ДНІПРОВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ОЛЕСЯ ГОНЧАРА

ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ КАФЕДРА МАТЕМАТИЧНОГО ЗАБЕЗПЕЧЕННЯ ЕОМ

Кваліфікаційна робота

перший (бакалаврський) рівень вищої освіти спеціальність 121 Інженерія програмного забезпечення освітня програма Інженерія програмного забезпечення

РОЗРОБЛЕННЯ ANDROID-ДОДАТКУ ДЛЯ КОУЧСЕРФІНГУ

Виконавець

студент групи ПЗ-21у-1 Войцехов Микола Олександрович

(підпис)

Керівник

доцент кафедри ЕОМ

Ємел’яненко Тетяна Георгіївна

(підпис)

В.о. завідувача кафедри МЗ ЕОМ, д-р техн. наук, професор

Байбуз Олег Григорович

(підпис)

Дніпро – 2024

**ДНІПРОВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ОЛЕСЯ ГОНЧАРА**

Факультет прикладної математики Кафедра математичного забезпечення ЕОМ Рівень вищої освіти перший (бакалаврський) Спеціальність 121 Інженерія програмного забезпечення Спеціалізація Освітня програма Інженерія програмного забезпечення

ЗАТВЕРДЖУЮ

В.о. завідувача кафедри математичного забезпечення ЕОМ

*Олег БАЙБУЗ*

(підпис)

« » 2023 року

***З А В Д А Н Н Я***

# НА КВАЛІФІКАЦІЙНУ РОБОТУ

Войцехову Миколі Олександровичу

(Прізвище, ім’я, по-батькові студента)

1. Тема роботи Розроблення Android-додатку для коучсерфінгу

керівник роботи Ємел’яненко Тетяна Георгіївна, доцент кафедри ЕОМ

(Прізвище, ім’я, по-батькові, науковий ступінь, вчене звання)

затверджені наказом по Університету від « » 202 року № C

1. Термін подання роботи
2. Вхідні дані до роботи технічне завдання на розробку Android-додатку, для коучсерфінгу
3. Перелік питань, які потрібно розробити
4. Аналіз актуальності проблеми.
5. Аналіз аналогів додатку.
6. Розробка мобільного додатку для коучсерфінгу.
7. Тестування працездатності додатку.
8. Керівник, консультант з окремих (спеціальних) розділів роботи

|  |  |  |  |
| --- | --- | --- | --- |
| Розділ | Ініціали прізвище та посада  керівника, консультанта | Підпис, дата | |
| завдання видав | завдання прийняв |
| 1 | Т.Г. Ємел’яненко, доцент кафедри ЕОМ |  |  |
| 2 | Т.Г. Ємел’яненко, доцент кафедри ЕОМ |  |  |
| 3 | Т.Г. Ємел’яненко, доцент кафедри ЕОМ |  |  |
| 4 | Т.Г. Ємел’яненко, доцент кафедри ЕОМ |  |  |

1. Дата видачі завдання

# КАЛЕНДАРНИЙ ПЛАН

|  |  |  |  |
| --- | --- | --- | --- |
| №  з/п | Зміст етапів роботи | Термін виконання етапів роботи | Примітка |
| 1. | Аналіз предметної області, пошук аналогів та визначення з вимогами для Android-додатку коучсерфінгу |  |  |
| 2. | Визначення інструментарію та розробка дизайну додатку |  |  |
| 3. | Розробка мобільного додатку, створення бази даних, підключення БД до додатку, створення дизайну |  |  |
| 4. | Тестування додатку |  |  |
| 5. | Робота з помилками мобільного додатку та тестування групою осіб |  |  |
| 6. | Оформлення кваліфікаційної роботи |  |  |

|  |  |  |
| --- | --- | --- |
| Студент | (підпис) | Микола ВОЙЦЕХОВ  (Власне ім’я ПРІЗВИЩЕ) |
| Керівник роботи | (підпис) | Тетяна ЄМЕЛ’ЯНЕНКО  (Власне ім’я ПРІЗВИЩЕ) |

**РЕФЕРАТ**

**Кваліфікація робота:** 46 c., 11 рис., 1 табл., 9 джерел.

**Об’єкт дослідження:** метаевристичні методи розв’язання задачі комівояжера та асиметричної задачі маршрутизації транспортних засобів.

**Мета розробки:** розробити програмне забезпечення для розв’язання задачі комівояжера та асиметричної задачі маршрутизації транспортних засобів з обмеженою місткістю методом мурашиної колонії.

**Одержані висновки та їх новизна:** розроблено програмне забезпечення, що реалізує алгоритми мурашиної колонії для розв’язання задачі комівояжера і задачі маршрутизації транспортних засобів методом мурашиної колонії; проведено тестування розроблених алгоритмів на тестових наборах даних; виявлені переваги і недоліки наведеного алгоритму, знайдені шляхи покращення алгоритму.

**Реалізація:** програмне забезпечення написано на С# використовуючи платформу .NET 7.0 у середовищі розробки Microsoft Visual Studio 2022.

**Результати дослідження можуть бути використані** для диспетчеризації транспортування товарів.

**Ключові слова:** ЗАДАЧА КОМІВОЯЖЕРА, ПРОБЛЕМА МАРШРУТИЗАЦІЇ ТРАНСПОРТНИХ ЗАСОБІВ З ОБМЕЖЕНОЮ ЄМНІСТЮ, МЕТОД МУРАШИНОЇ КОЛОНІЇ.

# ANNOTATION

The graduation research of the 4-year student V. Soloviov (Oles Honchar Dnipro National University, Faculty of Applied Mathematics, Department of Mathematical Support of Computers), focuses on the development of software solutions for two well-known optimization problems: the Traveling Salesman Problem (TSP) and the Asymmetric Capacitated Vehicle Routing Problem (ACVRP). Specifically, the research aims to utilize the Ant Colony Optimization (ACO) algorithm to solve these problems effectively.

The primary objective of the research is to adapt an ACO algorithm to handle the complexities associated with the ACVRP and TSP. Both problems are known to be computationally challenging and have real-world applications in various fields such as logistics, transportation, and supply chain management.

The ACVRP involves finding optimal routes for multiple vehicles with different capacities to serve a set of customers, considering the capacity limitations of each vehicle and the asymmetric nature of the problem (where the travel distance between two points may vary depending on the direction taken). On the other hand, the TSP focuses on determining the shortest possible route that visits a set of cities exactly once and returns to the starting point, with the goal of minimizing the overall distance traveled.

By adapting the ACO algorithm for these two problems, V. Soloviov's research aims to provide an efficient solution. The performance of the developed algorithm will be evaluated and compared with existing approaches for solving the ACVRP and TSP. This comparison will likely involve benchmarking the algorithm on established problem instances and assessing factors such as solution quality, computation time, and scalability.

The outcomes of this work can have practical implications for industries that deal with routing and logistics optimization, offering more efficient and effective solutions to their complex problems.

Pages 46, Bibliography 9, pictures 11, tables 1.

# ЗМІСТ

[ПОСТАНОВКА ЗАДАЧІ 8](#_bookmark0)

1. [ТЕОРЕТИЧНІ ВІДОМОСТІ 9](#_bookmark1)
   1. [Огляд існуючих рішень 9](#_bookmark2)
   2. [Огляд існуючих технологій та засобів мобільної розробки 11](#_bookmark3)
   3. [Обґрунтування вибору обраних технологій та засобів розробки 12](#_bookmark4)
2. [ОПИС РОЗРОБЛЕНОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ…………. 15](#_bookmark5)
   1. [Аналіз вимог до програмного забезпечення 15](#_bookmark6)
   2. [Опис архітектури програмного забезпечення 16](#_bookmark7)
   3. [Use case діаграма 17](#_bookmark8)
   4. [Створення дизайну мобільного додатку 18](#_bookmark8)
   5. [Монетизація додатку 26](#_bookmark8)
3. [ОПИС ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ 28](#_bookmark12)
   1. [Структура бази даних 28](#_bookmark13)
   2. [Структура клієнтської частини додатку 30](#_bookmark14)
   3. [Структура серверної частини додатку 31](#_bookmark15)
   4. [Інструкція для користувачів 33](#_bookmark16)

[СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ 51](#_bookmark22)

**ВСТУП**

У сучасному світі, мобільні технології стають невід'ємною частиною нашого повсякденного життя, вже майже кожна людина з народження контактує з ними. Багато інновацій прийшли до нас за останні декілька років, які спростили наше життя. Серед таких інновацій особливе місце займає коучсерфінг – це соціальна практика, що дозволяє мандрівникам знаходити тимчасове безкоштовне житло та підтримку від місцевих жителів, щоб більше зануритися у місцеві традиції. Розробка мобільного додатку для Android, що сприятиме ефективному зв'язку та взаємодії між користувачами у цій сфері, є актуальною та перспективною задачею.

Цей дипломний проект має на меті створити надійний, інтуїтивно зрозумілий додаток для коучсерфінгу, який дозволить користувачам легко знаходити господарів і гостей, обмінюватись відгуками. Основна увага приділяється розробці зручного користувацького інтерфейсу, щоб кожна людина змогла його використовувати.

Важливість розробки такого додатку також підкреслюється швидким розвитком мобільних технологій та зростанням популярності культури незалежних подорожей. Використання цифрових платформ для сприяння культурному обміну і розширення можливостей для мандрівників стає не просто зручністю, а необхідністю. Тому цей проект відіграє ключову роль у розвитку цифрових сервісів у галузі туризму.

Проект сприятиме не тільки технічному, але й соціальному прогресу, дозволяючи людям з усього світу знаходити нових друзів, досліджувати нові культури та робити свої подорожі більш приємними та економічно вигідними. Ця робота є кроком до створення більш відкритого та з'єднаного світу, де кожен може стати частиною глобальної спільноти мандрівників.

# ПОСТАНОВКА ЗАДАЧІ

Основною метою є розробка мобільного додатку, який надаватиме користувачам зручний інтерфейс, широкі функціональні можливості для пошуку житла та спілкування, а також високий рівень безпеки під час взаємодії між гостями та господарями.

Додаток повинен містити наступний функціонал:

1. Система профілю(реєстрація та авторизація);
2. Система рейтингу – після дати закінчення оренди, у обох сторін з’являється можливість виставити оцінку;
3. Система скарг­ ­– якщо на користувача за останній час прийшло багато скарг(більше 5 скарг за останній місяць) то профіль буде заблоковано. Тільки адміністратор зможе розблокувати;
4. Система оренди – з можливістю наглядно на мапі подивитися, де знаходиться житло;
5. Пошук профіля та об’яв( та їх сортування);
6. Система адміністратора;
7. Система сповіщень ­– орендодавець та орендар, за день та в день оренди будуть отримувати сповіщення.

# ТЕОРЕТИЧНІ ВІДОМОСТІ

Коучсерфінг, як соціальна практика, набуває все більшої популярності, проте існуючі платформи для коучсерфінгу не завжди відповідають зростаючим вимогам користувачів щодо зручності, безпеки та функціональності. Відтак, розробка нового мобільного додатку для коучсерфінгу, який буде враховувати сучасні тенденції та потреби користувачів, є актуальною і важливою задачею.

Основною метою даної дипломної роботи є створення мобільного додатку, що надаватиме користувачам зручний інтерфейс, широкий функціонал для пошуку житла і спілкування, а також високий рівень безпеки під час взаємодії між гостями і господарями. Це передбачає розробку системи профілів, рейтингу, скарг, оренди, пошуку та сортування оголошень, адміністратора, а також системи сповіщень[1].

## Огляд існуючих рішень

Couchsurfing є найбільш відомою та широко використовуваною платформою для коучсерфінгу, яка з'єднує мандрівників з місцевими жителями, які готові безкоштовно прийняти їх у своїх домівках. Сайт та мобільний додаток дозволяють користувачам шукати господарів, встановлювати з ними контакт та планувати свої подорожі.

Преваги**:**

* 1. Величезна глобальна спільнота, що легко дозволяє знайти господаря або гостя майже у будь-якому куточку світу.
  2. Різноманітні можливості для взаємодії та обміну культурним досвідом.
  3. Функції безпеки, включаючи верифікацію профілів, систему відгуків та рекомендацій.

Недоліки**:**

* 1. Необхідність платити за отримання статусу верифікованого користувача.
  2. Можливі складнощі з навігацією та використанням інтерфейсу сайту або додатку.
  3. Іноді зустрічаються скарги на проблеми з безпекою та конфіденційністю.

BeWelcome — це некомерційна платформа для коучсерфінгу, яка націлена на сприяння культурному обміну та гостинності по всьому світу. Вона зосереджена на створенні довірливих взаємовідносин між користувачами та пропонує зручні інструменти для пошуку місць проживання та спілкування.

Переваги**:**

* 1. Безкоштовне користування всіма функціями, включаючи верифікацію.
  2. Сильне почуття спільноти та підтримки серед користувачів.
  3. Відкритість та некомерційний характер платформи.

Недоліки**:**

* + 1. Менша база користувачів у порівнянні з Couchsurfing.
    2. Обмежені функціональні можливості та інструменти на сайті.

Trustroots створена з метою об'єднання людей, які поділяють цінності довіри, відкритості та взаємодопомоги. Це платформа спрямована на мандрівників, які шукають більш глибокого занурення у місцеві культури та спільноти.

Преваги**:**

* 1. Фокус на створенні міцних взаємин та довіри між користувачами.
  2. Простий та легкий у використанні інтерфейс.
  3. Повністю безкоштовний доступ до всіх функцій.

Недоліки**:**

* 1. Відносно мала кількість користувачів.
  2. Менше географічне покриття, особливо в менш популярних регіонах.
  3. Обмежені можливості для верифікації користувачів.

Кожна з цих платформ має свої унікальні особливості та відповідає певним потребам користувачів. Вибір платформи залежить від індивідуальних уподобань, цілей подорожі та бажаного рівня взаємодії з місцевими жителями та іншими мандрівниками[2].

## Огляд існуючих технологій та засобів мобільної розробки

**Android SDK (Software Development Kit)** є основним інструментарієм для розробки додатків під Android ОС. Він включає набір розробницьких інструментів, таких як дебагер, бібліотеки, емулятор пристрою, документацію, приклади коду, а також специфічні API для використання функцій Android пристроїв. Розробники можуть використовувати Android SDK через Android Studio, офіційне інтегроване середовище розробки (IDE) від Google, що спрощує процес розробки і тестування додатків.

Переваги:

* 1. Широкі можливості для розробки і тестування додатків.
  2. Підтримка від Google та великої спільноти розробників.
  3. Регулярні оновлення та вдосконалення інструментів.

Недоліки:

* 1. Вимагає часу на вивчення та адаптацію до постійно оновлюваних інструментів.
  2. Може виявитися складним для новачків у мобільній розробці.

**Java** — це універсальна, об'єктно-орієнтована мова програмування, яка використовується для розробки Android додатків. Є однією з найпопулярніших мов програмування для Android, завдяки своїй портативності, безпеці та легкості у використанні. Java дозволяє розробникам писати код, який може виконуватися на різних платформах без змін, що робить її ідеальною для мобільної розробки.

**Переваги:**

* 1. Велика спільнота та багато ресурсів для навчання.
  2. Висока переносимість коду між різними платформами.
  3. Сильна підтримка з боку розробників інструментів та бібліотек.

Недоліки:

* 1. В порівнянні з Kotlin, може мати більш високу складність і меншу виразність коду.
  2. Може вимагати більше ресурсів системи та часу на виконання.

Flutter — це відкритий фреймворк від Google для створення високоякісних нативних інтерфейсів на iOS та Android з єдиного кодової бази. Flutter використовує мову програмування Dart і забезпечує високу продуктивність додатків завдяки компіляції в нативний код.

Переваги:

* 1. Єдина кодова база для iOS та Android, що значно скорочує час розробки.
  2. Велика кількість готових до використання віджетів та висока продуктивність додатків.
  3. Гаряча перезавантаження (hot reload) для швидкого тестування змін у коді.

Недоліки:

* 1. Вимагає вивчення мови програмування Dart, яка менш відома порівняно з Java або Kotlin.
  2. Можуть виникати складнощі при інтеграції з деякими нативними бібліотеками або при використанні специфічних платформних можливостей[3].

## Обґрунтування вибору обраних технологій та засобів розробки

Flutter є відкритим фреймворком від Google для створення високоякісних нативних інтерфейсів на iOS і Android з єдиного коду. Він використовує мову програмування Dart і надає розробникам потужний набір інструментів та віджетів для створення інтуїтивно зрозумілих та візуально привабливих мобільних додатків.

Переваги:

Швидка розробка: Гаряче перезавантаження дозволяє миттєво побачити результати змін у коді.

Кросплатформність: Єдиний кодовий базис для розробки додатків під Android та iOS.

Висока продуктивність: Компіляція у нативний код забезпечує високу продуктивність додатків.

Гнучкий UI: Великий вибір віджетів та можливість створення складних UI елементів.

Недоліки:

Вивчення нової мови: Необхідність вивчення Dart, хоча вона і легка для освоєння.

Розмір додатку: Додатки на Flutter можуть бути більшими за розміром порівняно з нативними[4].

Обмежений доступ до деяких нативних функцій: Може вимагати додаткової роботи для інтеграції специфічних нативних функцій[2].

PHP — це популярна загальноцільова скриптова мова програмування, яка активно використовується для розробки веб-сайтів. PHP легко інтегрується з HTML і використовується на серверній стороні, що робить його ідеальним для розробки динамічних веб-сайтів і веб-додатків.

Переваги:

Широка підтримка: Підтримується більшістю хостингових провайдерів.

Гнучкість: Легко інтегрується з різними базами даних, такими як MySQL, PostgreSQL та іншими.

Велика спільнота: Багато ресурсів для навчання, фреймворків та готових рішень.

Простота вивчення: Легко почати для новачків у програмуванні.

Недоліки:

Безпека: Вимагає ретельного написання коду для уникнення вразливостей.

Продуктивність: У деяких сценаріях може працювати повільніше порівняно з іншими мовами програмування[5].

MySQL — це найпопулярніша система управління базами даних SQL (СУБД), яка використовується у всьому світі для зберігання, обробки та отримання даних. Є відкритим програмним забезпеченням і часто використовується разом з PHP для розробки веб-додатків.

Переваги:

Висока надійність та стабільність: Масштабується для обробки великих обсягів даних.

Широка підтримка: Інтегрується з багатьма мовами програмування та платформами.

Легкість використання: Простий у вивченні синтаксис SQL для управління даними.

Безпека: Включає розширені функції безпеки для захисту даних.

Недоліки:

Обмеження в розподіленій обробці даних: Не найкращий вибір для розподілених систем баз даних порівняно з NoSQL рішеннями.

Складність оптимізації: Для великих та складних баз даних може вимагати додаткових зусиль для оптимізації продуктивності[6].

# ОПИС РОЗРОБЛЕННОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

## Аналіз вимог до програмного забезпечення

Під час аналізу вимог до програмного забезпечення коучсерфінга було проведено глибоке обговорення з потенційними користувачами та зацікавленими сторонами з метою зрозуміти їх потреби та очікування від додатку. Основні вимоги до програмного забезпечення визначено наступним чином:

1. Реєстрація/Авторизація: Користувач має можливість створити обліковий запис, для цього йому потрібно ввести дані про себе, пошту та пароль, після чого отримає листа на пошту та повинен ввести код. Для авторизації потрібно ввести пароль та пошту.

2. Відновлення аккаунту: Для відновлення аккаунту користувачу потрібно вказати пошту, після чого отримати листа на пошту та вказати код. Далі буде можливість змінити пароль.

3. Система профілю: Користувач має профіль де відображається основна інформація про нього, а також його рейтинг. Він має можливість вийти з аккаунту або видалити. Також користувачі мають можливість шукати інших людей та переглядати їх профілі.

4. Створення об’яви: Користувач має можливість додати нову об’яву, вказати адресу, кількість людей, дату заселення та виселення, додати фото домівки. З можливістю подальшого редагування та видалення.

5. Перегляд розміщених об’яв: При перегляді об’яв користувач має змогу сортувати, за датою створенням об’яви, за кількістю людей. Також шукати за країною або містом. Для більшої інформації о розташуванні житла, користувач має можливість переглянути на мапі.

6. Система скарг: Користувачі мають можливість шукати облікові записи інших користувачів та залишати на них скарги. Якщо на користувача прийшло більше 5 скарг за останній місяць, то його обліковий запис буде заблоковано. Для подальшого використання потрібно зв’язатися з адміністратором.

7. Система рейтингу: Після завершення дати об’яви для обох сторін стає доступна можливість оцінювання по 5 бальній шкалі.

8. Система сповіщення: За день та в день оренди користувачам буде приходити нагадування про оренду.

9. Система адміністратора: адміністратор має можливість видаляти об’яви та користувачів, переглядати жалоби, розблоковувати користувачів та відправляти їм повідомлення.

## Опис архітектури програмного забезпечення

Додаток розроблено на основі сучасних технологій: Flutter для розробки мобільного інтерфейсу, PHP для обробки серверної логіки та MySQL для зберігання даних. Ця комбінація технологій забезпечує високу продуктивність, масштабованість та гнучкість додатку.

Архітектура клієнтської частини (Flutter)

Flutter дозволяє створювати високопродуктивний та візуально привабливий інтерфейс користувача для мобільних платформ (iOS та Android) з єдиного кодової бази. Архітектура клієнтської частини базується на використанні віджетів, які є основними будівельними блоками для створення користувацького інтерфейсу в Flutter.

Використовується архітектурний патерн BLoC (Business Logic Component), який дозволяє ефективно управляти станом додатку, розділяючи інтерфейс користувача та бізнес-логіку. BLoC спрощує розробку та тестування додатку, забезпечуючи чисту архітектуру та високу масштабованість.

Архітектура серверної частини (PHP)

Серверна частина реалізована на PHP, мові програмування, яка широко використовується для розробки веб-додатків. PHP пропонує широкі можливості для створення динамічних веб-сторінок і взаємодії з базами даних.

## Use Case діаграма

Діаграма варіантів використання – це динамічна діаграма або діаграма поведінки в UML. Діаграми варіантів використання моделюють функціональність системи з використанням діючих осіб та сценаріїв використання. Варіанти використання – це набір дій, функцій, які повинна виконувати система. «Діючі особи» - люди, організації або програмне забезпечення, що діють під певними ролями в системі. Відповідно до кожного актора були визначені їх сценарії взаємодії з мобільним додатком, які відповідають всім визначеним вимогам, що були визначені в аналізі вимог до програмного продукту[7].

Для розробки діаграми варіантів використання Android-додатку були визначені актори :

− Admin – адміністратор, що представляє особу, що буде слідкувати за правилами платформи, блокувати користувачів, видаляти об’яви, що містять не допустимий контент;

− Customer – користувач додатку.

Після формування переліку акторів наступним етапом необхідно сформувати перелік сценаріїв :

− Реєстрація/Авторизація;

− Відновлення паролю;

− Перегляд профілю;

− Редагування/Видалення профілю;

− Створення об’яв;

− Редагування/Видалення об’яв;

− Відклик на об’яву;

− Блокування/Розблокування користувача;

− Залишення відгуку на об’яву;

− Відправлення скарг на інших користувачів.

На рис. 2.5 представлено діаграма варіантів використання для веб– додатку.



Рисунок 2.1– Use Case діаграма

## Створення дизайну мобільного додатку

Після аналізу теми та конкурентів, наступний крок створення дизайну додатку для коучсерфінгу. Першим ділом було створення вікно «Авторизації», так як при першому заході користувач повинен створити обліковий запис чи авторизуватися. Дизайн повинен містити простий дизайн, щоб користувачу було легко орієнтуватися в додатку, показано на рисунку 2.3.

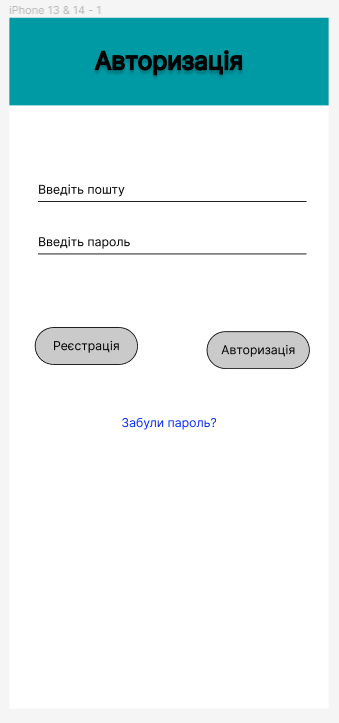


Рисунок 2.2– Дизайн «Авторизації»

Далі було створений макет «Реєстрації», це вікно повинне містити поля для вводу інформації про користувача, також містити кнопку для виходу на вікно «Авторизації», якщо випадково користувач зайшов до реєстрації, показано на рисунку 2.3.

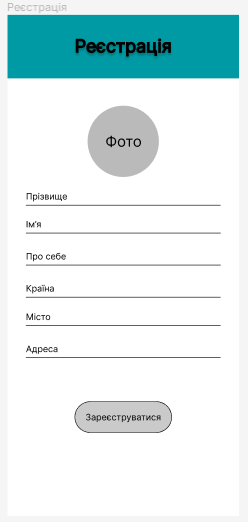


Рисунок 2.3 – Дизайн «Реєстрації»

Якщо користувач забув пароль від облікового профілю він повинен мати можливість відновити його, тому далі було створено вікно «Відновлення паролю». Там знаходиться заголовок, щоб користувач зрозумів, що за вікно, кнопка для виходу назад, а також поля для вводу пошти, коду та нового паролю, дизайн наведено на рисунку 2.4.

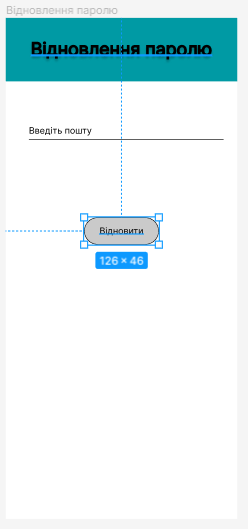


Рисунок 2.3 – Дизайн «Відновлення паролю»

Наступне вікно, це вікно профілю, воно повинно містити мінімум інформації та бути інформативним. Тому вікно містить наступну інформацію: фото, ім’я, прізвище користувача, рейтинг, адресу проживання та коротку інформацію о собі. Також знаходяться кнопки для видалення та виходу з профілю, якщо це перегляд іншого користувача, тоді цих можливостей не має, але є для відправлення скарги. Показано на рисунку 2.4.

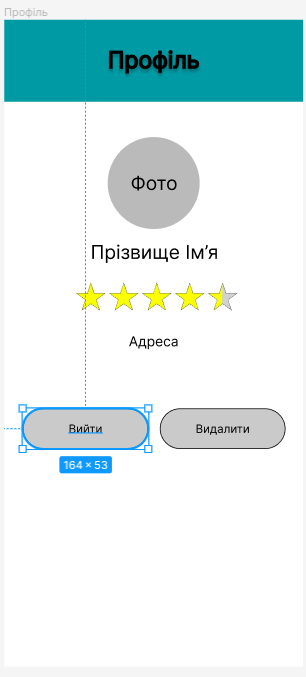


Рисунок 2.4 – Дизайн «Профіль»

Далі розроблено дизайн для вікна «Пошук користувачів». Вікно містить тільки поле для введення інформацію та виведення підходящих користувачів, приклад дизайну наведено на рисунку 2.5.

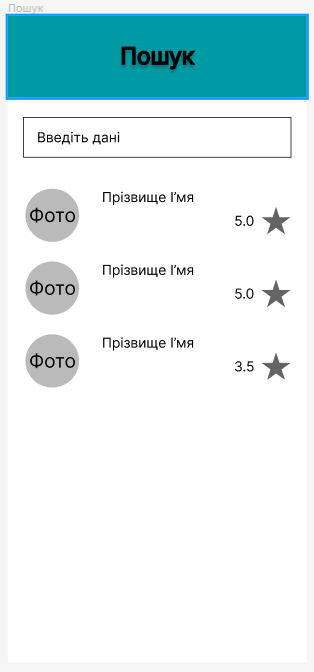


Рисунок 2.5 – Дизайн «Пошук користувача»

Наступне вікно «Мапа», воно містить тільки карту та маркер для мітки адреси. Показано на рисунку 2.6.

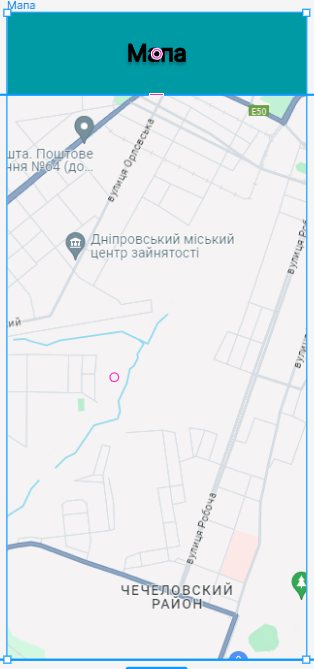


Рисунок 2.6 – Дизайн «Мапи»

Вікно «Об’яв» повинен містити багато дій, тому для його дизайну потрібно віднестись більш детально. Під заголовком вікна повинні знаходитись кнопки для перегляду всіх об’яв, створених об’яв користувачем та відгуки користувача. Далі повинно міститись об’яви з короткою інформацією: фото, назва, адреса житла та дати оренди. Майбутній дизайн показано на рисунку 2.7.



Рисунок 2.7 – Дизайн «Головної сторінки»

Останній великий макет це дизайн вікна «Об’ява». В цьому вікні користувач ознайомлюється з усією інформацією о житлі, та його власнику. Дизайн повинен бути зрозумілим та простим, тому спочатку розміщено фотографії житла, далі основна інформацію, потім кнопки для взаємодії та в самому кінці інформація об орендарі. Дизайн вікна «Об’ява» показано на рисунку 2.8.



Рисунок 2.8 – Дизайн «Об’яви»

## Монетизація додатку

Монетизація додатку для коучсерфінгу є важливим аспектом для забезпечення його довготривалої працездатності та майбутнього розвитку. Враховуючи цільову аудиторію, яка цінує час та доступність, необхідно знайти баланс між безкоштовними послугами та платними функціями, які користувачі вважатимуть вартих грошей. Далі наведено можливі стратегії монетизації, які можуть бути впроваджені у майбутньому.

1. Преміум підписка. За допомогою неї користувач зможе розширити функції для пошуку (більш розширені фільтри та опції пошуку об’яв), безрекламний інтерфейс (відсутність реклами), пріоритетна підтримка, ексклюзивний контент (навчальні матеріали, інформація про країни);
2. Розміщення реклами. Інтеграція реклами може бути ефективним способом отримання доходу, особливо якщо у додатку велика користувацька база. Для цільової аудиторії додатку можна використовувати тільки банерну рекламу (наприклад у вікні «Об’ява»), так як вона не буде заважати користувача і відволікати їх від ознайомлення.
3. Спонсорство та партнерські програми. Спонсорські угоди з компаніями, що надаюсь послуги або продукти, що можуть бути корисними для користувачів додатку. Наприклад розміщення відео або статей, інтеграція з іншими сервісами де користувачі отримують спеціальні пропозиції або знижки.

Ефективна монетизація для додатку вимагає ретельного планування та тестування різних стратегій. Важливо зберігати баланс між наданням цінних безкоштовних послуг та створенням преміум функцій.

# ОПИС ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

## Структура бази даних

База даних для мобільного додатку коучсерфінгу включає наступні таблиці: `users`, `rate`, `rental`, `report`. Кожна з таблиць має специфічні зв’язки та атрибути. Далі наведено детальний опис таблиць бази даних.

Таблиця **`users`** – зберігає всю інформацію про користувачів додатку. Містить наступні атрибути:

* **`UserID`**(int) – Унікальний ідентифікатор користувача, є primal key;
* **`Surname`**(varchar(50)) – Прізвище користувача;
* **`Name`**(varchar(50)) – Ім’я користувача;
* **`Email`**(varchar(50)) – Електронна пошта користувача;
* **`BIO`**(text) – Коротка інформація про себе;
* **`ProfilePicture`**(text) – Посилання на фотографію профілю.
* **`Password`**(varchar(100)) – Пароль до профілю, зберігається хеш пароль;
* **`CreateAt`**(date) – Дата створення профілю;
* **`Country`**(varchar(50)) – Країна проживання;
* **`City`**(varchar(50)) – Місто проживання;
* **`Street`**(varchar(100)) – Вулиця проживання.

Таблиця **`report` –** зберігає інформацію про скарги користувачів. Має наступні атрибути:

* **`ReportID`**(int) – Унікальний ідентифікатор скарги, є primal key;
* **`firstID`**(int) – Ідентифікатор користувача, який подає скаргу;
* **`secondID`**(int) – Ідентифікатор користувача, на якого подається скарга;
* **`Info`**(text) – Деталі скарги;
* **`Date`**(date) – Дата подачі скарги.

Таблиця **`rate` –** зберігає інформацію про оцінки користувачів. Має наступні атрибути:

* **`RateID`**(int) – Унікальний ідентифікатор оцінки, є primal key;
* **`firstID`**(int) – Ідентифікатор користувача, який виставляє оцінку;
* **`secondID`**(int) – Ідентифікатор користувача, якому виставляється оцінка;
* **`Stars`**(int) – Кількість зірок(оцінка);
* **`Info`**(text) – Коментар до оцінки;
* **`RentalID`**(int) – Ідентифікатор оренди, з якою пов'язана оцінка.

Таблиця **`rental` –** зберігає інформацію про оголошення оренди житла. Має наступні атрибути:

* **`RentalID`**(int) – Унікальний ідентифікатор оголошення, є primal key;
* **`Title`**(varchar(50)) – Назва оголошення;
* **`Description`**(text) – Опис оголошення;
* **`MaxPeople`**(int) – Максимальна кількість людей, що може розмістити в себе орендар;
* **`Country`**(varchar(50)) – Країна розташування житла;
* **`City`**(varchar(50)) – Місто розташування житла;
* **`Street`**(varchar(100)) – Адреса розташування житла;
* **`DateRental`**(date) – Дата початку оренди;
* **`DateEviction`**(date) – Дата завершення оренди;
* **`LessorID`**(int) – Ідентифікатор користувача, що здає житло;
* **`RenterID`**(int) – Ідентифікатор користувача, що знімає житло;
* **`DateCreate`**(date) – Дата створення оголошення;
* **`PhotoCount`**(int) – Кількість фотографій у оголошенні.

### Зв'язки між таблицями

* **`report`**: Зв'язана з таблицею `**users`** через `**firstID`** та `**secondID`**, які є зовнішніми ключами, що посилаються на `**UserID`** у таблиці `**users`**.
* **`rate`**: Зв'язана з таблицею `**users`** через `**firstID`** та `**secondID`**, які є зовнішніми ключами, що посилаються на `**UserID`** у таблиці `**users`**. Також має зв'язок з таблицею `**rental`** через `**RentalID`**.
* **`rental`**: Зв'язана з таблицею `**users`** через `**LessorID`** та `**RenterID`**, які є зовнішніми ключами, що посилаються на `**UserID`** у таблиці `**users`**.

Далі наведено графічне представлення бази даних, показано на рисунку 3.1.

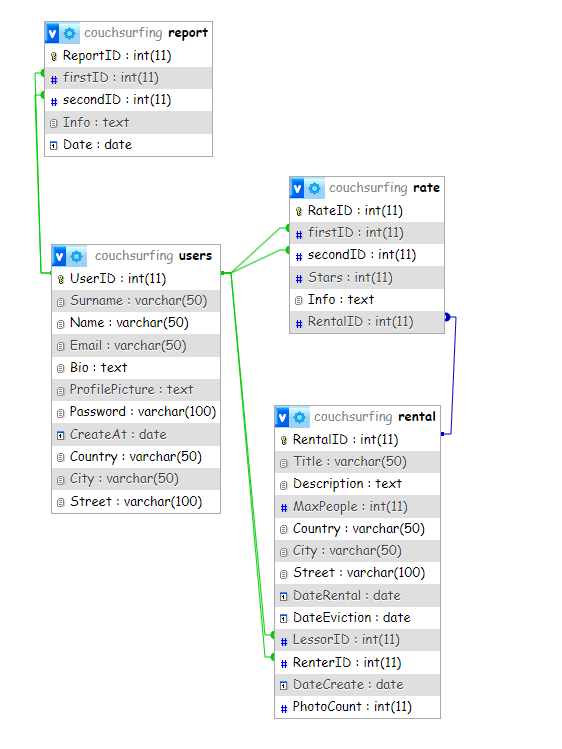


Рисунок 3.1 – Структура БД

## Структура клієнтської частини додатку

Клієнтська частина мобільного додатку для коучсерфінгу розроблена на мові програмування Dart з використанням фреймворку Flutter. Вона складається з кількох основних модулів(файлів), кожен з яких відповідає за певний функціонал додатку. Далі наведено детальний опис структури клієнтської частини Android-додатку.

* **`main.dart`** – Головний файл, який ініціалізує необхідні компоненти, перевіряє чи авторизований в додатку користувач;
* **`auth.dart`** – Модуль, в якому користувач авторизується, обробляється інформація;
* **`reg.dart` –** Файл, для вводу даних користувача для подальшої реєстрації(Прізвище, Ім’я, Електронна пошта, пароль тощо). Валідація даних, а також відправка листа на пошту, для підтвердження пошти;
* **`account.dart` –** Модуль призначений для роботи з профілем користувача, з можливістю наступних дій: перегляд основної інформації, видалення профілю з додатку, вихід з профілю на даному пристрої;
* **`home.dart` –** Файл, який відповідає за відображення об’яв про оренду житла, а також перегляду створених користувачем об’яв та об’яв з його відгуком;
* **`rental.dart` –** Сторінка для створення оголошення, загрузки зображень, з валідацією даних.
* **`rentalDetails.dart` –** Модуль , для взаємодії з об’явою. На даній сторінці можна подивитись основну інформацію про житло, автора об’яви та відгукнутися на неї;
* **`find.dart`** – Файл, який відповідає за пошук користувачів.
* **`map.dart`** – Модуль, відображення інтерактивної карти, для наглядного перегляду адреси орендованого житла;
* **`serverInfo.dart` –**  Файл, для отримання інформації про сервер додатку;
* **`editProfile.dart `** – Потрібен для редагування профілю користувача.

На рисунку 3.2 наведено структура клієнтської частини.

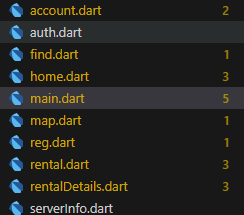


Рисунок 3.2 – Структура клієнтської частини

## Структура серверної частини додатку

Серверна частина мобільного додатку для коучсерфінгу розроблена на мові програмування PHP. Вона складається з кількох основних модулів(файлів), кожен з яких відповідає за певний функціонал додатку. Далі наведено детальний опис структури серверної частини мобільного додатку.

* **`addTable.php`** – Відповідає за додавання нового запису до таблиць бази даних;
* **`auth.php` –** Повертає хеш пароль з відповідного користувача за заданою поштою. Основна його функція це авторизація в додатку.
* **`destroyTable.php` –** Потрібен для видалення записів з бази даних.
* **`findTable.php` –** Повертає середній рейтинг користувача за заданою поштою;
* **`getID.php` –** Відповідає за отримання унікального ключа користувача за поштою;
* **`getRentalUser.php` –** Повертає всю інформацію про об’яву за заданими параметрами;
* **`getRateInfo.php` –** Для отримання інформації про оцінку користувача, за його унікальним номером та номером об’яви;
* **`getRental.php` –** Повертає усі об’яви;
* **`getReport.php` –** Відповідає за отримання кількості жалоб на певного користувача;
* **`getTable.php` –** Потрібен для отримання інформації про користувача за його поштою;
* **`getUserForID.php` –** Повертає інформацію про користувача за його унікальним номером;
* **`lastRental.php` –** Відповідає за повернення інформації про останні оголошення оренди;
* **`rate.php` –** Повертає середній рейтинг користувача за заданою поштою;
* **`respondRental.php` –** Відповідає за перевірку, чи не відгукнувся користувач на декілька об’яв в один день;
* **`sendEmail.php` –** Потрібен для відправки листа на пошту користувачу;
* **`updateInfo.php` –** Відповідає за зміну паролю користувачем;
* **`uploadPhoto.php`** – Завантажує фото профілю користувача на сервер додатку;
* **`editProfile.php`** – Відповідає за оновлення даних користувачів;
* **`uploadRental.php` –** Завантажує фото об’яви на сервер.

На рисунку 3.3 наведено структура серверної частини.

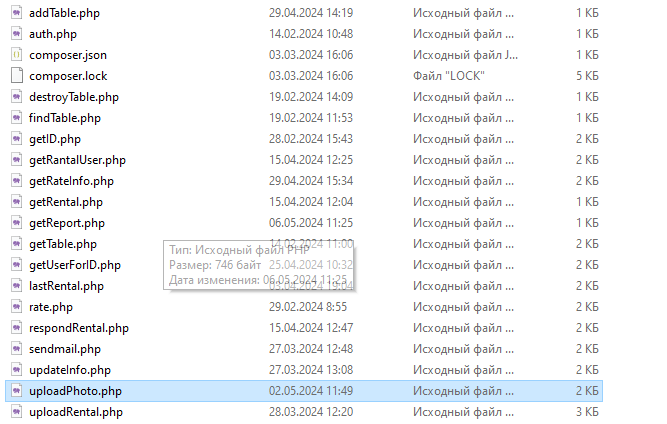


Рисунок 3.3 – Структура серверної частини

## Інструкція для користувача

При першому заході користувача в додаток його зустріне вікно авторизації, в якому користувач може : авторизуватися (за допомогою пошти та паролю), перейти у вікно реєстрації та відновити пароль(за допомогою пошти), показано на рисунку 3.4.

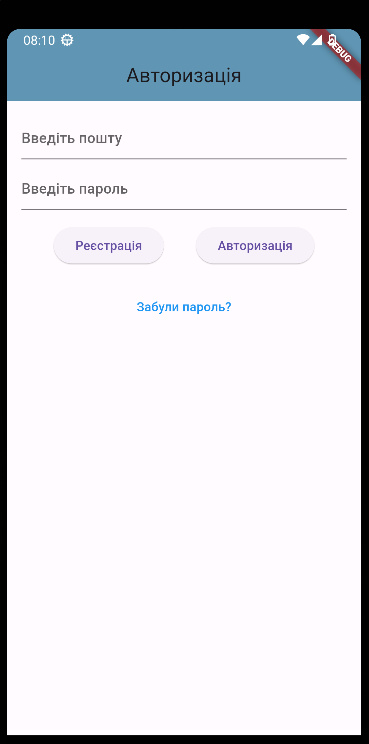


Рисунок 3.4 – Вікно авторизації

При авторизації, якщо користувач ввів не коректні данні, йому покажеться відповідне повідомлення, показано на рисунку 3.5.

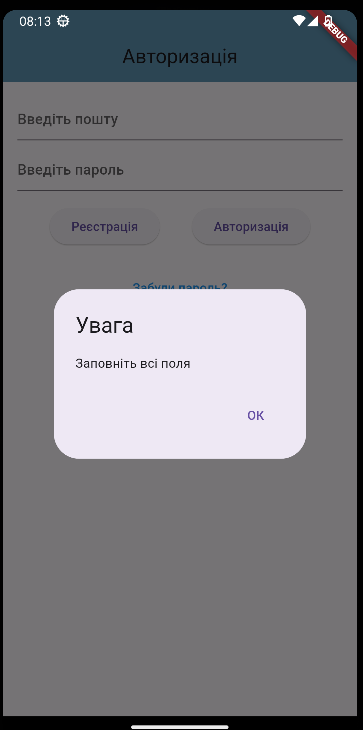


Рисунок 3.5 – Виведення помилки

Після того, як користувач ввів коректні дані від свого профілю, його перекине на вікно з об’явами, приклад приведено на рисунку 3.6.

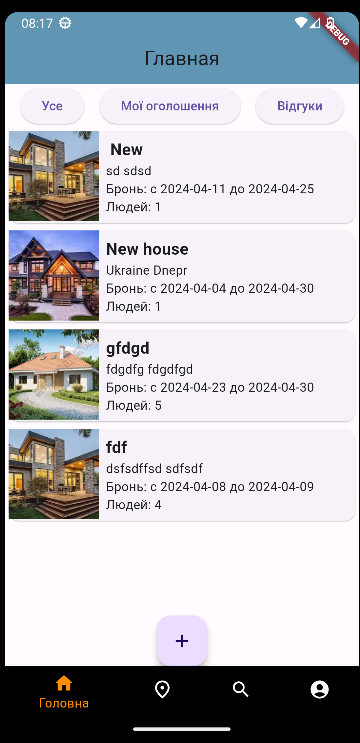


Рисунок 3.6 – Головне меню

Якщо користувач не має профілю в мобільному додатку, тоді він повинен натиснути кнопку «Реєстрація» у вікні авторизації, розміщення кнопки показано на рисунку 3.7.

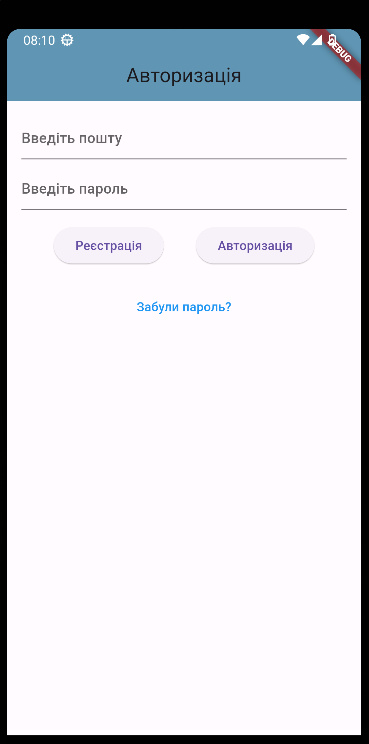


Рисунок 3.7 – Розміщення кнопки «Реєстрації»

Наступним кроком буде введення інформації про користувача та фото профілю, якщо користувач введе не коректну інформацію, тоді він отримає відповідне повідомлення про помилку, показано на рисунках 3.8, 3.9.

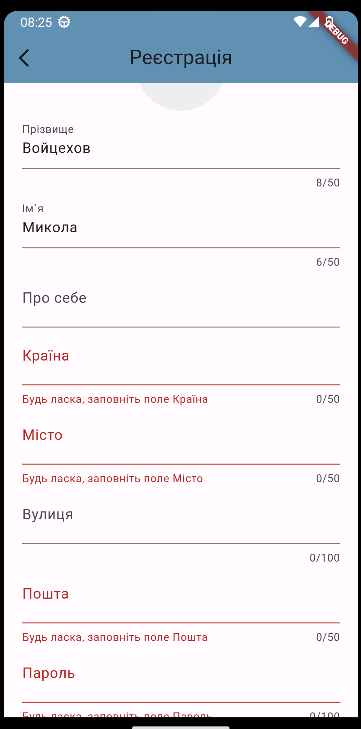


Рисунок 3.8 – Виведення помилки

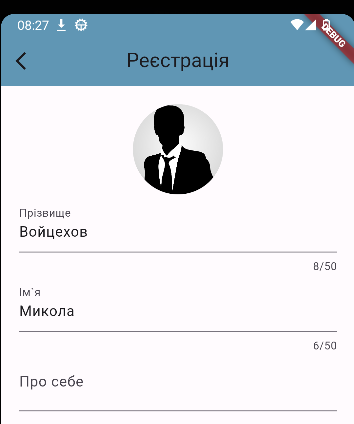


Рисунок 3.9 – Обране фото профілю

Після введення коректних даних, останнім кроком реєстрації є підтвердження пошти, користувачу прийде лист з кодом підтвердження, йому потрібно його ввести. Після введення коректного коду користувача перекине на головну сторінку, показано на рисунках 3.10, 3.11.

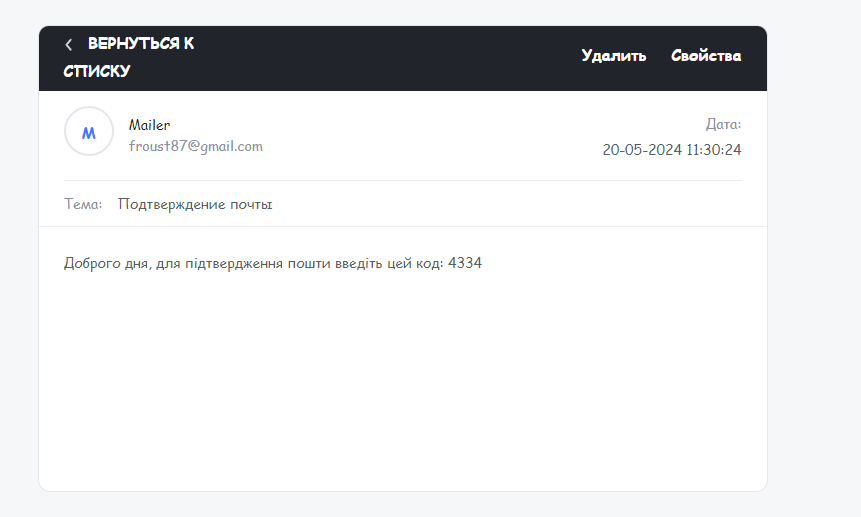


Рисунок 3.10 – Отримане письмо

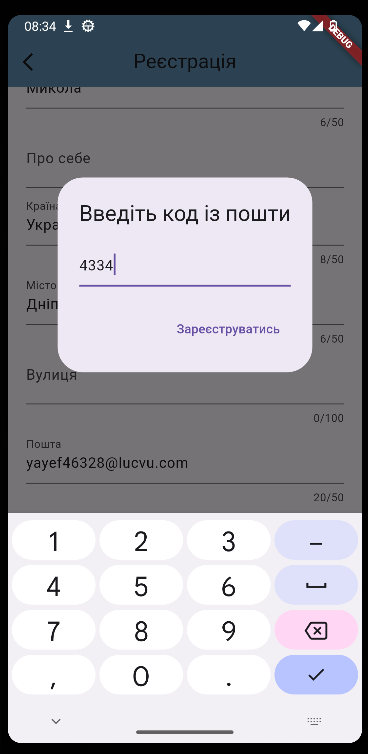


Рисунок 3.11 – Введення коду

Коли користувач забув пароль від свого профілю, він має можливість відновити його. Для цього потрібно натиснути «Забули пароль?», далі ввести пошту на яку зареєстровано профіль, після чого ввести код який прийшов на пошту та ввести новий пароль, показано на рисунках 3.12-3.14.

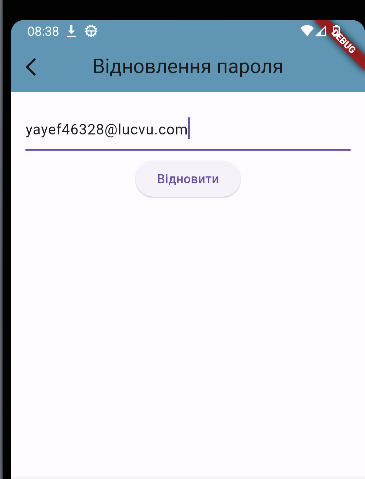


Рисунок 3.12 – Відновлення паролю

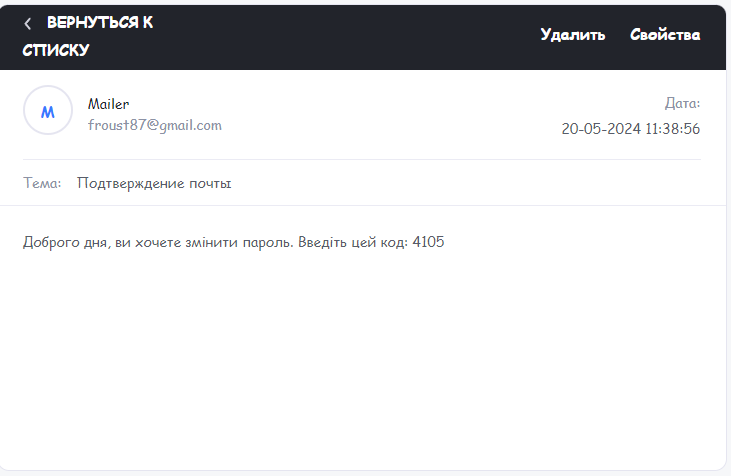


Рисунок 3.13 – Отриманий лист

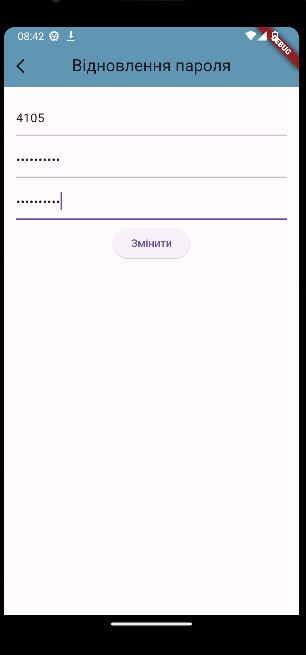


Рисунок 3.14 – Зміна паролю

За допомогою нижнього навігаційного меню користувач може перейти на :головну, мапу, пошук, профіль. Для переходу на сторінку свого профілю користувач повинен натиснути на останню іконку, показано на рисунку 3.15.

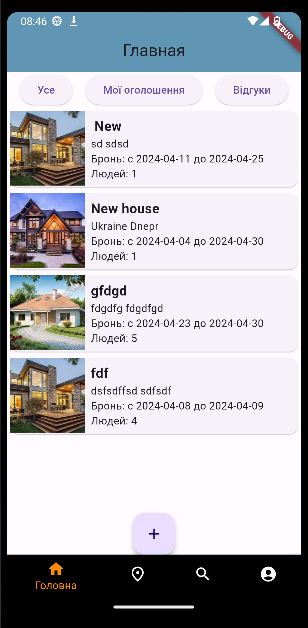


Рисунок 3.15 – Навігація в додатку

На даній сторінки користувач може подивитися інформацію про себе(фото профілю, свій рейтинг, коротку інформацію про себе), вийти з додатку та видалити свій профіль. Якщо користувач видаляє профіль, то спочатку отримає повідомлення і тільки після його підтвердження, профіль буде видалено, показано на рисунках 3.16,3.17.

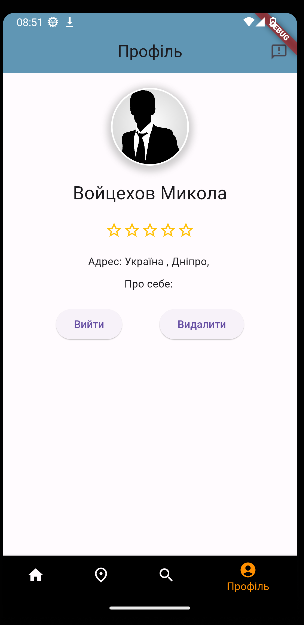


Рисунок 3.16 – Профіль користувача

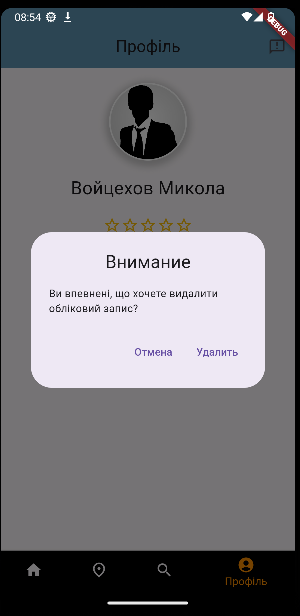


Рисунок 3.17 – Видалення профілю

Якщо при реєстрації було вказано не вірно дані або дані застарілі, тоді користувач натиснувши кнопку «Редагувати» у профілі перейде до вікна, де зможе оновити свої дані, показано на рисунках 3.18,3.19.

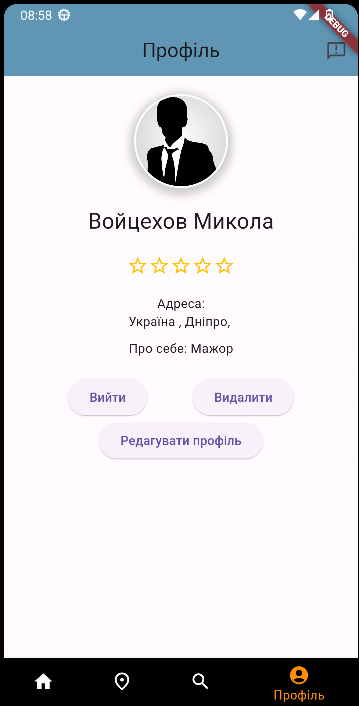


Рисунок 3.18 – Редагування профілю

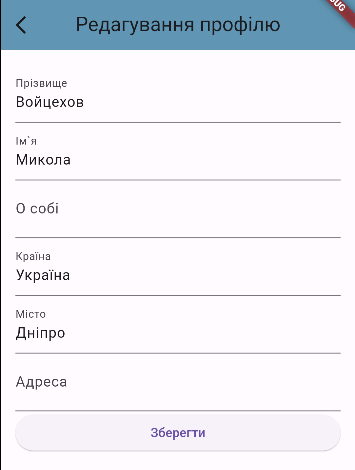


Рисунок 3.19 – Зміна даних

Для пошуку користувача, потрібно перейти на 4 вкладку у нижній панелі, після чого у відповідне поле почати вводити інформацію про особу що шукають, та натиснути лупу. Далі буде виведено користувачів, що підходять під запит, натиснувши на потрібний запис користувач перейде до профілю даної людини. Показано на рисунках 3.20-3.22.

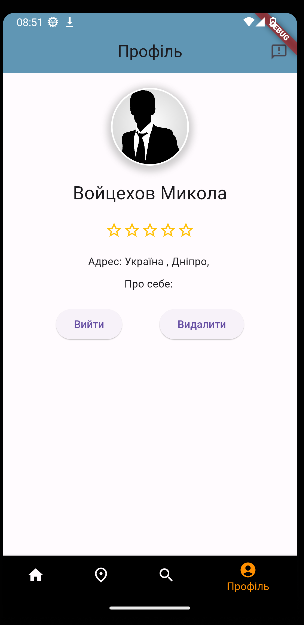


Рисунок 3.20 – Пошук профілю

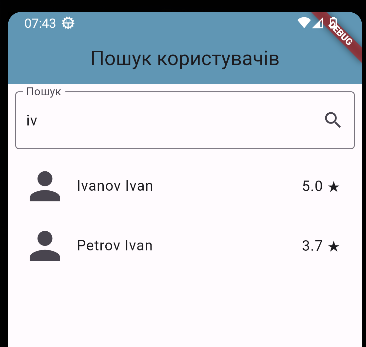


Рисунок 3.21 – Найдені профілі

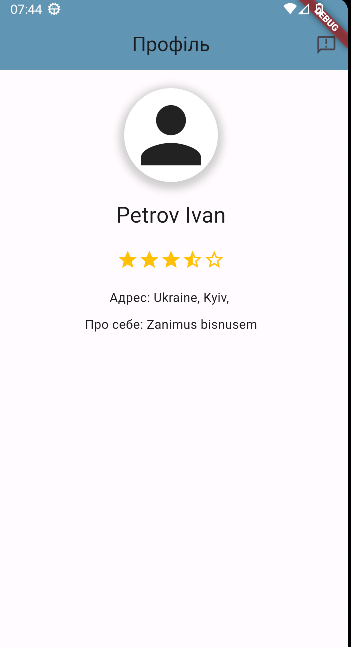


Рисунок 3.22 – Перегляд профілю

Щоб залишити скаргу на користувача, треба у його профілі натиснути кнопку у лівому правому краю, після чого обрати причину скарги та відправити її, приклад приведено на рисунку 3.23.

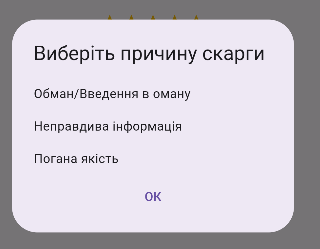


Рисунок 3.23 – Відправлення скарги

На головній сторінці користувач, за допомогою верхньої панелі переглянути усі об’яви, свої об’яви та відгуки, показано на рисунку 3.24.

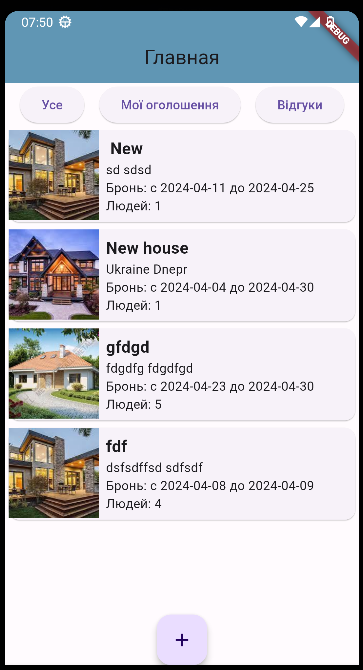


Рисунок 3.24 – Головна сторінка

Для перегляду інформацію про об’яву потрібно просто натиснути на об’яву, після чого буде показана повна інформація про неї та користувача, що створив її, показано на рисунку 3.25.

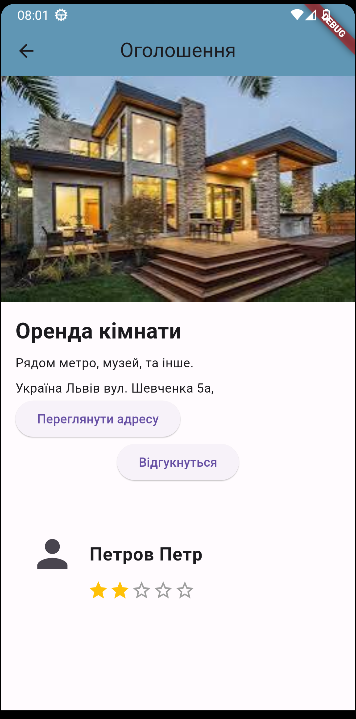


Рисунок 3.25 – Основна інформація

Для того щоб відгукнутися, потрібно натиснути кнопку «Відгукнутися», якщо користувач вже відгукнувся на об’яву та дати пересікаються виведеться відповідне повідомлення, в іншому випадку бронювання пройде успішно, показано на рисунках 3.26,3.27.

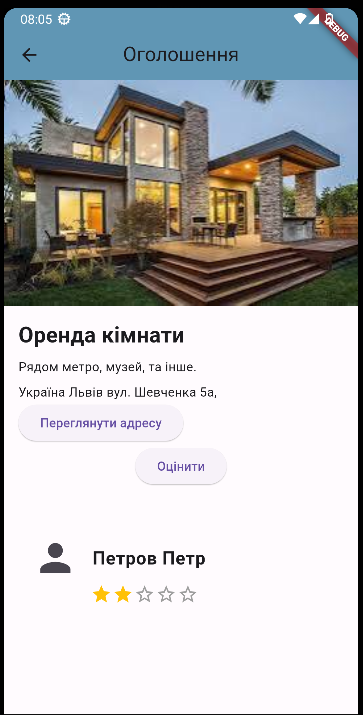


Рисунок 3.26 – Успішне бронювання



Рисунок 3.27 – Конфлікт дат

Щоб створити об’яву потрібно перейти на головну натиснути кнопку внизу, після чого ввести інформацію, обрати дату, фото та натиснути кнопку «Додати». Якщо користувач забув вести всю необхідну інформацію, він отримає помилку, показано на рисунках 3.28-3.30.

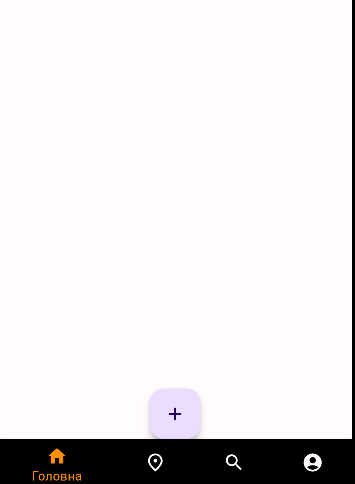


Рисунок 3.28 – Кнопка створення об’яви

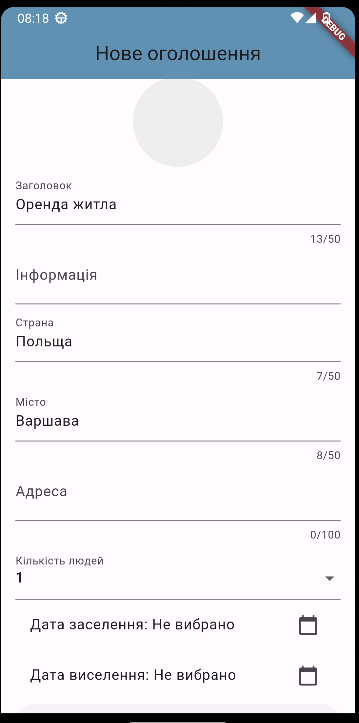


Рисунок 3.29 – Введення інформації для об’яви

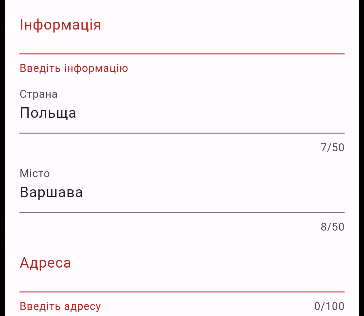


Рисунок 3.30 – Виведення помилки

Якщо переглянути створену об’яву, то є можливість її видалити, показано на рисунку 3.31.

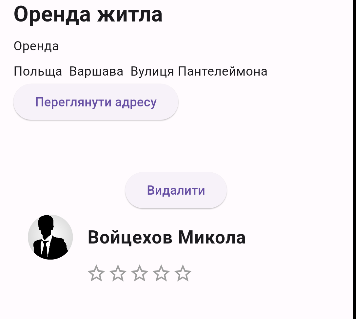


Рисунок 3.31 – Своя об’ява

Після того як закінчилася дата оренди для обох сторін доступна оцінка іншої сторони оренди. Для цього потрібно перейти на об’яву та натиснути «Оцінити» та обрати оцінку. показано на рисунках 3.32,3.33.

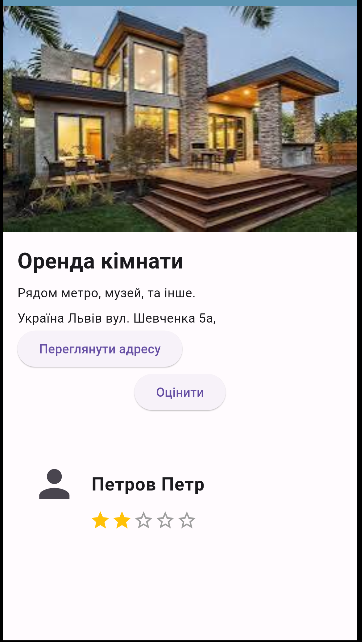


Рисунок 3.32 – Оцінювання оренди

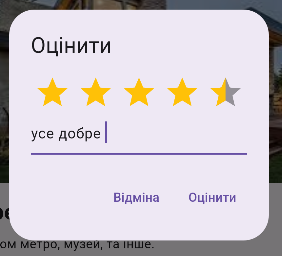


Рисунок 3.33 – Панель оцінювання

# ВИСНОВКИ

Під час роботи над проєктом я отримав безцінний досвід у різних аспектах програмування та розробки програмного забезпечення. Розробка цього проєкту дозволила мені ознайомитися з новими для мене технологіями, такими як Flutter для мобільної розробки, мови програмування Dart та PHP для роботи з сервером, також покращив навички роботи з базою даних MySQL.

Завдяки такому додатку, користувачі отримують можливість не тільки знайти безкоштовне житло на час своїх подорожей, але й зустріти нових друзів, обмінятися культурним досвідом та розширити свій світогляд.

Процес розробки такого додатку вимагає глибокого розуміння потреб цільової аудиторії, а також технічних аспектів мобільної розробки. Використання сучасних технологій, таких як Flutter для кросплатформної розробки, PHP і MySQL для створення серверної частини та обробки даних, дозволяє створити надійний, функціональний та привабливий для користувача продукт.

Важливим аспектом розробки є забезпечення безпеки та приватності даних користувачів, особливо враховуючи особистий характер інформації, що обмінюється між гостями та господарями. Також необхідно передбачити механізми верифікації користувачів і відгуків, щоб підвищити довіру та безпеку в спільноті коучсерфінгу.

Завершення проєкту мобільного додатку для коучсерфінгу відкриває перед спільнотою мандрівників нові можливості для взаємодії та підтримки Цей додаток не тільки спрощує пошук житла у подорожах, але й сприяє розвитку культурного обміну та поглибленню міжнародного розуміння та дружби.

# 

# СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

* 1. couchsurfing що це та як користуватися URL: https://www.phototravel.dp.ua/14-useful/298-couchsurfing.html. (дата звернення: 27.02.2024).
  2. Конкуренти коучсерфінгу URL: https://www.similarweb.com/ru/website/couchsurfing.com/competitors. (дата звернення: 05.03.2024).
  3. Які технології використовують для розробки Android додатків URL: https://smart-solutions.com.ua/archives/632. (дата звернення: 05.03.2024).
  4. Flutter, чи можливо їм користуватися. URL: https://www.purrweb.com/ru/blog/razrabotka-na-flutter-plyusi-minusi/ (дата звернення: 28.02.2024).
  5. PHP . URL: https://freehost.com.ua/ukr/faq/wiki/chto-takoe-php/ (дата звернення: 28.02.2024).
  6. MySQL vs MongoDB. URL: https://robotdreams.cc/blog/207-mysql-ili-mongodb-chto-vybrat (дата звернення: 28.02.2024).
  7. Що таке Use case діаграма . URL: https://dou.ua/lenta/articles/use-cases/ (дата звернення: 10.03.2024).