

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

**КВАЛІФІКАЦІЙНА РОБОТА
ЗА СТУПЕНЕМ ВИЩОЇ ОСВІТИ «БАКАЛАВР»**

**РОЗРОБКА МОБІЛЬНОГО ДОДАТКУ ДЛЯ ПОШУКУ РЕЦЕПТІВ ІЗ
НАЯВНИХ ІНГРЕДІЄНТІВ З ВИКОРИСТАННЯМ APACHE CORDOVA
ТА MYSQL**

**DEVELOPMENT OF A MOBILE APPLICATION FOR RECIPE SEARCH
USING AVAILABLE INGREDIENTS WITH APACHE CORDOVA AND
MYSQL**

спеціальність 121 «Інженерія програмного забезпечення»
освітня програма «Інженерія програмного забезпечення»

Виконав: здобувач вищої освіти
групи ІПЗс-21
Мартинюк Назар Ярославович

Керівник:
Падалко Анатолій Михайлович

Кваліфікаційну роботу
допущено до захисту
«10» 06 2025 р.
Гарант освітньої програми:
к.т.н., доцент
Ліщина Наталія Миколаївна



Луцьк – 2025 року

ЛУЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

Факультет комп'ютерних та інформаційних технологій

Кафедра інженерії програмного забезпечення

Ступінь вищої освіти бакалавр

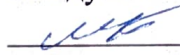
Галузь знань: 12 «Інформаційні технології»

Спеціальність: 121 «Інженерія програмного забезпечення»

Освітня програма: «Інженерія програмного забезпечення»

ЗАТВЕРДЖУЮ

Завідувач кафедри


«20» 12 2024 р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧУ ВИЩОЇ ОСВІТИ

Мартинюку Назару Ярославовичу

(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи «Розробка мобільного додатку для пошуку рецептів із наявних інгредієнтів з використанням Apache Cordova та MySQL».

Керівник роботи: к.фіз.-мат.н., доцент Падалко А. М.

затверджені наказом закладу вищої освіти від «19» грудня 2024 р. № 474/01-02

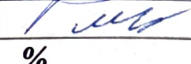
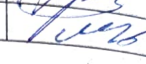
2. Строк подання здобувачем вищої освіти кваліфікаційної роботи бакалавра «10» червня 2025 р.

3. Вихідні дані до роботи: Apache Cordova, MySQL, методичні вказівки до виконання кваліфікаційної роботи бакалавра.

4. Зміст розрахунково-пояснювальної записки (перелік питань, що потрібно розробити): аналіз сучасного стану проблеми, специфікація функціональних і нефункціональних вимог, проектування архітектури додатку, обґрунтування вибору технологій, реалізація клієнтської та серверної частини, розробка бази даних, тестування і налагодження системи, заходи з інформаційної безпеки, формування висновків і рекомендацій.

5. Перелік графічного матеріалу: 6 рисунків, 1 таблиця.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис	
		завдання видав	завдання прийняв
Аналіз предметної області	Падалко А. М.		
Специфікація вимог до розробленої системи	Падалко А. М.		
Розробка об'єкта проектування	Падалко А. М.		
Нормоконтроль	Вознюк А. В.		
Гарант ОП	Ліщина Н. М.		
Показник запозичень тексту	0,84 %		
Академічна доброчесність	Падалко А. М.		

7. Дата видачі завдання «20» грудня 2024 р.

№	Назва етапів кваліфікаційної роботи бакалавра	Строк виконання етапів роботи	Примітка
1	Огляд літературних джерел по темі кваліфікаційної роботи бакалавра	до 04.02.2025 р.	<i>вик.</i>
2	Аналіз проблеми розробки та впровадження об'єкту проектування	до 01.03.2025 р.	<i>вик.</i>
3	Обґрунтування вибору шляхів, технологій і засобів вирішення поставленого завдання	до 15.03.2025 р.	<i>вик.</i>
4	Розробка функціонально-структурної схеми роботи об'єкта проектування та проектування бази даних	до 29.03.2025 р.	<i>вик.</i>
5	Практична реалізація об'єкта проектування та розробка бази даних	до 26.04.2025 р.	<i>вик.</i>
6	Тестування та налагодження об'єкта проектування	до 03.05.2025 р.	<i>вик.</i>
7	Здача чистового варіанту кваліфікаційної роботи бакалавра на кафедрі	до 10.06.2025 р.	<i>вик.</i>

Здобувач вищої освіти



Назар МАРТИНЮК

Керівник кваліфікаційної роботи



Анатолій ПАДАЛКО

АНОТАЦІЯ

Мартинюк Н. Я. Розробка мобільного додатку для пошуку рецептів із наявних інгредієнтів з використанням Apache Cordova та MySQL. Рукопис. Кваліфікаційна робота бакалавра ОП «Інженерія програмного забезпечення» спеціальності «Інженерія програмного забезпечення». Луцький національний технічний університет. Луцьк, 2025.

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

У першому розділі здійснено аналіз сучасного стану мобільних застосунків у кулінарній сфері, виявлено недоліки аналогів і сформульовано завдання. У другому розділі визначено вимоги до системи, обґрунтовано вибір технологій, побудовано UML-моделі та описано архітектуру. У третьому розділі представлено реалізацію додатку, тестування, побудову бази даних і основи захисту. У висновках узагальнено результати та подано перспективи розвитку.

Ключові слова: пошук рецептів, мобільний додаток, Cordova, інгредієнти, база даних.

ABSTRACT

Martyniuk N. Development of a mobile application for recipe search using available ingredients with Apache Cordova and MySQL. Manuscript. Qualification work of the bachelor's program "Software Engineering" in the specialty "Software Engineering". Lutsk National Technical University. Lutsk, 2025.

The bachelor's qualification thesis consists of an introduction, three chapters, conclusions and a list of references.

The first chapter presents the analysis of mobile applications in the culinary field, highlights the shortcomings of existing solutions, and formulates the objectives. The second chapter defines the system requirements, justifies the choice of technologies, provides UML modeling and architectural design. The third chapter demonstrates implementation, testing, database development and data protection mechanisms. The conclusions summarize the results and suggest future improvements.

Keywords: recipe search, mobile app, Cordova, ingredients, database.

ЗМІСТ

ВСТУП.....	7
РОЗДІЛ 1 АНАЛІЗ СУЧАСНОГО СТАНУ МОБІЛЬНИХ ДОДАТКІВ ДЛЯ ПРИГОТУВАННЯ СТРАВ І ПОСТАНОВКА ЗАВДАНЬ НА КВАЛІФІКАЦІЙНУ РОБОТУ	9
1.1 Аналіз сучасного стану проблеми	9
1.2 Постановка завдання на кваліфікаційну роботу бакалавра	12
Висновки до розділу 1	13
РОЗДІЛ 2 СПЕЦИФІКАЦІЯ ВИМОГ ДО РОЗРОБЛЮВАНОГО МОБІЛЬНОГО ДОДАТКУ	14
2.1 Аналіз, визначення вимог до розроблюваного програмного забезпечення та проектування	14
2.2 Вибір засобів, методів і алгоритмів вирішення поставленого завдання	18
Висновки до розділу 2	23
РОЗДІЛ 3 РОЗРОБКА МОБІЛЬНОГО ДОДАТКУ ДЛЯ ПІДБОРУ РЕЦЕПТІВ ІЗ НАЯВНИХ ІНГРЕДІЄНТІВ	24
3.1 Практична реалізація об'єкта проектування	24
3.2 Тестування та налагодження інформаційно-комп'ютерної системи	28
3.3 Розробка бази даних	30
3.4 Захист інформаційно-комп'ютерної системи	32
Висновки до розділу 3	34
ВИСНОВКИ	36
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	39

ВСТУП

Актуальність теми. Сфера мобільних технологій стрімко розвивається, проникаючи в усі аспекти повсякденного життя, включаючи харчування й побут. Особливої популярності набувають мобільні застосунки, які дозволяють спростити процес приготування їжі – від пошуку рецептів до складання меню на основі залишків продуктів у холодильнику. Програми такого типу дозволяють користувачам економити час, зменшувати харчові відходи та урізноманітнювати свій раціон. Проте в більшості існуючих сервісів відсутня підтримка локалізації українською мовою, обмежений функціонал підбору страв за залишковими інгредієнтами або необхідність постійного підключення до Інтернету.

Світові тенденції демонструють активне впровадження кросплатформних технологій (зокрема, Apache Cordova, Flutter, React Native) для прискорення процесу розробки мобільних додатків, що працюють одночасно на Android та iOS. Паралельно з цим, у сфері зберігання та обробки інформації перевагу надають реляційним базам даних, серед яких MySQL залишається однією з найпопулярніших завдяки своїй надійності, простоті адміністрування та відкритому коду.

На сьогодні у предметній галузі кулінарних застосунків частково реалізовані рішення, які дозволяють вводити інгредієнти вручну та переглядати доступні рецепти (наприклад, Yummly, SuperCook, Cookpad). Однак ці додатки часто не враховують локальні кулінарні вподобання, не підтримують офлайн-доступ до даних та не надають можливості редагування бази рецептів користувачем. Це створює певну прогалину в задоволенні запиту локального користувача на зручний, швидкий і автономний інструмент планування приготування їжі.

Відповідно, актуальність теми полягає у необхідності створення простого у використанні мобільного додатку, який би забезпечував добір рецептів із врахуванням наявних інгредієнтів користувача. Такий додаток має бути

зручним, україномовним, з можливістю офлайн-роботи, відкритим до розширення функціоналу та захищеним у плані збереження персональних даних.

Метою кваліфікаційної роботи є розробка мобільного додатку на базі Apache Cordova з використанням MySQL, який дозволяє здійснювати ефективний пошук рецептів на основі наявних інгредієнтів, з урахуванням потреб сучасного користувача.

Об'єктом кваліфікаційної роботи є процес автоматизованого пошуку кулінарних рецептів за заданими умовами (наявними інгредієнтами).

Предметом кваліфікаційної роботи виступають методи реалізації такого функціоналу в мобільному середовищі, структура бази даних рецептів і реалізація алгоритмів пошуку з використанням вебтехнологій та MySQL.

Для досягнення поставленої мети були визначені такі завдання:

- 1) проаналізувати сучасний стан проблеми, провести огляд існуючих мобільних додатків і літературних джерел;
- 2) сформулювати технічні вимоги до системи, спроектувати архітектуру системи;
- 3) провести аналіз та обрати технології розробки (Apache Cordova, MySQL);
- 4) розробити основний функціонал додатку;
- 5) провести тестування функціоналу й оптимізацію роботи;
- 6) розробити базу даних з використанням MySQL;
- 7) реалізувати базові засоби захисту даних.

Областю застосування створеного програмного продукту є побутові умови користувачів, які готують їжу вдома, а також потенційно – заклади громадського харчування (кафе, кухні самообслуговування, студентські гуртожитки), де потрібно оперативно приймати рішення щодо використання обмеженого набору продуктів.

РОЗДІЛ 1

АНАЛІЗ СУЧАСНОГО СТАНУ МОБІЛЬНИХ ДОДАТКІВ ДЛЯ ПРИГОТУВАННЯ СТРАВ І ПОСТАНОВКА ЗАВДАНЬ НА КВАЛІФІКАЦІЙНУ РОБОТУ

1.1 Аналіз сучасного стану проблеми

У сучасних умовах стрімкого розвитку мобільних технологій та зростаючої популярності здорового способу життя підвищується попит на інструменти, які допомагають планувати раціон та ефективно використовувати наявні продукти. Одним із найбільш затребуваних типів таких інструментів є мобільні додатки для пошуку рецептів за наявними інгредієнтами. Їх основна мета – автоматизувати процес підбору страв, мінімізуючи витрати часу користувача та зменшуючи кількість харчових відходів.

У межах дослідження було проаналізовано низку актуальних наукових і практичних джерел, присвячених застосуванню мобільних додатків у сфері харчування, кулінарії та цифровізації побуту. У статті F. Mandracchia та співавторів [1] здійснено систематичний огляд мобільних застосунків, орієнтованих на користувачів із харчовими алергіями та непереносимістю продуктів. Автори використовують шкалу MARS для оцінки якості застосунків і вказують на низький рівень інтерактивності, слабку персоналізацію та обмежений користувацький досвід – проблеми, що актуальні також і для додатків із підбору рецептів.

Дослідження С. Е. Mauch та співавторів [2] фокусується на корисності та прийнятності мобільних додатків для підтримки здорового харчування родин. У процесі тестування виявлено важливість візуальної привабливості, простоти навігації та зручного пошуку за критеріями. Ці висновки мають безпосереднє значення для побудови інтерфейсу користувача в розроблюваному додатку. Практичний підхід до вирішення кулінарних бар'єрів представлений у статті Т. К. Mohd та співавторів [3], де описано онлайн-застосунок Cooking PaPa, що допомагає молоді опанувати кулінарні навички. У дослідженні підкреслюється

важливість поєднання доступності, мультимедійного супроводу та адаптивного пошуку рецептів – саме ці функції також планується реалізувати в межах кваліфікаційної роботи бакалавра.

Український контекст досліджується у магістерській роботі О. К. Красковського [4], в якій створено інформаційну систему для пошуку та сортування рецептів. Автор реалізував веборієнтований функціонал з фільтрацією та категоризацією страв, однак система не адаптована під мобільні пристрої, що й зумовлює потребу у створенні додатку з відповідною платформенною підтримкою. Інший важливий приклад – робота В. О. Матвієнка [5], де реалізовано мобільний додаток із використанням алгоритмів штучного інтелекту для персоналізованого підбору страв. Попри потужний потенціал AI, автор зазначає складність адаптації таких рішень для широкого кола користувачів через високу ресурсоемність. У межах даної кваліфікаційної роботи було вирішено зробити акцент на простоті, доступності й автономності, залишаючи AI-функціонал як перспективу подальшого розвитку.

Також було враховано аспекти цифрової трансформації сфери громадського харчування, висвітлені в статті О. В. Олійника та співавторів [6]. Автори наголошують на потребі впровадження адаптивних мобільних рішень у ресторанному бізнесі, що підтверджує доцільність створення подібного застосунку і для побутового, і для професійного використання (наприклад, у закладах швидкого харчування, гуртожитках або кейтерингових службах). Проаналізовані джерела не лише підтверджують актуальність теми, але й окреслюють науково-практичне підґрунтя для розробки ефективного, функціонального та адаптивного мобільного застосунку для підбору рецептів на основі наявних інгредієнтів.

У результаті аналізу ринку були виявлені найпоширеніші рішення, що реалізують схожий функціонал. Yummly – один із найбільш популярних міжнародних сервісів, який дозволяє підбирати рецепти за смаками, алергенами та інгредієнтами [7]. Має гарний дизайн, проте потребує постійного

підключення до Інтернету, містить велику кількість платного контенту, не підтримує українську мову.

SuperCook – орієнтований саме на функцію пошуку страв за залишками продуктів [8]. Перевагою є простота інтерфейсу та велика база даних. Недоліками є англомовний інтерфейс, обмежені можливості персоналізації та відсутність офлайн-доступу. Cookpad – соціальний кулінарний застосунок, що дозволяє користувачам ділитися рецептами [9]. Має слабо реалізований пошук за інгредієнтами та вимагає реєстрації.

У межах патентного пошуку були виявлені кілька патентів, пов'язаних із автоматизованими системами підбору рецептів, однак більшість із них орієнтовані на великі комерційні платформи або інтеграцію з побутовими пристроями (розумна кухня), а не на мобільні рішення для індивідуального користування.

Аналіз зазначених рішень дозволяє виокремити такі основні переваги:

- 1) велика база рецептів;
- 2) можливість фільтрації за складом;
- 3) підтримка мультимедійного контенту.

Водночас наявні недоліки:

- 1) відсутність україномовного інтерфейсу;
- 2) прив'язка до онлайн-сервісів;
- 3) обмежений функціонал без реєстрації;
- 4) складність у використанні для людей похилого віку або осіб без технічної підготовки.

Ураховуючи ці фактори, виникає потреба у створенні нового мобільного додатку, який би мав повноцінну підтримку української мови, працював офлайн з локальною базою даних, мав простий інтерфейс, адаптований для будь-якої вікової категорії, дозволяв гнучко додавати нові рецепти та інгредієнти користувачем та забезпечував захист персональних даних.

Отже, доцільність створення нової системи зумовлена наявними обмеженнями існуючих рішень, локальними потребами українських

користувачів і прагненням до автономного, доступного та зручного використання застосунку в побуті.

1.2 Постановка завдання на кваліфікаційну роботу бакалавра

На основі аналізу предметної області, сучасних технологічних рішень, вимог потенційних користувачів та існуючих програмних аналогів було сформовано мету та основні завдання кваліфікаційної роботи. Кожне з поставлених завдань відповідає структурним підрозділам пояснювальної записки та реалізується в межах відповідного етапу розробки.

Для досягнення мети – створення мобільного додатку для підбору кулінарних рецептів на основі наявних інгредієнтів з використанням Apache Cordova та MySQL – необхідно виконати такі завдання:

1) здійснити аналіз сучасного стану проблеми шляхом огляду літературних джерел, патентних рішень і вже існуючих мобільних застосунків з подібним функціоналом; визначити переваги та недоліки аналогів; обґрунтувати доцільність створення нового рішення;

2) сформулювати технічні вимоги до програмного забезпечення, побудувати моделі системи, зокрема за допомогою UML-нотації, розробити специфікацію функціональних і нефункціональних характеристик, визначити інтерфейсні та інформаційні потоки системи;

3) обґрунтувати вибір інструментів розробки – Apache Cordova, HTML/CSS/JavaScript, MySQL – та алгоритмів реалізації пошуку рецептів; розробити загальну архітектуру програмного забезпечення та алгоритмічну модель обробки запитів;

4) розробити мобільний додаток, реалізувавши основні функції: введення інгредієнтів, пошук відповідних рецептів, відображення результатів у зручному вигляді; забезпечити адаптивність інтерфейсу;

5) здійснити тестування додатку, визначити мінімальні системні вимоги, провести усунення помилок та налагодження функціоналу для стабільної роботи системи;

6) спроектувати та розробити базу даних, що забезпечує зберігання інформації про рецепти, інгредієнти, зв'язки між ними; реалізувати основні SQL-запити до бази;

7) імплементувати базові елементи захисту розробленої системи, зокрема механізми перевірки введених даних та обмеження доступу до системних функцій.

Виконання зазначених завдань дозволить забезпечити повний цикл створення програмного продукту – від концептуального проєктування до реалізації, тестування, документування та підготовки до впровадження.

Висновки до розділу 1

У першому розділі було здійснено ґрунтовний аналіз сучасного стану проблеми, пов'язаної з автоматизованим підбором кулінарних рецептів за наявними інгредієнтами, а також проведено дослідження предметної області з урахуванням актуальних світових тенденцій у галузі мобільної розробки.

Виявлена потреба у створенні україномовного, автономного, інтуїтивно зрозумілого додатку для підбору рецептів підтверджує доцільність розробки нової мобільної системи на базі Apache Cordova з використанням MySQL. Така система повинна відповідати актуальним запитам користувачів щодо простоти, доступності та персоналізованого підходу до приготування їжі.

На основі проведеного аналізу було сформульовано мету та сформовані основні завдання кваліфікаційної роботи бакалавра, які відповідають етапам проєктування, реалізації, тестування та впровадження інформаційної системи. Здійснена постановка завдання створює методологічне підґрунтя для подальшої практичної реалізації мобільного додатку.

РОЗДІЛ 2

СПЕЦИФІКАЦІЯ ВИМОГ ДО РОЗРОБЛЮВАНОГО МОБІЛЬНОГО ДОДАТКУ

2.1 Аналіз, визначення вимог до розроблюваного програмного забезпечення та проектування

У процесі проектування мобільного додатку для пошуку рецептів із наявних інгредієнтів було прийнято рішення використати каскадну модель життєвого циклу програмного забезпечення, яка передбачає чітку послідовність етапів: аналіз вимог, проектування, реалізація, тестування, впровадження та супровід. Цей підхід дозволяє формалізувати процеси розробки та документування ПЗ, що є доцільним для кваліфікаційної роботи бакалавра.

Для виявлення вимог було застосовано комбіновану методику: вивчення аналогів (додатки типу Tasty, Supercook), аналіз очікувань цільової аудиторії та розробка користувацьких сценаріїв. На основі цього сформульовано специфікацію функціональних і нефункціональних вимог до додатку.

Цілі системи: надати користувачам можливість швидко знаходити рецепти на основі введених інгредієнтів та забезпечити просту навігацію, швидкий доступ до результатів і зручне редагування контенту. Основні функції системи:

- 1) введення нових рецептів через форму (назва, інгредієнти, інструкція, зображення);
- 2) збереження, редагування та видалення записів у локальній або віддаленій базі даних;
- 3) пошук рецептів за ключовими словами та категоріями;
- 4) фільтрація за наявними інгредієнтами;
- 5) візуалізація результатів у вигляді карток;
- 6) авторизація користувачів і розмежування прав доступу (адміністратор/користувач).

Для відображення структури та функціоналу ПЗ було створено моделі у нотації UML [10]. На діаграмі на рисунку 2.1 зображено основні сценарії взаємодії користувача з мобільним застосунком. Користувач має змогу здійснювати пошук рецептів, переглядати їх, додавати нові записи, редагувати їх або керувати обліковим записом. Адміністратор має розширений функціонал для керування всіма користувачами та рецептами.



Рисунок 2.1 – Use Case Diagram (взаємодія користувача з додатком)

Діаграма на рисунку 2.2 демонструє структуру основних класів: User, Recipe, Ingredient, Category. Вона описує атрибути об'єктів та зв'язки між ними. Один рецепт може містити багато інгредієнтів, а кожен користувач може створювати багато рецептів.

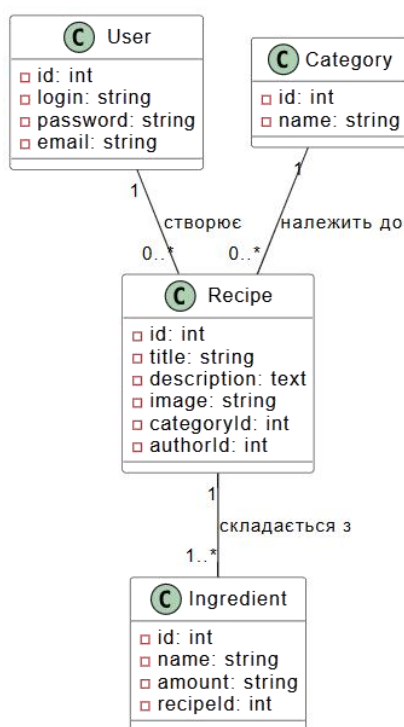


Рисунок 2.2 – Class Diagram (основні сутності додатку)

Діаграма на рисунку 2.3 описує логіку пошуку рецептів за введеними інгредієнтами. Користувач вводить список інгредієнтів, система формує запит до бази даних, фільтрує рецепти, і показує результати.

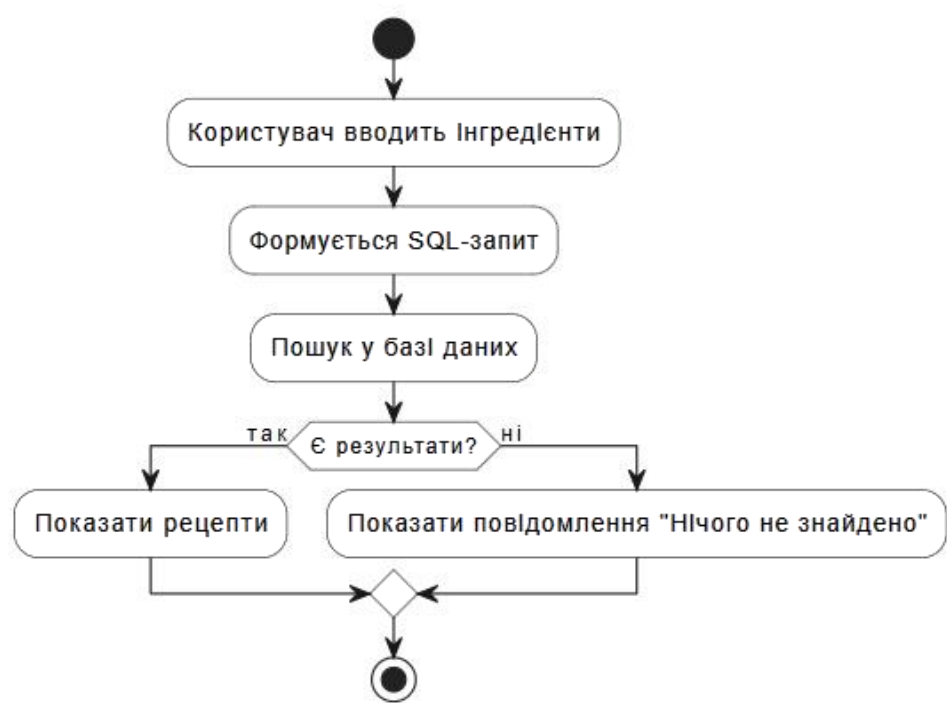


Рисунок 2.3 – Activity Diagram (алгоритм пошуку рецептів)

Діаграма на рисунку 2.4 відображає послідовність взаємодій між компонентами системи при виконанні пошуку.

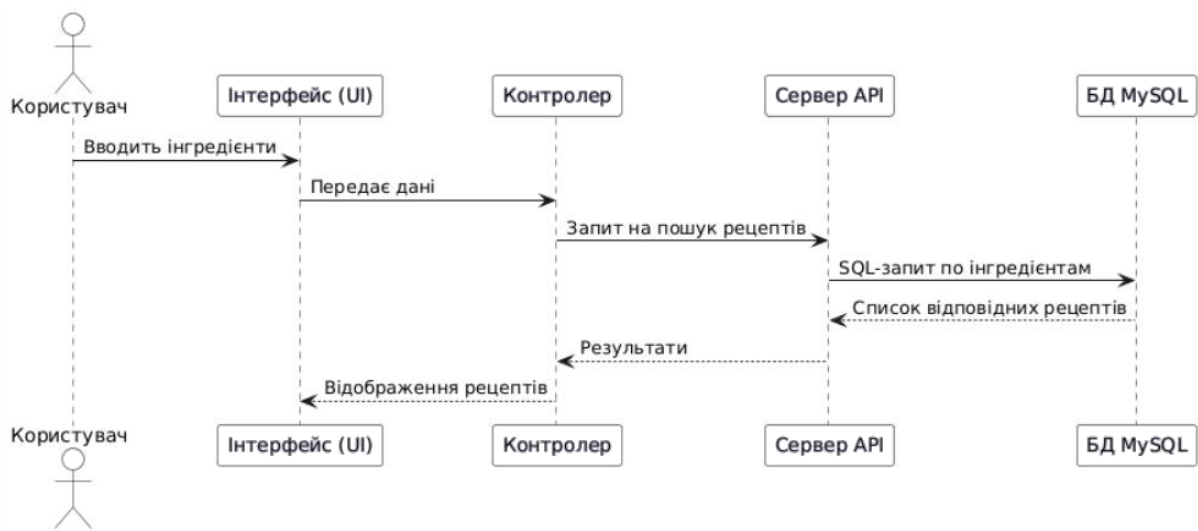


Рисунок 2.4 – Sequence Diagram: Сценарій «Пошук рецепту»

Ця діаграма показує, як користувач ініціює пошук, як формується запит до бази даних і як результат повертається на клієнт.

В свою чергу, основні події та відгуки системи представлено в таблиці 2.1.

Таблиця 2.1 – «Подія – Відгук»

Подія	Відгук системи
Користувач вводить інгредієнти	Система виконує запит до БД та повертає релевантні рецепти
Створення нового рецепту	Система додає новий запис у БД та оновлює інтерфейс
Вхід адміністратора	Відкривається панель редагування рецептів та користувачів
Відсутність інтернету	Система повідомляє про помилку зв'язку або працює в офлайн-режимі

Інтерфейс реалізовано відповідно до принципів мінімалізму, інтуїтивної зрозумілості та адаптивності. Основні екрани:

- 1) головна сторінка з категоріями;
- 2) пошуковий блок;
- 3) форма додавання нового рецепту;
- 4) детальна картка рецепту;
- 5) панель облікового запису.

Розробка ведеться з використанням HTML5/CSS3, фреймворку Apache Cordova, jQuery, а збереження даних – через API-запити до віддаленої бази даних MySQL.

Забезпечення якості та безпеки реалізовано через перевірку обов'язкових полів і типів даних, валідацію на клієнтському рівні та розмежування прав доступу до CRUD-операцій. Передача даних відбувається через протокол TCP/IP із підтримкою HTTPS, а цілісність бази даних забезпечується через дотримання реляційної структури.

Отримана модель програмного продукту охоплює як функціональні вимоги, так і взаємодію користувачів з системою. Проєктна структура дозволяє розширювати додаток у майбутньому, наприклад, шляхом інтеграції з API закупівель або формуванням списку покупок.

2.2 Вибір засобів, методів і алгоритмів вирішення поставленого завдання

У процесі створення мобільного додатку для пошуку рецептів з використанням наявних інгредієнтів було проаналізовано низку сучасних інструментів і платформ, що застосовуються в розробці інформаційних систем. У результаті обрано найбільш доцільні з погляду функціональності, продуктивності, зручності використання та доступності.

Apache Cordova – це платформа для розробки мобільних застосунків, яка дозволяє створювати кросплатформенні рішення з використанням стандартних вебтехнологій – HTML5, CSS3 та JavaScript [11]. Завдяки цьому значно спрощується процес розробки, оскільки не потребує окремого написання нативного коду для кожної операційної системи.

Серед переваг Cordova варто виділити:

- 1) можливість використання вже знайомих вебтехнологій, що скорочує час на навчання новим інструментам;
- 2) широкий набір плагінів для доступу до функцій пристрою, таких як файлові системи, камера, буфер обміну, інтернет-з'єднання тощо;
- 3) підтримку кількох платформ (Android, iOS, Windows) без необхідності переписування логіки;
- 4) активну спільноту розробників, документацію та оновлення.

Недоліками Cordova є:

- 1) відносно невисока продуктивність порівняно з нативними застосунками, особливо при роботі з великою кількістю візуальних елементів або обчислень;
- 2) залежність від браузерного рушія WebView, який може поводитися по-різному на різних пристроях;
- 3) складність у налаштуванні проєкту для публікації в офіційних маркетах через політики безпеки.

Незважаючи на ці обмеження, вибір Cordova є виправданим для створення MVP або невеликого функціонального застосунку, що не потребує високих графічних чи апаратних ресурсів.

MySQL – одна з найпоширеніших систем управління реляційними базами даних із відкритим вихідним кодом [12]. Вона використовується в тисячах проєктів і підтримується провідними ІТ-компаніями.

Переваги MySQL включають:

- 1) висока швидкість виконання запитів, особливо при правильній структурі таблиць і індексах;
- 2) підтримка складних SQL-запитів, транзакцій, зовнішніх ключів і реляцій між таблицями;
- 3) доступність великої кількості навчальних матеріалів і засобів адміністрування (наприклад, phpMyAdmin);
- 4) сумісність із багатьма мовами програмування (PHP, Node.js, Python тощо).

Серед недоліків:

- 1) відсутність підтримки гнучкої схемної структури, властивій документоорієнтованим базам, як-от MongoDB;
- 2) потреба в регулярному адмініструванні для оптимізації продуктивності;
- 3) обмежена масштабованість у порівнянні з деякими сучасними хмарними СУБД.

MySQL у цьому проєкті використовується для зберігання інформації про рецепти, користувачів, категорії та інгредієнти, а також для реалізації пошуку за фільтрами. Вибір саме цієї СУБД обумовлений стабільністю, підтримкою зв'язків між таблицями та швидким доступом до даних.

JavaScript – основна мова клієнтської логіки, що дозволяє створювати динамічну поведінку елементів інтерфейсу [13]. У поєднанні з бібліотекою jQuery забезпечується спрощена робота з DOM-деревом, подіями, анімацією та AJAX-запитами [14].

Переваги цього вибору:

- 1) простота у вивченні й широке поширення;
- 2) підтримка великої кількості бібліотек і фреймворків;
- 3) швидка розробка односторінкових додатків (SPA);
- 4) можливість інтеграції з іншими інструментами, такими як TinyMCE або Cordova API.

Недоліки:

- 1) відсутність суворої структури коду, що з часом ускладнює масштабування і підтримку;
- 2) нижчий рівень безпеки (клієнтський код відкритий і доступний у браузері);
- 3) застарілість jQuery у контексті сучасних SPA-фреймворків (наприклад, React чи Vue.js), хоча для простих мобільних рішень ця бібліотека є ефективною;
- 4) JavaScript і jQuery обрано для забезпечення логіки переходів між сторінками, взаємодії з DOM та обробки дій користувача (додавання, редагування, фільтрація, перегляд).

Додатково в проєкті застосовано редактор TinyMCE, який дозволяє створювати та редагувати рецепти з використанням WYSIWYG-підходу [15]. Це забезпечує візуальну зручність користувачам при введенні даних.

Для доступу до функціоналу пристрою використано плагіни Cordova:

- 1) cordova-plugin-advanced-http – для надійної роботи з API, у т.ч. з заголовками, токенами, SSL;
- 2) cordova-plugin-file і cordova-clipboard – для взаємодії з локальною пам'яттю пристрою та буфером обміну;
- 3) cordova-plugin-network-information – для перевірки наявності інтернет-з'єднання перед надсиланням запитів до серверу.

Обраний технологічний стек є оптимальним для реалізації мобільного додатку середнього рівня складності. Він дозволяє швидко розгорнути функціональний продукт, забезпечуючи при цьому масштабованість, гнучкість, доступність та зручність для користувача.

Також опишемо деякі алгоритми, що використовуються в системі. Алгоритм пошуку рецептів за інгредієнтами реалізований у вигляді SQL-запиту до бази даних, що враховує часткові входження інгредієнтів у текстовий список. Умовна логіка цього алгоритму наведена в лістингу 2.1.

Лістинг 2.1 – Алгоритм пошуку рецептів за інгредієнтами

```
SELECT * FROM recipes
WHERE ingredients LIKE '%молоко%' AND ingredients LIKE '%яйце%';
```

кінець лістингу 2.1

Перевірка прав доступу – на рівні інтерфейсу реалізовано розмежування:

1) якщо користувач має роль admin – відкриваються функції редагування, створення, видалення;

2) інакше – лише перегляд і пошук.

Блок-схема алгоритму пошуку рецептів наведена на рисунку 2.5.

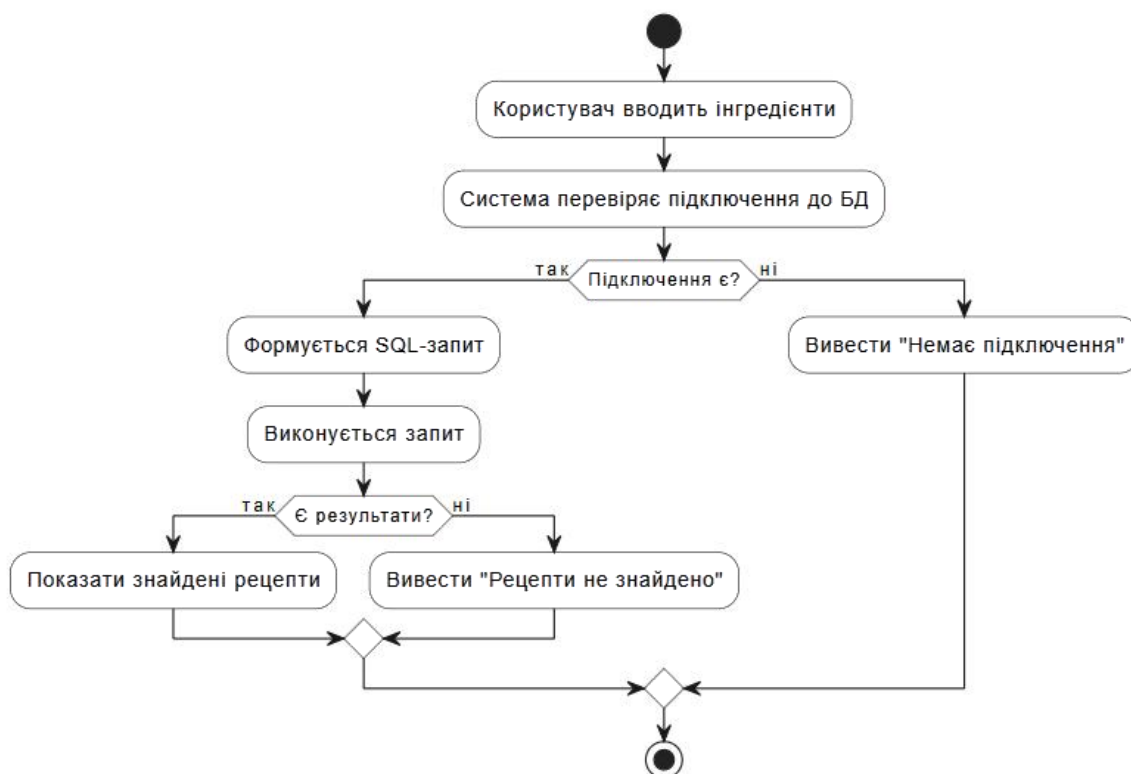


Рисунок 2.5 – Блок-схема алгоритму пошуку рецептів за введеними інгредієнтами

На цій блок-схемі зображено логіку роботи системи під час пошуку рецептів за інгредієнтами. Користувач вводить дані, після чого система перевіряє підключення до бази даних. Якщо підключення встановлено, формується SQL-запит і виконується пошук. У разі наявності результатів вони виводяться користувачу, інакше – відображається повідомлення про відсутність збігів. Якщо з'єднання з базою відсутнє, система повідомляє про помилку доступу.

Діаграма розгортання системи наведена на рисунку 2.6. Ця діаграма ілюструє архітектуру інформаційної системи та взаємодію її компонентів. Користувач працює зі смартфоном, на якому запущено інтерфейс користувача (Cordova App) та клієнтську логіку (JavaScript/jQuery). Дані передаються на веб-сервер через HTTP-запити до REST API, яке, в свою чергу, обробляє SQL-запити до серверу бази даних MySQL. Така структура забезпечує розподіленість, масштабованість і можливість централізованого збереження інформації.



Рисунок 2.6 – Діаграма розміщення компонентів системи

У межах реалізації поставленого завдання було обґрунтовано використання стеку: Apache Cordova, MySQL, HTML/CSS/JavaScript та супутніх плагінів. Застосовані інструменти дозволяють реалізувати функціональний, зручний і розширюваний мобільний застосунок із підтримкою пошуку рецептів за інгредієнтами, керування даними, автентифікації та редагування вмісту.

Висновки до розділу 2

У цьому розділі було проведено глибокий аналіз вимог до мобільного додатку для пошуку рецептів за наявними інгредієнтами, визначено цілі, функціональність та архітектурну структуру майбутнього програмного продукту. Для формалізації вимог було використано методи моделювання у нотації UML: побудовано діаграми варіантів використання, класів, активностей та послідовностей, що дозволило чітко визначити основні сценарії взаємодії користувача з системою.

З урахуванням поставлених завдань було обґрунтовано вибір відповідного інструментарію. Для створення кросплатформеного мобільного додатку обрано Apache Cordova, який забезпечує швидку розробку із застосуванням вебтехнологій. Для збереження даних про рецепти, інгредієнти та користувачів використано MySQL як реляційну базу даних. Реалізацію клієнтської логіки забезпечено мовою JavaScript у поєднанні з бібліотекою jQuery, що значно спрощує обробку подій і динамічну роботу з елементами DOM.

Описано логіку основного алгоритму пошуку рецептів, реалізованого на основі SQL-запитів з фільтрацією за інгредієнтами. Також було представлено блок-схеми процесів пошуку та взаємодії з базою даних. Визначено й реалізовано основні функціональні компоненти, включно з механізмами введення, редагування, перегляду й управління користувачами, що повністю відповідають функціональній специфікації, викладеній на етапі проектування.

РОЗДІЛ 3

РОЗРОБКА МОБІЛЬНОГО ДОДАТКУ ДЛЯ ПІДБОРУ РЕЦЕПТІВ ІЗ НАЯВНИХ ІНГРЕДІЄНТІВ

3.1 Практична реалізація об'єкта проєктування

Процес розробки мобільного застосунку для пошуку рецептів із наявних інгредієнтів реалізовано з використанням кросплатформеного фреймворку Apache Cordova, мови JavaScript та бібліотеки jQuery, а також мови розмітки HTML5 і стилів CSS3. Вебчастину інтегровано у Cordova-застосунок за допомогою WebView. Серверна частина працює з MySQL базою даних через REST API.

Після ініціалізації події `deviceready`, що є стандартною для Cordova, відбувається перевірка доступу до Інтернету та підвантаження вмісту через HTTP-запити (ліст. 3.1).

Лістинг 3.1 – Ініціалізація застосунку та перевірка мережі

```
document.addEventListener('deviceready', onDeviceReady, false);
function onDeviceReady() {
    userId = localStorage.getItem('userId');
    checkConnection();
    if (userId == null && !internet) {
        alert('Internet connection error!');
    } else {
        loadDataJson();
    }
}
```

кінець лістингу 3.1

Цей код демонструє використання Cordova API для визначення стану мережі. Метод `checkConnection()` використовує об'єкт `navigator.connection`, щоб з'ясувати, чи пристрій має доступ до мережі. Надалі функція `loadDataJson()` здійснює HTTP-запит до REST-сервера й отримує JSON-дані з категоріями рецептів і самими рецептами.

Для динамічного заповнення інтерфейсу застосовано шаблони HTML (<template>) та маніпуляції з DOM за допомогою jQuery (ліст. 3.2).

Лістинг 3.2 – Формування категорій на основі даних

```
let part = $("#part-temp");  
let partClone = part[0].content.cloneNode(true);  
$(partClone).find('.text').text(item.name);  
$('.main .inner').append(partClone);
```

кінець лістингу 3.2

Відповідно, у залежності від даних, отриманих з API, застосунок автоматично формує категорії рецептів на головному екрані.

Реєстрація користувача відбувається покроково, із валідацією введених даних. Дані зберігаються в localStorage, після чого користувач автоматично переходить у профіль. Під час реєстрації використовується кілька етапів введення: ім'я, прізвище, номер телефону, дата народження. Віконце вибору дати реалізовано як власний UI-елемент з кастомним прокруткою, без використання сторонніх бібліотек (ліст. 3.3).

Лістинг 3.3 – Зчитування вибраної дати народження

```
$('.input-over .btns .ok').on("click", function(e){  
    let day = $('.birth .day .selected').text();  
    let month = $('.birth .month .selected').text();  
    let year = $('.birth .year .selected').text();  
    let date = `${day}.${month}.${year}`;  
    $('.profile-page .date-js').val(date);  
});
```

кінець лістингу 3.3

Дані користувача можна змінювати в особистому кабінеті. Для цього реалізовано функцію dataEditUser(), яка формує запит до API із параметрами, що передаються в URL, і виконує оновлення інформації на сервері (ліст. 3.4).

Лістинг 3.4 – Запит на оновлення даних користувача

```
let url = 'https://server.Intuuu.pp.ua/data/edit?name=' + name + '&sname=' + soname +
'&phone=' + phone + '&date=' + date + '&id=' + userId;

cordova.plugin.http.get(url, {}, { Authorization: 'OAuth2: token' }, function(response) {
    userJson = JSON.parse(response.data);
});
```

кінець лістингу 3.4

Особливу увагу приділено побудові фільтрів. Рецепти фільтруються за країною походження та діапазоном дат (ліст. 3.5).

Лістинг 3.5 – Алгоритм фільтрації рецептів

```
function filterItems(category) {
    let selectedCountry = category.find('.country').val();
    let fromDate = new Date(category.find('.from').val());
    let toDate = new Date(category.find('.to').val());

    category.find('.item').each(function () {
        let itemDate = new Date($(this).data('date'));
        // логіка фільтрації
    });
}
```

кінець лістингу 3.5

Візуальну складову реалізовано через кастомізовану структуру HTML-елементів та CSS-класи, що відповідають сучасним вимогам адаптивності. За потреби застосовуються стилі для візуалізації анімацій, затемнення фону, preload-стану та спливаючих вікон.

На рисунку 3.1 представлено фрагмент інтерфейсу мобільного додатку. Інтерфейс реалізовано у вигляді форми, де користувач може обрати категорію страви, ввести її назву, вказати фільтр (наприклад, дієтична, веганська тощо), завантажити зображення страви, а також увести сам рецепт за допомогою текстового редактора з розширеним форматуванням. У прикладі наведено рецепт, який містить блок «Інгредієнти» з маркованим списком та блок «Приготування» з покроковою інструкцією. Кнопка «Зберегти» внизу форми

дає змогу підтвердити та зберегти внесені зміни. Інтерфейс адаптований під мобільні екрани, є інтуїтивно зрозумілим і підтримує мультимедіа.

Рис. 3.1 – Фрагмент інтерфейсу додатку

Загалом, практична частина демонструє повноцінну реалізацію мобільного застосунку, який використовує повнофункціональний стек технологій для обробки введених даних, комунікації з сервером, управління інтерфейсом користувача та забезпечення UX-оптимізованої логіки.

3.2 Тестування та налагодження інформаційно-комп'ютерної системи

Після завершення основного етапу розробки мобільного додатку для пошуку рецептів із наявних інгредієнтів було проведено систематичне тестування з метою перевірки коректності функціонування всіх основних модулів застосунку. Основними напрямками перевірки стали: функціональне тестування, перевірка UX/UI-елементів, тестування мережевих запитів, перевірка взаємодії з базою даних і налагодження збереження локальних даних у localStorage.

Функціональне тестування охоплювало перевірку сценаріїв реєстрації та авторизації користувача, перегляду категорій рецептів, перегляду інформації про конкретні страви, використання фільтрів, а також редагування профілю. Окрему увагу приділено коректній обробці подій у разі втрати мережевого з'єднання, відсутності введених даних або введення їх у неправильному форматі. Було протестовано понад 20 базових сценаріїв взаємодії користувача з системою.

Тестування сумісності здійснювалось у середовищі Android Studio на емуляторах Android 8.1, 10 та 12, а також на фізичному пристрої з Android 11. Застосунок демонструє стабільну роботу в середовищі WebView з підтримкою усіх базових подій і методів Cordova.

UX/UI-тестування дозволило перевірити, наскільки інтерфейс є інтуїтивно зрозумілим та адаптивним на різних розмірах екранів. Була виявлена помилка відображення блоку реєстрації на екранах шириною менше 360 px, яку було усунуто завдяки додатковим медіазапитам у CSS.

Мережеве тестування перевіряло правильність формування REST-запитів та обробки відповіді у форматі JSON. Застосунок використовує плагін cordova-plugin-advanced-http, який успішно взаємодіє з бекендом через захищені HTTP-запити з авторизацією. Проблема з некоректною обробкою порожньої відповіді була вирішена через додаткову перевірку на null перед парсингом JSON.

Налагодження проводилось через логування `console.log()`, відладку в режимі `WebView Debugging` та ручну перевірку логіки дій. Найчастіше усувалися помилки некоректного доступу до властивостей `undefined`, помилки у форматуванні дат, а також некоректне повторне використання шаблонів `DOM`.

Мінімальні системні вимоги для роботи застосунку:

- 1) операційна система: Android 8.0 або новіша;
- 2) процесор: ARM з частотою від 1.5 ГГц;
- 3) обсяг оперативної пам'яті: 1 ГБ;
- 4) вільний простір на диску: 150 МБ;
- 5) доступ до Інтернету (Wi-Fi або мобільні дані).

Рекомендовані параметри:

- 1) Android 10 або вище;
- 2) 2+ ГБ оперативної пам'яті;
- 3) стабільне підключення до Інтернету (HTTPS/TCP/IP);
- 4) дисплей з роздільною здатністю 720×1280 px або вище.

Виявлені недоліки та усунення:

1) проблема з валідацією дати народження – введення порожніх значень призводило до зависання. Вирішено додаванням перевірки на наявність значень перед формуванням дати;

2) некоректна обробка втрати з'єднання з мережею – додано сповіщення через `alert()` у разі відсутності інтернету;

3) подвійне додавання HTML-шаблонів – виявлено при оновленні вмісту категорій, усунуто через очищення контейнера перед вставкою нових елементів.

Після завершення всіх етапів тестування система відповідає вимогам до стабільної роботи, сумісності та надійності. Отриманий програмний продукт забезпечує повний набір функцій, передбачених технічним завданням, і демонструє готовність до використання кінцевими користувачами.

3.3 Розробка бази даних

Для збереження, фільтрації та обробки інформації про рецепти, користувачів і категорії в мобільному додатку була розроблена серверна база даних на основі системи керування базами даних MySQL.

Взаємодія з базою здійснюється через REST API за допомогою HTTP-запитів, сформованих з клієнтської частини додатку. Інтерфейсна частина звертається до API-запитів, які, у свою чергу, маніпулюють таблицями в базі.

Для розробки структури бази даних було використано середовище phpMyAdmin, яке дозволяє створювати таблиці, редагувати схеми, а також переглядати й тестувати SQL-запити у візуальному інтерфейсі.

Така архітектура забезпечує розподілення відповідальності між клієнтом і сервером, спрощує масштабування додатку та полегшує підтримку й оновлення даних.

На основі аналізу коду додатку, що виконує запити на редагування користувача (`edit?name=...`), завантаження даних (`loadDataJson()`) та фільтрацію рецептів за категоріями, було реконструйовано й реалізовано наступну структуру бази даних. Таблиця `users` наведена в лістингу 3.6.

Лістинг 3.6 – Таблиця `users`

```
CREATE TABLE users (  
  id INT AUTO_INCREMENT PRIMARY KEY,  
  name VARCHAR(100),  
  sname VARCHAR(100),  
  phone VARCHAR(20),  
  birth_date DATE  
);
```

кінець лістингу 3.6

Ця таблиця містить облікову інформацію про користувачів, яка вводиться під час реєстрації та редагується в особистому кабінеті. Значення `id` використовується як зовнішній ключ для прив'язки рецептів до користувача. Таблиця `recipes` наведена в лістингу 3.7.

Лістинг 3.7 – Таблиця recipes

```
CREATE TABLE recipes (  
  id INT AUTO_INCREMENT PRIMARY KEY,  
  title VARCHAR(255),  
  description TEXT,  
  country VARCHAR(100),  
  created_date DATE,  
  category_id INT,  
  user_id INT,  
  FOREIGN KEY (user_id) REFERENCES users(id)  
);
```

кінець лістингу 3.7

Тут зберігаються всі рецепти, що відображаються в застосунку. Поля country та created_date використовуються для фільтрації. Зв'язок з таблицею users дозволяє фіксувати автора рецепту. Таблиця categories наведена в лістингу 3.8.

Лістинг 3.8 – Таблиця categories

```
CREATE TABLE categories (  
  id INT AUTO_INCREMENT PRIMARY KEY,  
  name VARCHAR(100)  
);
```

кінець лістингу 3.8

Таблиця з переліком категорій (наприклад, «Перші страви», «Десерти»), які потім відображаються в інтерфейсі.

Наведемо приклади SQL-запитів. Для отримання рецептів за конкретною категорією використовується наступний запит (ліст. 3.9).

Лістинг 3.9 – Отримання рецептів за категорією

```
SELECT * FROM recipes  
WHERE category_id = 2;
```

кінець лістингу 3.9

В лістингу 3.10 наведено запит для оновлення профілю користувача.

Лістинг 3.10 – Оновлення профілю

```
UPDATE users  
SET name = 'Іван', sname = 'Петренко', phone = '380999999999', birth_date = '1999-  
10-01'  
WHERE id = 1;
```

кінець лістингу 3.10

Для додавання нового рецепту використовується запит, наведений в лістингу 3.11.

Лістинг 3.11 – Додавання рецепту

```
INSERT INTO recipes (title, description, country, created_date, category_id, user_id)  
VALUES ('Борщ', 'Опис рецепту борщу...', 'Україна', '2025-05-01', 1, 1);
```

кінець лістингу 3.11

Загалом, база даних відповідає структурі предметної області й повністю підтримує основні функції: зберігання користувачів, додавання та фільтрацію рецептів, а також категоризацію. Надалі структура може бути розширена для збереження фото, рейтингу страв, коментарів тощо.

3.4 Захист інформаційно-комп'ютерної системи

У процесі розробки мобільного додатку для пошуку рецептів із наявних інгредієнтів було враховано базові аспекти захисту даних та користувацької інформації. Зважаючи на клієнт-серверну архітектуру проєкту, що передбачає взаємодію з REST API через мережу, особливу увагу було приділено безпеці передачі даних, контролю доступу та валідації введеної інформації.

Для забезпечення захищеної комунікації між мобільним клієнтом і сервером застосовується протокол HTTPS, який шифрує передавані HTTP-запити за допомогою TLS (Transport Layer Security). Усі звернення до API відбуваються за допомогою повного шифрованого шляху (наприклад:

<https://server.lntuuu.pp.ua/data/edit?...>), що захищає дані від перехоплення під час передавання через незахищені мережі.

Реалізовано елементарний механізм контролю доступу до CRUD-операцій. Усі основні дії, пов'язані з редагуванням персональних даних (наприклад, зміна імені, номера телефону або дати народження), дозволяються лише після перевірки автентифікації користувача за локальним ідентифікатором `userId`, що зберігається в `localStorage`. Цей підхід дозволяє уникнути несанкціонованих змін з боку неавторизованих користувачів. Лістинг 3.12 демонструє елементарну перевірку автентифікації користувача шляхом зчитування значення `userId` з локального сховища браузера (`localStorage`).

Лістинг 3.12 – Фрагмент коду перевірки наявності авторизованого користувача у локальному сховищі застосунку

```
userId = localStorage.getItem('userId');
if (userId == null) {
    alert('Access denied');
}
```

кінець лістингу 3.12

Для запобігання SQL-ін'єкціям та логічним помилкам у запитах реалізовано клієнтську валідацію введених даних:

- 1) перевірка заповнення обов'язкових полів (наприклад, імені, телефону);
- 2) контроль формату дати народження;
- 3) обробка подій натискання кнопки підтвердження лише при валідних даних.

Також обрано метод GET-запитів з параметрами, що дозволяє відносно прозоро контролювати запити на серверній стороні та здійснювати фільтрацію вхідних параметрів на рівні API.

Дані користувача, такі як `userId`, тимчасово зберігаються у браузерному сховищі `localStorage`, що обмежено контекстом застосунку. Хоча цей спосіб не є криптографічно захищеним, для MVP-рівня додатку він є прийнятним

рішенням за умови відсутності критично чутливої інформації (паролів або токенів).

Крім того, використано Cordova плагін `cordova-plugin-network-information`, який дозволяє перевіряти наявність підключення до мережі перед виконанням будь-яких API-запитів, що знижує ризик обробки помилкових даних через нестабільне підключення.

Хоча в додатку не реалізовано повноцінну багаторівневу систему безпеки з хешуванням паролів чи захистом від DDoS-атак (через відсутність реєстрації з логінами й паролями), реалізовані базові механізми відповідають рівню застосунку на етапі MVP:

- 1) безпечна передача даних через HTTPS;
- 2) авторизація через локальні маркери;
- 3) валідація введених даних;
- 4) контроль доступу до API-операцій.

Ці заходи забезпечують належний рівень інформаційної безпеки в контексті простого мобільного рішення та створюють основу для подальшого розширення й зміцнення захисту в наступних версіях програмного забезпечення.

Висновки до розділу 3

У процесі реалізації мобільного додатку для підбору рецептів із наявних інгредієнтів було створено функціональний програмний продукт, що поєднує клієнтську частину на базі Apache Cordova з серверною частиною, яка працює з базою даних MySQL через REST API. Застосунок успішно виконує основні задачі: реєстрацію та авторизацію користувача, завантаження та фільтрацію рецептів, редагування профілю, перегляд інформації та інтерактивну навігацію.

Реалізація здійснювалась із застосуванням сучасних вебтехнологій – HTML, CSS, JavaScript і бібліотеки jQuery, що забезпечило кросплатформену

сумісність і гнучкість інтерфейсу. Інтерфейс створений з урахуванням вимог до адаптивності та зручності користування.

Для забезпечення стабільності роботи застосунку проведено тестування функціоналу, включаючи перевірку мережевого з'єднання, динамічного оновлення даних та функцій фільтрації. Виявлені недоліки усунені в процесі налагодження.

Структура бази даних передбачає зв'язки між користувачами, категоріями та рецептами, що дозволяє ефективно зберігати й опрацьовувати інформацію. Реалізовано стандартні SQL-запити для доступу до даних та їх оновлення.

Застосунок включає базові механізми захисту, зокрема перевірку ідентифікатора користувача, зберігання його у сховищі пристрою, використання токенів і безпечного протоколу передачі даних. Ці заходи забезпечують достатній рівень безпеки для MVP-рівня проєкту.

ВИСНОВКИ

У результаті виконання кваліфікаційної роботи досягнуто поставленої мети – розроблено мобільний застосунок для пошуку кулінарних рецептів на основі наявних інгредієнтів. Застосунок створено з використанням кросплатформеного фреймворку Apache Cordova, мов HTML, CSS, JavaScript і jQuery, а також бази даних MySQL та REST API для обміну інформацією між клієнтською й серверною частинами. Реалізовано функціонал реєстрації, перегляду рецептів, редагування профілю, фільтрації даних, динамічного формування контенту, локального збереження інформації та базових засобів захисту.

У процесі аналізу сучасного стану мобільних застосунків для приготування страв встановлено, що більшість існуючих рішень зосереджуються на рецептах за категоріями або популярності, але лише деякі з них пропонують фільтрацію за наявними інгредієнтами. Це підтверджує актуальність поставленої задачі та обґрунтовує вибір напрямку дослідження.

На основі проведеного аналізу функціональних і нефункціональних вимог до програмного продукту сформовано чітку специфікацію, UML-діаграми та сценарії використання, що дозволило точно описати поведінку системи. Було визначено основні модулі: взаємодія з користувачем, обробка даних, зв'язок із сервером і базою даних, що стали основою подальшого проєктування.

У ході аналізу інструментів розробки обґрунтовано доцільність використання Apache Cordova, MySQL та JavaScript-технологій для реалізації поставленого завдання. Враховуючи переваги обраних засобів, зокрема їх доступність, кросплатформеність і активну підтримку спільноти, сформовано ефективне технічне рішення для розробки інформаційної системи.

Практичну реалізацію проєкту здійснено шляхом створення інтерфейсу користувача, обробки запитів до сервера, реалізації фільтрації, реєстрації та редагування даних. Розробка підтвердила застосовність обраного

технологічного стеку для реалізації повноцінного функціоналу мобільного застосунку з доступом до серверної БД.

Під час тестування й налагодження програмного продукту було виявлено окремі недоліки в логіці роботи фільтрації та валідації даних, які успішно усунуто. Проведено функціональне тестування інтерфейсів і сценаріїв, що підтвердило стабільність роботи застосунку в умовах з різним рівнем доступу до мережі.

Розробка бази даних здійснювалася з урахуванням логічної структури предметної області: користувачі, рецепти, інгредієнти та зв'язки між ними. Створено реляційну схему в MySQL, що дозволяє ефективно обробляти запити, фільтрувати рецепти й забезпечувати цілісність даних при взаємодії з REST API.

Для забезпечення захищеної взаємодії між клієнтською та серверною частинами було реалізовано базові засоби безпеки: валідацію введених даних, перевірку мережевого підключення, захист від несанкціонованого доступу до API-методів та збереження ідентифікатора користувача лише локально. Це дозволило досягти прийнятного рівня безпеки без ускладнення логіки системи.

Робота узгоджується з актуальними напрямками світових тенденцій розробки мобільного ПЗ, що передбачають використання вебтехнологій для створення кросплатформених застосунків, оптимізацію під UX/UI-вимоги та інтеграцію з віддаленими серверними ресурсами через REST-інтерфейси. Програмний продукт відповідає сучасним вимогам до MVP-рівня застосунку та може бути розширений у майбутньому з урахуванням складніших функцій: збереження рецептів у профіль, рейтингів, рекомендаційної системи на основі попередніх вподобань користувача.

Зв'язок роботи з науково-дослідною діяльністю кафедр та освітніх програм університету проявляється в активному використанні принципів розробки інформаційних систем, дослідженнях у сфері UX/UI-дизайну, впровадженні принципів кросплатформеного проєктування та моделюванні архітектури систем. Під час виконання проєкту отримано нові результати в частині побудови простого, але функціонального REST-застосунку з

динамічним відображенням даних на мобільному пристрої без застосування важкої серверної логіки.

Практична цінність роботи полягає в можливості безпосереднього застосування розробленого продукту для спрощення кулінарного планування в побуті, ресторанному господарстві, для навчальних потреб у галузі технологій харчування або побутового обслуговування. Застосунок є потенційною базою для запуску повноцінного стартапу, що може зацікавити користувачів у сегменті low-cost рішень для підбору рецептів.

У майбутньому можливе вдосконалення роботи у напрямках розширення функціоналу (додавання голосового пошуку, підтримка фотоінгредієнтів, режим offline-кешування), поліпшення захисту даних (використання JWT, хешування паролів, авторизація через OAuth2), підвищення масштабованості за рахунок перенесення серверної логіки до хмарної інфраструктури. Також варто розглянути інтеграцію з OpenFoodFacts API або аналогами для розпізнавання складників продуктів.

Загалом, кваліфікаційна робота має як практичну, так і науково-прикладну цінність, підтверджує професійну компетентність автора в питаннях проектування, реалізації та тестування сучасних інформаційних систем.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Mobile Phone Apps for Food Allergies or Intolerances in App Stores: Systematic Search and Quality Assessment Using the Mobile App Rating Scale (MARS) / F. Mandracchia et al. JMIR mHealth and uHealth. 2020. Vol. 8, no. 9. P. e18339. URL: <https://doi.org/10.2196/18339> (date of access: 07.03.2025).
2. Commercially Available Apps to Support Healthy Family Meals: User Testing of App Utility, Acceptability, and Engagement / C. E. Mauch et al. JMIR mHealth and uHealth. 2021. Vol. 9, no. 5. P. e22990. URL: <https://doi.org/10.2196/22990> (date of access: 08.03.2025).
3. Cooking PaPa: An online application helps removing the barrier between modern generation and cooking / T. K. Mohd et al. In Artificial Intelligence and Applications. 2023. Vol. 1, no. 3, P. 145-153.
4. Красковський, О. К. Інформаційна система з пошуку та упорядкування кулінарних рецептів : магістерська дисертація : 126 Інформаційні системи та технології. Київ, 2024. 111 с.
5. Матвієнко В. О. Мобільний додаток підбору рецептів з використанням штучного інтелекту : робота на здобуття кваліфікаційного ступеня магістра : спец. 122 - комп'ютерні науки, наук. кер. О. В. Бойко. Суми : Сумський державний університет, 2023. 80 с.
6. Олійник О. В., Шестакова А. В., Ярмолюк Д. І. Напрями цифровізації ресторанного бізнесу. Економіка, управління та адміністрування. 2023. № 1(103). С. 15-21. URL: [https://doi.org/10.26642/ema-2023-1\(103\)-15-21](https://doi.org/10.26642/ema-2023-1(103)-15-21) (дата звернення: 12.03.2025).
7. Yummly Recipes & Cooking Tools - Завантажити APK для Android. URL: <https://yummly.ua.aptoide.com/app> (дата звернення: 14.03.2025).
8. Supercook: recipe search by ingredients you have at home. SuperCook - Zero Waste Recipe Generator. URL: <https://www.supercook.com/#/desktop> (date of access: 15.03.2025).

9. Cookpad: Зберігайте, публікуйте та діліться рецептами. Client Challenge. URL: <https://cookpad.com/ua> (date of access: 16.03.2025).
10. UML Diagrams in Software Engineering Research: A Systematic Literature Review / Н. Коç et al. Proceedings. 2021. Vol. 74, no. 1. P. 13. URL: <https://doi.org/10.3390/proceedings2021074013> (date of access: 10.04.2025).
11. Apache Cordova. URL: <https://cordova.apache.org/> (date of access: 11.04.2025).
12. MySQL. URL: <https://www.mysql.com/> (date of access: 13.04.2025).
13. Сучасний підручник з JavaScript. URL: <https://uk.javascript.info/> (дата звернення: 15.04.2025).
14. jQuery. URL: <https://jquery.com/> (date of access: 17.04.2025).
15. The Most Trusted & Feature-rich WYSIWYG Rich Text Editor : TinyMCE. URL: <https://www.tiny.cloud/> (date of access: 22.04.2025).