

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

Факультет інформатики та обчислювальної техніки

Кафедра інформаційних систем та технологій

«На правах рукопису»
УДК 004.45

До захисту допущено:
Завідувач кафедри
_____ Олександр РОЛІК
« » _____ 2024 р.

Магістерська дисертація

на здобуття ступеня магістра

за освітньо-професійною програмою

«Інформаційне забезпечення робототехнічних систем»

зі спеціальності 126 «Інформаційні системи та технології»

**на тему: «Інформаційна система для підтримки здорового способу
життя на основі Інтернету речей»**

Виконав:
студент 2 курсу, групи ІК-32мп
Кравець Микола Олександрович _____

Керівник:
доцент каф. ІСТ, к.т.н., доц.
Батрак Євгеній Олександрович _____

Рецензент:
зав. кафедри ІСТ ФІТ
КНУ ім. Тараса Шевченка, д.т.н., проф.
Дружинін Володимир Анатолійович _____

Засвідчую, що у цій магістерській
дисертації немає запозичень з праць
інших авторів без відповідних
посилань.
Студент _____

Київ – 2024 року

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра інформаційних систем та технологій

Рівень вищої освіти – другий (магістерський)

Спеціальність – 126 «Інформаційні системи та технології»

Освітньо-професійна програма «Інформаційне забезпечення
робототехнічних систем»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Олександр РОЛІК

«__» _____ 2024 р.

ЗАВДАННЯ
на магістерську дисертацію студенту
Кравцю Миколі Олександровичу

1. Тема дисертації «Інформаційна система для підтримки здорового способу життя на основі Інтернету речей», науковий керівник дисертації Батрак Євгеній Олександрович, к.т.н., доц., затверджені наказом по університету від «08» 11 2024 р. № 5016-с
2. Термін подання студентом дисертації «09» 12 2024 р.
3. Об'єкт дослідження: Інформаційні системи для підтримки здорового способу життя на основі Інтернету речей.
4. Вихідні дані: Мова програмування Java, СКБД SQLite, середовище програмування Android Studio, підтримка мобільних телефонів на базі Android, можливість використання Bluetooth та виявлення підключених пристроїв, можливість взаємодії з пристроями IoT.
5. Перелік завдань, які потрібно розробити: аналіз наявних рішень інформаційних систем для підтримки здорового способу життя на основі Інтернету речей, розробка алгоритмів, проектування та реалізація інформаційної системи підтримки здорового способу життя на основі Інтернету речей, проведення тестувань розробленої системи, написання пояснювальної записки.

6. Орієнтовний перелік графічного (ілюстративного) матеріалу: діаграма варіантів використання, структурна схема, ER-діаграма, діаграма послідовності, схема потоків даних, діаграма класів, BPMN діаграма, блок-схема рекомендації меню на основі інгредієнтів.

7. Орієнтовний перелік публікацій: Огляд автоматизованих систем охорони здоров'я.

8. Дата видачі завдання 02.09.2024 р.

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації	Примітка
1.	Ознайомлення із застосуваннями інформаційних систем догляду за здоров'ям на основі Інтернету речей	08.09.2024	
2.	Дослідження основних проблем при роботі з оцінкою стану здоров'я	15.09.2024	
3.	Огляд існуючих рішень для вирішення задачі	22.09.2024	
4.	Формування вимог та критерії до інформаційної системи	27.09.2024	
5.	Розробка архітектури інформаційної системи	06.10.2024	
6.	Вибір методу зчитування, обробки та зберігання даних, що вводяться користувачем та створюються системою	13.10.2024	
7.	Ознайомлення із принципами роботи у SQLite	13.10.2024	
8.	Розробка інформаційної системи	31.10.2024	
9.	Тестування та вдосконалення інформаційної системи	05.11.2024	
10.	Написання документації до розробленої системи	25.11.2024	

Студент

Микола КРАВЕЦЬ

Науковий керівник

Євгеній БАТРАК

РЕФЕРАТ

Інформаційна система для підтримки здорового способу життя на основі Інтернету речей: 105 с., 31 табл., 26 рис., 8 дод., 30 джерел.

ІНФОРМАЦІЙНА СИСТЕМА, ІНТЕРНЕТ РЕЧЕЙ, РЕКОМЕНДАЦІЯ МЕНЮ, ХАРЧУВАННЯ, BLUETOOTH, ПЕРСОНАЛІЗОВАНІ РЕКОМЕНДАЦІЇ, КАЛОРІЇ, ПУЛЬС, МОНІТОРИНГ ПОКАЗНИКІВ.

Сучасний світ стикається з численними викликами у сфері здоров'я, що спричинені малорухливим способом життя, незбалансованим харчуванням та стресовими умовами. Пошук ефективних рішень для підтримки здорового способу життя стає важливим завданням для інноваційної індустрії, особливо в контексті стрімкого розвитку технологій Інтернету речей (IoT). Інтеграція IoT у щоденну рутину відкриває нові можливості для персоналізованого моніторингу та управління фізичною активністю, що дозволяє не лише підвищувати рівень усвідомленості про стан власного здоров'я, але й сприяти досягненню конкретних цілей – схуднення, підтримання ваги чи покращення фізичної форми.

Метою роботи є розробка інформаційної системи, яка включатиме в себе всі необхідні процеси для підтримання здорового способу життя.

Для реалізації даної системи було виділено наступні завдання:

- проаналізувати існуючі програмні рішення даної задачі;
- сформулювати чіткі вимоги до функціоналу інформаційної системи;
- дослідити можливості IoT для збирання та обробки даних;
- спроектувати архітектуру системи, що забезпечить інтеграцію пристроїв IoT;
- розробити алгоритми персоналізованого аналізу та рекомендацій;
- протестувати систему для оцінки її ефективності.

Об'єкт дослідження – інформаційні системи для підтримки здорового способу життя на основі Інтернету речей.

Предмет дослідження – технології для розробки інформаційних систем розрахунку оцінки стану здоров'я з використанням технологій Інтернету речей.

ABSTRACT

Internet of Things-based information system for supporting a healthy lifestyle:
105 p., 31 tab., 26 draw., 8 app., 30 sources.

INFORMATION SYSTEM, INTERNET OF THINGS, MENU RECOMMENDATION, NUTRITION, BLUETOOTH, PERSONALIZED RECOMMENDATIONS, CALORIES, PULSE, INDICATORS MONITORING.

The modern world faces numerous health challenges caused by a sedentary lifestyle, unbalanced nutrition and stressful conditions. Finding effective solutions to support a healthy lifestyle is becoming an important task for the innovation industry, especially in the context of the rapid development of Internet of Things (IoT) technologies. The integration of IoT into daily routines opens up new opportunities for personalized monitoring and management of physical activity, which allows not only to increase awareness of one's own health, but also to contribute to the achievement of specific goals – losing weight, maintaining weight or improving physical fitness.

The purpose of the work is to develop an information system that will include all the necessary processes for maintaining a healthy lifestyle.

The following tasks were identified for the implementation of this system:

- analyze existing software solutions for this task;
- formulate clear requirements for the functionality of the information system;
- explore the capabilities of IoT for data collection and processing;
- design a system architecture that will ensure the integration of IoT devices;
- develop algorithms for personalized analysis and recommendations;
- test the system to assess its effectiveness.

The object of the study is Internet of Things-based information systems for supporting a healthy lifestyle.

The subject of the study is technologies for developing information systems for calculating health assessment using Internet of Things technologies.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАЧЕНЬ.....	8
ВСТУП.....	10
1 ОПИС ПРЕДМЕТНОЇ ОБЛАСТІ	13
1.1 Загальний огляд об'єкта дослідження.....	13
1.2 Обґрунтування доцільності розробки	14
1.3 Аналіз функціональних особливостей системи	15
1.4 Огляд існуючих рішень	16
1.4.1 Інформаційна система «Calorie Counter by Lose It!»	16
1.4.2 Інформаційна система «Calorie Counter App: Fooducate»	17
1.4.3 Інформаційна система «My diet meal plan»	17
1.4.4 Інформаційна система «SuperCook».....	18
1.5 Постановка задачі.....	19
1.5.1 Функціональні вимоги до системи	19
1.5.2 Нефункціональні вимоги до системи	20
Висновки до розділу 1	21
2 АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ ПОБУДОВИ ІНФОРМАЦІЙНИХ СИСТЕМ... ..	22
2.1 Загальні вимоги до розроблювального прототипу	22
2.2 Аналіз проблем прототипу та їх рішень	22
2.2.1 Монолітна архітектура.....	23
2.2.2 Архітектура клієнт-сервер.....	25
2.2.3 Архітектура MVC	27
2.2.4 Архітектура MVP	29
2.2.5 Архітектура MVVM.....	31
2.2.6 Архітектура MVI	33
2.2.7 Чиста архітектура	35
2.2.8 Мікросервісна архітектура	36
2.2.9 Порівняльна характеристика архітектур	37
2.3 Вибір та обґрунтування технічних рішень	40

2.4 Вибір та обґрунтування можливих варіантів технічної реалізації.....	42
Висновки до розділу 2	45
3 ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ.....	47
3.1 Вхідні дані.....	47
3.2 Математичні формули для розрахунків.....	48
3.3 Вихідні дані.....	52
3.4 Структура бази даних	52
Висновки до розділу 3	56
4 РЕАЛІЗАЦІЯ СИСТЕМИ.....	57
4.1 Опис апаратної бази та середовища розробки	57
4.2 Підходи до розробки клієнтського інтерфейсу.....	58
4.3 Опис інтерфейсу користувача.....	59
Висновок до розділу 4.....	75
5 РОЗРОБЛЕННЯ СТАРТАП-ПРОЄКТУ	76
5.1 Опис ідеї проєкту	76
5.2 Технологічний аудит ідеї проєкту.....	80
5.3 Аналіз ринкових можливостей запуску стартап-проєкту.....	81
5.4 Розроблення ринкової стратегії проєкту.....	92
5.5 Розроблення маркетингової програми стартап-проєкту	94
Висновки до розділу 5	98
ВИСНОВКИ.....	100
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	103
ДОДАТОК А.....	106
ДОДАТОК Б	107
ДОДАТОК В.....	108
ДОДАТОК Г	109
ДОДАТОК Д.....	110
ДОДАТОК Е	111
ДОДАТОК Ж.....	112
ДОДАТОК З.....	113

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАЧЕНЬ

БД – база даних.

СУБД – система управління базою даних.

AI – Artificial Intelligence – розділ комп'ютерної лінгвістики й інформатики, зосереджений на розробці інтелектуальних машин, здатних виконувати завдання, які зазвичай потребують людського інтелекту.

API – Application Programming Interface – це спосіб завдяки якому дві або більше комп'ютерні програми можуть спілкуватися між собою.

BLE – Bluetooth Low Energy – Bluetooth з низьким енергоспоживанням

BMI – Body Mass Index – величина, що дозволяє оцінити ступінь відповідності маси людини до її зросту, й тим самим, непрямо оцінити, чи є маса недостатньою, нормальною, надмірною

BMR – Basal Metabolic Rate – мінімальна кількість енергії (калорій), що витрачається людським організмом для підтримки власного життя в спокої.

BPMN – Business Process Model and Notation – система умовних позначень (нотація) для моделювання бізнес-процесів.

CRUD – Create Read Update Delete – 4 основні функції управління даними «створення, читання, оновлення і вилучення».

CPU – Central Processing Unit – функціональна частина комп'ютера, що призначена для інтерпретації команд.

HTTP – HyperText Transfer Protocol – протокол передачі даних, що використовується в комп'ютерних мережах.

IoT – Internet of Things – концепція мережі, що складається із взаємозв'язаних фізичних пристроїв, які мають вбудовані засоби та технології для взаємодії один з одним або із зовнішнім середовищем.

MVI – Model-View-Intent – Модель-Вид-Намір – шаблон проектування, що застосовується під час проектування архітектури застосунків.

MVC – Model-View-Controller – Модель-Вид-Контролер – шаблон проектування, що застосовується під час проектування архітектури застосунків.

MVP – Model-View-Presenter – Модель-Вид-Презентер – шаблон проектування, що застосовується під час проектування архітектури застосунків.

MVVM – Model-View-Viewmodel – Модель-Вигляд-Виглядмодель – шаблон проектування, що застосовується під час проектування архітектури застосунків.

SQL – Structured Query Language – мова написання запитів, за допомогою якої можна взаємодіяти із базами даних.

SWOT – Strengths Weaknesses Opportunities Threats – метод аналізу, що полягає у виявленні факторів внутрішнього та зовнішнього середовища організації та поділу їх на чотири категорії.

TDEE – Total Daily Energy Expenditure – це кількість калорій, які людина спалює за день, враховуючи фізичну активність.

UI – User Interface – засіб зручної взаємодії користувача з інформаційною системою.

Wi-Fi – Wireless Fidelity – технологія бездротової локальної мережі із пристроями на основі стандартів IEEE 802.11.

ВСТУП

У сучасному світі здоров'я та спосіб життя стали важливими пріоритетами для більшості людей. Поширення різноманітних захворювань, зокрема серцево-судинних і метаболічних розладів, пов'язаних з малорухливим способом життя, неправильним харчуванням та великою кількістю стресу, змушує людей звертати більше уваги на свій спосіб життя. Однак традиційні підходи до підтримки здоров'я вимагають значних ресурсів і часто не враховують індивідуальних потреб людей.

Сучасний розвиток інформаційних технологій, зокрема Інтернету речей (IoT), відкриває нові можливості для вирішення цих завдань. Він дозволяє створювати систему пристроїв, які можуть збирати, аналізувати та передавати дані в реальному часі. Це робить можливим персоналізований підхід до моніторингу здоров'я та спрощує процес збору та обробки необхідних даних для формування рекомендацій щодо підтримання здорового способу життя.

Розумні пристрої, такі як фітнес-трекери, розумні годинники, сенсори для моніторингу різних показників, стають дедалі популярнішими, що вказує на високий інтерес до технологій, які сприяють підтримці здоров'я. Додатковою мотивацією для розробки таких систем є вплив глобальних викликів, пов'язаних із пандеміями та необхідністю швидкої адаптації до змін в умовах життя. Ці чинники стимулюють розвиток інформаційних систем, які полегшують аналіз поточного стану здоров'я та сприяють підтримці здорового способу життя.

Однак зараз існує велика кількість інформаційних систем, що спеціалізуються лише на певних аспектах догляду за здоров'ям. Хоч вони й полегшують процес догляду за здоров'ям, але потрібно використовувати різні інформаційні системи для комплексного моніторингу показників здоров'я. Використання декількох систем потребують виділення великої кількості часу, що в умовах сучасного життя не завжди є можливим.

Таким чином, дослідження в області створення інформаційної системи для підтримки здорового способу життя на основі Інтернету речей є актуальним як із

соціальної, так і з науково-технічної точки зору. Ця тема відповідає сучасним тенденціям розвитку технологій, спрямованих на підвищення якості життя, та є важливою складовою забезпечення суспільного здоров'я.

У магістерській дисертації буде розглянута дана проблема та створена інформаційна система підтримки здорового способу життя на основі Інтернету речей. Ця система базується на сучасних технологіях і використовує Bluetooth з низьким енергоспоживанням (BLE) для взаємодії з пристроями Інтернету речей.

Мета дослідження полягає в розробці інформаційної системи для підтримки здорового способу життя на основі Інтернету речей, яка дозволяє ефективно здійснювати персоналізований моніторинг здоров'я, аналізувати отримані дані та формувати рекомендації для підтримки здорового способу життя.

Для досягнення цієї мети були визначені такі завдання:

- проаналізувати існуючі програмні рішення даної задачі;
- сформулювати чіткі вимоги до функціоналу інформаційної системи;
- дослідити можливості IoT для збирання та обробки даних;
- спроектувати архітектуру системи, що забезпечить інтеграцію пристроїв IoT;
- розробити алгоритми персоналізованого аналізу та рекомендацій;
- протестувати систему для оцінки її ефективності.

Об'єктом дослідження є інформаційні системи, призначені для підтримки здорового способу життя на основі Інтернету речей.

Предметом дослідження є технології, які використовуються для створення інформаційних систем, що здійснюють оцінку стану здоров'я на основі даних, зібраних за допомогою пристроїв Інтернету речей. Це включає методи збору, обробки та аналізу даних, а також розробку алгоритмів для формування рекомендацій.

Розроблена інформаційна система має високе практичне значення, оскільки сприяє підвищенню обізнаності користувачів про їхній фізичний стан, допомагає автоматизувати процеси збору та обробки даних і надає персоналізовані рекомендації для підтримки здоров'я. Її можна використовувати у повсякденному

житті для зручного та швидкого моніторингу стану здоров'я та підтримання здорового способу життя. Така система також може знижувати ризики розвитку хронічних захворювань шляхом відображення статистики пульсу та інших показників, розрахованих системою та зчитаних з пристроїв Інтернету речей, що, у свою чергу, сприятиме зменшенню навантаження на систему охорони здоров'я.

Нижче наведено структуру магістерської дисертації, яка складається з наступних розділів: вступ, опис предметної області, аналіз існуючих рішень побудови інформаційних систем, інформаційне забезпечення, реалізація системи, розроблення стартап-проєкту, висновків, списку джерел, та 8 додатків. Графічна частина включає 8 креслеників формату А3.

Крім того, було опубліковано статтю, яка досліджує переваги та недоліки існуючих рішень, а також алгоритми розрахунку показників якості здоров'я[1].

1 ОПИС ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Загальний огляд об'єкта дослідження

З ростом інтересу суспільства до підтримки здорового способу життя з'являється необхідність впровадження нових технологічних рішень для підтримання здорового способу життя та надання рекомендацій щодо його покращення. Одним із таких рішень є розробка інформаційних систем для підтримки здорового способу життя на основі Інтернету речей (IoT), які забезпечують збір, аналіз і використання даних для досягнення індивідуальних цілей користувачів. Основною метою таких систем є інтеграція технологій IoT для створення єдиної екосистеми, яка дозволяє автоматизовано та персоналізовано контролювати фізичний стан користувачів і сприяти зміцненню їхнього здоров'я.

Існуючі інформаційні системи для підтримки здорового способу життя мають ряд недоліків. Зокрема, вони часто обмежені в можливостях інтеграції різноманітних IoT-пристроїв, не забезпечують високого рівня персоналізації, а також можуть мати складний інтерфейс, що ускладнює їх використання для користувачів. Ще однією проблемою є недостатня обробка та аналіз даних у режимі реального часу, через що користувачі не отримують актуальних рекомендацій. Крім того, багато існуючих рішень орієнтовані лише на окремі аспекти здоров'я, що не дозволяє забезпечити комплексний підхід.

Розробка сучасної інформаційної системи на основі Інтернету речей дозволить не лише усунути зазначені недоліки, а й створити ефективну платформу для збирання та аналізу фізіологічних і поведінкових даних. Така система забезпечить користувачів персоналізованими рекомендаціями та полегшить підтримку здорового способу життя. Вона також дозволить інтегрувати різні IoT-пристрої та технології для забезпечення зручного й безперервного моніторингу здоров'я. Це сприятиме популяризації здорового способу життя, покращенню якості життя населення та зменшенню навантаження на систему охорони здоров'я.

1.2 Обґрунтування доцільності розробки

У сучасному суспільстві, де спостерігається зростання попиту на технології, спрямовані на підтримку здорового способу життя, виникає необхідність створення інформаційної системи, яка б дозволяла здійснювати комплексний моніторинг фізичного стану користувачів і забезпечувала їх персоналізованими рекомендаціями. Використання Інтернету речей (IoT) у такій системі відкриває нові можливості для ефективного збирання та аналізу даних, що робить дану розробку актуальною та корисною для широкого кола користувачів.

Розроблювана інформаційна система допоможе людям слідкувати за своїм здоров'ям у зручній та інтерактивній формі. Завдяки інтеграції IoT-пристроїв, таких як розумні годинники чи фітнес-трекери, система зможе автоматично збирати необхідні дані, такі як частота серцевих скорочень, рівень активності та кількість спалених калорій, що значно пришвидшить збір та обробку даних. Інтуїтивно зрозумілий інтерфейс забезпечить легкий доступ до отриманих показників, а алгоритми персоналізації нададуть рекомендації щодо меню, яке допоможе отримувати необхідну кількість калорій.

Інформаційна система для підтримки здорового способу життя на основі Інтернету речей має значний потенціал у підтримці правильного харчування та дієти. Завдяки інтеграції з IoT-пристроєм та впровадженню алгоритмів і розрахунків, система дозволить користувачам здійснювати моніторинг споживаних калорій, макро- і мікроелементів, а також порівнювати їх з рекомендаціями щодо здорового харчування. Користувачі зможуть отримувати персоналізовані поради щодо складу раціону, що сприятиме досягненню їхніх цілей, таких як зниження ваги, покращення енергійності чи підтримка загального здоров'я. Завдяки таким можливостям система допоможе ефективно керувати дієтою і забезпечить користувачів необхідною інформацією для прийняття обґрунтованих рішень про їхнє харчування.

Система також сприятиме популяризації здорового способу життя серед населення, особливо в умовах, коли фізична активність та контроль за станом

здоров'я є важливими елементами профілактики багатьох хронічних захворювань. Використання новітніх технологій IoT значно зменшить затрати часу на збір та аналіз даних, що зробить систему доступною, інтерактивною та ефективною.

Таким чином, розробка інформаційної системи для підтримки здорового способу життя на основі Інтернету речей відповідає сучасним потребам суспільства. Вона полегшить процес моніторингу здоров'я, зробить його доступним і зручним для користувачів, сприятиме покращенню якості життя та зменшенню навантаження на систему охорони здоров'я. Також вона допоможе заощадити час обробки даних, який користувач зможе використати на безпосереднє покращення способу життя.

1.3 Аналіз функціональних особливостей системи

Інформаційна система для підтримки здорового способу життя на основі Інтернету речей буде орієнтована на кінцевого користувача – людину, яка прагне відслідковувати свій фізичний стан і покращити спосіб життя. Система дозволить користувачам інтегрувати різноманітні пристрої, такі як фітнес-трекери та смарт-годинники, для збору даних про їхній рівень активності. Ці дані будуть використовуватися для аналізу поточного способу життя користувача та надання йому рекомендацій щодо харчування.

Користувач, маючи доступ до системи, зможе вносити і переглядати свої особисті дані, зокрема інформацію про дієту, показники здоров'я (пульс, кількість спалених калорій, пройденої відстань). За допомогою інтуїтивно зрозумілого інтерфейсу система автоматично збирає дані з пристроїв, надаючи персоналізовані поради для поліпшення фізичного стану, з урахуванням поточних показників та цілей користувача.

Користувачі зможуть отримувати інформативні звіти про свій прогрес, зокрема про кількість витрачених калорій, пройденої кроки та відстань. Система також пропонуватиме рекомендації щодо корекції раціону в залежності від цілі користувача. Всі ці дані будуть зібрані в одному місці та зручно представлені у

вигляді графіків, що дозволить користувачам з легкістю відслідковувати їхній прогрес.

Для зручності використання система також повинна мати фільтри для налаштування інтерфейсу, що дозволяє користувачам швидко знаходити потрібну інформацію. Усі функції системи повинні бути адаптовані під інтерфейс мобільних пристроїв для забезпечення доступності та зручності в будь-яких умовах. До того ж, дані повинні зберігатися в базі даних для швидко і зручного їх аналізу.

Інтуїтивно зрозумілий дизайн інтерфейсу та можливість автоматичного збору та обробки даних дозволять користувачам без зайвих зусиль відслідковувати своє здоров'я, отримувати персоналізовані рекомендації і підтримувати здоровий спосіб життя на щоденній основі.

1.4 Огляд існуючих рішень

Серед найбільш популярних рішень, що виконують функцію відслідковування стану здоров'я та підтримки здорового способу життя було виділено наступні інформаційні системи: «Calorie Counter by Lose It!», «Calorie Counter App: Fooducate», «My diet meal plan», «SuperCook». Більш детально про переваги та недоліки даних систем можна прочитати в статті[1].

1.4.1 Інформаційна система «Calorie Counter by Lose It!»

Calorie Counter by Lose It! – це популярний мобільний застосунок, створений для допомоги користувачам у відстеженні їхнього харчування та контролю за калоріями[2]. Він орієнтований на людей, які прагнуть контролювати свою вагу та покращити здоров'я. Lose It! дозволяє відстежувати калорії, макроелементи та інші важливі показники.

Lose It! – це інструмент для підрахунку калорій, відстеження харчування та періодичного голодування, який допоможе вам відмовитися від небажаної

поведінки, яка заважає вам досягти цільової ваги. Завдяки простоті використання, багатій базі даних і можливості синхронізації з іншими платформами, Lose It! допомагає зробити процес відстеження харчування легким і зручним.

Основні переваги для майбутнього проекту:

- розрахунок кількості спожитих калорій відповідно до прийому їжі (сніданок, обід, вечеря);
- отримання короткої інформацію про кількість спожитих корисних речовин.

1.4.2 Інформаційна система «Calorie Counter App: Fooducate»

Fooducate – це мобільний застосунок, який сприяє здоровому харчуванню за допомогою простого відстеження їжі та аналізу її складу[3]. Застосунок також дозволяє відстежувати прийоми їжі і кількість спожитих калорій. Користувачі можуть фіксувати, що саме вони їдять, а система автоматично підраховує калорії, білки, жири та вуглеводи.

Основні переваги для майбутнього проекту:

- отримання числових показників необхідної кількості поживних речовин;
- отримання розрахунку індексу маси тіла на основі введених даних.

1.4.3 Інформаційна система «My diet meal plan»

My Diet Meal Plan – це онлайн-платформа, яка допомагає користувачам створювати персоналізовані плани харчування для досягнення різних цілей, таких як схуднення, підтримка здоров'я або набір м'язової маси[4]. Сайт пропонує зручний інтерфейс для складання меню на день, тиждень або місяць, враховуючи індивідуальні вподобання та деякі дієтичні обмеження. Користувачі можуть вибирати з великої бази рецептів або створювати власні страви, що дозволяє легко підтримувати збалансовану і різноманітну дієту.

Однією з основних особливостей сайту є функція відстеження калорій та макроелементів (білки, жири, вуглеводи), що дозволяє користувачам стежити за їх споживанням та адаптувати дієту відповідно до своїх потреб.

Основні переваги для майбутнього проекту:

- перегляд рецептів страв;
- можливість обирання наявних продуктів.

1.4.4 Інформаційна система «SuperCook»

SuperCook – це онлайн-платформа, яка допомагає користувачам знаходити рецепти на основі інгредієнтів, які є у них вдома[5]. Це зручний інструмент для тих, хто хоче готувати, не витрачаючи час на покупки додаткових продуктів або шукаючи рецепти з певними інгредієнтами. Користувачі можуть вводити або вибирати інгредієнти, які є в їхньому холодильнику або на полицях, і SuperCook автоматично генерує рецепти, що підходять для цих продуктів.

Сайт має величезну базу даних рецептів, що охоплює різноманітні кухні світу та стилі харчування, від веганських і безглютенових страв до класичних домашніх рецептів. Користувачі можуть налаштовувати пошук за різними параметрами, такими як час приготування, складність страви, калорійність або тип дієти. Це дозволяє знайти оптимальні варіанти для будь-якої ситуації – від швидкого обіду до складної вечері.

SuperCook також допомагає з економією часу та ресурсів, оскільки користувачі можуть планувати свої страви, базуючись лише на тих інгредієнтах, які вже є вдома. Це знижує ймовірність харчових відходів та дозволяє з легкістю використовувати всі наявні продукти.

Основні переваги для майбутнього проекту:

- рекомендація рецептів на основі наявних продуктів;
- взаємодія з пристроями Інтернету речей;
- незалежність від підключення до мережі Інтернет.

1.5 Постановка задачі

Дослідивши переваги існуючих аналогів інформаційних систем, можна сформувати набір вимог, які дозволять створити зручну та інтуїтивно зрозумілу інформаційну систему для підтримки здорового способу життя на основі Інтернету речей.

1.5.1 Функціональні вимоги до системи

Для досягнення мети, було сформовано ряд функціональних вимог, які переймуть переваги кожного з аналогів та виокремить розроблювану систему за рахунок новизни:

- система має бути розрахована на використання людьми, які хочуть підтримувати здоровий спосіб життя;
- можливість створення облікового запису користувача;
- авторизація за допомогою логіна та пароля;
- надання користувачу можливості перегляду, редагування та видалення власних даних;
- підтримка підключення пристроїв Інтернету речей через Bluetooth;
- можливість автоматичного зчитування даних з пристроїв Інтернету речей;
- наявність адаптивний та інтуїтивно зрозумілого інтерфейсу;
- автоматичний розрахунок щоденної потреби в калоріях, білках, жирах, вуглеводах та воді на основі введених параметрів, таких як вік, стать, висота, вага, рівень активності;
- відображення результатів розрахунків та основних показників стану здоров'я, таких як пульс, кількість кроків, спожиті калорії у зручному для користувача форматі;
- збереження історії даних для аналізу змін у часі;
- користувач повинен мати можливість вводити наявні та заборонені інгредієнти;

- автоматична генерація меню на день на основі розрахунків та введених користувачем продуктів або особистих уподобань;
- рекомендовані меню повинні надавати можливість перегляду рецептів страв;
- відображення калорійності, білків, жирів та вуглеводів у кожній страві;
- система повинна розраховувати кількість спожитих калорій на основі обраних страв та інгредієнтів;
- можливість роботи без підключення до Інтернету;
- введення даних має бути валідованим.

1.5.2 Нефункціональні вимоги до системи

Крім функціональних вимог, потрібно також звернути увагу на нефункціональні вимоги, які можна застосувати до розроблюваної інформаційної системи.

Нижче наведено список нефункціональних вимог, які повинна задовольняти система:

- система повинна обробляти дані у режимі реального часу з мінімальною затримкою (не більше 1–2 секунд);
- система має бути спроектована з урахуванням можливості збільшення кількості користувачів та підключених пристроїв Інтернету речей без зниження продуктивності;
- архітектура повинна дозволяти легке додавання нових функцій або інтеграцію з іншими платформами та сервісами;
- система повинна забезпечувати безперервну роботу без помилок навіть у разі відсутності підключення до Інтернету;
- забезпечення мінімального часу відновлення роботи у разі збою (не більше 5 хвилин);
- інтерфейс системи повинен бути інтуїтивно зрозумілим, адаптованим до користувачів із різним рівнем технічної грамотності;

– мобільний застосунок повинен мінімізувати використання батареї пристрою навіть при активному використанні (не більше 10% заряду за 1 годину активної роботи);

– формат одиниць вимірювання повинен бути адаптований під користувачів з різних територій.

Висновки до розділу 1

У цьому розділі було обґрунтовано розробку інформаційної системи для підтримки здорового способу життя на основі Інтернету речей.

У рамках розділу було визначено основну мету та завдання, які планується вирішувати за допомогою створюваної інформаційної системи, а також розглянуто її функціональні характеристики. Проведено аналіз існуючих аналогів, спрямованих на моніторинг стану здоров'я та дотримання здорового способу життя, що дозволило виокремити їхні сильні й слабкі сторони. Отримані результати стали основою для формування ключових вимог до системи.

Головною метою розробки є створення системи, яка дозволить користувачам відстежувати показники свого здоров'я та оптимізувати харчування. Завдяки інтеграції з пристроями Інтернету речей система забезпечить автоматизований збір і обробку даних про фізичний стан користувачів, включаючи показники активності, серцевий ритм та кількість пройдених кроків.

На основі проведеного аналізу було визначено ключові функції майбутньої системи. Вони включають інтеграцію з IoT-пристроями, відображення результатів роботи системи та даних користувача, формування персоналізованих рекомендацій та створення простого і зручного інтерфейсу. Ці можливості покликані забезпечити ефективне використання технологій для підтримки здорового способу життя, а також задовольнити потреби користувачів.

2 АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ ПОБУДОВИ ІНФОРМАЦІЙНИХ СИСТЕМ

2.1 Загальні вимоги до розроблювального прототипу

Щоб вирішити проблему підтримання здорового способу життя, планується розробити інформаційну систему на основі Інтернету речей, яка буде збирати дані про фізичний стан користувача і обробляти їх. Основною метою системи є обробка даних користувача та надання рекомендацій, які допоможуть користувачу слідкувати за своїм здоров'ям.

Розробка прототипу передбачає використання надійних обчислювальних ресурсів для забезпечення стабільної роботи системи, а також створення відповідного операційного середовища. Система має бути налаштована для підтримки безперервного обміну даними між пристроями Інтернету речей та центральною системою. Це дозволить досягти високої продуктивності, швидкості обробки інформації та відповідності сучасним вимогам до подібних систем.

Інформаційна система повинна бути розроблена для пристроїв Android, що дозволить використовувати її в будь-який час та в будь-якому місці. Також система повинна мати доступ до налаштувань Bluetooth та не повинна залежати від Інтернет зв'язку. Такий підхід сприятиме максимальному комфорту та широкому охопленню цільової аудиторії.

Крім того, важливо забезпечити сумісність з популярними IoT-пристроями для моніторингу здоров'я, такими як фітнес-трекери та розумні годинники, та надавати можливість зберігання отриманих даних. Це необхідно для відстеження динаміки змін стану здоров'я користувача і формування індивідуальних рекомендацій для підтримки його здоров'я.

2.2 Аналіз проблем прототипу та їх рішень

Існує кілька архітектурних підходів, які широко використовуються при розробці програмного забезпечення, і кожен з них має свої специфічні переваги та

недоліки, в залежності від типу системи, її вимог та потреб користувачів. Вибір оптимальної архітектури для конкретного проєкту вимагає врахування різноманітних факторів, таких як тип взаємодії з користувачем, складність обробки даних, масштабованість і тестованість програми.

З урахуванням того, що інформаційна система буде реалізована для пристроїв на базі Android, існує декілька основних архітектур розроблення даних систем:

- монолітна архітектура;
- архітектура клієнт-сервер;
- архітектура MVC (Model-View-Controller);
- архітектура MVP (Model-View-Presenter);
- архітектура MVVM (Model-View-ViewModel);
- архітектура MVI (Model-View-Intent);
- чиста архітектура (Clean Architecture);
- мікросервісна архітектура.

Перед вибором оптимальної архітектури для розроблюваної інформаційної системи, дослідимо кожен з наведених архітектур.

2.2.1 Монолітна архітектура

Монолітна архітектура – це традиційна модель програмного забезпечення, яка побудована як уніфікована одиниця, самодостатня та незалежна від інших програм. [6]. Дана архітектура зображена на рисунку 2.1.

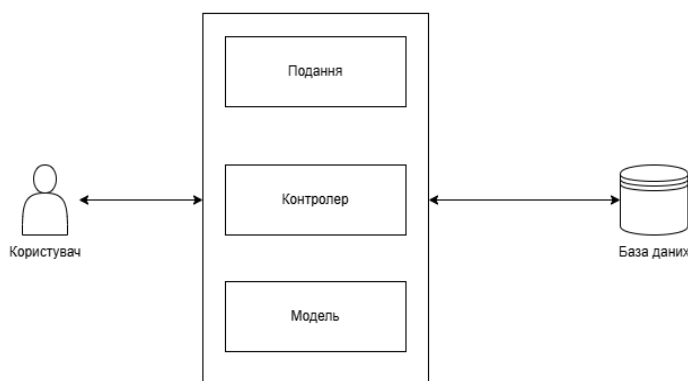


Рисунок 2.1 – Монолітна архітектура

У межах монолітної архітектури код і функціональність, які відповідають за користувацький інтерфейс, бізнес-логіку та обробку даних, звичайно розміщуються в одному пакеті або програмі. Це означає, що всі компоненти системи функціонують разом і взаємодіють між собою без чіткої відокремленості або інтерфейсів для взаємодії між частинами.

Монолітна архітектура найбільше підходить для проєктів, які є відносно невеликими за обсягом і не вимагають високої масштабованості або складної інфраструктури. Це може бути хорошим вибором для прототипів або продуктів на початкових етапах розвитку, коли швидкість виходу на ринок важливіша за потенційну складність у майбутньому[7]. Якщо система передбачає невеликий набір функцій і не має високих вимог до масштабованості, монолітна архітектура може бути зручним і ефективним підходом.

Однією з основних переваг монолітної архітектури є простота розробки та розгортання. Оскільки весь код зберігається в одному місці, розробникам не потрібно вирішувати питання інтеграції та взаємодії між різними мікросервісами або модулями. Це може значно зменшити час на розробку та тестування, оскільки все знаходиться в одній кодовій базі, і зібрати або запустити систему можна з мінімальними зусиллями.

Монолітна архітектура також дозволяє легко управляти даними і станами програми, оскільки всі компоненти мають доступ до спільного простору пам'яті та бази даних. Це може бути зручним для менших проєктів або для тих, хто тільки розпочинає розробку інформаційної системи, оскільки дозволяє уникнути складностей, пов'язаних із розподіленою системою.

Однак монолітна архітектура має і свої недоліки. Усі компоненти в монолітній архітектурі дуже взаємозалежні, тому впроваджувати будь-які нові функції або змінювати систему з таким високим рівнем зв'язку складно. Будь-які зміни в коді вплинуть на всю систему, що робить практично неможливим експериментувати з новими функціями[8].

Ще однією проблемою монолітної архітектури є складність масштабування. Оскільки система складається з єдиного коду, масштабування одного з його

аспектів (наприклад, лише бізнес-логіки) може вимагати масштабування всієї системи, навіть якщо інші частини не потребують таких змін. Це може бути економічно неефективним і привести до надмірного використання ресурсів.

2.2.2 Архітектура клієнт-сервер

Архітектура клієнт-сервер є однією з найпоширеніших моделей для розробки інформаційних систем, яка передбачає розподіл системи на дві основні частини: клієнтську і серверну[9]. На рисунку 2.2 зображено приклад архітектури клієнт-сервер.



Рисунок 2.2 – Архітектура клієнт-сервер

Клієнт-серверна архітектура зазвичай складається з таких основних компонентів[10]:

- клієнти, такі як персональні комп'ютери, робочі станції або інші розумні пристрої, які надсилають запити на сервери, а потім отримують інформацію;
- сервери – великі пристрої, які використовуються для зберігання й обробки файлів, баз даних і програм;
- мережеві пристрої, які з'єднують робочі станції та сервери та дозволяють їм спілкуватися та ефективно надсилати інформацію.

У цьому підході клієнтська частина інформаційної системи, тобто застосунок, взаємодіє з сервером через мережу для отримання необхідних даних

або виконання певних операцій. Основна перевага такої архітектури полягає в тому, що більша частина обробки даних і логіки відбувається на сервері, що дозволяє пристрою зберігати ресурси та забезпечує централізовану обробку і зберігання інформації.

Архітектура клієнт-сервер є надзвичайно популярною для розробки інформаційних систем, які потребують централізованої обробки даних. Наприклад, такі системи як соціальні мережі, банківські застосунки або онлайн-магазини часто використовують цю архітектуру для взаємодії з користувачами. Клієнт взаємодіє з сервером для отримання даних про акаунти, транзакції, товари або новини, а сервер обробляє ці запити, зберігає дані в базі даних і забезпечує логіку роботи системи.

Одна з основних переваг цієї архітектури – централізоване управління та контроль над усіма даними. Це дає можливість зручно управляти доступом до інформації, захищати дані за допомогою централізованих засобів безпеки, а також зберігати інформацію на сервері. Така організація дозволяє швидко оновлювати систему, адже зміни можуть бути внесені лише на сервері, без необхідності оновлювати клієнтські програми.

Іншою перевагою є можливість масштабування. Клієнт-серверна архітектура дозволяє збільшувати потужності серверів залежно від навантаження, що важливо для підтримки великої кількості одночасних користувачів. Сервер може бути налаштований для обробки великої кількості запитів, а також для балансування навантаження між кількома серверами, що забезпечує високу продуктивність навіть при значних навантаженнях.

Головним недоліком моделі клієнт-сервер є небезпека перевантаження системи через брак ресурсів для обслуговування всіх клієнтів. Якщо занадто багато різних клієнтів намагаються підключитися до спільної мережі одночасно, з'єднання може бути невдалим або сповільненим[11].

Крім того, у клієнт-серверних мережах вартість налаштування та обслуговування сервера зазвичай вища, ніж вартість роботи мережі.

2.2.3 Архітектура MVC

Архітектура MVC (Model-View-Controller) є одним із найбільш популярних і широко використовуваних підходів у розробці інформаційних систем[12]. Дана архітектура зображена на рисунку 2.3.

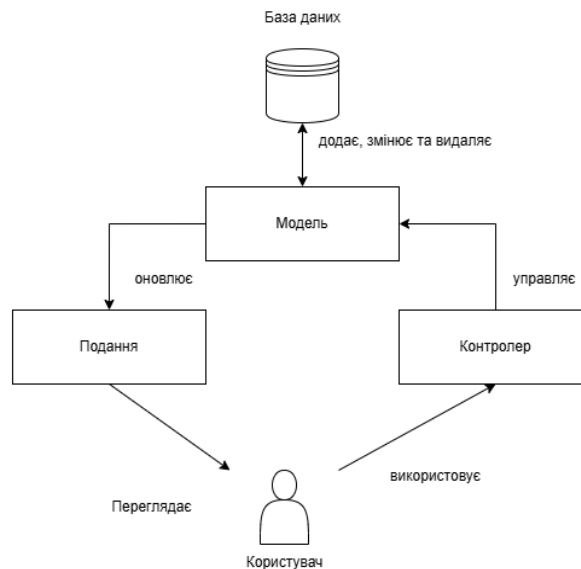


Рисунок 2.3 – Архітектура MVC

Вона передбачає розподіл програми на три основні компоненти: Модель (Model), Подання (View) та Контролер (Controller). Така архітектура дозволяє ефективно організувати взаємодію між даними, користувацьким інтерфейсом і бізнес-логікою, сприяючи чистому та підтримуваному коду.

Модель представляє логіку даних і дані, які зберігаються у вашій програмі. Модель зазвичай зберігає інформацію про те, що сталося в одній або кількох подіях, таких як введення користувача або запити до бази даних. Модель також забезпечує спосіб доступу до даних з інтерфейсу користувача. Це включає збереження введених користувачем даних, отримання даних із бази даних та виконання інших завдань, необхідних для функціональності вашої програми.

Подання – це компонент, який відповідає за відображення даних користувачу. Воно бере дані, надані моделлю, і відображає їх на екрані таким чином, щоб користувачі могли з ними взаємодіяти. Важливо відзначити, що подання не містить жодної логіки. Воно лише відображає дані та обробляє дані користувача.

Контролер – це компонент, який розташований між моделлю та видом. Він відповідає за обробку взаємодії користувача з представленням, а також за виконання будь-якої логіки, необхідної для підготовки даних для відображення. Наприклад, якщо користувач натискає кнопку в інтерфейсі користувача програми, це ініціює подію, яку оброблятиме контролер, у свою чергу, він може використовувати цю інформацію для отримання даних із бази даних або виконання інших завдань[13].

MVC є ідеальним підходом для мобільних застосунків, де основний акцент зроблений на простоту реалізації та чітку структуру програми. Ця архітектура широко застосовується в застосунках, де потрібне чітке розділення даних і представлення. Вона підходить для застосунків з не надто складною логікою бізнес-правил і які повинні мати добре структурований інтерфейс користувача. Наприклад, застосунки для відображення новин, погодних застосунків, калькулятори, мобільні версії блогів, онлайн-магазини тощо.

Архітектура MVC допомагає чітко розділити функціональні частини програми, що полегшує підтримку та масштабування коду. Оскільки модель, подання і контролер відокремлені, зміни в одному компоненті не повинні впливати на інші частини системи. Ще одна перевага архітектури MVC – це полегшене тестування. Оскільки компоненти мають чітке розділення відповідальності, можна тестувати кожен частину програми окремо.

Попри переваги, MVC має і деякі обмеження. Якщо інформаційна система має складну логіку інтерфейсу, то подання може стати занадто громіздким, оскільки воно повинне включати в себе велику кількість коду для управління різними станами користувацького інтерфейсу. Це може призвести до того, що код стане важким для обслуговування та змін.

Ще одна проблема, з якою можна зіткнутися при використанні MVC, це необхідність управління великою кількістю зв'язків між контролером і моделлю. Це може ускладнити програмування, оскільки контролер має керувати великою кількістю різних подій і даних, що вимагає значних зусиль для підтримки цілісності логіки.

2.2.4 Архітектура MVP

Архітектура MVP (Model-View-Presenter) є однією з популярних архітектур для розробки мобільних застосунків, яка фокусується на чіткому розподілі відповідальностей між компонентами програми[14]. Ця архітектура зображена на рисунку 2.4.

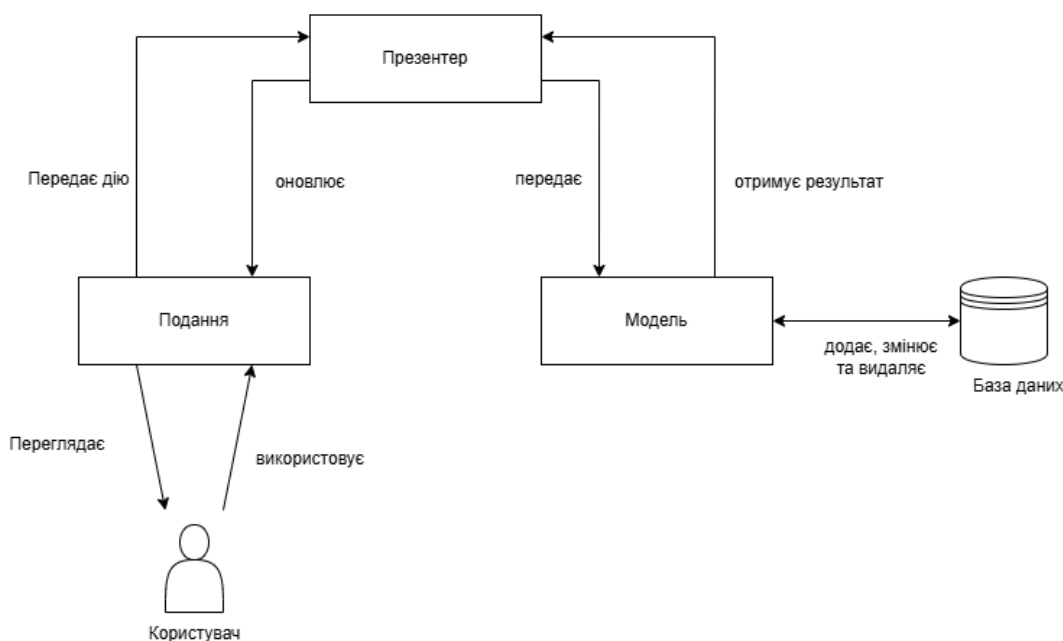


Рисунок 2.4 – Архітектура MVP

MVP є еволюцією архітектури MVC і розподіляє обов'язки ще більш детально, створюючи окремі ролі для обробки логіки програми. В основі цієї архітектури знаходяться три ключових компоненти: Модель (Model), Подання (View) та Презентер (Presenter).

Модель – як і в інших архітектурних підходах, модель містить бізнес-логіку програми та відповідає за обробку даних.

Подання – цей компонент відповідає за відображення інформації користувачу. Подання приймає дані від презентера і представляє їх на екрані. Важливо, що подання не має доступу до бізнес-логіки чи даних, вона лише відображає те, що їй передає презентер.

Презентер – це компонент, який є серцем архітектури MVP. Презентер виступає між моделлю і поданням, бере на себе більшу частину логіки. Він

отримує дані з моделі, обробляє їх, готує для відображення і передає в подання. Також презентер обробляє введення від користувача через подання, викликаючи відповідні функції моделі для виконання операцій.

MVP особливо корисна в інформаційних системах, де важлива чітка організація коду та тестування. Ця архітектура добре підходить для застосунків з великими або складними інтерфейсами користувача, де є потреба у розподілі обов'язків між компонентами для зручності обслуговування і модифікації. Оскільки MVP дозволяє чітко визначити, де повинні знаходитися бізнес-логіка та UI-логіка, вона підходить для застосунків, де інтерфейс повинен змінюватися незалежно від обробки даних.

Основною перевагою MVP є чітке розділення відповідальностей між компонентами, що значно спрощує тестування і підтримку коду. Оскільки презентер не має залежності від конкретного виду представлення, його можна легко тестувати окремо, без необхідності взаємодії з інтерфейсом користувача. Це дає можливість виконувати юніт-тестування логіки бізнес-процесів, не займаючись тестуванням самого інтерфейсу.

Іншою перевагою є гнучкість у зміні інтерфейсу. Оскільки логіка представлення відокремлена від бізнес-логіки, можна змінювати вигляд інтерфейсу користувача без впливу на інші компоненти. Це корисно при розробці систем, що повинні підтримувати різні платформи або адаптивні інтерфейси.

З іншої сторони, MVP має деякі недоліки. Одним з них є те, що презентер може стати дуже громіздким, якщо логіка системи стає складною. Окрім того, якщо модель або подання змінюється, то це може вимагати змін у презентері, що робить систему менш гнучкою, ніж, наприклад, в архітектурі MVVM.

Іншим недоліком є необхідність організації комунікації між компонентами, що може ускладнити розробку на етапі початкової реалізації.

Реалізація архітектури часто вимагає створення додаткових класів та інтерфейсів (часто з подібним вмістом), що часто пов'язано з надлишковим кодом.

До того ж, дана архітектура є занадто складною для простих невеликих програм[15].

2.2.5 Архітектура MVVM

Архітектура MVVM (Model-View-ViewModel) є однією з архітектур для розробки інформаційних систем, яка забезпечує чітке розділення обов'язків між компонентами системи, що сприяє простоті тестування та розширюваності коду[16]. Архітектура MVVM зображена на рисунку 2.5.

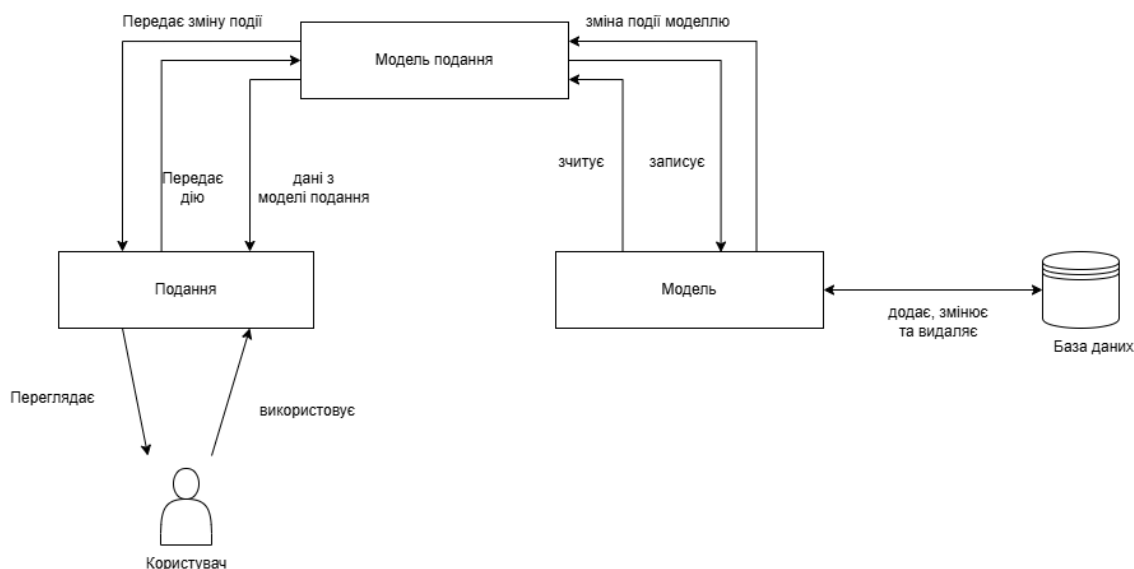


Рисунок 2.5 – Архітектура MVP

Вона складається з трьох основних частин: модель, подання та модель подання.

Модель містить бізнес-логіку системи, а також відповідає за взаємодію з джерелами даних.

Подання – компонент подання відповідає за відображення інтерфейсу користувача та отримання даних від користувача. Важливо, що подання у MVVM пасивне, тобто воно тільки відображає зміни, не втручаючись у їх обробку.

Модель подання (ViewModel) – це центральний елемент у архітектурі MVVM, що виконує роль посередника. Модель подання отримує дані з моделі, обробляє їх і надає у зручному для відображення вигляді. За допомогою двосторонньої прив'язки даних модель подання автоматично передає зміни даних у подання, що забезпечує актуальність і швидкість оновлень без написання додаткового коду для взаємодії між поданням і моделлю.

MVVM найбільш ефективна у системах з складними або динамічними інтерфейсами, де важлива швидка реакція на зміни даних і необхідність автоматичного оновлення інтерфейсу. Така архітектура широко використовується у застосунках, де необхідно працювати з великими обсягами даних або даними в реальному часі. Наприклад, це можуть бути фінансові застосунки, застосунки для моніторингу здоров'я, соціальні мережі або застосунки для електронної комерції.

Однією з основних переваг MVVM є двостороння прив'язка даних, яка значно спрощує код і знижує складність взаємодії між поданням та моделлю подання. Це дозволяє автоматично оновлювати інтерфейс користувача в разі змін у даних, без необхідності вручну оновлювати елементи інтерфейсу після кожної зміни в моделі. Це також спрощує тестування, оскільки модель подання містить всю логіку застосунка і може бути протестована без необхідності запуску інформаційну систему.

MVVM дозволяє відокремити представлення від його моделі даних. Іншими словами, ви можете використовувати одностороннє зв'язування даних, щоб підтримувати їх синхронізацію. Це полегшує тестування вашого коду, оскільки він модульний і слабо пов'язаний. Поділ проблем, який надає MVVM, спрощує обслуговування та тестування вашого коду. Ви можете замінити один фрагмент коду, не впливаючи на інший[17].

Незважаючи на численні переваги, MVVM має і свої недоліки. Один з них полягає в тому, що двостороння прив'язка даних може призвести до неочікуваних проблем при масштабуванні систем. Якщо неправильно реалізувати прив'язку даних або не контролювати її, це може призвести до надмірних оновлень інтерфейсу, що негативно впливає на продуктивність програми.

Іншим недоліком є те, що при великій кількості елементів інтерфейсу, які мають двосторонню прив'язку, проект може стати дуже складним для розуміння і підтримки. Для невеликих застосунків, які не потребують складних UI або динамічних оновлень, MVVM може бути надмірною архітектурою.

2.2.6 Архітектура MVI

Архітектура MVI (Model-View-Intent) є однією з архітектур для проектування інформаційних систем, що популярна завдяки своїй ясності та передбачуваності при обробці користувацьких подій[18]. Ця архітектура зображена на рисунку 2.6.

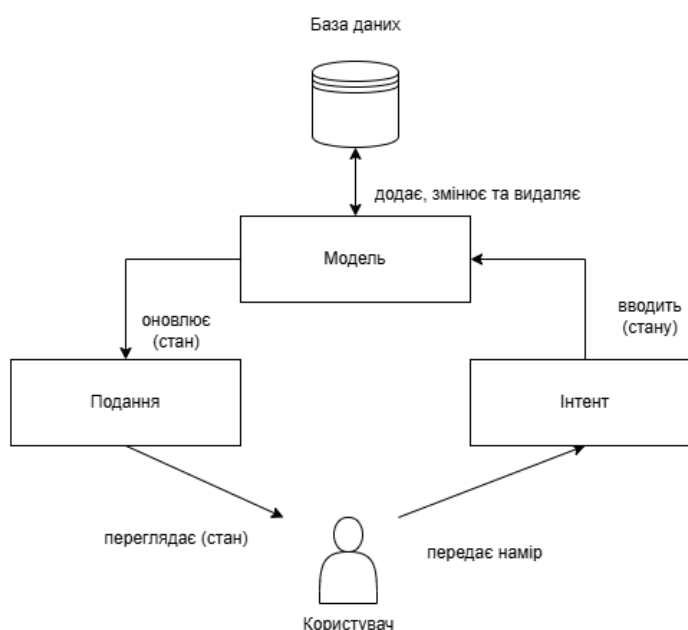


Рисунок 2.6 – Архітектура MVI

Вона підходить для систем з великою кількістю взаємодій, де кожна зміна стану повинна бути чітко відслідковувана і оброблена. Основною метою MVI є забезпечення передбачуваності поведінки системи через чітке розмежування між поданням, станами та подіями, що відбуваються в системі.

Модель – модель в архітектурі MVI відповідає за обробку бізнес-логіки застосунка та управління даними.

Подання – це компонент, що відображає інформації користувачу. У MVI подання є пасивним, відображаючи лише поточний стан системи.

Намір (Intent) – ключовий компоненті MVI, що представляє собою намір користувача взаємодіяти з системою, наприклад, натискання кнопки, введення тексту чи вибір елемента. Інтент описує бажану дію користувача, на основі чого

відбувається оновлення стану системи. Таким чином, інтерфейс користувача не має прямої взаємодії з моделлю, лише через описані інтенти.

Архітектура MVI чудово підходить для розробки інформаційних систем, де важливо точно відслідковувати і управляти всіма змінами стану в системі. Це можуть бути системи з високою динамічністю або системи, де кожна зміна в інтерфейсі залежить від конкретних дій користувача, таких як фільтрація даних, навігація між екранами або інтерактивні елементи. Архітектура MVI може бути корисною також у складних системах з багатьма активними користувацькими подіями, де потрібне чітке управління станами, щоб уникнути непередбачуваних змін або складних помилок у користувацькому інтерфейсі.

Однією з найбільших переваг MVI є передбачуваність станів застосунка. Завдяки чітко структурованому процесу обробки інтенцій користувача та змін стану, система легко передбачувана та спрощена для тестування. MVI також дозволяє значно спростити управління станом у застосунку, особливо для великих програм з великою кількістю взаємодій, оскільки кожен стан застосунку є незмінним.

Один з недоліків MVI полягає в тому, що архітектура часто вимагає написання додаткового коду для обробки односпрямованого потоку даних, обробки подій і управління станом. Це може призвести до збільшення шаблонного коду.

Крім того, реалізація MVI з нуля може потребувати додаткового часу розробки порівняно з використанням більш звичних шаблонів. Розробникам необхідно спроектувати односпрямований потік даних, встановити зв'язок між компонентами та обробляти оновлення стану та візуалізацію інтерфейсу користувача[19].

Іншим недоліком є те, що на великих екранах або з багатьма подіями система може стати складною для масштабування, оскільки кожен змістовний елемент вимагає точної відповідності між інтентом та оновленням стану.

2.2.7 Чиста архітектура

Чиста архітектура (Clean Architecture) – це принцип розробки програмного забезпечення, представлений Робертом С. Мартіном, якого часто називають «дядько Боб». Це спосіб організації коду та визначення архітектури програмного застосунку для досягнення кращої зручності обслуговування, гнучкості та тестування[20]. Приклад такої архітектури зображений на рисунку 2.7.



Рисунок 2.7 – Чиста архітектура

Основною метою цієї архітектури є створення модульного, масштабованого і легко підтримуваного коду, що дозволяє зменшити залежності між компонентами та підвищити тестованість і стійкість застосунку до змін.

У чистій архітектурі передбачається, що всі важливі частини програми поділяються на кілька шарів, з яких основний – це шар бізнес-логіки. Цей шар містить правила і логіку, які не залежать від місця надходження даних. Завдяки такому підходу бізнес-логіка є незалежною від конкретних технологій, таких як бази даних або інтерфейси користувача.

Крім бізнес-логіки, чиста архітектура передбачає чітке розділення між різними рівнями системи. Це зазвичай включає такі шари: шар представлення (Presentation Layer), шар використання (Use Case Layer), шар домену (Domain Layer) і шар даних (Data Layer). Шар представлення відповідає за інтерфейс користувача і взаємодію з ним, Use Case Layer визначає основні функціональні

вимоги до системи, а Data Layer займається зберіганням і отриманням даних. Кожен з цих шарів має чітко визначену роль і взаємодіє з іншими шарами лише через добре визначені інтерфейси.

Одна з головних переваг чистої архітектури полягає в тому, що вона дозволяє зберігати високу гнучкість і можливість розширення застосунку. Крім того, тестування компонентів стає простим завдяки чітким межах між шарами і відсутності складних залежностей.

Проте важливо зазначити, що використання чистої архітектури вимагає ретельного планування і розуміння принципів поділу відповідальності та залежностей між компонентами.

Зі збільшенням кількості шарів, загальна складність системи може зрости. Для невеликих застосунків чиста архітектура може бути надто складною та призвести до зниження ефективності розробки через збільшення взаємодії між шарами.

Використання кількох рівнів та інтерфейсів може призвести до певних витрат на продуктивність. Хоча це зазвичай є незначним, у системах реального часу або в програмах, де критично важлива максимальна продуктивність, це може викликати занепокоєння та має бути ретельно оцінено[21].

2.2.8 Мікросервісна архітектура

Мікросервісна архітектура – це підхід до розробки інформаційних систем, що полягає в розподілі системи на невеликі, самостійні сервіси, кожен з яких відповідає за виконання певної бізнес-логіки[22].

У мікросервісній архітектурі кожен сервіс є окремим застосунком, який має свою власну базу даних і комунікує з іншими сервісами через добре визначені API. Це дозволяє розділяти логіку застосунка на дрібні, легко керовані компоненти.

Однією з основних переваг мікросервісної архітектури є здатність до масштабування. Кожен сервіс можна масштабувати окремо, в залежності від навантаження.

Ще однією важливою перевагою є гнучкість у виборі технологій. Оскільки мікросервіси є незалежними, кожен сервіс може бути розроблений із застосуванням різних технологій або мов програмування, в залежності від вимог і цілей.

Мікросервіси також забезпечують більшу стійкість до помилок. Оскільки сервіси є незалежними, збої в одному з них не призводять до повного відключення системи.

Однак мікросервісна архітектура має і свої недоліки, зокрема складність у розгортанні та управлінні великими системами. Оскільки кожен сервіс є самостійним компонентом, потрібно налаштовувати додаткові інструменти для моніторингу та взаємодії між сервісами.

Крім того, мікросервіси потребують висококваліфікованих фахівців, які можуть управляти складною інфраструктурою і забезпечувати ефективну комунікацію між сервісами.

Ще одним недоліком є тестування та налагодження програми на основі мікросервісної архітектури, оскільки програма може бути розподілена між кількома серверами і пристроями. Тому для ефективного тестування та налагодження програми потрібно мати доступ до всіх серверів і пристроїв, які є частиною системи[23].

2.2.9 Порівняльна характеристика архітектур

Порівняємо розглянуті архітектурні рішення для проектування інформаційних систем. Порівняльна характеристика архітектур наведена в таблиці 2.1.

Таблиця 2.1 – Порівняльна характеристика архітектур

Архітектура	Переваги	Недоліки	Використання
Монолітна архітектура	Простота розробки та тестування, менш складна інфраструктура	Мала гнучкість, важко масштабувати, зміни в одній частині можуть впливати на всю систему.	Підходить для невеликих та середніх застосунків, де масштабування не є критичним.
Клієнт-Серверна архітектура	Централізоване зберігання даних, легкість масштабування	Залежність від сервера, можливі проблеми з безпекою та продуктивністю при високому навантаженні	Підходить для застосунків, що потребують постійної комунікації з сервером і зберіганням даних
Архітектура MVC	Чіткий поділ обов'язків, полегшує тестування та підтримку	Складність у великих проєктах, велика кількість кодових файлів, ускладнене управління станами.	Підходить для застосунків, де важливий чіткий поділ між інтерфейсом і логікою
Архітектура MVP	Відокремленість логіки від UI, полегшене тестування, презентер може бути повторно використаний.	Потенційно більше шаблонного коду, складність у великих проєктах, де кілька презентер.	Підходить для застосунків з необхідністю тісної взаємодії між поданням і моделлю, наприклад, мобільні застосунки.
Архітектура MVVM	Полегшує тестування, відокремлює	Складність у впровадженні двостороннього	Підходить для мобільних застосунків, особливо з

Архітектура	Переваги	Недоліки	Використання
	логіку бізнесу від UI, добре працює з двостороннім зв'язком даних.	зв'язку, зайва складність для простих застосунків.	використанням архітектур, як Android
Архітектура MVI	Чіткий потік даних і управління станами, легка адаптація UI, підтримка складних сценаріїв взаємодії	Складність у масштабуванні та обробці складних станів, велика кількість обробки UI	Підходить для застосунків з високою взаємодією з користувачем та складними бізнес-процесами
Чиста архітектура	Модульність і гнучкість, полегшує тестування та обслуговування, добре ізольовані шари.	Потребує значних зусиль при реалізації, велика кількість шаблонного коду.	Підходить для великих, складних систем, де важлива модульність і незалежність компонентів.
Мікро-сервіси	Гнучкість, масштабованість, незалежність компонентів, можливість використання різних технологій.	Складність в управлінні, великий обсяг інфраструктури.	Підходить для великих, високонавантажених застосунків, що потребують масштабування та гнучкості.

2.3 Вибір та обґрунтування технічних рішень

Для розробки інформаційної системи для підтримки здорового способу життя на основі Інтернету речей необхідно врахувати низку технічних аспектів, що відповідають поставленим завданням. Основною метою є створення системи, яка дозволить користувачам легко відстежувати свій фізичний стан, отримувати рекомендації щодо здорового способу життя та аналізувати дані, отримані з IoT-пристроїв.

Для реалізації цієї інформаційної системи найоптимальнішим є використання архітектури MVVM, тому що такий підхід дозволяє створювати масштабовані інформаційні системи на основі Інтернету речей, які будуть швидко отримувати та обробляти дані з IoT-пристроїв.

Вибір даної архітектури обґрунтовується її ключовими перевагами, які ідеально відповідають вимогам системи. Одна з основних причин полягає у двосторонньому зв'язуванні даних між поданням і моделлю подання. Завдяки цьому дані, отримані від IoT-пристроїв, такі як показники пульсу та рівня фізичної активності, автоматично синхронізуються з інтерфейсом користувача. Це підвищує швидкість обробки інформації та зручність використання системи.

Також MVVM дозволяє чітко розмежувати компоненти системи: бізнес-логіка зосереджена у моделі, логіка представлення даних – у моделі подання, а UI реалізується у поданні. Це спрощує розробку, тестування та масштабування системи, особливо в умовах великої кількості функціональних модулів, таких як моніторинг здоров'я, аналіз фізичної активності, планування дієти та створення рекомендацій. Відокремлення логіки від інтерфейсу також робить систему гнучкішою, дозволяючи вносити зміни без впливу на всю архітектуру.

Системи IoT зазвичай потребують обробки великого обсягу даних, що надходять із пристроїв. Завдяки моделі подання ці дані обробляються та готуються до відображення в поданні, що зменшує навантаження на інтерфейс і забезпечує високу продуктивність. Це особливо важливо для IoT-застосунків, де швидкість обробки даних може впливати на користувацький досвід.

Крім того, MVVM добре інтегрується з сучасними інструментами розробки, такими як прив'язка даних (Data Binding) у платформі Android, що автоматизує процес синхронізації даних і зменшує кількість рутини в коді. Це дає змогу створювати ефективні та інтуїтивно зрозумілі інтерфейси для користувачів, які спрощують взаємодію із системою.

Бібліотека прив'язки даних – це бібліотека підтримки, яка дозволяє вам прив'язувати компоненти інтерфейсу користувача у макетах до джерел даних у програмі, використовуючи декларативний формат, а не програмно. Макети часто визначаються в діях з кодом, який викликає методи інфраструктури інтерфейсу користувача[24].

Для зберігання всіх необхідних даних інформаційної системи для підтримки здорового способу життя на основі Інтернету речей буде використовуватися реляційна база даних.

Реляційна база даних – це набір інформації, яка організовує дані в попередньо визначених зв'язках, де дані зберігаються в одній або кількох таблицях (або зв'язках) стовпців і рядків, що дає змогу легко побачити та зрозуміти, як різні структури даних пов'язані одна з одною[25].

Використання реляційної СКБД є найоптимальнішим варіантом через ряд переваг:

- реляційна СКБД пропонує організацію даних у структурованому форматі, тобто дозволяє ефективно організувати дані у вигляді таблиць, де кожен запис відображає ключову інформацію про користувача або IoT-пристрій;
- реляційна СКБД пропонує можливість зв'язку між даними, що дозволяє у системі, яка працює з даними від IoT-пристроїв, зв'язувати показники здоров'я з конкретними користувачами;
- забезпечення цілісності даних;
- легка інтеграція з мобільними застосунками;
- підтримка роботи з великими обсягами даних;
- масштабованість.

2.4 Вибір та обґрунтування можливих варіантів технічної реалізації

Ураховуючи вимоги до розроблюваної інформаційної системи та аналіз архітектур проєктування систем, було сформовано технічну реалізацію системи.

Відповідно до використання Bluetooth технологій та розроблення мобільного застосунку, який зчитує дані з пристроїв Інтернету речей, було обрано архітектуру MVVM побудови системи. Структурна схема архітектури інформаційної системи наведено в додатку А. Реалізація даної системи буде базуватися на мові програмування Java.

Java – це багатоплатформна, об'єктно-орієнтована мова програмування, яка сама по собі може використовуватися як платформа[26]. Це швидка, безпечна та надійна мова програмування для всього: від мобільних застосунків та корпоративного ПЗ до застосунків для роботи з великими даними та серверних технологій.

Переваги використання Java:

- широка підтримка Android – Java є основною мовою розробки для Android-застосунків, що забезпечує доступ до вбудованих API та фреймворків;
- переносимість, тобто код, написаний на Java, може працювати на різних платформах із мінімальними змінами;
- продуктивність і безпека – Java забезпечує надійність програмного коду завдяки перевірці типів на етапі компіляції та автоматичному управлінню пам'яттю.

Також у Java є бібліотека для взаємодії з Bluetooth, яка забезпечує можливість бездротового з'єднання між пристроями для обміну даними на короткій відстані. За допомогою Android Bluetooth API розробники можуть здійснювати пошук доступних пристроїв, підключатися до них та передавати дані. Основні класи включають BluetoothAdapter для керування Bluetooth-модулем пристрою, BluetoothDevice для представлення знайдених пристроїв, а також BluetoothSocket для встановлення з'єднання. Android Studio надає зручне середовище для інтеграції Bluetooth у мобільний застосунок, дозволяючи

створювати функції для зчитування даних із пристроїв Інтернету речей (IoT), таких як фітнес-трекери чи розумні годинники.

Взаємодію класів зображено на діаграмі класів інформаційної системи в додатку Д.

Для зберігання даних, інформаційна система буде використовувати SQLite.

SQLite – це компактна, вбудована система керування реляційними базами даних (СКБД). Вона відрізняється від багатьох інших СКБД, таких як MySQL або PostgreSQL, тим, що вона вбудовується безпосередньо в застосунок, не вимагаючи окремого сервера баз даних[27].

Вона є оптимальним рішенням, оскільки має низку переваг:

- легка інтеграція з мобільними застосунками – SQLite є вбудованою СКБД, яка не потребує встановлення додаткового сервісу або серверного середовища для функціонування;

- висока продуктивність для локального зберігання – SQLite дозволяє швидко виконувати операції читання та запису, що є критично важливим для забезпечення безперебійної роботи застосунку;

- підтримка офлайн-режиму – інформаційна система повинна відстежувати показники здоров'я, тому вона може використовуватися під час фізичних активностей, не маючи підключення до стабільного Інтернету;

- простота в налаштуванні та використанні – SQLite не потребує складної конфігурації чи налаштувань, що пришвидшує її впровадження;

- ефективне використання ресурсів – SQLite є надзвичайно компактною та ефективною з точки зору споживання пам'яті та процесорного часу;

- гнучкість – SQLite підтримує усі базові операції SQL, що дозволяє виконувати CRUD-операції, а також працювати з більш складними запитам.

Ураховуючи потреби користувачів до розроблюваної системи, взаємодія з інформаційною системою для підтримки здорового способу життя на основі Інтернету речей буде відбуватися у вигляді мобільного застосунку.

Мобільний застосунок – це програмний застосунок, розроблений спеціально для використання на невеликих бездротових обчислювальних пристроях, таких як

смартфони та планшети, а не на настільних або портативних комп'ютерах. [28]. Він надає користувачам доступ до різних сервісів, функціоналу або контенту, що можуть бути як автономними, так і залежними від підключення до Інтернету. Мобільні застосунки можуть бути реалізовані для різних платформ, наприклад, Android або iOS, і часто мають зручний графічний інтерфейс для взаємодії з користувачами.

Переваги мобільного застосунку:

- мобільність і зручність доступу, тобто мобільні застосунки можна використовувати будь-де та будь-коли, особливо якщо вони не потребують доступу до Інтернету;

- інтеграція з пристроями Інтернету речей;

- мінімальне апаратне забезпечення, оскільки зазвичай мобільні застосунки не потребують великих апаратних обрахунків, що робить систему більш доступною для використання;

- автономність – завдяки використанню локальної бази даних, мобільний застосунок може функціонувати без постійного підключення до Інтернету.

У додатку В наведено діаграму послідовності інтеграції системи, яка показує як користувач та пристрої Інтернету речей взаємодіють із системою.

Для розробки інформаційної системи було обрано IDE Android Studio. Android Studio – це офіційне інтегроване середовище розробки (IDE) для створення мобільних застосунків під операційну систему Android, розроблене компанією Google. Відповідальність Android Studio полягає в наданні інтерфейсу для створення програм і керування більшою частиною складного керування файлами за лаштунками. Використовується мова програмування Java або Kotlin[29].

Однією з ключових переваг Android Studio є інтеграція з інструментами Інтернету речей. Це дозволяє розробляти застосунки, які можуть взаємодіяти з фітнес-трекерами, розумними годинниками та іншими пристроями. Використовуючи протоколи Bluetooth, Wi-Fi або HTTP, система може легко

збирати, аналізувати й відображати дані про здоров'я користувачів, такі як рівень фізичної активності або пульс.

Середовище пропонує зручний графічний редактор для створення інтерфейсів користувача. Цей інструмент спрощує розробку інтуїтивно зрозумілих і візуально привабливих застосунків.

Android Studio також забезпечує можливість тестування застосунків за допомогою вбудованого емулятора, який дозволяє перевіряти роботу на віртуальних пристроях різних конфігурацій. Це особливо корисно для перевірки сумісності застосунка з різними пристроями Інтернету речей та оптимізації його роботи в реальному часі.

Крім того, Android Studio має потужні інструменти для управління проектами, включаючи Gradle для роботи із залежностями та інтеграцію з системами контролю версій, такими як Git. Це забезпечує зручність і швидкість у розробці.

Завдяки регулярним оновленням Android Studio підтримує новітні технології та забезпечує сумісність із сучасними версіями Android. Це робить платформу надійною для розробки інформаційної системи для підтримки здорового способу життя на основі Інтернету речей, забезпечуючи високу якість і адаптивність програмного продукту до сучасних потреб користувачів.

Висновки до розділу 2

У даному розділі було проведено детальний аналіз сучасних технічних рішень, які можуть бути використані для створення інформаційної системи підтримки здорового способу життя на основі Інтернету речей. Було досліджено різні архітектурні підходи, зокрема монолітний, клієнт-серверний, мікросервісний, чистий, MVC, MVP, MVI та MVVM. Кожен із цих підходів був оцінений з точки зору його переваг і недоліків у контексті розроблюваної системи.

На основі порівняння технічних рішень для розробки системи було обрано архітектуру MVVM. Цей підхід дозволяє чітко розділити логіку взаємодії з користувачем і бізнес-логіку, що значно спрощує розробку та тестування, особливо при інтеграції з пристроями IoT. Архітектура MVVM забезпечує високий рівень масштабованості і гнучкості, що важливо для майбутнього розширення системи. Реалізація клієнтської частини у вигляді мобільного застосунка забезпечує зручний доступ до функцій системи, дозволяючи користувачам ефективно взаємодіяти з застосунком на різних платформах. Для зберігання даних було обрано реляційну СКБД SQLite, що є оптимальним рішенням для роботи з локальними даними на мобільних пристроях.

Таким чином, обрані технічні рішення повністю відповідають вимогам проєкту, забезпечуючи надійність, масштабованість і зручність у використанні для кінцевих користувачів.

3 ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ

3.1 Вхідні дані

Інформаційна система для підтримки здорового способу життя на основі Інтернету речей має забезпечувати наступні три види введення даних:

- введення даних користувачем;
- автоматичний збір даних з пристроїв Інтернету речей;
- автоматичне введення даних з бази даних.

Далі наведено більш детальну інформацію про дані, які вводяться в інформаційну систему.

Нижче наведено перелік даних, які вводяться користувачем:

- пошта користувача;
- ім'я користувача;
- пароль та підтвердження пароля користувача;
- підтвердження зберігання даних для входу;
- стать користувача;
- дата народження користувача;
- система вимірювань (міжнародна або імперська);
- зріст в сантиметрах або в футах та дюймах;
- вага в кілограмах або фунтах;
- спосіб життя;
- ціль;
- наявні в домі інгредієнти;
- інгредієнти, заборонені користувачем;
- інгредієнти та/або страви, спожиті протягом дня.

Решта інформації буде вводиться в інформаційну систему автоматично. Частина автоматичного введення даних буде зчитуватися з пристроїв Інтернету речей, таких як фітнес-трекери або розумні годинники. До таких належать наступні дані:

- заряд батареї Інтернет пристрою;

- значення пульсу користувача;
- кількість кроків, пройдених користувачем;
- дистанція в метрах, пройдена користувачем;
- кількість калорій, спалених користувачем під час активності.

Також інформація буде вводитися в систему з бази даних. До такої інформації належать:

- дані, введені в систему раніше (введені користувачем та зчитані з пристроїв Інтернету речей);
- назва рецепту;
- калорійність на 100г страви;
- час приготування;
- інгредієнти, що використовуються для приготування страви;
- інструкція до приготування страви;
- категорія, до якої відноситься страву;
- вміст протеїну в страві;
- вміст жирів в страві;
- вміст вуглеводів у страві;
- вміст клітковини у страві;
- порція страви в грамах;
- загальна кількість калорій на порцію.

3.2 Математичні формули для розрахунків

Інформаційна система для підтримки здорового способу життя на основі Інтернету речей надає рекомендації щодо меню на основі введених даних. Для цього вона використовує математичні формули, що представлені у даному підрозділі.

Основною формулою є розрахунок базального метаболізму (BMR) за переглянутим рівнянням Харріса-Бенедикта, дослідженою в статті «Огляд

автоматизованих систем охорони здоров'я»[1]. Нижче представлено дане рівняння (3.1–3.2):

Міжнародна система одиниць:

$$\begin{aligned} \text{MaleBMR} &= (9.65 \times \text{weight}) + (5.73 \times \text{height}) - (5.08 \times \text{age}) + 260 \\ \text{FemaleBMR} &= (7.38 \times \text{weight}) + (6.07 \times \text{height}) - (2.31 \times \text{age}) + 43. \end{aligned} \quad (3.1)$$

Імперська система:

$$\begin{aligned} \text{MaleBMR} &= (4.38 \times \text{weight}) + (14.55 \times \text{height}) - (5.08 \times \text{age}) + 260 \\ \text{FemaleBMR} &= (3.35 \times \text{weight}) + (15.42 \times \text{height}) - (2.31 \times \text{age}) + 43, \end{aligned} \quad (3.2)$$

де MaleBMR – добове споживання калорій організмом чоловіка;

weight – вага людини в кілограмах для міжнародної системи та фунтах для імперської;

height – висота людини в сантиметрах для міжнародної системи та в дюймах для імперської;

age – вік людини;

FemaleBMR – добове споживання калорій організмом жінки.

Дані формули розраховують добове споживання калорій організмом людини. Також потрібно врахувати похибки вимірювання BMR, оскільки вони можуть суттєво впливати на результат.

Похибка вимірювання BMR – це різниця між теоретичними і реальними значеннями, яку можуть спричиняти різні фактори. Оскільки для розрахунку BMR використовується рівнянням Харріса-Бенедикта, то на похибку можуть вплинути наступні фактори:

– індивідуальні характеристики людини, такі як вік, стать, рівень фізичної активності;

– час доби та харчування, оскільки результати вимірювань можуть змінюватися в залежності від часу доби, коли проводиться вимірювання, а також після прийому їжі.

Це важливо врахувати для досягнення максимальної точності.

Для мінімізації похибок вимірювання BMR важливо враховувати фізіологічні особливості користувача, такі як вік, стать, рівень фізичної активності та інші фактори, що впливають на обмін речовин.

Для більшої точності розрахунків, потрібно домножити результати на коефіцієнт, який відповідає способу життя користувача.

Нижче наведено коефіцієнти, які відповідають різним способам життя:

– малорухливий спосіб життя (мало або без фізичних вправ), коефіцієнт активності = 1,2;

– легко активний (легкі вправи/спорт 1–3 дні/тиждень), коефіцієнт активності = 1,375;

– помірно активний (помірні вправи/спорт 3–5 днів на тиждень), коефіцієнт активності = 1,55;

– дуже активний (інтенсивні вправи/спорт 6–7 днів на тиждень), коефіцієнт активності = 1,725;

– надзвичайно активний (дуже важкі вправи/спорт і фізична робота або подвійне тренування), коефіцієнт активності = 1,9.

Також потрібно враховувати значення спалених калорій, отримане з пристроїв Інтернету речей.

Урахувавши дані фактори, формула розрахунку калорій має виглядати так (3.3):

$$TDEE = BMR \times k + n, \quad (3.3)$$

де TDEE – загальні добові витрати енергії;

BMR – базальний метаболізм;

k – коефіцієнт активності;

n – кількість спалених калорій, отриманих з пристрою Інтернету речей.

Також система допоможе визначити фізичний стан користувача за індексом маси тіла (3.4–3.5):

Міжнародна система одиниць:

$$BMI = \frac{weight}{height^2}. \quad (3.4)$$

Імперська система:

$$BMI = 703 \times \frac{weight}{height^2}, \quad (3.5)$$

де BMI – індекс маси тіла.

За результатами даної формули можна виділити наступні категорії[30]:

- недостатня вага (сильна худорлявість) $< 16,0$;
- недостатня вага (помірна худорлявість) $16,0 - 16,9$;
- недостатня вага (помірна худорлявість) $17,0 - 18,4$;
- нормальний діапазон $18,5 - 24,9$;
- надмірна вага (перед ожирінням) $25,0 - 29,9$;
- ожиріння (I клас) $30,0 - 34,9$;
- ожиріння (II клас) $35,0 - 39,9$;
- ожиріння (клас III) $\geq 40,0$.

Також система буде розраховувати кількість поживних речовин, які необхідно споживати протягом дня.

Розрахунок споживання білка (3.6):

$$Protein = \frac{TDEE}{4} \times 0.2, \quad (3.6)$$

де Protein – кількість білків, необхідних для споживання в грамах.

Розрахунок споживання жирів (3.7):

$$Fat = \frac{TDEE}{9} \times 0.27, \quad (3.7)$$

де Fat – кількість жирів, необхідних для споживання в грамах.

Розрахунок споживання вуглеводів (3.8):

$$Carbohydrates = \frac{TDEE}{4} \times 0.55, \quad (3.8)$$

де Carbohydrates – кількість вуглеводів, необхідних для споживання в грамах.

Розрахунок споживання клітковини (3.9):

$$Fiber = \frac{TDEE}{1000} \times 14, \quad (3.9)$$

де Fiber – кількість клітковини, необхідної для споживання в грамах.

Розрахунок споживання води (3.10–3.11):

Міжнародна система одиниць:

$$\text{WaterIntake} = m \times 0.03. \quad (3.10)$$

Імперська система:

$$\text{WaterIntake} = \frac{2 \times m}{4} \times 0.03, \quad (3.11)$$

де WaterIntake – кількість води, необхідної для споживання в літрах;
 m – вага користувача.

3.3 Вихідні дані

Вихідні дані, це дані які відображаються користувачу. Нижче наведено перелік вихідних даних системи:

- введені дані користувача;
- пульс;
- кількість пройдених кроків за день;
- кількість спалених калорій за день;
- графіки пульсу та пройдених кроків;
- поточний стан здоров'я на основі ІМТ;
- кількість корисних речовин, які потрібно спожити протягом дня;
- рекомендаційне меню з описом страв;
- кількість калорій, спожитих за день.

Також при неправильно введених даних, система повинна повідомити користувача про помилку та її можливі причини.

3.4 Структура бази даних

У якості реляційної системи керування базами даних для зберігання інформації буде використовуватися SQLite.

Графічний матеріал включає в себе схему бази даних в Додатку Б.

База даних буде складатися з семи сутностей, які наведені в таблиці 3.1.

Таблиця 3.1 – Сутності бази даних

№	Назва таблиці	Назва сутності
1	Recipes	Рецепти
2	User	Користувач
3	Calculations	Розрахунки
4	UserAvailableIngredients	Наявні інгредієнти
5	UserRestrictedIngredients	Заборонені інгредієнти
6	UserMeasurements	Вимірювання
7	UserConsumedCalories	Спожиті калорії

Детальний опис сутностей представлено в таблицях 3.2–3.8.

Таблиця 3.2 – Опис сутності «Рецепти»

Назва параметру	Тип даних	Опис
id	INTEGER	Унікальний код рецепту
name	TEXT	Назва рецепту
calorie_per_100	TEXT	Кількість калорій на 100 грам страви
preparation_time	TEXT	Час приготування
recipe_ingredientparts	TEXT	Інгредієнти для страви
recipe_instructions	TEXT	Інструкція до приготування страви
recipe_category	TEXT	Категорія страви
protein_content	TEXT	Кількість білків у порції страви
fat_content	TEXT	Кількість жирів у порції страви
carbohydrate_content	TEXT	Кількість вуглеводів у порції страви
fiber_content	TEXT	Кількість клітковини у порції страви
portion	TEXT	Середня вага однієї порції страви
calories	TEXT	Кількість калорій на порцію страви

Таблиця 3.3 – Опис сутності «Користувач»

Назва колонки	Тип колонки	Опис колонки
id	INTEGER	Унікальний код користувача
user_nickname	TEXT	Ім'я користувача
user_mail	TEXT	Пошта користувача
user_password	TEXT	Пароль користувача
user_gender	TEXT	Стать користувача
user_birthday	TEXT	Дата народження користувача
user_system	TEXT	Система вимірювання
user_height1	TEXT	Зріст у сантиметрах або футах користувача
user_height2	TEXT	Зріст у дюймах користувача
user_weight	TEXT	Вага кілограмах або фунтах користувача
user_lifestyle	TEXT	Стиль життя користувача
user_goal	TEXT	Ціль користувача

Таблиця 3.4 – Опис сутності «Розрахунки»

Назва колонки	Тип колонки	Опис колонки
id	INTEGER	Унікальний код вимірювань
user_id	INTEGER	Унікальний код користувача
date	TEXT	Актуальна дата
time	TEXT	Актуальний час
user_steps	INTEGER	Кількість кроків, пройдених користувачем
user_burnt_calories	INTEGER	Кількість калорій, спалених користувачем
user_distance	REAL	Дистанція, яку пройшов користувач
user_hearttrate	INTEGER	Серцевий ритм користувача

Таблиця 3.5 – Опис сутності «Наявні інгредієнти»

Назва колонки	Тип колонки	Опис колонки
id	INTEGER	Унікальний код наявного інгредієнта
user_id	INTEGER	Унікальний код користувача
available_ingredients	TEXT	Назва наявного інгредієнта

Таблиця 3.6 – Опис сутності «Заборонені інгредієнти»

Назва колонки	Тип колонки	Опис колонки
id	INTEGER	Унікальний код забороненого інгредієнта
user_id	INTEGER	Унікальний код користувача
restricted_ingredients	TEXT	Назва забороненого інгредієнта

Таблиця 3.7 – Опис сутності «Вимірювання»

Назва колонки	Тип колонки	Опис колонки
id	INTEGER	Унікальний код вимірювань
user_id	INTEGER	Унікальний код користувача
date	TEXT	Дата вимірювань
time	TEXT	Час вимірювань
user_steps	INTEGER	Кількість кроків
user_burtncalories	INTEGER	Кількість спалених калорій
user_distance	REAL	Пройдена відстань
user_heartrate	INTEGER	Пульс

Таблиця 3.8 – Опис сутності «Спожиті калорії»

Назва колонки	Тип колонки	Опис колонки
id	INTEGER	Унікальний код запису спожитих калорій

Назва колонки	Тип колонки	Опис колонки
user_id	INTEGER	Унікальний код користувача
food	TEXT	Їда, яку було спожито
calories	TEXT	Кількість калорій, що відповідає їді

Взаємодію з даними сутностями зображено на схемі потоків даних, яка знаходиться в додатку Г.

Висновки до розділу 3

У даному розділі було розглянуто інформаційне забезпечення для інформаційної системи для підтримки здорового способу життя на базі Інтернету речей. У першому підрозділі було розглянути вхідні дані, які поділяються на три вида: дані, введені користувачем, дані, зібрані з пристроїв Інтернету речей та дані, завантажені з бази даних.

У другому підрозділі було описано математичну модель розрахунку калорій та поживних речовин, що використовується для надання рекомендацій. Модель враховує введені дані та розраховує кількість калорій, що витрачаються протягом дня та кількість необхідних поживних речовин, таких як: білки, жири, вуглеводи, клітковина та вода.

У третьому підрозділі було описано вихідні дані інформаційної системи, що відображаються користувачу. Також система повинна виводити повідомлення в разі помилки та можливі подолання проблеми.

У четвертому підрозділі описано структуру бази даних з детальним описом сутностей та їх параметрів. Дана база даних використовується системою для зберігання вхідних та вихідних даних, а також результатів роботи інформаційної системи.

4 РЕАЛІЗАЦІЯ СИСТЕМИ

4.1 Опис апаратної бази та середовища розробки

Розробка інформаційної системи для підтримки здорового способу життя на основі Інтернету речей вимагає підбору відповідної апаратної бази та середовища розробки. Основними складовими апаратної бази є пристрої IoT, такі як фітнес-трекери та розумні годинники, а також мобільні пристрої, на яких працюватиме система.

Для забезпечення коректної та плавної роботи інформаційної системи, необхідно враховувати мінімальні апаратні вимоги до всіх компонентів системи.

Мінімальні апаратні вимоги для мобільних пристроїв:

- операційна система Android 8.0 або новіша;
- чотириядерний процесор CPU з тактовою частотою не менше 1.5 ГГц;
- мінімальна оперативна пам'ять 2 ГБ;
- мінімальна вільна пам'ять щонайменш 50 МБ для зберігання застосунку, кешу та локальних даних;
- підключення Bluetooth 4.2 або новіше;

Мінімальні апаратні вимоги для IoT-пристроїв:

- підтримування протоколу Bluetooth Low Energy (BLE) для енергоефективного з'єднання з системою;
- акумулятор з тривалістю роботи не менше 7 днів при середньому використанні;
- наявність датчиків руху, пульсометрів та акселерометрів;
- сумісність з Android через відкриті API.

Інформаційна система базується на мові програмування Java та використовує бібліотеку для протоколу зв'язку Bluetooth Low Energy (BLE). Інформаційна система розроблюється в середовищі Android Studio.

4.2 Підходи до розробки клієнтського інтерфейсу

Дизайн інтерфейсу користувача (UI) є ключовим аспектом розробки інформаційної системи для підтримки здорового способу життя на основі Інтернету речей (IoT). Оскільки система спрямована на широку аудиторію користувачів, включаючи людей із різним рівнем технічної обізнаності, інтерфейс повинен бути інтуїтивно зрозумілим і зручним. Головна мета дизайну UI – забезпечити користувачам простий доступ до функціоналу системи, включаючи моніторинг даних із IoT-пристроїв, перегляд рекомендацій та відстеження прогресу.

Нижче наведено основні принципи дизайну інтерфейсу користувача, які слід використовувати під час розробки інформаційних систем:

- простота та зрозумілість, що означає, що інтерфейс повинен бути максимально простим, із мінімальною кількістю елементів на екрані, щоб користувачі могли легко виконувати основні завдання, наприклад, переглядати дані про фізичну активність або налаштовувати IoT-пристрої;

- консистентність, тобто всі елементи UI, такі як кольори, шрифти, стилі кнопок та меню, мають бути однаковими на всіх екранах застосунку, що забезпечить користувачам почуття послідовності та передбачуваності, полегшуючи їхню взаємодію з системою;

- відповідність завданням, тобто інтерфейс повинен концентруватися на ключових функціях системи, таких як моніторинг здоров'я та отримання рекомендацій;

- зручність взаємодій – основні елементи, такі як великі кнопки, зручні для натискання на сенсорних екранах, і прості форми введення, забезпечуватимуть легкість у використанні навіть на невеликих пристроях;

- врахування контексту використання, що передбачає адаптацію інтерфейсу для різних мобільних пристроїв, таких як смартфони та планшети, а також для сценаріїв, коли користувач перебуває в русі.

Таким чином, врахувавши всі вище вказані принципи, дизайн інтерфейсу для інформаційної системи буде орієнтованим на користувача, зрозумілим та функціональним, що забезпечить позитивний досвід і ефективність у використанні.

4.3 Опис інтерфейсу користувача

Інформаційна система для підтримки здорового способу життя на основі Інтернету речей повинна бути інтуїтивно зрозуміла для користувачів із різним рівнем технічної обізнаності.

Інтерфейс користувача інформаційної системи для підтримки здорового способу життя на основі Інтернету речей має бути створений з урахуванням сучасних принципів зручності, естетичності та інтуїтивності.

Він має забезпечувати швидкий доступ до основних функцій, таких як моніторинг показників здоров'я, відстеження фізичної активності, розрахунок калорій і формування персоналізованих рекомендацій. Для зручності користувачів у застосунку має бути передбачено зрозумілу навігацію, графічні елементи (діаграми, графіки) для візуалізації даних.

Можливу взаємодію користувача з системою зображено на діаграмі варіантів використання в додатку Е.

Нижче наведено опис користувацького інтерфейсу системи.

При запуску застосунку, з'являється екран авторизації в систему, зображений на рисунку 4.1.

По середині екрану відображається текстове поле для введення електронної пошти з підказкою «Email». Далі є поле для введення пароля з підказкою «Password», яке приховує введені символи. Нижче знаходяться дві кнопки: кнопка «Register» для реєстрації та кнопка «Log in» для входу в застосунок. Під полем для пароля є чекбокс із підписом «Save my Input», що вказує на можливість зберегти введені дані для зручності наступного входу.

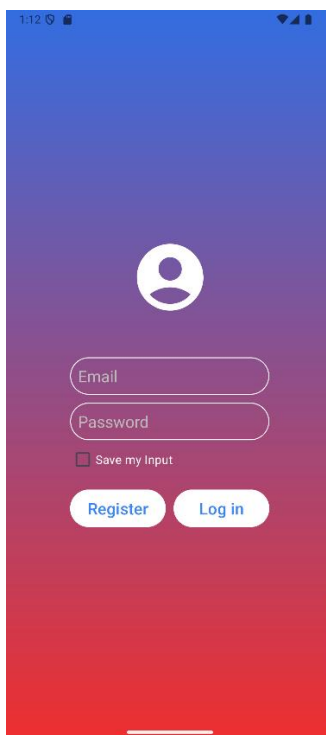


Рисунок 4.1 – Екран авторизації в системі

Якщо користувач залишив хоча б одне з полів пустим та натисне «Log in», з'явиться спливаюче повідомлення про необхідність введення даних, як показано на рисунку 4.2.

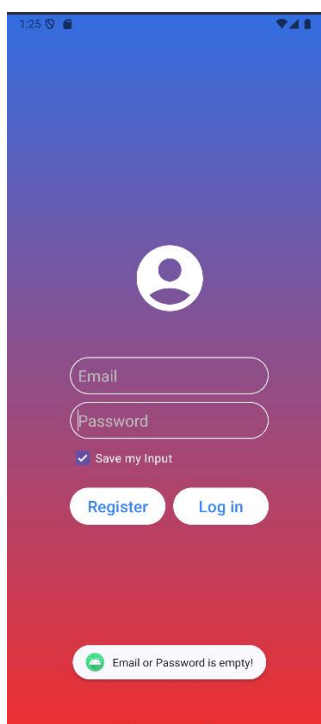


Рисунок 4.2 – Повідомлення користувача про необхідність заповнення даних

Також наявна валідація даних, яка обробляє варіант, коли користувач ввів неіснуючі в системі дані, або якщо користувач ввів некоректний пароль. Дані повідомлення зображені на рисунках 4.3–4.4.

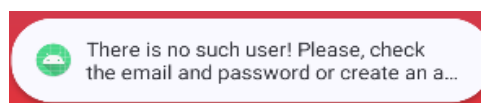


Рисунок 4.3 – Повідомлення про відсутність користувача в системі

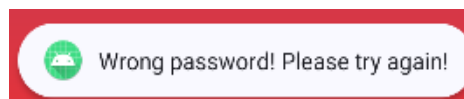


Рисунок 4.4 – Повідомлення про неправильно введений пароль

Якщо при вході в систему, користувач ставить прапорець «Save my input», логін і пароль зберігаються в файлах програми. При наступному запуску застосунку, система автоматично введе дані користувача.

Якщо користувач ще не має акаунту, він має натиснути на кнопку «Register». Після цього з'явиться екран з формою для введення даних про користувача, необхідних для роботи системи. В залежності від вибору системи одиниць, форма має 6 текстових полів при виборі міжнародної системи (рисунок 4.5), та 7 текстових полів при імперській системі одиниць (рисунок 4.6).

A vertical registration form on a mobile device. The background is a gradient from blue at the top to red at the bottom. The form contains the following fields from top to bottom: "Email", "Password", "Confirm password", "Nickname", "Male" (with a dropdown arrow), "Select Birthday date:" with "NOV 27 2024" below it, "International" (with a dropdown arrow), "Height max 250" and "Weight max 250" (with small icons), "Sedentary lifestyle" (with a dropdown arrow), "Lose weight" (with a dropdown arrow), and a "Confirm" button at the bottom.

Рисунок 4.5 – Екран з формою для реєстрації користувача при міжнародній системі одиниць

1:57

Email

Password

Confirm password

Nickname

Male

Select Birthday date:
NOV 27 2024

Imperial

Height max feet in inches Weight max pounds

Sedentary lifestyle

Lose weight

Confirm

Рисунок.4.6 – Екран з формою для реєстрації користувача при імперській системі одиниць

Також форма має 4 випадючих списки, вибір дати народження та та кнопку підтвердження. Випадаючі списки по замовчуванню приймають перше значення зі списку.

Текстові поля розроблені для наступної інформації:

- пошта користувача;
- пароль користувача;
- підтвердження паролю;
- ім'я користувача;
- висота користувача в см (міжнародна система одиниць);
- вага в кг (міжнародна система одиниць);
- висота в футах (імперська система одиниць);
- висота в дюймах (імперська система одиниць);
- вага в фунтах (імперська система одиниць).

Випадаючий список статі має два значення:

- male (чоловік);
- female (жінка).

Випадаючий список систем одиниць має два значення:

- international (міжнародна система одиниць);
- imperial (імперська система одиниць).

Випадаючий список фізичної активності має п'ять значення:

- sedentary lifestyle;
- lightly active;
- moderately active;
- very active;
- extremely active.

Випадаючий список цілі має три значення:

- lose weight;
- maintain weight;
- gain weight.

При натисканні на дату, відкривається спливаюче вікно з трьома сувіями вибору дати: дня, місяця та року.

Якщо користувач не введе якийсь із полів, на екрані з'явиться повідомлення з проханням заповнити всі поля форми. Також у системі оброблений випадок введення пошти, яка вже є в системі та введення різних паролів.

Якщо користувач ввів усі дані коректно і натиснув кнопку «Confirm», його дані зберігаються в базі даних, після чого відкривається головне меню застосунку, зображене на рисунку 4.7.

У верхній частині головного меню розташовані кнопки для редагування профілю та підключення пристроїв через Bluetooth, що дозволяє синхронізувати дані для більш точного аналізу.

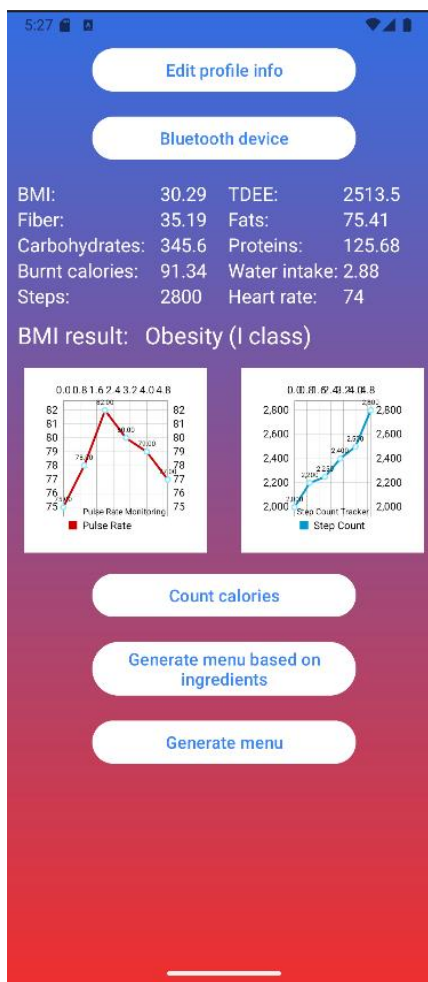


Рисунок 4.7 – Головне меню застосунку

Центральна частина екрану містить детальну інформацію про фізичні показники. Тут вказані наступні дані:

- індекс маси тіла (BMI);
- загальні добові витрати енергії (TDEE);
- добова кількість клітковини в грамах;
- добова кількість жирів в грамах;
- добова кількість вуглеводів в грамах;
- добова кількість протеїну в грамах;
- кількість спалених калорій;
- добова кількість води в літрах.
- кількість пройдених кроків;
- пульс;
- оцінка результатів індексу маси тіла.

Нижче розташовані два графіки. Лівий графік відображає зміну серцевого ритму (пульсу) протягом доби, а правий – кількість пройдених протягом доби. Ці візуалізації дозволяють краще оцінити фізичну активність та відслідковувати свої досягнення.

У нижній частині екрану розміщені кнопки для додаткових функцій. Користувач може розрахувати калорійність раціону, згенерувати меню на основі доступних інгредієнтів або просто згенерувати меню. Це забезпечує персоналізований підхід до планування харчування.

Якщо користувач ще не підключив до системи пристрій Інтернету речей, деякі дані будуть недоступні, як показано на рисунку 4.8.

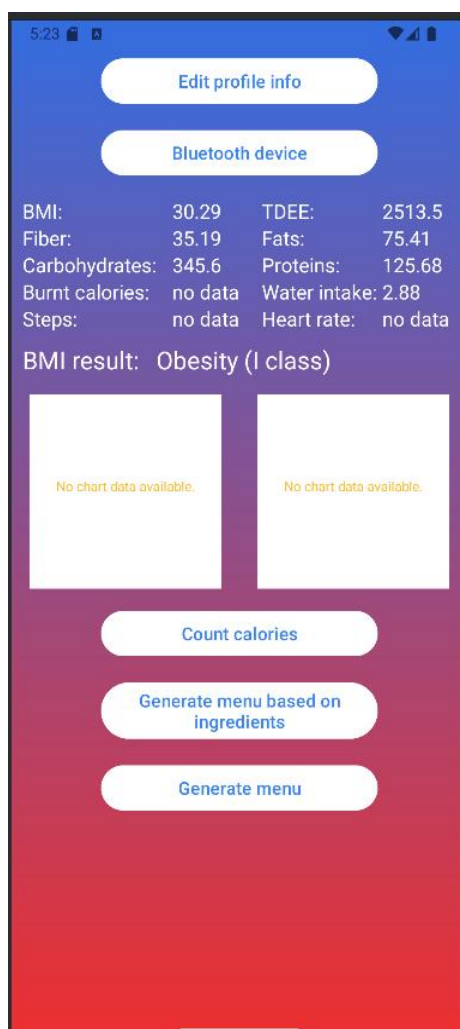


Рисунок 4.8 – Головне меню, якщо користувач не під'єднавав пристрій Інтернету речей

При натисканні на кнопку «Edit profile info», користувачу відкривається екран з формою для зміни дани, як показано на рисунку 4.9.

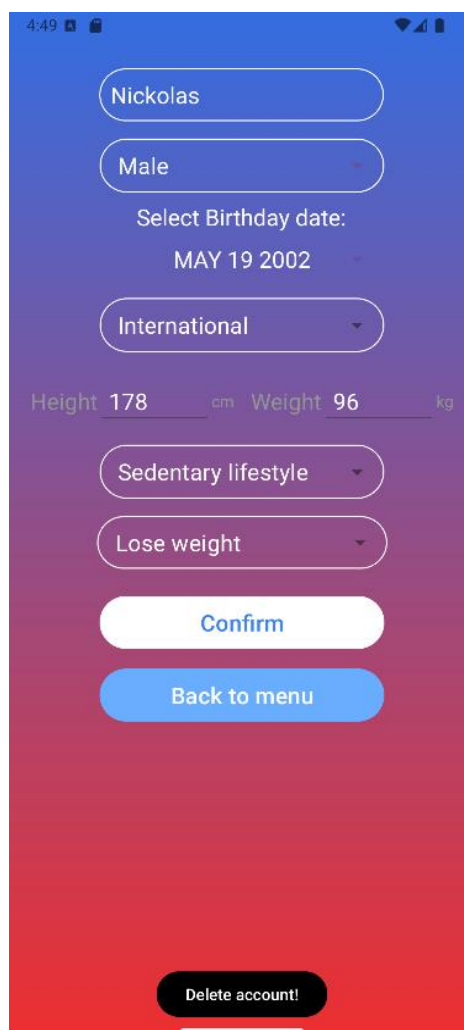


Рисунок 4.9 – Екран оновлення інформації користувача

Інформація про користувача підтягується з бази даних. Таким чином, всі поля заповнені та користувач може бачити попередньо введені дані для порівняння.

Дана форма має 3 текстових поля при міжнародній системі одиниць та 4 текстових полів при імперській системі оцінювання, 4 випадючих списка, вибір дати народження та 3 кнопки.

У формі представлені наступні текстові поля:

- ім'я користувача;
- висота користувача в см (міжнародна система одиниць);
- вага в кг (міжнародна система одиниць);

- висота в футах (імперська система одиниць);
- висота в дюймах (імперська система одиниць);
- вага в фунтах (імперська система одиниць).

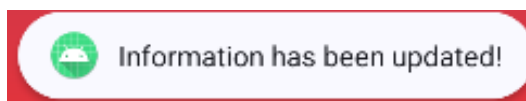
У формі представлено наступні випадуючі списки:

- стать користувача;
- система одиниць;
- фізична активність;
- ціль.

У формі представлені наступні кнопки:

- підтвердження змін;
- повернутися до меню;
- видалити аккаунт.

При натисканні на кнопку «Confirm», якщо всі поля заповнені, застосунок повідомить про оновлення інформації як показано на рисунку 4.10.



Рисуннок 4.10 – Повідомлення про оновлення інформації

У іншому випадку, застосунок повідомить про те, що потрібно заповнити всі дані, як показано на рисунку 4.11.

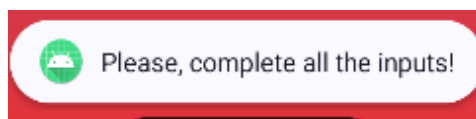


Рисунок 4.11 – Повідомлення про необхідність заповнення полів

При натисканні на кнопку «Back to menu», застосунок повернеться до головного меню.

Якщо користувач натисне кнопку «Delete account!», з'явиться спливаюче вікно з попередженням про видалення аккаунту, як показано на рисунку 4.12.

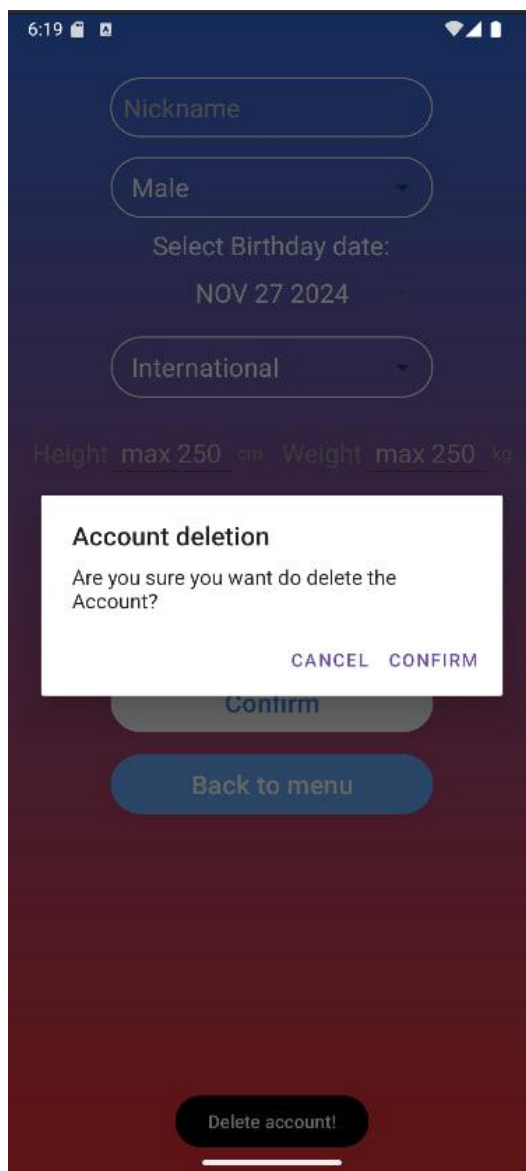


Рисунок 4.12 – Попередження про видалення акаунту

Якщо користувач натисне «Cancel», вспливаюче вікно закриється без жодних змін. Якщо користувач натисне «Confirm», акаунт буде видалено та відкриється екран входу.

Наступною кнопкою головного меню є «Bluetooth device». При натисканні на неї відкривається екран з даними про Bluetooth пристрій.

Якщо жодного пристрою Інтернету речей не виявлено, застосунок виведе повідомлення про відсутність пристрої, як показано на рисунку 4.13.



Рисунок 4.13 – Екран підключень до пристроїв Інтернету речей при відсутності пристроїв Інтернету речей

Якщо є наявні пристрої Інтернету речей, застосунок покаже його назву та надасть можливість підключитися до пристрою, як показано на рисунку 4.14.

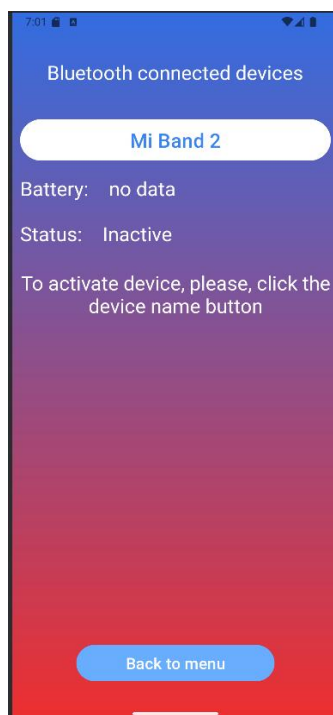


Рисунок 4.14 – Екран підключень до пристроїв Інтернету речей при наявності пристроїв Інтернету речей

Після натискання на кнопку з назвою пристрою, інформація про заряд пристрою і підключення оновиться, як показано на рисунку 4.15.

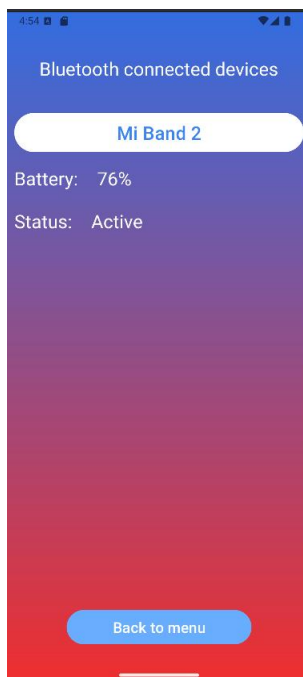


Рисунок 4.15 – Оновлення інформації про пристрій після підключення.

Кнопка «Back to menu» повертає до головного меню при натисканні.

Наступною кнопкою головного меню є кнопка «Count calories». При натисканні на неї, відкривається екран для підрахунку спожитих калорій, як показано на рисунку 4.16.

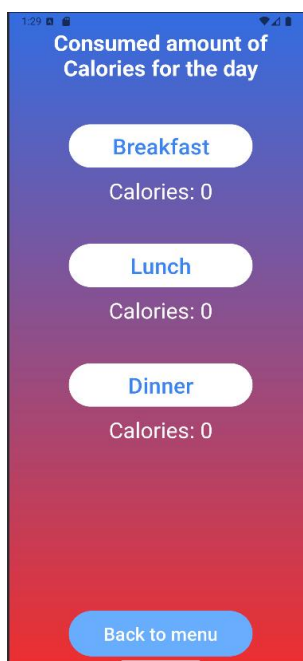


Рисунок 4.16 – Екран для розрахунку спожитих калорій

Екран для підрахунку спожитих калорій має 4 кнопки. Перша кнопка відповідає сніданку, друга – обіду, третя – вечері. Кожна з описаних кнопок має підпис, який відповідає кількості калорій, спожитих за певний прийом їжі. Остання кнопка «Back to menu» відповідає за повернення на головний екран застосунку.

При натисканні на кнопку, що відповідає прийому їжі, відкривається екран з можливістю вибору спожитих страв та інгредієнтів, на випадок відсутності необхідної страви. На рисунку 4.17 показано приклад такого екрану.

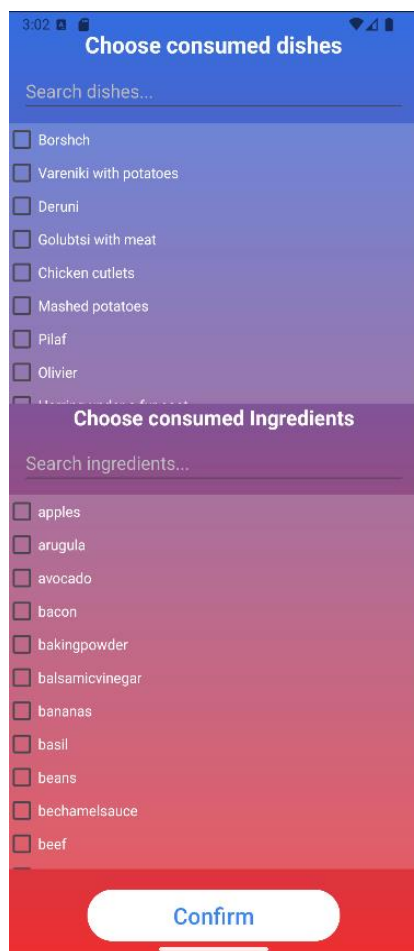


Рисунок 4.17 – Екран вибору страв та інгредієнтів, спожитих за певний прийом їжі

На даному екрані, користувач може обрати страви зі списку флажків зверху, або інгредієнти, зі списку флажків знизу. Дані списки мають функцію прокручування, тобто користувач може гортати список до необхідного йому продукту. Крім того, є поля пошуку страв та інгредієнтів. При введенні даних,

страви чи інгредієнти відфільтровуються, полегшуючи пошук необхідного продукту.

Внизу сторінки розміщена кнопка «Confirm», натиснувши на яку, застосунок зберігає обрані страви та інгредієнти та підраховує їх калорійність. Після чого, відкривається екран підрахунку калорій, де відображається кількість спожитих калорій під відповідною кнопкою.

Наступною кнопкою головного меню є кнопка «Generate menu based on ingredients». При натисканні, дана кнопка переносить на екран вибору наяних та заборонених інгредієнтів, як показано на рисунку 4.18.

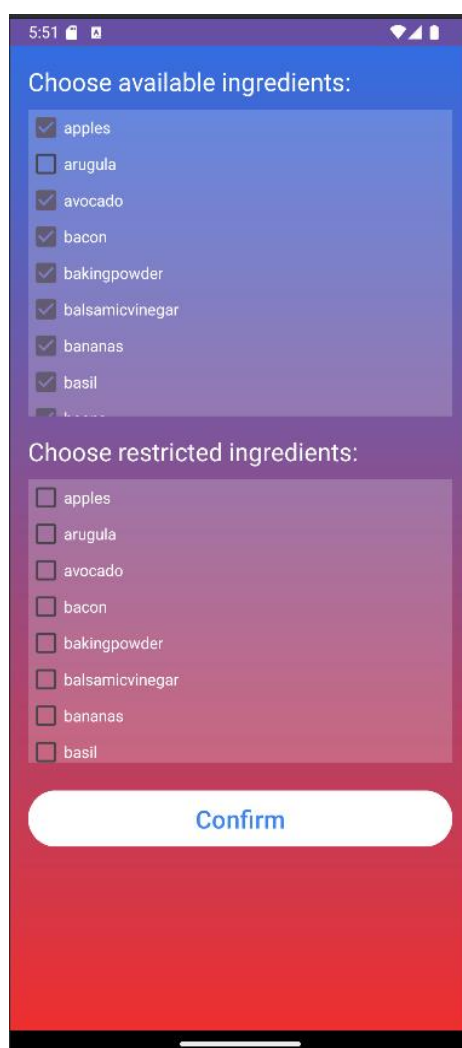


Рисунок 4.18 – Екран вибору наяних та заборонених інгредієнтів

Даний екран має два поля з флажками з можливістю прокручування. Зверху рошташовані флажки для наяних інгредієнтів, знизу – для заборонених.

Також внизу екрану є кнопка «Confirm». При натисканні на дану кнопку, застосунок зберігає дані в базі даних, після чого виконує пошук оптимальних блюд на основі наявних і відсутніх інгредієнтів, а також загальної витрати калорій протягом дня і цілі, обраної користувачем.

При наступному відкритті даного екрану, флажки будуть підтянуті з бази даних, що спростить повторний вибір інгредієнтів.

Після проведення розрахунків, застосунок відкриє екран з рекомендаціями меню на день, погрупованих по 5 страв, як показано на рисунку 4.19.

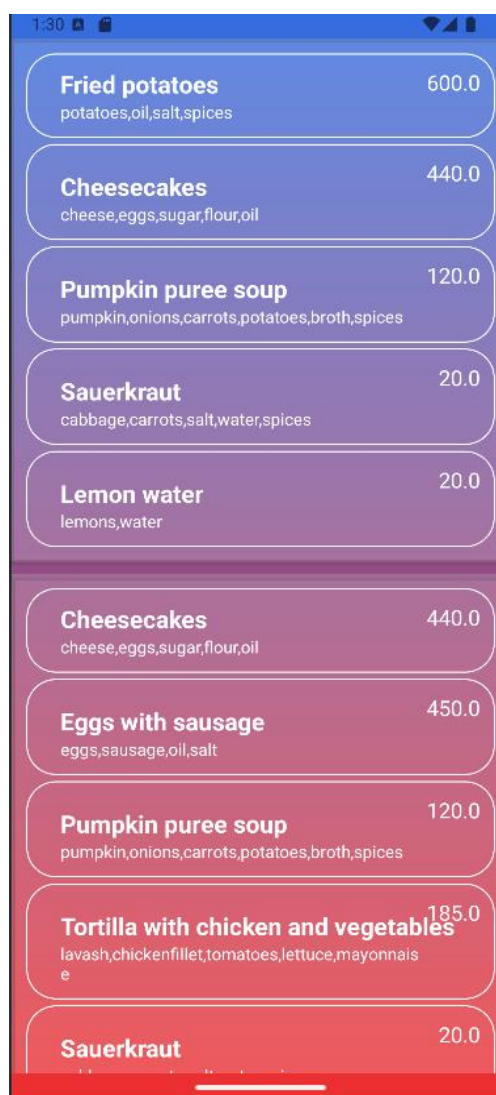


Рисунок 4.19 – Екран з рекомендаціями меню на день

Як працює рекомендація меню зображено на блок-схемі рекомендації меню на основі інгредієнтів в додатку 3.

Кожна страва має короткий опис, який складається з таких даних:

- назва страви;
- кількість калорій на одну порцію;
- перелік інгредієнтів.

При натисканні на будь-яку зі страв відкривається екран з більш детальним описом, як показано на рисунку 4.20.

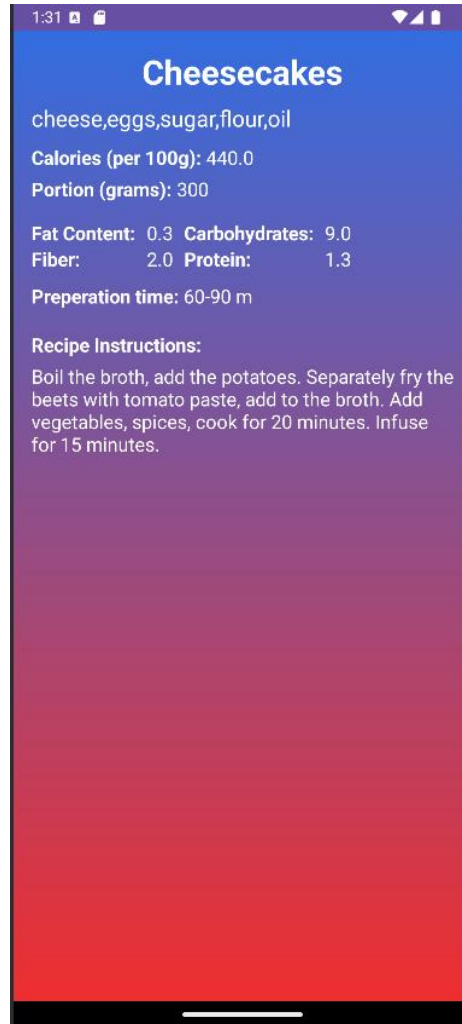


Рисунок 4.20 – Екран з описом страви

Даний екран має наступну інформацію про страву:

- назва страви;
- інгредієнти, необхідні для страви;
- кількість калорій на 100 грам;
- середня порція страви;
- кількість жирів в грамах;
- кількість вуглеводів в грамах;

- кількість клітковини в грамах;
- кількість протеїну в грамах;
- середній час приготування страви;
- рецепт для приготування страви.

Останньою кнопкою головного меню є кнопка «Generate menu». Вони відкриває екран з рекомендаціями меню на день, погруповних по 5 страв.

Дані меню відрізняються від попередніх тим, що ці меню створюються лише на основі загальної добової витрати енергії та цілі користувача, без врахування наявних та заборонених інгредієнтів. Таким чином, користувач може проглянути всі можливі меню, що підходять по калорійності.

Більш детально взаємодію користувача та пристроїв Інтернету речей з системою зображено на BPMN діаграмі в додатку Ж.

Висновок до розділу 4

У даному розділі було здійснено комплексний аналіз розробленої інформаційної системи для підтримки здорового способу життя на основі Інтернету речей. Описано апаратну складову системи та середовище розробки. Також описано мінімальні вимоги до пристроїв Інтернету речей для забезпечення нормальної роботи системи.

Особливу увагу приділено розгляду підходів до створення клієнтського інтерфейсу. Описано принципи дизайну, які дозволяють зробити взаємодію з системою інтуїтивно зрозумілою, зручною та привабливою для користувачів, зокрема для студентів, які прагнуть оптимізувати свій спосіб життя.

Крім того, наведено детальний опис інтерфейсу користувача, включаючи графічне зображення компонентів системи, їх функціональні можливості та взаємодію між ними. Це дозволило продемонструвати, як система підтримує вирішення завдань, пов'язаних із моніторингом фізичної активності, контролем харчування та іншими аспектами здорового способу життя.

5 РОЗРОБЛЕННЯ СТАРТАП-ПРОЄКТУ

Наступні розділи цього стартап-проєкту деталізують ідею, ринковий потенціал, стратегії впровадження та маркетинговий план.

Процес створення стартапу включає в себе такі етапи:

- опис ідеї проєкту;
- технологічний аудит ідеї проєкту;
- аналіз ринкових можливостей запуску стартап-проєкту;
- розроблення ринкової стратегії проєкту;
- розроблення маркетингової програми стартап-проєкту;
- висновки.

5.1 Опис ідеї проєкту

Проєкт «Інформаційна система для підтримки здорового способу життя на основі IoT» спрямований на розробку мобільного застосунку, який дозволить користувачам контролювати власну фізичну активність і харчування завдяки інтеграції зі смарт-годинниками. Система автоматично відстежуватиме кількість кроків, пульс та витрачені калорії, надаючи індивідуальні рекомендації щодо харчування на основі введених даних про вагу, зріст та вік. Таким чином, вона забезпечує користувача чітким планом для підтримання здорового способу життя.

Створення такого застосунку є актуальним у контексті зростаючого попиту на інструменти цифрового здоров'я, що поєднують зручність, доступність та індивідуалізацію. Завдяки поєднанню технологічних можливостей IoT та науково обґрунтованих підходів до здорового способу життя, проєкт має всі передумови для успішного ринкового старту та широкого впровадження.

Основні функції застосунку включають:

- автоматичне отримання даних зі смарт-годиннику;
- автоматичний підрахунок кількості кроків та спалювання калорій;
- моніторинг пульсу;

- персоналізовані рекомендації щодо споживання калорій на основі введених даних про вагу, зріст, вік та мету;
- визначення індексу маси тіла (ВМІ), базального метаболізму (ВМР) і загальної добової витрати енергії (ТДЕЕ);
- створення персоналізованого плану харчування на день з урахуванням доступних інгредієнтів, заборонених продуктів та необхідної калорійності;
- відображення даних користувача у вигляді графіків та статистики.

Застосунок орієнтований на споживачів, які прагнуть вести здоровий спосіб життя та ефективно слідкувати за своїм фізичним станом.

Після аналізу ідеї проекту, його можливих сфер застосування та потенційної користі для користувачів, було отримано результати, представлені в таблиці 5.1.

Таблиця 5.1 – Опис ідеї стартап-проекту

Зміст ідеї	Напрямки застосування	Вигоди для користувача
Інформаційна система для підтримки здорового способу життя на основі Інтернету речей	Фітнес та фізична активність	Зручний контроль за фізичною активністю та пульсом, мотивація досягнення цілей.
	Контроль ваги	Точний розрахунок необхідних калорій для схуднення, підтримки або набору ваги.
	Здоровий спосіб життя	Персоналізовані рекомендації щодо харчування та фізичних навантажень.
	Оптимізація використання продуктів	Планування страв із наявних інгредієнтів, економія часу.

Нижче наведено головні відмінності системи від існуючих аналогів:

- гнучке планування харчування;
- інтеграція зі смарт-годинниками;
- урахування індивідуальних параметрів для розрахунків і рекомендацій;
- інтуїтивно зрозумілий інтерфейс.

Зведені дані про сильні, слабкі та нейтральні сторони ідеї проекту в порівнянні з її конкурентами наведено в Таблиці 5.2.

Таблиця 5.2 – Визначення сильних, слабких та нейтральних характеристик ідеї проекту

№ п/п	Техніко-економічні характеристики ідеї	(потенційні) товари/концепції конкурентів					W	N	S
		Мій проєкт	Lose It!	Calorie Counter App: Fooducate	My diet meal plan	Super-Cook			
1	Інтеграція з IoT-пристроями	так	так	ні	ні	ні			S
2	Автоматичний розрахунок витрачених калорій	так	так	так	так	ні		N	
2	Автоматичний розрахунок витрачених калорій	так	так	так	так	ні		N	
3	Генерація меню на день з урахуванням наявних інгредієнтів	Враховує калорії та продукти	ні	ні	Не враховує продукти	Ручний вибір меню			S
4	Розрахунок BMI, BMR, TDEE	так	так	ні	так	ні		N	

№ п/п	Техніко-економічні характеристики ідеї	(потенційні) товари/концепції конкурентів					W	N	S
		Мій проєкт	Lose It!	Fooducate	My diet meal plan	SuperCook			
5	Надання рекомендацій щодо харчування	з урахуванням цілей та наявних продуктів	Лише вживання калорій	Лише вживання калорій	Частковий підхід	лише рецептів			S
6	Інтерфейс користувача	Інтуїтивний зрозумілий	Довге налаштування	Незручний	Інтуїтивний зрозумілий	Незручний		N	
7	Введення заборонених продуктів	так	ні	ні	ні	ні			S
8	Візуалізація результатів та прогресу	Графіки, динамічні звіти	лише в преміум	лише в преміум	Звіти	ні			S
9	Вартість	низка	середня	Висока	низька	Безкоштовно		N	
10	Офлайн-режим	так	так	так	ні	ні			S

Система вирізняється своєю функціональною гнучкістю, глибокою персоналізацією і підтримкою ключових функцій, які не реалізовані або слабо розвинені у конкурентів. Вона має великий потенціал завдяки своїм сильним

сторонам, однак потребує нейтральної цінової політики для утримання конкурентної переваги.

5.2 Технологічний аудит ідеї проєкту

У рамках стартап-проєкту було проведено технологічний аудит ідеї проєкту. Результати аналізу якого наведено в Таблиці 5.3.

Таблиця 5.3 – Технологічна здійсненність ідеї проєкту

№ п/п	Ідея проєкту	Технології її реалізації	Наявність технологій	Доступність технологій
1	Мова програмування	Java	Наявна	Доступна
		Kotlin	Наявна	Доступна
Обрана технологія реалізації ідеї проєкту: Java				
2	Робота з BLE	Android Bluetooth API	Наявна	Доступна
		Nordic BLE Library	Наявна	Доступна
Обрана технологія реалізації ідеї проєкту: Android Bluetooth API				
3	Управління даними в застосунку	SQLite	Наявна	Доступна
		Room Database	Наявна	Доступна
Обрана технологія реалізації ідеї проєкту: SQLite				
4	Користувацький інтерфейс	Android Studio + XML	Наявна	Доступна
		Jetpack Compose	Наявна	Доступна
Обрана технологія реалізації ідеї проєкту: Android Studio + XML				
5	Архітектура для розробки	MVVM	Наявна	Доступна
		MVP	Наявна	Доступна
Обрана технологія реалізації ідеї проєкту: MVVM				

Технологічна здійсненність ідеї проєкту є високою завдяки наявним інструментам, можливості їх адаптування та платформам для реалізації.

5.3 Аналіз ринкових можливостей запуску стартап-проєкту

Ринок систем для підтримання здорового способу життя є динамічним, з високим потенціалом зростання. Ринок демонструє середнє зростання на рівні 20–25% щорічно, що пов'язано з інтересом до здорового способу життя, інноваційних технологій, та збільшенням використання IoT-пристроїв.

Для аналізу ринкових можливостей запуску стартап-проєкту було проаналізовано попит і отримано результати, які наведені в таблиці 5.4.

Таблиця 5.4 – Попередня характеристика потенційного ринку стартап-проєкту

№ п/п	Показники стану ринку (найменування)	Характеристика
1	Кількість головних гравців, од	10–15
2	Загальний обсяг продаж, грн/ум.од	12–15 млрд доларів на рік.
3	Динаміка ринку (якісна оцінка)	Зростає
4	Наявність обмежень для входу (вказати характер обмежень)	Висока конкуренція, вимоги до захисту даних користувачів (GDPR, CCPA), витрати на маркетинг та просування.
5	Специфічні вимоги до стандартизації та сертифікації	Дотримання норм захисту персональних даних, стандартів безпеки IoT-пристроїв, вимог магазинів застосунків
6	Середня норма рентабельності в галузі (або по ринку), %	15–25%, залежно від монетизації

Потенційний ринок характеризується швидким зростанням і високою конкуренцією, проте завдяки інтеграції IoT-пристроїв, стартап має значні перспективи для зайняття своєї ніші.

Далі проведено аналіз потенційних груп клієнтів та їх характеристики, що дало можливість сформулювати перелік вимог до товару для кожної групи. Результати аналізу наведено в таблиці 5.5.

Таблиця 5.5 – Характеристика потенційних клієнтів стартап-проєкту

№ п/п	Потреба, що формує ринок	Цільова аудиторія (цільові сегменти ринку)	Відмінності у поведінці різних потенційних цільових груп клієнтів	Вимоги споживачів до товару
1	Підтримка здорового способу життя та персоналізовані рекомендації	– люди, що бажають змінити вагу; – люди, що ведуть активний спосіб життя; – люди старшого віку.	– орієнтація на відстеження калорій; – орієнтуються на контроль активності; – збереженні здоров'я.	– просте та зручне відстеження харчування; – зрозумілий інтерфейс; – урахування обмеження продуктів; – безпека даних.

Цільова аудиторія стартапу складається з людей різних вікових груп і з різними цілями щодо здоров'я. Особливо зацікавлені у застосуванні застосунку користувачі, які хочуть контролювати своє здоров'я, активність та харчування.

Далі було проведено аналіз ринкового середовища. У процесі дослідження було виділено основні фактори, які сприяють ринковому впровадженню проєкту та ті, які йому перешкоджають. Результати аналізу наведено в таблицях 5.6–5.7.

Таблиця 5.6 – Фактори загроз

п/п	Фактор	Зміст загрози	Можлива реакція компанії
1	Висока конкуренція	Присутність численних відомих брендів на ринку застосунків для здоров'я може ускладнити вихід на ринок.	Підвищення інноваційності продукту, акцент на унікальності, персоналізації, інтеграція з IoT-пристроями

п/п	Фактор	Зміст загрози	Можлива реакція компанії
2	Проблеми з безпекою	Загроза витоку персональних даних користувачів.	Забезпечення відповідності стандартам безпеки.
3	Зміна законодавства	Потенційні зміни в щодо захисту персональних даних та здоров'я	Моніторинг змін у законодавстві, адаптація продукту під вимоги.
4	Технічні проблеми з інтеграцією IoT	Проблеми з інтеграцією або сумісністю смарт-годинників можуть призвести до неефективної роботи продукту.	Постійне оновлення програмного забезпечення для підтримки нових моделей пристроїв, співпраця з провідними виробниками.
5	Низька адаптація користувачів до нових технологій	Споживачі можуть бути не готові до інтеграції з IoT-пристроями або нового підходу до контролю здоров'я.	Спрощення інтерфейсу, розробка інтерактивних інструкцій, навчальних матеріалів, підтримка 24/7.
6	Економічні спади	Економічні труднощі можуть призвести до зменшення витрат користувачів на застосунки для здоров'я.	Пропозиція доступних версій застосунку з основними функціями, лояльність до різних цінових сегментів.
7	Технологічна застарілість	Швидкий розвиток технологій IoT може зробити продукт застарілим через відсутність оновлень.	Постійні інвестиції в розробку нових функцій та оновлення технологій.
8	Негативні відгуки користувачів	Негативні відгуки можуть вплинути на репутацію бренду, знизити довіру.	Покращення якості продукту, оперативне реагування на відгуки.

Таблиця 5.7 – Фактори можливостей

п/п	Фактор	Зміст можливості	Можлива реакція компанії
1	Розширення ринку	Зростаючий попит на здоровий спосіб життя та фітнес-технології.	Активне просування продукту через маркетингові кампанії
2	Поширення IoT-пристроїв	Збільшення кількості IoT-пристроїв, що взаємодіють з застосунками для здоров'я.	Розширення підтримки нових моделей пристроїв, співпраця з виробниками пристроїв.
3	Підвищення інтересу до персоналізації	Зростаючий попит на персоналізовані рішення для здоров'я, харчування та фізичної активності.	Покращення персоналізованих алгоритмів та функцій для рекомендацій, що враховують індивідуальні особливості.
4	Віддалене здоров'я	Популяризація віддалених медичних консультацій.	Інтеграція з платформами надання консультацій онлацн.
5	Зростання попиту на здорове харчування	Збільшення уваги до здорового харчування, органічних продуктів та дієт.	Розширення застосунку для підтримки здорового харчування та рекомендацій меню.
6	Адаптація до нових ринків та культур	Вихід на нові ринки з адаптацією для культурних і харчових звичок.	Локалізація застосунку для різних мовних та культурних регіонів.
7	Інвестування в технології здоров'я	Підвищена увага до інвестицій в технології, що покращують здоров'я.	Залучення інвестицій для розвитку нових функцій та пошуку нових партнерів.

Даний проєкт має значні можливості для розвитку в умовах зростаючого попиту на здоровий спосіб життя, персоналізацію та інтеграцію з новими IoT-пристроями. Окрім того, зростаючий інтерес до онлайн-послуг та технологій здоров'я відкриває можливості для подальшого вдосконалення продукту.

Для аналізу пропозиції було визначено загальні риси конкуренції проєкту на ринку. Результати аналізу об'єднано в таблиці 5.8.

Таблиця 5.8 – Ступеневий аналіз конкуренції на ринку

Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)
1. Тип конкуренції Чиста конкуренція	Компанії встановлюють власні ціни на свої мобільні застосунки.	Створення унікальних переваг: інтеграція з більшою кількістю IoT-пристроїв, покращення функцій.
2. За рівнем конкурентної боротьби Міжнародний рівень	Ринок мобільних застосунків для підтримання здорового способу життя не зосереджений на певній території.	Акцент на підтримці різних культурних особливостей. Просування через онлайн-маркетинг, співпраця з місцевими брендами IoT-пристроїв.
3. За галузевою ознакою Внутрішньогалузева	Конкуренція не виходить за межі застосунків для підтримки здорового способу життя.	Забезпечення комплексного моніторингу для покращення конкурентної позиції на ринку.
4. Конкуренція за видами товарів Товарно-видова	Конкуренція між схожими застосунками, які пропонують частково схожі функції.	Розробка та покращення унікальних функцій, розширена аналітика з графіками, персоналізовані рекомендації.
5. За характером конкурентних переваг Нецінова конкуренція	Конкуренція базується на функціоналі, зручності інтерфейсу, персоналізації та додаткових послугах.	Забезпечення зрозумілого інтерфейсу, високої продуктивності та широких можливостей.

Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)
6. За інтенсивністю Не марочна	Користувачі часто віддають перевагу зручності, ніж бренду в плані застосунків.	Покращення інтерфейсу, партнерства з іншими застосунками, розвиток підтримки користувачів для підвищення рівня задоволеності.

Ринок характеризується високою конкуренцією. Для забезпечення конкурентоспроможності проекту основну увагу акцентовано на створенні унікальних функцій, персоналізації та комплексному підході.

Далі було проведено аналіз конкурентного середовища за моделлю М. Портера, щоб оцінити інтенсивність конкуренції в галузі. Розглянуто такі фактори: прямі конкуренти, потенційні нові конкуренти, постачальники, клієнти та загроза з боку заміників. Результати цього аналізу представлені в таблиці 5.9.

Таблиця 5.9 – Аналіз конкуренції в галузі за М. Портером

	Прямі конкуренти в галузі	Потенційні конкуренти	Постачальники	Клієнти	Товари-замінники
Складові і аналізу	Lose It!, Fooducate, Supercook, My Diet Meal Plan.	Високий рівень конкуренції, технічні та регуляторні вимоги.	Висока сила впливу виробників IoT та Android пристроїв через сумісність.	Прибутки користувача, вимоги до персоналізації, контроль безпеки даних.	Альтернативні рішення, ручний підрахунок калорій, безкоштовні онлайн-інструменти

	Прямі конкуренти в галуз	Потенційні конкуренти	Постачальники	Клієнти	Товари-замінники
Висновки	Інтенсивна конкуренція: багато гравців, постійна розробка нових функцій.	Для входу в ринок потрібні інновації, потенційні конкуренти є.	встановлення умов та обмеження доступу до деяких функцій платформ.	Очікування простого інтерфейсу, високої якості та функцій без додаткової плати.	Низька вартість замінників та їхня простота у використанні.

На основі проведеного аналізу можна зробити кілька важливих висновків.

Ринок характеризується високою інтенсивністю конкуренції, оскільки на ньому вже працюють відомі гравці, які здобули визнання серед користувачів і створюють значний бар'єр для нових учасників.

Можливості входу в ринок є відкритими. Однак, для успішного входу необхідно запропонувати унікальний продукт із функціями, які відрізняються від конкурентів. Наприклад, це можуть бути автоматичне створення меню на основі наявних інгредієнтів або використання AI-рекомендацій.

Сила постачальників IoT-пристроїв є значною, оскільки вони контролюють доступ до API, що може обмежувати можливості нових застосунків. Для зниження цього впливу необхідно забезпечити інтеграцію з найпопулярнішими пристроями.

Клієнти також мають значну силу, оскільки мають широкий вибір застосунків. Це змушує компанії зосереджуватися на простоті, функціональності та привабливих цінах, щоб задовольнити високі вимоги користувачів.

Загрози з боку замінників оцінюються на середньому рівні. Традиційні методи є менш функціональними, але водночас простішими у використанні.

Для аналізу і оцінки конкурентоспроможності проєкту було проведено обґрунтування факторів конкурентоспроможності і висвітлено результати в таблиці 5.10.

Таблиця 5.10 – Обґрунтування факторів конкурентоспроможності

п/п	Фактор конкурентоспроможності	Обґрунтування
1	Інтеграція з IoT-пристроями	Забезпечує автоматизований збір даних, що полегшує моніторинг здоров'я. Конкуренти часто мають обмежену інтеграцію з пристроями або підтримують лише власні гаджети.
2	Персоналізація рекомендацій	Автоматичне генерування рекомендації щодо харчування на основі даних користувача, чого бракує конкурентам.
3	Унікальний функціонал (меню з інгредієнтів)	Генерація меню на основі продуктів, які є в наявності у користувача, виділяє продукт серед конкурентів.
4	Простота та інтуїтивність інтерфейсу	Простий і зрозумілий інтерфейс підвищує привабливість продукту, враховуючи, що багато конкурентів мають складні або перевантажені інтерфейси.
5	Комплексний підхід (харчування, активність, вода)	Надає користувачам можливість стежити за багатьма аспектами здоров'я в одному застосунку, що зменшує потребу в кількох різних програмах.
6	Цінова доступність	Забезпечення безкоштовного доступу до основних функцій або низька вартість преміум-функцій робить продукт більш привабливим для масової аудиторії.

п/п	Фактор конкурентоспроможності	Обґрунтування
7	Висока якість обслуговування клієнтів	Забезпечення підтримки та надання корисної інформації клієнтам сприяє формуванню лояльності, що є ключовим для довгострокового успіху на ринку.

Фактори конкурентоспроможності проекту базуються на аналізі ринку, характеристиках проекту та очікуваннях споживачів. Основними сильними сторонами є інтеграція з IoT-пристроями, унікальний функціонал, комплексний підхід до здоров'я, а також простота інтерфейсу. Ці фактори дозволяють створити продукт, що виділяється на фоні існуючих рішень і задовольняє потреби клієнтів.

За результатами попереднього дослідження було проведено аналіз сильних та слабких сторін проекту. Результати наведено в Таблиці 5.11.

Таблиця 5.11 – Порівняльний аналіз сильних та слабких сторін стартап-проекту «Інформаційна система для підтримки здорового способу життя на основі Інтернету речей»

№ п/п	Фактор конкурентоспроможності	Бали (1–20)	Рейтинг товарів-конкурентів у порівнянні з проектом						
			-3	-2	-1	0	+1	+2	+3
1	Інтеграція з IoT-пристроями	18						+	
2	Персоналізація рекомендацій	19							+
3	Унікальний функціонал (меню з інгредієнтів)	20							+
4	Простота та інтуїтивність інтерфейсу	17						+	
5	Комплексний підхід	18							+
6	Доступність на різних платформах	16					+		
7	Інтеграція штучного інтелекту для аналітики	18						+	

№ п/п	Фактор конкурентоспроможності	Бали (1–20)	Рейтинг товарів-конкурентів у порівнянні з проектом						
			–3	–2	–1	0	+1	+2	+3
8	Локалізація продукту	17						+	
9	Цінова доступність	15					+		
10	Висока якість обслуговування клієнтів	16					+		

Зібрані дані дозволяють провести детальний SWOT-аналіз проекту, виявивши його сильні та слабкі сторони, а також можливості та загрози. Результати аналізу представлені в таблиці 5.12.

Таблиця 5.12 – SWOT-аналіз стартап-проекту

Сильні сторони	Слабкі сторони
<ul style="list-style-type: none"> – інтеграція з IoT-пристроями, що забезпечує високу технологічність; – простий і зручний інтерфейс для користувачів; – інтеграція аналітичних даних; – рекомендація меню на основі розрахунків та наявних інгредієнтів. 	<ul style="list-style-type: none"> – обмежені фінансові ресурси; – невеликий початковий асортимент функцій у порівнянні з деякими конкурентами; – залежність від точності та доступності IoT-пристроїв.
Можливості	Загрози
<ul style="list-style-type: none"> – зростання попиту на технології, що підтримують здоровий спосіб життя; – потенційна інтеграція з новими IoT-пристроями (розширення аудиторії); – розширення функціоналу для більш комплексного моніторингу здоров'я; – залучення партнерів із медичного або фітнес-сектору. 	<ul style="list-style-type: none"> – висока конкуренція з боку вже існуючих; – ризик збоїв або втрати довіри через проблеми з безпекою даних; – залежність від змін законодавства щодо захисту персональних даних.

Стартап-проект має сильні конкурентні переваги завдяки інтеграції IoT, простому інтерфейсу та аналітичним можливостям. Щоб бути конкурентоспроможним, слід зосередитись на вирішенні слабких сторін (масштабування функціоналу, фінансування) та реалізації потенційних можливостей (партнерства, розширення функцій). Особливу увагу потрібно приділити подоланню загроз конкуренції та захисту даних.

На основі SWOT-аналізу проведено дослідження альтернатив ринкового впровадження проекту, розглянуто різні стратегії виходу на ринок для залучення інвестицій або клієнтів. Перелік можливих підходів до впровадження стартапу наведено в таблиці 5.13.

Таблиця 5.13 – Альтернативи ринкового впровадження стартап-проекту

№ п/п	Альтернатива (орієнтовний комплекс заходів) ринкової поведінки	Ймовірність отримання ресурсів	Строки реалізації
1	Початковий запуск продукту із базовими функціями.	Висока	3–6 місяців
2	Розробка та впровадження розширеного функціоналу: персоналізовані рекомендації, розширення бази даних харчування та фізичної активності.	Середня	6–12 місяців
3	Співпраця з партнерами: створення акційних пропозицій для користувачів партнерів.	Мала	9–15 місяців
4	Впровадження функціоналу для синхронізації з новими IoT-пристроями та інтеграція з іншими платформами.	Висока	9–15 місяців
5	Проведення маркетингової кампанії: просування у соціальних мережах, співпраця з інфлюенсерами у сфері здоров'я та фітнесу.	Висока	3–6 місяців

Обрана альтернатива – початковий запуск продукту із базовими функціями. Цей вибір обґрунтований тим, що є висока ймовірність отримання ресурсів через оригінальний функціонал та доступність IoT-пристроїв. Також стислі строки реалізації сприяють зайнятті ніші на ринку та залученню початкової аудиторії. Успішна реалізація цієї альтернативи створить основу для подальшого розвитку проекту, включаючи впровадження розширеного функціоналу та партнерств.

5.4 Розроблення ринкової стратегії проекту

Для розроблення ринкової стратегії, в першу чергу було проаналізовано та зроблено опис цільових груп потенційних споживачів. Результати аналізу представлено в таблиці 5.14.

Таблиця 5.14 – Вибір цільових груп потенційних споживачів

№ п/п	Опис профілю цільової групи потенційних клієнтів	Готовність споживачів сприйняти продукт	Орієнтовний попит в межах цільової групи (сегменту)	Інтенсивність конкуренції в сегменті	Простота входу у сегмент
3	Молодь та студенти, що цікавляться здоровим способом життя та експериментують з новими технологіями для моніторингу своєї фізичної активності.	Висока	Середній	Висока	Легка
Які цільові групи обрано: Молодь та студенти, що цікавляться здоровим способом життя					

Для успішної роботи в обраних сегментах ринку було проведено дослідження базових стратегій розвитку та обрано одну із стратегій. Результати визначення базової стратегії розвитку стартап-проекту наведено в таблиці 5.15.

Таблиця 5.15 – Визначення базової стратегії розвитку

№ п/п	Обрана альтернатива розвитку проекту	Стратегія охоплення ринку	Ключові конкуренто-спроможні позиції відповідно до обраної альтернативи	Базова стратегія розвитку
1	Проект, що пропонує унікальний продукт для певного сегмента	Стратегія диференціації	– унікальні характеристики товару; – висока відмінність від конкурентів; – підвищена лояльність клієнтів.	Стратегія диференціації, орієнтована на створення товару, який буде відрізнятися від конкурентів.

Далі було вибрано стратегії конкурентної поведінки і зроблено аналіз, результати якого висвітлено в Таблиці 5.16.

Таблиця 5.16 – Визначення базової стратегії конкурентної поведінки

№ п/п	Чи є проект «першопрохідцем» на ринку?	Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?	Чи буде компанія копіювати основні характеристики товару конкурента, і які?	Стратегія конкурентної поведінки
1	Ні.	Компанія буде орієнтуватися на існуючих споживачів.	Компанія буде копіювати конкурентів, але додаватиме нові можливості	Стратегія наслідування лідеру.

Наведена вище таблиця допомагає оцінити, яку конкурентну стратегію було обрано для успішного виходу на ринок, зважаючи на характеристики конкурентів та ринкові можливості.

Наступним кроком було визначення стратегії позиціонування. Результати наведено в таблиці 5.17.

Таблиця 5.17 – Визначення стратегії позиціонування

№ п/п	Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкуренто-спроможні позиції власного стартап-проєкту	Вибір асоціацій, які мають сформувати комплексну позицію власного проєкту (три ключових)
1	Можливість отримати рекомендації на основі персональних даних	Стратегія заняття конкурентної ніші	– генерація меню на основі інгредієнтів; – пропозиції для здорового способу життя; – комплексний підхід.	Піклування про здоров'я, персоналізація рішень

Стартап-проєкт визначив для себе чітку ринкову позицію, яка забезпечить його ідентифікацію споживачами, відповідність їх очікуванням і конкурентоспроможність на ринку.

5.5 Розроблення маркетингової програми стартап-проєкту

Для успішного розроблення маркетингової програми було проаналізовано та визначено ключові переваги концепції потенційного товару, які наведено в таблиці 5.18.

Таблиця 5.18 – Визначення ключових переваг концепції потенційного товару

№ п/п	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити)
1	Персоналізація рекомендацій щодо харчування та фізичної активності	Точні рекомендації на основі даних про вагу, вік, активність та цілі користувача	– унікальна функція генерації меню на основі наявних інгредієнтів; – персоналізовані тренувальні програми.
2	Інтеграція з IoT-пристроями	Автоматичне отримання даних у реальному часі (кроки, пульс, сон)	– широка інтеграція з популярними IoT-пристроями; – можливість синхронізації з медичними гаджетами.
3	Простий та зручний інтерфейс	Інтуїтивне управління застосунком, зрозумілі графіки та аналітика	– простота у використанні навіть для новачків; – візуалізація даних у зручному для розуміння вигляді.
4	Мотивація до досягнення здорового способу життя	Гейміфікація (змагання, виклики) та соціальна інтеграція	– інтерактивні функції, спільноти користувачів, мотиваційні програми.
5	Комплексний підхід до моніторингу здоров'я	Моніторинг харчування, фізичної активності, сну та водного балансу	– універсальність продукту – всі функції в одному застосунку, що усуває потребу використовувати кілька програм.
6	Доступність	Безкоштовна базова версія з розширеними функціями у преміумі	– конкурентоспроможна ціна; – висока цінність продукту для користувача.

Даний аналіз допомагає чітко сформулювати маркетингову концепцію товару, яка виділяє стартап-проект серед конкурентів і орієнтується на задоволення ключових потреб споживачів.

Далі розроблено трирівневу маркетингову модель товару, яка показана в таблиці 5.19.

Таблиця 5.19 – Опис трьох рівнів моделі товару

Рівні товару	Сутність та складова		
I. Товар за задумом	Інформаційна система для підтримки здорового способу життя на основі Інтернету речей		
II. Товар у реальному виконанні	Властивості/характеристики	М/Нм	Вр/Тх /Тл/Е/Ор
	Якість: стандарти		
	Пакування: арк		
III. Товар із підкріпленням	До продажу: інформаційна система		
	Після продажу: технічна підтримка		
За рахунок чого потенційний товар буде захищено від копіювання: за рахунок захисту інтелектуальної власності.			

Наступним кроком було визначено межі встановлення ціни і наведено їх в таблиці 5.20.

Таблиця 5.20 – Визначення меж встановлення ціни

№ п/п	Рівень цін на товари-замінники	Рівень цін на товари-аналоги	Рівень доходів цільової групи споживачів	Верхня та нижня межі встановлення ціни на товар/послугу
1	Від 300 до 400 грн/місяць	Від 900 до 2000 грн/місяць	Середній дохід: 40 000 грн/місяць	400 грн – 1700 грн/місяць

Аналіз показав, що замітники, як правило, пропонуються за більш низькою ціною порівняно з товарами-аналогами. Даний аналіз враховує цінові межі на основі аналізу ринку, рівня цін конкурентів і доходів цільової аудиторії.

Для ефективної реалізації стартап-проекту необхідно розробити чітку систему збуту. Можливі варіанти представлені в таблиці 5.21.

Таблиця 5.21 – Формування системи збуту

№ п/п	Специфіка закупівельної поведінки цільових клієнтів	Функції збуту, які має виконувати постачальник товару	Глибина каналу збуту	Оптимальна система збуту
1	Клієнти віддають перевагу онлайн-замовленням і цифровим платформам	<ul style="list-style-type: none"> – залучати сторонніх посередників для збуту; – пошук нових клієнтів шляхом маркетингових дій в соціальних мережах; – формування попиту; – дослідження економічного стану галузі для корекції цін на послуги. 	Один рівень	Система збуту через мобільні магазини (App Store, Google Play)

Ця таблиця враховує специфіку клієнтів, функції збуту, вибір глибини каналу збуту та оптимальну систему для ефективного продажу товару.

Останнім етапом у складанні маркетингової програми є аналіз концепцій маркетингових комунікацій. Обрану для даного проєкту концепцію наведено в Таблиці 5.22.

Таблиця 5.22 – Концепція маркетингових комунікацій

№ п/п	Специфіка поведінки цільових клієнтів	Канали комунікацій, якими користуються цільові клієнти	Ключові позиції, обрані для позиціонування	Завдання рекламного повідомлення	Концепція рекламного звернення
1	Клієнти віддають перевагу онлайн-замовленням через мобільні магазини	Соціальні мережі (Facebook, Instagram, TikTok), email-розсилки	Інноваційність, зручність, сучасний стиль	Привернути увагу до унікальних переваг товару	Емоційне звернення з акцентом на зручність та сучасність

Ця таблиця враховує різні аспекти поведінки клієнтів, їх улюблені канали комунікації, а також завдання рекламного повідомлення і основну концепцію.

Висновки до розділу 5

У цьому розділі було розглянуто створення стартап-проєкту інформаційної системи для підтримки здорового способу життя на основі Інтернету речей.

Аналіз ринку продемонстрував високий попит на продукти для підтримки здорового способу життя, особливо ті, які інтегруються з пристроями Інтернету речей. Ринок систем для моніторингу здоров'я демонструє середньорічне зростання на рівні 20–25%, що пов'язано з популярністю фітнес-технологій і цифрових рішень. Враховуючи унікальні функції проєкту, такі як персоналізовані

рекомендації, генерація меню на основі наявних інгредієнтів та інтеграція з різноманітними IoT-пристроями, проєкт має значний комерційний потенціал. Правильна цінова політика (поєднання безкоштовного функціоналу з преміум-підписками) забезпечить рентабельність і конкурентоспроможність.

Проєкт має чітко визначені цільові групи, серед яких молодь, що активно користується фітнес-гаджетами, та люди, які прагнуть контролювати своє здоров'я. Бар'єри входження, зокрема висока конкуренція та вимоги до захисту даних, долаються завдяки унікальному функціоналу. Конкурентоспроможність проєкту забезпечується комплексним підходом до здоров'я (моніторинг активності, харчування), інтуїтивним інтерфейсом та високою ціновою доступністю.

Для ринкової реалізації найдоцільнішою є стратегія диференціації, що зосереджується на унікальності функцій продукту. Генерація персоналізованих рекомендацій щодо харчування, інтеграція з IoT-пристроями та комплексний підхід виділяють проєкт серед конкурентів. Початковий запуск з базовим функціоналом дозволить швидко зайняти свою нішу, після чого можливе поступове розширення функцій через оновлення.

Проєкт доцільно впроваджувати, оскільки він має високий потенціал для успіху завдяки зростаючому ринку Інтернету речей та мобільних застосунків для підтримки здорового способу життя, значним конкурентним перевагам і запиту серед споживачів. Подальша імплементація повинна включати завершення розробки базової версії застосунку, запуск маркетингової кампанії через соціальні мережі та пошук партнерів серед виробників IoT-пристроїв. Це дозволить зайняти нішу проєкту на ринку та його розвиток у майбутньому.

ВИСНОВКИ

При написанні магістерської дисертації було розроблено інформаційну систему для підтримки здорового способу життя на основі Інтернету речей. Для демонстрації її роботи було розроблено мобільний застосунок з можливістю підключення до пристроїв Інтернету речей, таких як розумні годинники або фітнес-трекери.

Перед розробкою цієї інформаційної системи було розглянуто існуючі аналоги інформаційних систем для підтримки здорового способу життя, досліджено їх переваги та недоліки. На основі виявлених особливостей було сформовано ряд функціональних особливостей системи, які мають бути впроваджені в розробленій інформаційній системі. Виконання цих умов має задовільнити вимоги користувачів та надати їм готову систему зі зручним інтерфейсом та точним розрахунком результатів.

У магістерській дисертації було проаналізовано існуючі рішення побудови інформаційних систем, а саме вид архітектури системи, мова програмування, середовище розробки та база даних, які використовувалися для розробки інформаційної системи. На основі аналізу було обрано розробку інформаційної системи у вигляді мобільного застосунку за допомогою Android Studio на мові програмування Java. Також було використано бібліотеку для взаємодії з Bluetooth пристроями. Для зберігання інформації було обрано реляційну СКБД SQLite. Дана система використовує архітектуру MVVM.

Також було описано інформаційне забезпечення системи. Даний пункт включає в себе вхідні дані, математичні формули для розрахунків, вихідні дані та структура бази даних. До вхідних даних системи належать дані, що вводяться користувачем, дані, що підтягуються з бази даних та дані, які зчитуються з пристроїв Інтернету речей.

До математичних формул було віднесено розрахунок індексу маси тіла, базального метаболізму, загальних добових витрат енергії та розрахунок

споживання води й корисних речовин у грамах. До цих корисних речовин відносяться білки, жири, вуглеводи та клітковина.

До вихідних даних належать дані, які відображаються користувачу. До них входять результати розрахунків, графіки, побудовані на основі отриманих даних, рекомендаційні меню, згенеровані на основі розрахунків та інгредієнтів, введених користувачем, дані, що зчитуються з пристроїв Інтернету речей, а також всі повідомлення успіху чи помилки роботи системи, які відображаються користувачу.

На основі результатів всіх попередніх пунктів було реалізовано інформаційну систему для підтримки здорового способу життя на основі Інтернету речей. Після чого було описано апаратну базу, середовище розробки та основні вимоги до пристроїв, у даному випадку для мобільних пристроїв на базі операційної системи Android та пристроїв Інтернету речей. Також було описано основні підходи до розробки клієнтського інтерфейсу, які враховувалися при розробці системи.

У магістерській дисертації було докладно описано інтерфейс користувача, його екрани та функції. Також було описано як працює кожен з елементів інтерфейсу. Всього в мобільному застосунку використано 10 екранів, кожен з яких охоплює певний функціонал і дає користувачу зручно взаємодіяти з розробленою системою.

Останнім розділом магістерської дисертації було розроблено стартап проєкт на основі реалізованої інформаційної системи. Під час розробки стартап-проєкту було проведено детальний аналіз ринку для визначення його основних тенденцій, конкурентного середовища та потреб цільової аудиторії. На основі отриманих даних сформульовано унікальну ціннісну пропозицію, що включає інноваційні функції, такі як автоматичне створення меню за наявними інгредієнтами. Також було розроблено базову архітектуру продукту, що враховує можливість інтеграції з популярними пристроями Інтернету речей. Крім того, створено бізнес-модель, яка відображає потенційні джерела доходу та витрати, а також стратегічний план виходу на ринок для ефективного залучення перших клієнтів.

Результати, досягнуті під час виконання магістерської дисертації, вказують на виконання всіх поставлених задач. У процесі роботи розроблено інформаційну систему для підтримки здорового способу життя на основі Інтернету речей. Система дозволяє відслідковувати ключові показники здоров'я, такі як фізична активність, серцевий ритм та споживання калорій. Також інформаційна система рекомендує користувачу харчування, яке генерується на основі загального добового споживання енергії, цілі та інгредієнтів, обраних користувачем.

Розроблена система повністю відповідає завданням, поставленим у рамках магістерської дисертації. Усі ключові вимоги, висунуті до функціональності, надійності та зручності використання системи, були реалізовані. Інтерфейс користувача спроектовано відповідно до стандартів дизайну, що сприяє інтуїтивності використання. Крім того, архітектура системи побудована з урахуванням можливості подальшого розширення функціоналу, що відповідає вимогам гнучкості програмного забезпечення.

Результати роботи мають високий потенціал для впровадження. На це вказують результати розробки стартап проєкту.

Наукова значущість роботи полягає в інтеграції технологій Інтернету речей у сферу здорового способу життя, що є актуальним напрямком сучасних досліджень. У науково-технічному аспекті робота цінна через впровадження алгоритмів обробки даних та створення персоналізованих рекомендацій щодо харчування.

Тематика роботи має значний потенціал для подальших досліджень. Зокрема, перспективним є удосконалення алгоритмів аналізу даних, використання штучного інтелекту для прогнозування ризиків для здоров'я та персоналізації рекомендацій. Також актуальним напрямком є забезпечення більшої сумісності системи з різними платформами та пристроями. Крім того, дана тема надає можливість впровадження великої кількості нових функцій, які полегшать догляд за здоров'ям. Продовження досліджень дозволить не лише підвищити ефективність роботи системи, а й зробити її доступною для ширшого кола користувачів.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Огляд автоматизованих систем охорони здоров'я / М. Кравець та ін. Адаптивні системи автоматичного управління. 2024. Т. 1, № 44. С. 16–23. URL: <https://doi.org/10.20535/1560-8956.44.2024.302195>.
2. Lose It! – Calorie counting made easy. Lose It! – Weight Loss That Fits. URL: <https://www.loseit.com/how-it-works/>.
3. Calorie Counter App: Fooducate. Version 6.7.0. Maple Media. URL: <https://play.google.com/store/apps/details?id=com.fooducate.nutritionapp>.
4. Meal Planner For Custom Diets & Macros | My Diet Meal Plan. My Diet Meal Plan. URL: <https://www.mydietmealplan.com>.
5. Supercook: recipe search by ingredients you have at home. SuperCook – Zero Waste Recipe Generator. URL: <https://www.supercook.com>.
6. Microservices vs. monolithic architecture | Atlassian. Atlassian. URL: <https://www.atlassian.com/microservices/microservices-architecture/microservices-vs-monolith>.
7. Monolithic vs Microservices Architecture: Pros and Cons | OpenLegacy. Legacy Integration for Core Legacy Systems | OpenLegacy. URL: <https://www.openlegacy.com/blog/monolithic-application>.
8. Mitrofanskiy K. Monolithic vs. Microservices Architecture: Main Differences. Intellisoft. URL: <https://intellisoft.io/monolith-vs-microservices-architecture-pros-and-cons/>.
9. Terra J. What is Client-Server Architecture? Everything You Should Know | Simplilearn. Simplilearn.com. URL: <https://www.simplilearn.com/what-is-client-server-architecture-article>.
10. What Is Client-Server Architecture? Coursera. URL: <https://www.coursera.org/articles/client-server-architecture>.
11. What is Client-Server Network? Definition, Advantages, and Disadvantages - zenarmor.com. Zenarmor - Agile Service Edge Security. URL: <https://www.zenarmor.com/docs/network-basics/what-is-client-server-network>.

12. Torunoğlu Z. MVC in Android. Medium. URL: <https://medium.com/@zorbeytorunoglu/mvc-in-android-c731f2e2b844>.

13. MVC Architecture | Ramotion Agency. Web Design, UI/UX, Branding, and App Development Blog. URL: <https://www.ramotion.com/blog/mvc-architecture-in-web-application/#section-mvc-architecture>.

14. MVP (Model View Presenter) Architecture Pattern in Android with Example – GeeksforGeeks. GeeksforGeeks. URL: <https://www.geeksforgeeks.org/mvp-model-view-presenter-architecture-pattern-in-android-with-example/>.

15. MVC, MVP and MVVM architecture pattern - Introduction. Mobile & Web App Development Company | USA, UK, Norway. URL: <https://itcraftapps.com/blog/mvc-mvp-and-mvvm-architecture-pattern-introduction/>.

16. Простая архитектура с использованием MVVM и делегатов в Android. Оптимальное решение для малых проектов. Хабр. URL: <https://habr.com/ru/articles/776344/>.

17. MVVM: Model-View-ViewModel Architecture | Ramotion Agency. Web Design, UI/UX, Branding, and App Development Blog. URL: <https://www.ramotion.com/blog/what-is-mvvm/#section-advantages-of-mvvm>.

18. ilyas ipek. How to Implement MVI with Delegates on Android. Medium. URL: <https://engineering.teknasyon.com/how-to-implement-mvi-with-delegates-on-android-f2aa1a842b73>.

19. Mishra A. MVI Architecture in Android - Scaler Topics. Scaler Topics. URL: <https://www.scaler.com/topics/android/mvi-architecture-android/>.

20. Tekin S. What is the Clean Architecture?. Medium. URL: <https://semihtekin.medium.com/what-is-the-clean-architecture-c80c2a2ff69a>.

21. YukiOnishi. Benefits and Drawbacks of Adopting Clean Architecture. DEV Community. URL: <https://dev.to/yukionishi1129/benefits-and-drawbacks-of-adopting-clean-architecture-2pd1>.

22. Pandya J. Microservices Architecture in Mobile App Development. Expert Mobile App Development Company | App Developers for Hire. URL:

<https://www.expertappdevs.com/blog/microservices-architecture-in-mobile-app-development>.

23. Advantages and Disadvantages of Microservices. Top Software Outsourcing Company in Vietnam - Orient Software. URL: <https://www.orientsoftware.com/blog/advantages-and-disadvantages-of-microservices/>.

24. Data Binding Library | Android Developers. Android Developers. URL: <https://developer.android.com/topic/libraries/data-binding>.

25. Реляційні бази даних. Дія.Освіта – IT-студії. URL: <https://it-osvita.diia.gov.ua/task/item/c1d56d2b-718f-4490-8498-ee209eac30d8>.

26. Sushama Pawar Chetashri Bhusari Yogita Jore. Java Programming. Independently Published, 2020.

27. Програмування баз даних на SQLite: перші кроки в освоєнні. FoxmindEd. URL: <https://foxminded.ua/prohramuvannia-baz-danykh-na-sqlite/>.

28. Hanna K. T., Wigmore I. What is a mobile app (mobile application)? – TechTarget Definition. WhatIs. URL: <https://www.techtarget.com/whatis/definition/mobile-app>.

29. Mastering Android Studio. Taylor & Francis Group, 2022. 248 p.

30. Çakmur H. Body Mass Index: Overweight, Normal Weight, Underweight. IntechOpen, 2023.