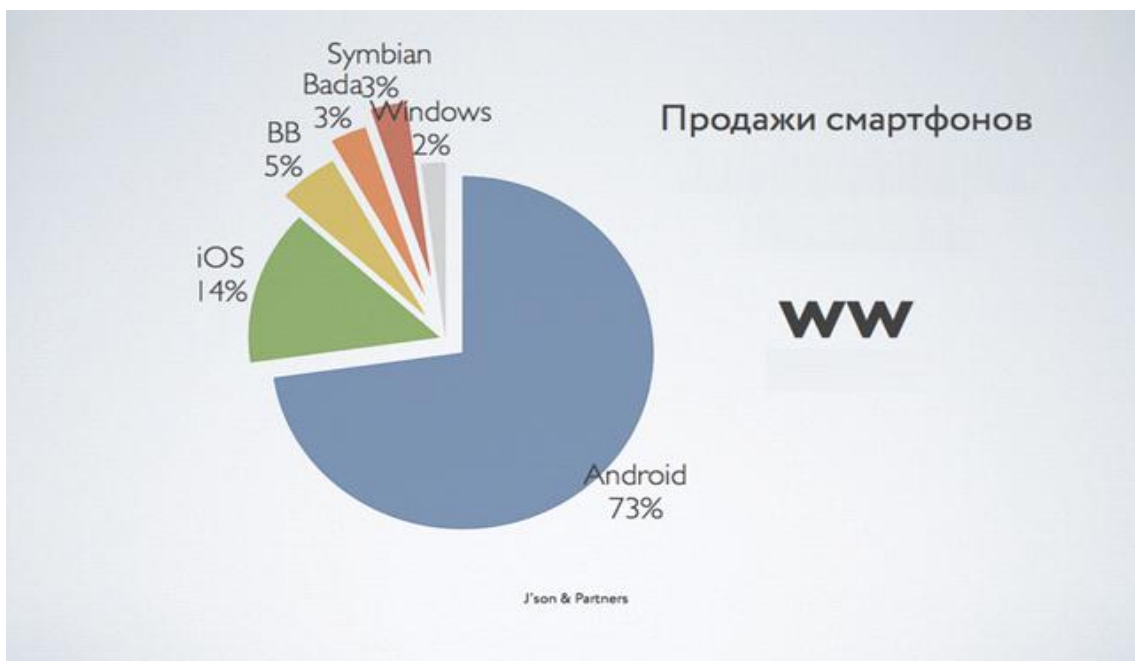


Рисунок 1.1 – Відсоток операційних систем [5]

Багатоплатформний (CLI) робочий процес: цей робочий процес дозволяє використовувати безліч різних мобільних операційних систем без особливої потреби в розробці конкретної платформи, цей робочий процес орієнтований на CordovaCLI. CLI – це інструмент високого рівня, який абстрагує більшість функціональних можливостей сценаріїв оболонки низького рівня та дозволяє створювати проекти для багатьох платформ одночасно. Інтерфейс командного рядка копіює загальний набір веб-ресурсів у підкаталог кожної мобільної платформи, вносить необхідні зміни конфігурації в кожну з них і створює сценарії. Фактичний інтерфейс командного рядка також надає загальний інтерфейс для реалізації надбудов у додатках. Робочий процес, орієнтований на платформу: цей робочий процес дає можливість створити додаток для 1 платформи та змінити його на нижчому рівні. Цей метод дозволяє додатку змішувати власний контент з веб-сайту з компонентом Cordova.

Під час першого запуску може бути найпростішим створити додаток за допомогою міжплатформенного робочого процесу декомунізації. У майбутньому, якщо вам знадобиться більше елементів керування, наданих SDK, ви можете перейти на робочий процес, орієнтований на платформу. Якщо ви

використовуєте іншу версію операційної системи платформи, у вас може бути інший веб-перегляд або той самий вигляд, але у вас інший набір функцій. Наприклад, сучасні пристрої Android підтримують функції WebRTC, але старіші версії Android не мають таких функцій.

На основі зібраних даних інформації потрібно скласти список функціональних вимог до додатку. Основні функції можуть включати в себе:

- планування тренувань: можливість створювати і зберігати розклад тренувань, вказуючи типи вправ, тривалість та інтенсивність;
- моніторинг харчування: можливість ведення журналу харчування, вказуючи прийоми їжі, калорійність та склад продуктів;
- відстеження прогресу: ведення статистики досягнень, включаючи зміни у вазі, обсязі тіла, фізичних показників та досягнення в тренуваннях;
- персоналізовані рекомендації: аналіз даних користувача та надання індивідуальних рекомендацій щодо тренувань та харчування;
- мотиваційні функції: надання позитивного підтримки та стимулювання користувача до досягнення своїх цілей.

Крім того, можливою функцією може бути інтеграція з іншими фітнес-пристроями та додатками для отримання додаткової інформації про фізичну активність та здоров'я користувача.

Визначення потреб користувачів та функціональних вимог на цьому етапі дозволить забезпечити те, що розроблений фітнес-додаток буде відповідати потребам користувачів і матиме необхідний функціонал для досягнення їхніх цілей у схудненні та поліпшенні здоров'я.

Висновки до розділу 1

В цьому розділі відбулося формулювання та постановка задачі. Це надасть змогу визначити потреби, функціонал, недоліки, а також мати уявлення про структуровану логічну та фізичну систему.

Аналіз сучасного стану проблеми у контексті фітнес-додатків для схуднення на мові програмування Python допомагає зрозуміти потреби користувачів, виявити наявні технологічні рішення та їхні обмеження, а також обґрунтувати необхідність розробки нової системи, яка відповідатиме потребам користувачів та використовуватиме передові технології для досягнення цілей схуднення та поліпшення здоров'я.

Також проаналізовано наявні методи розробки та переваги використання WebView з підтримкою Cordova адже це дозволяє програмі користуватися повним призначенням для користувача інтерфейсом. На деяких платформах це також може бути компонентом більш гібридного додатку, який змішує WebView з власними компонентами програми.

З цього аналізу стало очевидно, що на сьогоднішній день популярність фітнес-додатків зростає, оскільки вони надають користувачам зручні та ефективні інструменти для планування тренувань, ведення статистики та моніторингу здоров'я.

Проведений аналіз дозволив виявити прогалини в існуючих аналогах систем, такі як обмежені можливості планування тренувань, відсутність інструментів для моніторингу харчування та недостатня інтерактивність інтерфейсу.

Висновок з цього аналізу полягає в тому, що розробка нового фітнес-додатка, який буде спрямований на схуднення та поліпшення фізичного здоров'я, є актуальним завданням. Проведення цієї роботи має сенс, оскільки вона дозволить заповнити прогалини в існуючих рішеннях та надати користувачам нові можливості для досягнення своїх цілей у сфері фітнесу та здорового способу життя.

РОЗДІЛ 2

СПЕЦИФІКАЦІЯ ВИМОГ ДО РОЗРОБЛЮВАЛЬНОЇ СИСТЕМИ

2.1 Аналіз визначення вимог до розроблюваного програмного забезпечення та проектування програмного забезпечення

На даному етапі важливо сформулювати основні вимоги до додатку, враховуючи специфіку предмету розробки та результати порівняльного аналізу існуючих аналогів [6]. Детальний аналіз вимог допоможе визначити ключові функції та особливості, які зроблять додаток ефективним і зручним для користувачів:

- безкоштовний додаток без реклами. Багато користувачів віддають перевагу безкоштовним додаткам, оскільки вони доступні для широкої аудиторії без фінансових бар'єрів. Відсутність реклами покращує користувацький досвід, оскільки відсутні відволікаючі фактори, що робить додаток більш приємним у використанні. Фінансування може бути здійснене через інші моделі монетизації, наприклад, через продаж преміум-функцій або донати;

- зручний, зрозумілий та візуально приємний користувацький інтерфейс. Інтуїтивно зрозумілий та привабливий інтерфейс сприяє зростанню залученості користувачів і зменшенню відсотка відмов. Високоякісний UX/UI дизайн підвищує задоволеність користувачів та їхню довіру до додатку. Реалізація включає дослідження потреб користувачів, створення макетів інтерфейсу та проведення тестування для збору зворотного зв'язку і подальшого покращення дизайну;

- функція пошуку продуктів. Користувачі повинні мати можливість швидко знаходити продукти, щоб додавати їх до свого раціону, що спрощує процес підрахунку калорій. Зручний пошук дозволяє швидше заповнювати дані про споживання їжі, роблячи додаток більш корисним. Реалізація включає створення бази даних продуктів з відповідною інформацією про калорійність та інші нутрієнти, а також розробку функціоналу для швидкого та точного пошуку;

– зберігання меню на кожен день тижня. Це допомагає користувачам планувати своє харчування на тиждень вперед, що сприяє кращому контролю за споживаними калоріями. Планування меню підвищує ймовірність досягнення користувачами своїх дієтичних цілей. Реалізація включає розробку функції для збереження та редагування щоденних меню, з можливістю перегляду, додавання та видалення продуктів;

– можливість редагування раціону. Користувачі повинні мати змогу вносити зміни у свій раціон, щоб відображати актуальні дані про споживання їжі. Гнучкість у редагуванні раціону забезпечує актуальність даних і дозволяє користувачам більш точно відстежувати своє харчування. Реалізація включає розробку інтуїтивно зрозумілого інтерфейсу для внесення змін, додавання нових продуктів та видалення старих;

– зручний підрахунок калорій. Основна функція додатку полягає у підрахунку калорій, тому цей процес повинен бути максимально простим і точним. Легкість у підрахунку калорій підвищує ймовірність того, що користувачі регулярно використовуватимуть додаток. Реалізація включає автоматичний підрахунок калорій на основі введених даних про продукти, з можливістю вказівки порцій та кількості наведена в таблиці 2.1 та на рисунку 2.1.

Таблиця 2.1 – Демонстрація прецедентів різних користувачів та їх опис

Прецеденти	Опис прецедентів
Перегляд калорійності багатьох продуктів	Користувач має можливість ознайомитись з харчовою цінністю продуктів
Натискання кнопок	Користувач може взаємодіяти з кнопками у додатку
Перехід на інші сторінки	Користувач може переходити на інші сторінки додатку щоб знайти потрібну йому функцію
Функція пошуку продуктів для користувача	Дозволяє швидше заповнювати дані про споживання їжі
Можливість редагування раціону для користувача	Користувачі можуть вносити зміни у свій раціон щоб відображати актуальні дані про споживання їжі
Зберігання меню на кожен день неділі	Дає можливість користувачам планувати своє харчування на тиждень вперед що сприяє кращому контролю за споживаними калоріями

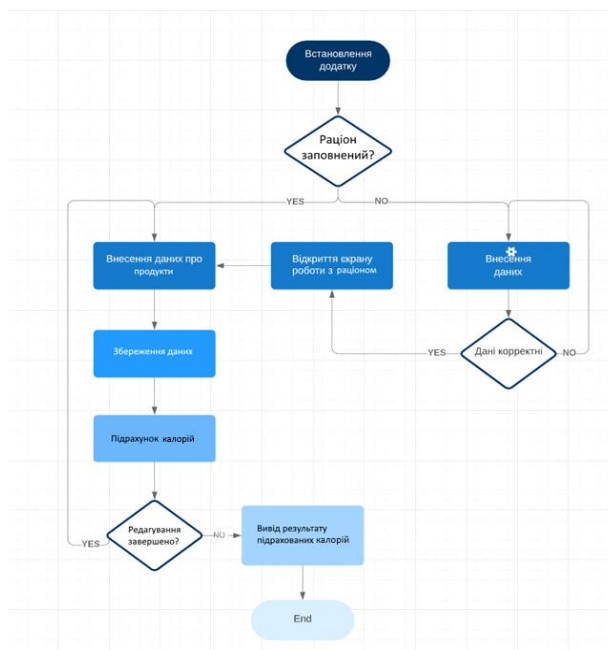


Рисунок 2.1 – Діаграма прецедентів з особливостями роботи сайту

2.2 Вибір засобів, методів і алгоритмів вирішення поставленого завдання

Застосування шаблону MVC на початку розробки структури програмного забезпечення забезпечить зручність розробки. Цей шаблон описує, як створюється структура програми. Його мета – відокремити бізнес-логіку від зазначеного інтерфейсу користувача. Це полегшує масштабування, тестування та, звичайно, випуск програми (рис. 2.2).

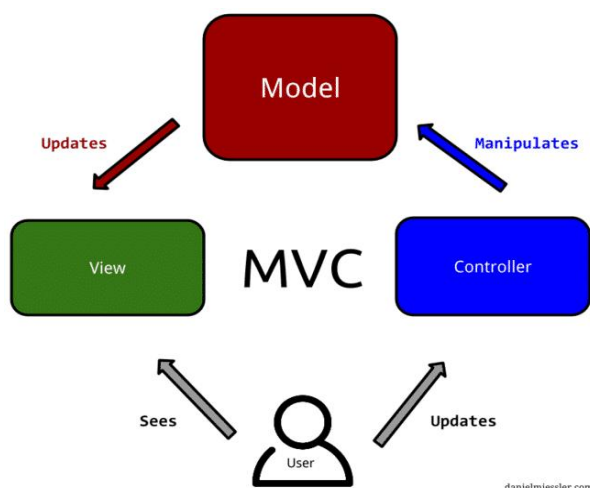


Рисунок 2.2 – MVC архітектура [7]

MVC характеризується схемою контролера представлення моделі. Кожен несе відповідальність за себе. У цій конфігурації модель MVC пропонує правила для даних і бізнес декомунізації, відповідає за інтерфейс взаємодії з користувачем і відповідає за процес взаємодії між моделлю і поданням.

Він відповідає за бізнес-логіку реалізації моделі. Він орієнтований на те, як вибирати, обробляти та надавати відповідні дані, і зазвичай робить його дуже товстим, що дуже нормально.

Модель не взаємодіє безпосередньо з користувачем. Всі операції зі зміни, налаштовані користувачем, повинні виконуватися контролером. Ця модель не генерує HTML-код або будь-який інший демонстраційний код, який був змінений відповідно до потреб користувача. Він відповідає за зовнішній перегляд даних подання та представлення завдань, отриманих від контролерів та моделей. Перегляд обробляє розмітку HTML та додавання певних PHP-кодів для сканування, форматування та відображення даних.

Подання зазвичай розбивається на загальний шаблон із загальною розміткою для всіх сторінок (наприклад, верхній або нижній колонтитули) або на будь-яку частину шаблону, яка може бути застосована для відображення даних шляхом виходу з моделі або відображення форми введення даних.

Контролер відповідає за обробку запитів від користувачів і відзначає логіку, необхідну для обробки запитів користувачів.

JavaScript – це динамічний об'єктно-орієнтований прототип мови програмування, заснованої на стандарті ECM3 [8]. Розробники спочатку створили JavaScript, щоб зробити веб-сторінку активною та «живою». Оскільки сьогодні веб-сторінка використовується при написанні сценаріїв, клієнт взаємодіє з браузером, процес обміну даними з сервером і процес зміни структури і зовнішнього вигляду веб-сторінки.

Коли з'явився AJAX, ситуація повернулася з зовсім іншого боку, і майбутні зміни відбулися відповідно до стандартів ES2015 та ES2017. З їх допомогою відбулася реструктуризація і з'явилися нові можливості для більш ефективного програмування. Ось чому JavaScript сьогодні настільки сильний,

що він бореться з проблемами за допомогою непереборного процесу впровадження, тестування та збоїв, тестування та налагодження, створення бібліотек та фреймворків та використання JavaScript поза браузером.

Цей синтаксис має деякі спільні риси з мовою Сі. Однак у нього є наступні відмінності:

- об'єкт імпортується та динамічно змінює свій тип за допомогою механізму прототипування;
- функціонує як об'єкти першого класу;
- лікування рідкісних випадків;
- тип самоповернення;
- автоматичне очищення;
- незалежна функція.

Реалізація веб-програми за допомогою JavaScript може відрізнитися через відповідний налагоджувач, який повинен бути у браузері. Тому особливо важливо, щоб ви мали повний доступ до набору інструментів налагодження. Сьогодні актуальні такі браузери, як Firefox, Opera, Google Chrome, Edge і Safari.

Програмний код називається сценарієм. Вони вже вбудовані та скомпільовані до моменту завантаження веб-сторінки.

Сценарії можуть бути легко розгорнуті і запущені у вигляді звичайного тексту. Вони не вимагають спеціальної підготовки або компіляції перед запуском. Це різниця декомунізації між JavaScript та іншою мовою, Java.

JavaScript – це сучасна «безпечна» мова програмування. JavaScript спочатку був створений для браузера, тому він не забезпечує низький рівень доступу до процесора або пам'яті. Здатність JavaScript працювати багато в чому залежить від середовища, в якому він обробляє дані.

Наприклад, можна використовувати JavaScript у своєму браузері:

- додайте новий HTML – код на сторінку або змініть готовий вміст і змініть стиль;
- редагуйте на дії користувача, клацання миші і переміщайте покажчики, натискання клавіш;

- встановіть мережеве з'єднання, відправляйте мережеві запити на віддалені сервери і завантажуйте файли (технології AJAX і COMET);
- отримуйте та завантажуйте файли cookie, задавайте питання відвідувачам та переглядайте повідомлення;
- зберігайте дані на стороні клієнта («локальне сховище»).

У мене є прототип, заснований на класифікації JavaScript. JavaScript також частково підтримує інші парадигми програмування. Наприклад, обов'язковий і частково функціональний. Він підтримує такі архітектурні функції, як динамічний і слабкий запис, автономне управління пам'яттю, емуляція прототипів і функціонування як об'єкти преміум-класу.

Мова JavaScript використовується для:

- створення сценаріїв для «живих» веб-сторінок;
- створення односторінкових веб-сайтів, таких як React, AngularJS та Vue.js;
- програмування на серверах, таких як вузли.js;
- електрон, КБ і т. д. програми, пов'язані з додатками.js;
- розробка мобільних додатків, таких як React Native і Cordova;
- сценарії прикладного програмного забезпечення;
- форматування тощо у документах PDF.

Вона має наступні характеристики:

- повна інтеграція з HTML/CSS;
- простота у використанні;
- доступність у будь-якому браузері та включена за замовчуванням.

Версія ES2016 функції є новою з додаванням операторів і послідовностей піднесення до ступеня. Прототип. Створено метод `includes`, який дозволяє перевірити, чи переданий аргумент знаходиться в масиві. А пізніше випущена версія ES2017 – це остання на сьогоднішній день версія стандарту, що розширює його функціональність, додає використання різних функцій, додає «зависання» параметрів функції, атомарні об'єкти та інші методи при роботі з рядками.

При розробці back-end функціоналу для фітнес додатка на мові програмування Python для схуднення важливо вибрати оптимальні технології, які забезпечать швидку та ефективну роботу системи, а також забезпечать безпеку та надійність обробки та збереження даних користувачів [9].

Одним з основних виборів є вибір фреймворку для розробки back-end частини додатка. У світі Python існує кілька популярних фреймворків, таких як Django, Flask, FastAPI та інші. Django є повноцінним фреймворком, який надає велику кількість готових модулів для різних завдань, включаючи аутентифікацію користувачів, роботу з базою даних, адміністративний інтерфейс, тощо. Flask, з іншого боку, є більш легким і гнучким фреймворком, що дозволяє розробляти back-end за допомогою більш простого та мінімалістичного підходу. FastAPI, знову ж таки, є новим фреймворком, який надає швидкий та продуктивний спосіб для створення API на Python, забезпечуючи високу швидкість роботи завдяки використанню асинхронного програмування. Вибір між цими фреймворками залежить від конкретних потреб проекту, а також від рівня складності та масштабованості системи.

Окрім вибору фреймворку, важливо визначити також базу даних, яка буде використовуватися для зберігання і обробки даних. У зв'язку з великим обсягом структурованої інформації про користувачів, їхні тренування та харчування, рекомендовано використовувати реляційну базу даних. MySQL, PostgreSQL та SQLite є популярними виборами для реляційних баз даних у світі Python. Іншим альтернативним варіантом може бути використання NoSQL бази даних, таких як MongoDB або Firebase Firestore, якщо система потребує великої гнучкості у структурі даних та можливості масштабування.

З урахуванням потреб проекту у високій швидкості та ефективності роботи, а також забезпечення безпеки даних, можна розглянути використання асинхронного програмування та інструментів, таких як Celery для асинхронної обробки завдань [10]. Це дозволить оптимізувати роботу back-end системи та забезпечити високу відповідність системи на великий обсяг одночасних запитів.

Важливим аспектом в розробці back-end є також забезпечення безпеки даних. Для цього можна використовувати різні методи, такі як хешування паролів, використання SSL для захищеної передачі даних, обробка та валідація вхідних даних для запобігання SQL-ін'єкціям та іншим атакам.

Отже, вибір технологій розробки back-end функціоналу для фітнес додатка на мові програмування Python для схуднення залежить від потреб проекту, рівня складності та масштабованості системи, а також від критичної ваги швидкості, ефективності та безпеки обробки даних.

Вибір бази даних та моделювання її структури є критично важливим етапом у розробці back-end функціоналу фітнес додатка на мові програмування Python для схуднення. Під час цього етапу визначається не лише технічна архітектура системи, а й способи зберігання та організації даних, які використовуються для забезпечення функціональності додатку.

Почнемо з вибору бази даних. На сьогоднішній день на ринку існує велика кількість різних баз даних з різними характеристиками та можливостями. Серед найпопулярніших рішень можна виокремити реляційні бази даних, такі як MySQL, PostgreSQL, SQLite, а також NoSQL бази даних, такі як MongoDB, Firebase Firestore тощо. Вибір бази даних залежить від конкретних потреб проекту, включаючи обсяг даних, швидкодію, доступність, сумісність та інші фактори. Наприклад, реляційні бази даних часто використовуються в проектах, де потрібна структурованість даних та підтримка транзакцій, тоді як NoSQL бази даних можуть бути корисними для проектів, де потрібна гнучкість у зберіганні неструктурованих даних та масштабованість системи.

Після вибору бази даних наступним кроком є моделювання її структури. Це включає в себе проектування схеми бази даних, визначення таблиць, їхніх взаємозв'язків та атрибутів. На цьому етапі важливо врахувати всі потреби додатку та забезпечити оптимальну структуру бази даних, яка буде ефективною для зберігання та опрацювання даних.

При моделюванні структури бази даних для фітнес-додатка на мові програмування Python для схуднення, можливо, будуть використані такі

таблиці як «Користувачі», «Тренування», «Харчування», «Статистика», «Цілі» тощо. Кожна з цих таблиць може мати свої атрибути та взаємозв'язки з іншими таблицями для забезпечення повноцінної функціональності додатку. Наприклад, таблиця «Користувачі» може містити дані про користувачів, такі як ім'я, електронна пошта, пароль тощо, а таблиця «Тренування» може містити дані про тренування користувачів, такі як дата, тип тренування, тривалість тощо. Крім того, важливо правильно визначити взаємозв'язки між таблицями, наприклад, зв'язок «один до багатьох» між таблицею «Користувачі» та «Тренування», де кожен користувач може мати багато тренувань, але кожне тренування належить лише одному користувачеві.

У результаті, вибір бази даних та моделювання її структури для back-end функціоналу фітнес-додатка на мові програмування Python для схуднення є ключовим етапом у розробці проекту. Від правильного вибору технологій та оптимальної структури бази даних залежить швидкодія, масштабованість та ефективність системи.

Висновки до розділу 2

У другому розділі нашої роботи було проведено проектування системи фітнес-додатка на мові програмування Python для схуднення. Ми ретельно розглянули вимоги до системи, її структуру та архітектуру, а також здійснили обґрунтування вибору технологій для розробки back-end та front-end функціоналу.

Формування та аналіз вимог до додатку про підрахунок калорій на Cordova є критично важливим етапом у розробці, який визначає функціональність та зручність використання додатку. Врахування цих вимог допоможе створити ефективний інструмент для контролю харчування, що сприятиме досягненню здорового способу життя користувачів.

У цьому розділі ми проаналізували методи розробки, вибрали та обґрунтували технології для створення майбутніх додатків. Це дозволяє

визначити потреби, функціональність, недоліки та отримати уявлення про структуровану логічну та фізичну систему. Дано архітектурний опис використання обраної моделі. Завдяки отриманим даним про майбутні проекти можна приступити до реалізації програми.

У результаті проведеного аналізу було сформульовано чіткі вимоги до функціональності додатка, зокрема, планування та відстеження тренувань, моніторинг харчування, встановлення особистих цілей, а також створення системи статистики та аналітики для відображення прогресу користувачів.

Щоб почати розробку функціональної частини мобільного додатка, необхідно поліпшити його структуру. Визначте основні частини інтерфейсу, їх розташування та приблизне розташування щодо інших функціональних частин. Добре продумана структура інтерфейсу важлива для зручності використання програми.

На головному екрані документ містить список днів. Їжа дня буде видно після відкриття дня. Також буде опублікована форма для введення користувачем. Після введення інформації калорії будуть підраховані (рис. 3.2-3.4).



Рисунок 3.2 – Структура продуктів на планшеті



Рисунок 3.3 – Структура додатку на телефоні

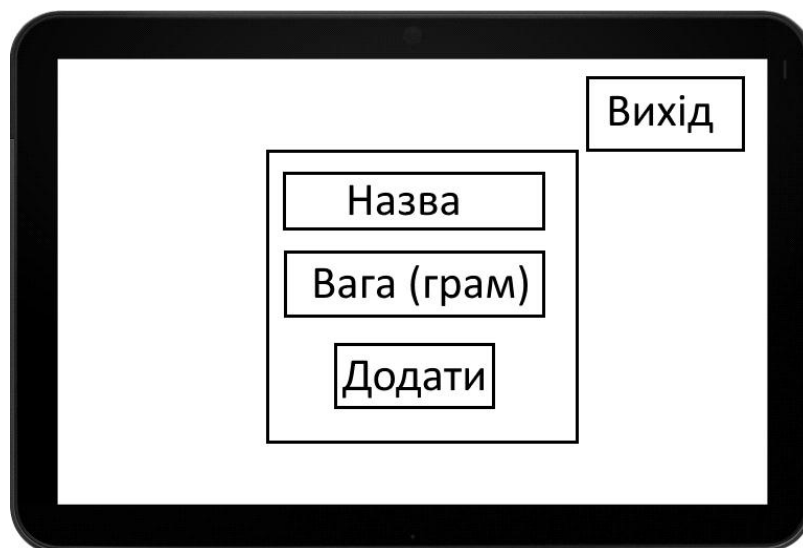


Рисунок 3.4 – Структура товару

Другий екран містить в собі головний функціонал для вибору списку продуктів. На ньому розміщений список з продуктів на цей день, також знаходиться список категорій і зручний пошук. При натисканні на категорію користувач бачить список продуктів і може додати новий.

Прототип дизайну включає реалізовані веб-сторінки з тестовим вмістом, створені за допомогою гіпертекстової розмітки. Ці сторінки будуть використовуватися для відображення інформації. Головна сторінка містить список днів для вибору раціону на день (рис. 3.5-3.6).

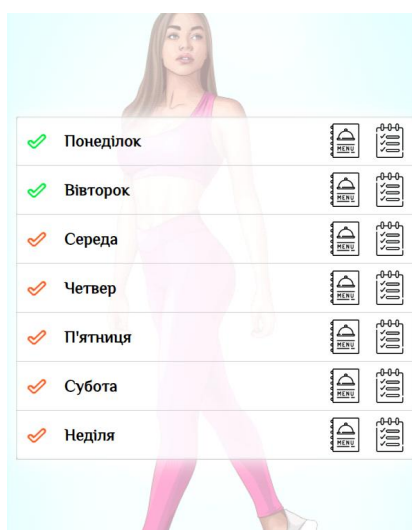


Рисунок 3.5 – HTML прототип головного екрану додатку

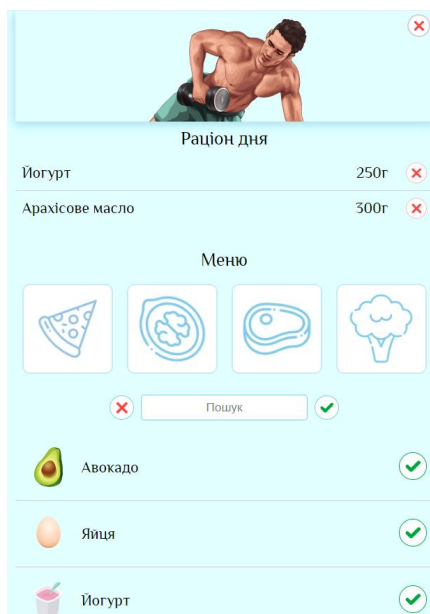


Рисунок 3.6 – HTML прототип екрану продуктів

3.2 Тестування та налагодження інформаційно-комп'ютерної системи

Встановивши додаток і відкривши його спочатку користувач побачить початковий екран в якому йому буде запропоновано список днів для встановлення добового раціону (рис. 3.7-3.8).

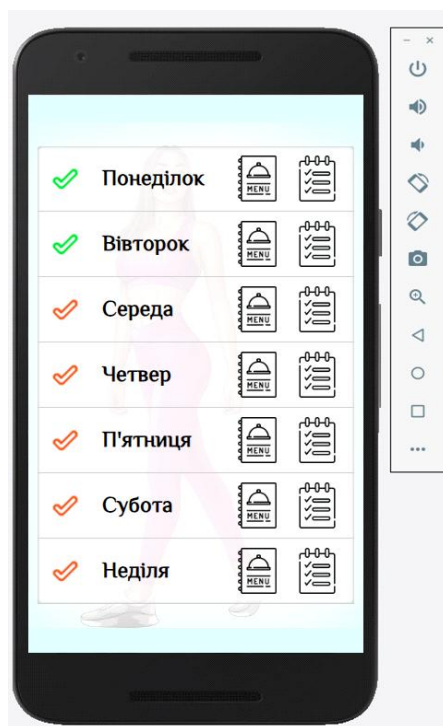


Рисунок 3.7 – Початок роботи з додатком

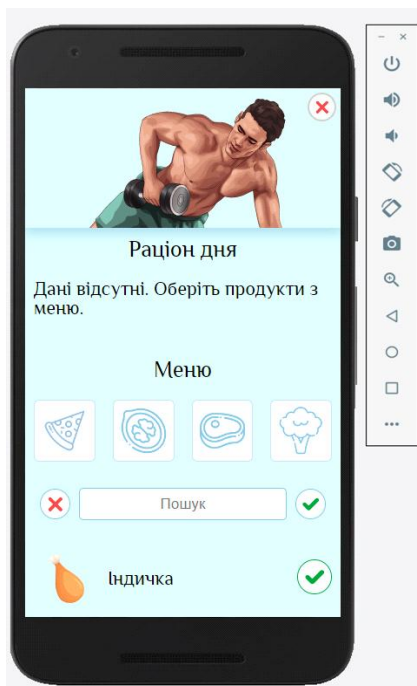


Рисунок 3.8 – Вибір раціону

Натиснувши на бажаний день користувач побачить екран для створення раціону. У нього є можливість пошуку і вибору необхідного продукту. Після вибору бажаного продукту користувач бачить форму для внесення інформації про добове вживання продукту (рис. 3.9-3.10).

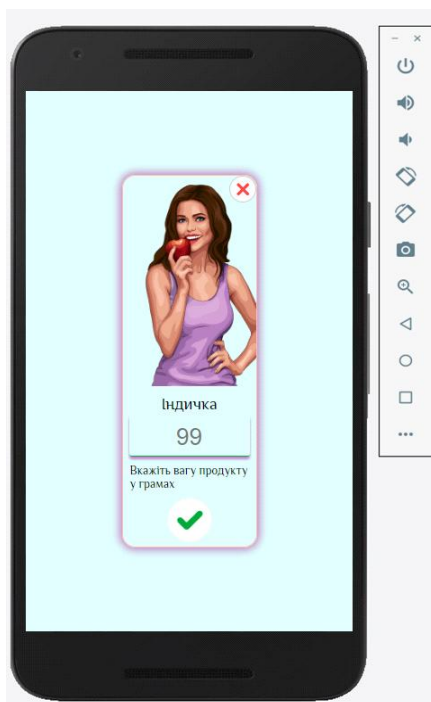


Рисунок 3.9 – Внесення ваги продукту

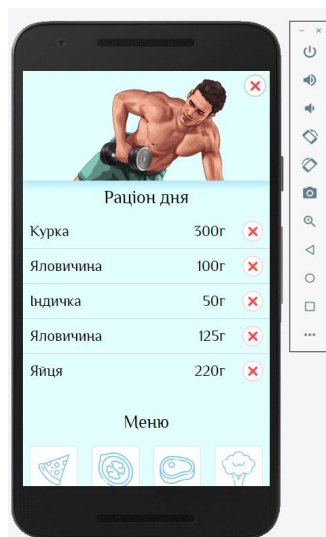


Рисунок 3.10 – Екран з раціоном дня

Всі внесені дані зберігаються і після натискання кнопки «додати» прораховуються відповідно до даних про корисність продукту. Користувач має змогу видалити продукти які внесені помилково або вже не вживаються.

Після всіх маніпуляцій з продуктами і раціоном на день на головній сторінці додатку день з внесеними даними змінює колір на зелений. Після натискання кнопки «перегляду» – іконки в правій частині екрану напроти кожного дня буде показано кількість спожитих вітамінів при такому раціоні (рис. 3.11).

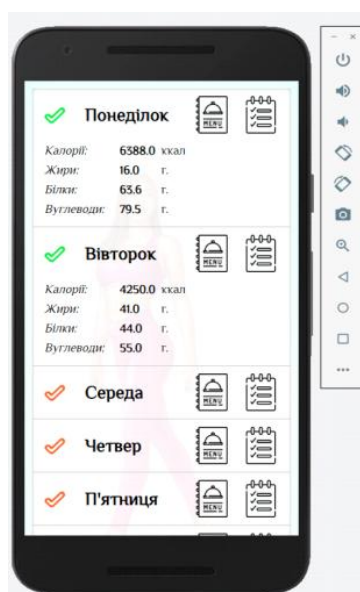


Рисунок 3.11 – Споживання вітамінів

Також користувач може скористатися «пошуком» для швидкого знаходження продуктів (рис. 3.12).

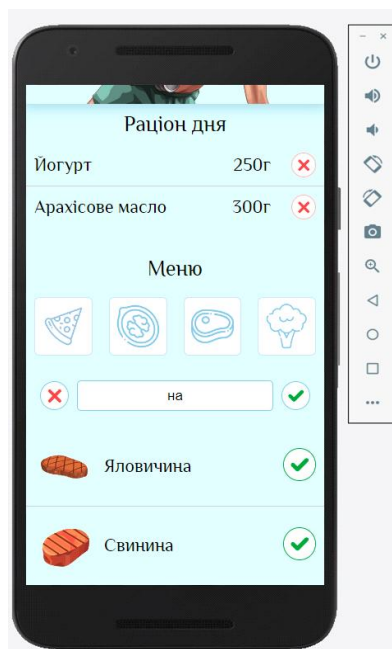


Рисунок 3.12 – Екран пошуку по продуктам

Тестування та валідація програмного забезпечення є критичними етапами в розробці фітнес-додатка на мові програмування Python для схуднення. Ці процеси спрямовані на перевірку та підтвердження відповідності розробленого програмного забезпечення вимогам, специфікаціям та очікуванням. Вони також допомагають виявити помилки та дефекти, що можуть виникнути в процесі роботи додатка, і забезпечити його стабільність, надійність та високу якість [11].

Першим кроком у тестуванні та валідації програмного забезпечення є розробка плану тестування. Цей план визначає стратегію тестування, обсяг та призначення кожного виду тесту, ресурси, необхідні для його виконання, а також критерії прийняття для визначення успішності тестування. Наприклад, план тестування може включати функціональне тестування, що перевіряє відповідність функціональності додатку вимогам, інтеграційне тестування, що перевіряє правильність взаємодії між компонентами системи; та навантажувальне тестування, що визначає максимальне навантаження, яке система може витримати без втрати продуктивності.

Після розробки плану тестування проводиться реалізація тестів. Цей етап включає в себе написання тестових сценаріїв та скриптів, розробку тестових наборів даних, налаштування тестових середовищ та інструментів, необхідних для виконання тестів. Наприклад, для функціонального тестування можуть бути розроблені автоматизовані тестові сценарії, які перевіряють роботу кожного модуля додатку, а для навантажувального тестування можуть бути використані спеціальні інструменти для створення навантаження на систему та моніторингу її реакції на цей навантаження.

Після виконання тестів здійснюється аналіз результатів. Цей етап включає в себе перевірку відповідності отриманих результатів очікуваним, виявлення та реєстрацію помилок та дефектів, а також прийняття рішення про необхідність внесення змін або виправлення в програмне забезпечення. Наприклад, якщо під час функціонального тестування було виявлено, що деякі функції додатку не працюють правильно, розробники повинні виправити ці помилки та перевірити їхню правильність під час повторного тестування.

Загальний висновок полягає в тому, що тестування та валідація програмного забезпечення є важливим етапом в розробці фітнес-додатка, який допомагає забезпечити його стабільність, надійність та високу якість. Виконання цих процесів дозволяє виявити та виправити помилки та дефекти ще на ранніх етапах розробки, що в свою чергу забезпечує успішну експлуатацію та задоволення потреб користувачів.

3.3 Розробка бази даних

Мова програмування Python є практичною у використанні, що обумовлено п'ятьма важливими критеріями для роботи:

- традиційністю;
- простотою;
- ефективністю;
- безпекою;

- гнучкістю.

Додаткова ще одна «характеристика», яка додає Python особливої привабливості: він поширюється безкоштовно, причому з відкритими початковими кодами (Open Source).

Python має спільний синтаксис з мовами C або Pascal, які часто зустрічаються у класичних програмах. Це знижує зусилля при початковому знайомстві та вивченні Python. Python – мова, яка успадкувала позитивні аспекти від Perl і C, та була спеціально розроблена для роботи з чітким синтаксисом і універсальною мовою.

Python – мова програмування, яка вже зайняла високе місце серед програмістів. Тому наразі є однією з найпопулярніших мов для розробки та створення веб-додатків.

Python інтегрується прямо в HTML – код сторінок, які оброблюються правильно Python-інтерпретатором. Можна легко залучати Python для формування HTML-документів, позбувшись викликів зовнішніх сценаріїв.

Широкий набір готових функцій дозволяє використовувати готові скрипти базових призначень функцій на різних мовах.

Мова програмування Python забезпечує гнучкі та ефективні методи безпеки, які можна поділити на дві групи:

- засоби системного рівня;
- засоби рівня додатку.

Уже готовий механізм безпеки в Python керується адміністраторами, що при коректному налаштуванні дозволяє максимально забезпечити безпеку даних. Python може перейти на роботу у безпечний режим, який корегує можливості роботи користувачами щодо важливих критеріїв, наприклад, за допомогою скорочення максимального часу виконання та використання пам'яті. Разом з ним можна використовувати базу даних.

База даних MySQL має свої переваги:

- база даних, яка дозволяє відкривати 1 логічний сервер бази даних на будь-якій кількості серверів або віртуальних машин;

– HTTP та JSON використовують протокол HTTP та дані JSON types.It він може працювати з будь-яким програмним забезпеченням. Він чудово працює із зовнішніми інструментами, включаючи проксі-сервери HTTP, і дозволяє налаштувати відповідне навантаження;

– початкова синхронізація даних в автономному режимі;

– хороша репутація. База даних вважається надійною і безпроблемною. Деякі вузли реалізують відмовостійкі структури даних, засновані лише на агрегуванні.

Обрана архітектура дає можливість комфортно розвивати логічну архітектуру проекту і можливість розвивати проект в майбутньому. Вибрана база даних із побудованою архітектурою також виконує червону лінію на малюнку нижче, що вказує на хорошу продуктивність бази даних (рис. 3.13).

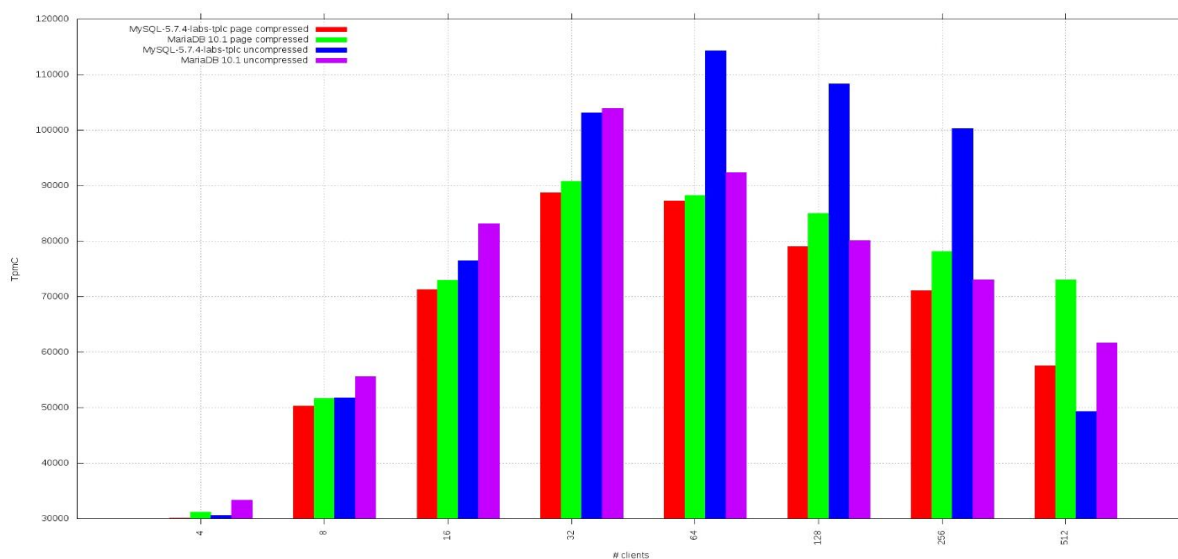


Рисунок 3.13 – Швидкодія бази даних

Швидкість роботи проекту залежить від ресурсів сервера на якому він розміщений.

При однакових умовах обрані технології показують гарний результат швидкості виконання (рис. 3.14).

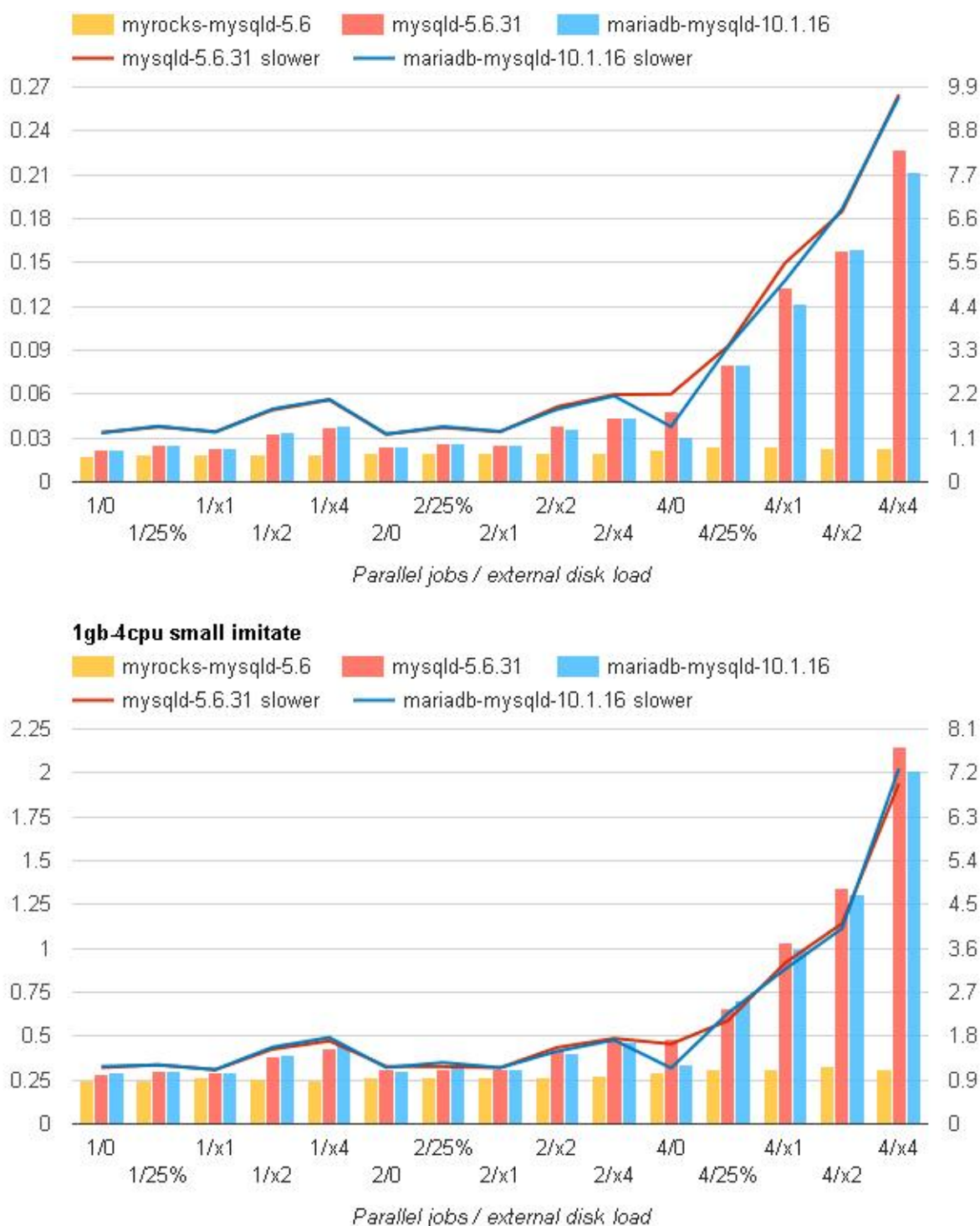


Рисунок 3.14 – Швидкодія бази даних

Обрана база даних відповідає вимогам швидкої роботи з даними. Щоб розпочати розробку мобільного додатка, потрібно створити прототип дизайну на основі приблизної структури програми.

По-перше, розробляється приблизний зовнішній вигляд і інтерфейс, створюється прототип зовнішнього вигляду, і такий дизайн.

Потім на основі розробленого подання створюється розмітка html і css у вигляді веб-сторінки. Кожен розділ створюється шляхом об'єднання даних та зображень html та css для розміщення елементів взаємодії.

При роботі з готовими додатками ці шаблони відображають динамічні дані, отримані від користувача.

Програма, яка розробляється, має файл конфігурації, і потрібно внести певні зміни до цього файлу, щоб компілятор міг створити певну тестову функцію:

- встановлення мінімальних і максимальних значень версії Android, в якій запущено додаток;
- доступ до зовнішніх веб-ресурсів (через сервер);
- можливість відкриття телефонної книги за посиланням;
- запит доступу до геолокації;
- відрегулюйте орієнтацію програми;
- встановіть додаток для відображення в повноекранному режимі;
- налаштування значків.

Після виконання цих налаштувань можна керувати основними функціями телефону в середовищі розробки. Після налаштування відображення сторінки, створеної додатком, дані, які ви хочете зберегти, спочатку надсилаються через форму. Додаток використовує перегляд веб-сторінок для відображення сторінки в браузері. Технологія також має доступ до файлової системи зберігання, але оскільки обсяг даних, які потрібно зберігати в додатку, дуже малий, localStorage можна використовувати як локальний пристрій зберігання даних.

Наступний екран програми містить форму для додавання даних. Створення нових даних, редагування та вилучення старих даних та видалення інформації з бази даних здійснюється за допомогою класів сценаріїв. Клас сценарію відповідає за певний набір функцій і методів, описаних тут, а клас контролера надсилає дані з локальної бази даних.

Нижче наведена одна з функцій для обробки даних. Він перераховує вагу введеного користувача і розраховує кількість грамів вітамінів відповідно до таблиці сумісності продукту (ліст. 3.1).

Лістинг 3.1 – Скрипт функції для обробки даних

```

$( '.weight-add-js' ).on("click", function(e) {
    // Отримуємо значення введеної ваги
    let val = $( '.get-weight-js' ).val(),
    // Розраховуємо відсоток введеної ваги
    percent = val / 100,
    // Створюємо порожній об'єкт
    obj = {};
    // Додаємо властивості до об'єкта obj
    obj.weight = val;
    obj.data = [];
    obj.name = nameSelectedSave;
    // Перебираємо масив та обчислюємо нові значення
    products[posSelectSave].forEach(function (item) {
        obj.data.push(item * percent);
    });
    // Додаємо об'єкт до відповідного масиву
    days[daySelectSave].push(obj);
    // Приховуємо елементи з класом 'weight' та
показуємо елементи з класом 'menu'
    $( '.weight' ).hide();
    $( '.menu' ).show();
    // Очищуємо значення поля вводу
    $( '.get-weight-js' ).val('');

    // Зберігаємо змінну days у локальному сховищі браузера
    localStorage.setItem('days', JSON.stringify(days));
    // Викликаємо функцію для оновлення меню
    setMenu();
});

```

Кінець лістингу 3.1

3.4 Розробка інсталяційного пакета

Розроблений додаток на технології Cordova, реалізовану на мові Java. Базові класи забезпечують функціональність всього додатку та взаємодію з сервером, все це розроблено у взаємопов'язаній системі (рис. 3.15).

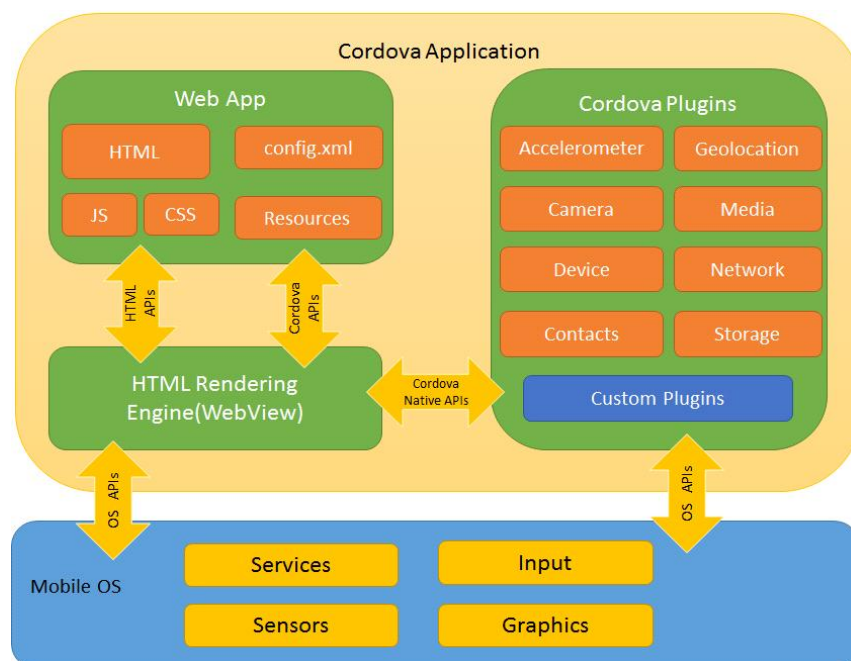


Рисунок 3.15 – Архітектура взаємодії елементів Cordova [12]

Основні методи та змінні класу ядра Cordova, що використовуються у програмі, наведено у таблиці 3.1.

Таблиця 3.1 – Список наявних робочих методів

Назва	Призначення
CordovaSystemWebViewEngine	
CordovaWebView	Надає представлення WebView
getCookieManager	Реалізовує об'єкт для зручної роботи з куками
initWebViewSettings	Ініціалізує WebView з параметрами гаджета
enableRemoteDebugging	Виводити логи помилок
loadUrl	Завантажує адресу url у WebView
stopLoading	Зупиняє завантаження вказаного url
CordovaSystemWebViewClient	
TargetApi	Визначення поточної версії Api пристрою
shouldOverrideUrlLoading	Цей метод дає можливість перехопити цю навігацію і виконати спеціальні дії.
onPageFinished	Метод дає можливість виконати додаткові дії після завершення завантаження сторінки.

Продовження таблиці 3.1

Назва	Призначення
onReceivedError	Метод дає можливість обробити цю помилку та виконати відповідні дії.
onLoadResource	Метод викликається, для відстеження завантажених ресурсів або виконання певних дій під час їх завантаження
onReceivedSslError	Метод викликається, коли сторінка містить SSL-помилки.
onReceivedHttpError	Метод викликається, коли відбувається HTTP-помилка під час завантаження сторінки.
isCurrentlyLoading	Перевірка чи був завантажений плагін для роботи додатку

Висновок до розділу 3

У третьому розділі нашої роботи була описана розробка системи фітнес-додатка на мові програмування Python для схуднення. Ми детально проаналізували архітектуру додатку, структуру його інтерфейсу, описали реалізацію основних функцій та класів, а також представили прототип дизайну і описали процес початку роботи з веб-ресурсом.

У результаті цього розділу ми розробили докладну архітектуру додатку, яка враховує потреби користувачів та вимоги функціональності. Також ми визначили структуру інтерфейсу, яка забезпечить зручне та ефективне використання додатку користувачами різних категорій. Описана реалізація основних функцій та класів дозволяє нам крок за кроком приступити до розробки додатку на практиці, враховуючи всі необхідні деталі та особливості.

Прототип дизайну, який був розроблений у цьому розділі, демонструє загальну концепцію та вигляд майбутнього додатку, що дозволяє зрозуміти його функціональність та можливості. Опис початку роботи з веб-ресурсом дозволяє нам зрозуміти, яким чином додаток буде взаємодіяти з іншими компонентами та сервісами Інтернету.

ВИСНОВКИ

Завершивши роботу над розробкою фітнес-додатку на мові програмування Python для схуднення, ми можемо зробити декілька важливих висновків.

По-перше, в ході виконання роботи були досягнуті всі поставлені цілі і завдання. Розроблено та детально проаналізовано архітектуру, функціональність та інтерфейс додатку, що дозволяє користувачам планувати тренування, вести статистику та моніторити свій прогрес у сфері фітнесу.

Далі, важливо відзначити, що розробка даного додатку відповідає світовим тенденціям у сфері розвитку програмного забезпечення для здоров'я та фітнесу. За останні роки спостерігається зростання популярності мобільних додатків для здоров'я та фітнесу, які допомагають користувачам досягати своїх цілей у здоровому способі життя.

Метою даної роботи є розробка мобільного додатку для підрахунку калорій. У ході реалізації проекту було проведено детальний аналіз технологій, які можуть бути використані для створення такого додатку, і обрано найбільш підходящі з них. Розроблений додаток може бути встановлений на мобільні телефони та планшети, що працюють під управлінням п'ятої або вищої версії операційної системи Android.

Для розробки додатку було використано ряд технологій: Cordova, Python, MySQL, HTML, CSS та JavaScript. Також було використано середовище розробки Android Studio. Використання цих технологій забезпечило простоту розробки, а поширеність інструментів полегшила майбутню підтримку програмного забезпечення.

Результатом роботи став готовий і працюючий Android-додаток. Процес розробки відбувався за визначеними етапами. Детальний аналіз цільової області розробки додатку:

- вибір методології розробки додатку на основі проаналізованих даних;

- розробка структури інтерфейсу додатку;
- розробка додатку;
- налагодження взаємодії між компонентами додатку;
- тестування та модифікація/покращення.

Результатом роботи став готовий і працюючий Android-додаток. Процес розробки відбувався за визначеними етапами. Детальний аналіз цільової області розробки додатку:

- вибір методології розробки додатку на основі проаналізованих даних;
- розробка структури інтерфейсу додатку;
- розробка додатку;
- налагодження взаємодії між компонентами додатку;
- тестування та модифікація/покращення.

Розроблений додаток відповідає всім поставленим вимогам і готовий до використання для підрахунку калорій, що допоможе користувачам ефективно контролювати своє харчування та досягати своїх цілей у здоровому способі життя.

Наслідки даної роботи можуть бути широко застосовані у різних галузях. Наприклад, фітнес-додаток може бути використаний в спортивних клубах для планування тренувань або в медичних установах для контролю здоров'я пацієнтів.

У майбутньому можливість вдосконалення розробленого додатка полягає у розширенні функціоналу, вдосконаленні інтерфейсу та оптимізації продуктивності. Також можливе розширення платформи, наприклад, створення версій для інших мобільних платформ або розробка веб-версії додатка.

У цілому, дана робота є важливим внеском у сферу розробки програмного забезпечення для здоров'я та фітнесу. Вона показує, як сучасні технології можуть бути використані для поліпшення нашого фізичного здоров'я та підтримки активного способу життя.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Мобільний додаток це? що таке, види, типи та особливості – IT рейтинг UA. IT рейтинг України. URL: <https://it-rating.ua/mobilniy-dodatok-tse-scho-take-vidi-tipi-ta-osoblivosti> (дата звернення: 01.03.2024).
2. Дизайн мобільних додатків: етапи створення та поради | Wezom. URL: <https://wezom.com.ua/ua/blog/dizajn-mobilnyh-prilozhenij-pochemu-on-vazhen-i-gde-zakazat> (дата звернення: 03.03.2024).
3. Android | Do More With Google on Android Phones & Devices. Android | Do More With Google on Android Phones & Devices. URL: <https://www.android.com/> (дата звернення: 05.03.2024).
4. 8 кращих додатків для схуднення. URL: <https://fitcurves.org/ua/blog/8-prilozhenij-dlya-pohudeniya/> (дата звернення: 07.03.2024).
5. Розробка мобільних застосунків: з чого почати. Хабр. URL: <https://habr.com/ru/companies/vk/articles/179113/> (дата звернення: 13.03.2024).
6. Світ технологій: Мобільні програми, які допоможуть схуднути. URL: <https://marieclaire.ua/uk/beauty/4-mobilnyh-prilozhenij-kotorye-pomogut-pohudet> (дата звернення: 03.04.2024).
7. Deploying Deep Learning Algorithm On AWS Cloud Platform.pdf. SlideShare. URL: <https://www.slideshare.net/slideshow/deploying-deep-learning-algorithm-on-aws-cloud-platformpdf/251701277> (дата звернення: 07.04.2024).
8. Сучасний підручник з JavaScript. Сучасний підручник з JavaScript. URL: <https://uk.javascript.info/> (дата звернення: 11.04.2024).
9. Створення мобільного додатка на Python – Wezom. URL: <https://wezom.com.ua/ua/blog/sozдание-mobilnogo-prilozhenija-na-python> (дата звернення: 15.04.2024).
10. Introduction to Celery – Celery 5.4.0 documentation. Celery - Distributed Task Queue – Celery 5.4.0 documentation. URL: <https://docs.celeryq.dev/en/stable/getting-started/introduction.html> (дата звернення: 16.04.2024).

11. Основні етапи тестування мобільних додатків. ITVDN. URL: <https://itvdn.com/ua/blog/article/mob-test-blog> (дата звернення: 02.05.2024).

12. Ammattikorkeakoulut - Theseus. URL: https://www.theseus.fi/bitstream/handle/10024/110286/Mahato_Rakesh.pdf;jsessionid=1A21639F7E8D69E0499D8559239D46F1?sequence=1 (дата звернення: 11.05.2024).

ДОДАТКИ

Додаток А

Лістинг класу CordovaActivity

```
package org.apache.cordova;

import java.util.ArrayList;
import java.util.Locale;

import org.json.JSONException;
import org.json.JSONObject;

import android.app.Activity;
import android.app.AlertDialog;
import android.annotation.SuppressLint;
import android.content.DialogInterface;
import android.content.Intent;
import android.content.res.Configuration;
import android.graphics.Color;
import android.media.AudioManager;
import android.os.Build;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.view.ViewGroup;
import android.view.Window;
import android.view.WindowManager;
import android.webkit.WebViewClient;
import android.widget.FrameLayout;

/**
 * This class is the main Android activity that represents
the Cordova
```

* application. It should be extended by the user to load the specific

* html file that contains the application.

*

* As an example:

*

* <pre>

* package org.apache.cordova.examples;

*

* import android.os.Bundle;

* import org.apache.cordova.*;

*

* public class Example extends CordovaActivity {

* @Override

* public void onCreate(Bundle savedInstanceState) {

* super.onCreate(savedInstanceState);

* super.init();

* // Load your application

* loadUrl(launchUrl);

* }

* }

* </pre>

*

* Cordova xml configuration: Cordova uses a configuration file at

* res/xml/config.xml to specify its settings. See "The config.xml File"

* guide in cordova-docs at <http://cordova.apache.org/docs> for the documentation

* for the configuration. The use of the set*Property() methods is

* deprecated in favor of the config.xml file.

*

*/

public class CordovaActivity extends Activity {

 public static String TAG = "CordovaActivity";

```

// The webview for our app
protected CordovaWebView appView;

private static int ACTIVITY_STARTING = 0;
private static int ACTIVITY_RUNNING = 1;
private static int ACTIVITY_EXITING = 2;

// Keep app running when pause is received. (default = true)
// If true, then the JavaScript and native code continue to
run in the background
// when another application (activity) is started.
protected boolean keepRunning = true;

// Flag to keep immersive mode if set to fullscreen
protected boolean immersiveMode;

// Read from config.xml:
protected CordovaPreferences preferences;
protected String launchUrl;
protected ArrayList<PluginEntry> pluginEntries;
protected CordovaInterfaceImpl cordovaInterface;

/**
 * Construct the default web view object.
 * <p/>
 * Override this to customize the webview that is used.
 */
protected CordovaWebView makeWebView() {
    return new CordovaWebViewImpl(makeWebViewEngine());
}

protected CordovaWebViewEngine makeWebViewEngine() {
    return CordovaWebViewImpl.createEngine(this, preferences);
}

```

```

protected CordovaInterfaceImpl makeCordovaInterface() {
    return new CordovaInterfaceImpl(this) {
        @Override
        public Object onMessage(String id, Object data) {
            // Plumb this to CordovaActivity.onMessage for backwards
compatibility
            return CordovaActivity.this.onMessage(id, data);
        }
    };
}

/**
 * Load the url into the webview.
 */
public void loadUrl(String url) {
    if (appView == null) {
        init();
    }

    // If keepRunning
    this.keepRunning = preferences.getBoolean("KeepRunning",
true);

    appView.loadUrlIntoView(url, true);
}

/**
 * Called when the system is about to start resuming a
previous activity.
 */
@Override
protected void onPause() {
    super.onPause();
    LOG.d(TAG, "Paused the activity.");

    if (this.appView != null) {

```

```

        // CB-9382 If there is an activity that started for result
        and main activity is waiting for callback
        // result, we shouldn't stop WebView Javascript timers, as
        activity for result might be using them
        boolean keepRunning = this.keepRunning ||
this.cordovaInterface.activityResultCallback != null;
        this.appView.handlePause(keepRunning);
    }
}

/**
 * Called when the activity receives a new intent
 */
@Override
protected void onNewIntent(Intent intent) {
    super.onNewIntent(intent);
    //Forward to plugins
    if (this.appView != null)
        this.appView.onNewIntent(intent);
}

/**
 * Called when the activity will start interacting with the
user.
 */
@Override
protected void onResume() {
    super.onResume();
    LOG.d(TAG, "Resumed the activity.");

    if (this.appView == null) {
        return;
    }
    if (! this.getWindow().getDecorView().hasFocus()) {
        // Force window to have focus, so application always

```

```
// receive user input. Workaround for some devices (Samsung
Galaxy Note 3 at least)
this.getWindow().getDecorView().requestFocus();
}

this.appView.handleResume(this.keepRunning);
}

/**
 * Called when the activity is no longer visible to the user.
 */
@Override
protected void onStop() {
    super.onStop();
    LOG.d(TAG, "Stopped the activity.");

    if (this.appView == null) {
        return;
    }
    this.appView.handleStop();
}
```

Додаток Б

Лістинг класу CordovaPlugin

```
package org.apache.cordova;

import org.apache.cordova.CordovaArgs;
import org.apache.cordova.CordovaWebView;
import org.apache.cordova.CordovaInterface;
import org.apache.cordova.CallbackContext;
import org.json.JSONArray;
import org.json.JSONException;

import android.content.Intent;
import android.content.pm.PackageManager;
import android.content.res.Configuration;
import android.net.Uri;
import android.os.Build;
import android.os.Bundle;

import java.io.FileNotFoundException;
import java.io.IOException;

/**
 * Plugins must extend this class and override one of the
execute methods.
 */
public class CordovaPlugin {
    public CordovaWebView webView;
    public CordovaInterface cordova;
    protected CordovaPreferences preferences;
    private String serviceName;

    /**
     * Call this after constructing to initialize the plugin.
     * Final because we want to be able to change args without
breaking plugins.
```



```

    */

    public final void privateInitialize(String serviceName,
CordovaInterface cordova, CordovaWebView webView,
CordovaPreferences preferences) {
        assert this.cordova == null;
        this.serviceName = serviceName;
        this.cordova = cordova;
        this.webView = webView;
        this.preferences = preferences;
        initialize(cordova, webView);
        pluginInitialize();
    }

    /**
     * Called after plugin construction and fields have been
initialized.
     * Prefer to use pluginInitialize instead since there is no
value in
     * having parameters on the initialize() function.
    */
    public void initialize(CordovaInterface cordova,
CordovaWebView webView) {
    }

    /**
     * Called after plugin construction and fields have been
initialized.
    */
    protected void pluginInitialize() {
    }

    /**
     * Returns the plugin's service name (what you'd use when
calling pluginManger.getPlugin())
    */
    public String getServiceName() {

```

```

return serviceName;
}

/**
 * Executes the request.
 *
 * This method is called from the WebView thread. To do a
non-trivial amount of work, use:
 * cordova.getThreadPool().execute(runnable);
 *
 * To run on the UI thread, use:
 * cordova.getActivity().runOnUiThread(runnable);
 *
 * @param action The action to execute.
 * @param rawArgs The exec() arguments in JSON form.
 * @param callbackContext The callback context used when
calling back into JavaScript.
 * @return Whether the action was valid.
 */
public boolean execute(String action, String rawArgs,
CallbackContext callbackContext) throws JSONException {
    JSONArray args = new JSONArray(rawArgs);
    return execute(action, args, callbackContext);
}

/**
 * Executes the request.
 *
 * This method is called from the WebView thread. To do a
non-trivial amount of work, use:
 * cordova.getThreadPool().execute(runnable);
 *
 * To run on the UI thread, use:
 * cordova.getActivity().runOnUiThread(runnable);
 *
 * @param action The action to execute.

```

```

    * @param args The exec() arguments.
    * @param callbackContext The callback context used when
calling back into JavaScript.
    * @return Whether the action was valid.
    */

    public boolean execute(String action, JSONArray args,
CallbackContext callbackContext) throws JSONException {
        CordovaArgs cordovaArgs = new CordovaArgs(args);
        return execute(action, cordovaArgs, callbackContext);
    }

/**
 * Executes the request.
 *
 * This method is called from the WebView thread. To do a
non-trivial amount of work, use:
 * cordova.getThreadPool().execute(runnable);
 *
 * To run on the UI thread, use:
 * cordova.getActivity().runOnUiThread(runnable);
 *
 * @param action The action to execute.
 * @param args The exec() arguments, wrapped with some
Cordova helpers.
 * @param callbackContext The callback context used when
calling back into JavaScript.
 * @return Whether the action was valid.
 */

    public boolean execute(String action, CordovaArgs args,
CallbackContext callbackContext) throws JSONException {
        return false;
    }

/**
 * Called when the system is about to start resuming a
previous activity.

```

```

*
* @param multitasking      Flag indicating if multitasking
is turned on for app
*/
public void onPause(boolean multitasking) {
}

/**
* Called when the activity will start interacting with the
user.
*
* @param multitasking      Flag indicating if multitasking
is turned on for app
*/
public void onResume(boolean multitasking) {
}

/**
* Called when the activity is becoming visible to the user.
*/
public void onStart() {
}

/**
* Called when the activity is no longer visible to the user.
*/
public void onStop() {
}

/**
* Called when the activity receives a new intent.
*/
public void onNewIntent(Intent intent) {
}

/**

```

```

    * The final call you receive before your activity is
    destroyed.
    */
    public void onDestroy() {
    }

    /**
    * Called when the Activity is being destroyed (e.g. if a
    plugin calls out to an external
    * Activity and the OS kills the CordovaActivity in the
    background). The plugin should save its
    * state in this method only if it is awaiting the result of
    an external Activity and needs
    * to preserve some information so as to handle that result;
    onRestoreStateForActivityResult()
    * will only be called if the plugin is the recipient of an
    Activity result
    *
    * @return Bundle containing the state of the plugin or null
    if state does not need to be saved
    */
    public Bundle onSaveInstanceState() {
    return null;
    }

    /**
    * Called when a plugin is the recipient of an Activity
    result after the CordovaActivity has
    * been destroyed. The Bundle will be the same as the one the
    plugin returned in
    * onSaveInstanceState()
    *
    * @param state Bundle containing the state of the plugin
    * @param callbackContext Replacement Context to return the
    plugin result to
    */

```

```

    public void onRestoreStateForActivityResult(Bundle state,
        CallbackContext callbackContext) {}

    /**
     * Called when a message is sent to plugin.
     *
     * @param id The message id
     * @param data The message data
     * @return Object to stop propagation or null
     */
    public Object onMessage(String id, Object data) {
        return null;
    }

    /**
     * Called when an activity you launched exits, giving you the
     * requestCode you started it with,
     * the resultCode it returned, and any additional data from
     * it.
     *
     * @param requestCode The request code originally supplied to
     * startActivityResult(),
     * allowing you to identify who this result came from.
     * @param resultCode The integer result code returned by the
     * child activity through its setResult().
     * @param intent An Intent, which can return result data to
     * the caller (various data can be
     * attached to Intent "extras").
     */
    public void onActivityResult(int requestCode, int resultCode,
        Intent intent) {}

```