

Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра програмного забезпечення

Бакалаврська дипломна робота

на тему: Розробка програмного мобільного застосунку для мешканців ОСББ

Виконав: студент IV курсу
групи ЗПІ-206
спеціальності

121 – Інженерія програмного забезпечення

(шифр і назва напрямку підготовки, спеціальності)

Ковальчук М.В.

(прізвище та ініціали)

Керівник: к.т.н., доц. каф. ПЗ Ракитянська Г.Б.

(прізвище та ініціали)

Рецензент: к.т.н., доц. каф. ОТ Городецька О.С.

(прізвище та ініціали)

Допущено до захисту

Зав. кафедри ПЗ

« 10 » червня 2024 р.

Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра програмного забезпечення
Рівень вищої освіти перший (бакалаврський)
Галузь знань 12 «Інформаційні технології»
Спеціальність 121 «Інженерія програмного забезпечення»
Освітньо-професійна програма «Інженерія програмного забезпечення»

ЗАТВЕРДЖУЮ
Завідувач кафедри ПЗ
О. Н. Романюк
« 12 » травня 2024 р.

ЗАВДАННЯ НА БАКАЛАВРСЬКУ ДИПЛОМНУ РОБОТУ СТУДЕНТУ

Ковальчуку Максиму Володимировичу

1. Тема роботи – «Розробка програмного мобільного застосунку для мешканців ОСББ».

Керівник роботи: Ракитянська Ганна Борисівна, кандидат техн. наук, доцент кафедри програмного забезпечення, затверджені наказом вищого навчального закладу від 11 березня 2024 р. № 80.

2. Строк подання студентом роботи 3 червня 2024.

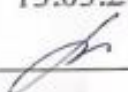
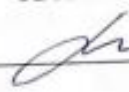
3. Вихідні дані до роботи: середовище розробки – Visual Studio Code, мови розробки – Dart, фреймворк – Flutter засоби організації даних програми – база даних Firebase, операційна система – Android.

4. Зміст текстової частини: вступ; Аналіз розвитку мобільних систем з використанням програмних засобів та постановка задач дослідження; розробка структури та алгоритмів програмного продукту; розробка програмних модулів реалізація в мобільному застосунку; тестування програми; висновки; перелік посилань; додатки; графічна частина.

5. Перелік ілюстративного матеріалу: титульний слайд; мета, предмет і об'єкт дослідження; новизна та практична цінність; актуальність розробки; задачі дослідження; розробка структури інтерфейсу програмного застосунку; блок-схема алгоритму авторизації дій користувача; використані технології;

навчальна програма; перевірка завдань; тестування програми; апробація та публікація роботи.

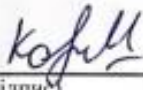
6. Консультанти розділів роботи

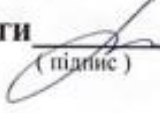
Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		Завдання видав	Завдання прийняв
1-4	Ракитянська Г.Б., к.т.н., доцент кафедри ПЗ	13.03.24 	03.06.24 

7. Дата видачі завдання 13 березня 2024р.

КАЛЕНДАРНИЙ ПЛАН

Ч.ч.	Назва етапів бакалаврської дипломної роботи	Строк виконання етапів роботи	Примітка
1	Обґрунтування вибору методу розробки та постановка задач дослідження	14.03.23 – 02.04.23	Виконано
2	Розробка структури та алгоритмів програмного продукту	03.04.23 – 10.04.23	Виконано
3	Розробка інтерфейсу	10.04.23 – 15.04.23	Виконано
4	Розробка модулів в мобільному застосунку	15.04.23 – 05.05.23	Виконано
5	Тестування програми	05.05.23 – 10.05.23	Виконано
6	Оформлення матеріалів до захисту БДР	11.05.23 – 30.05.23	Виконано

Студент  Ковальчук М.В.
(підпис) (ініціали та прізвище)

Керівник бакалаврської дипломної роботи  Ракитянська Г.Б.
(підпис) (ініціали та прізвище)

АНОТАЦІЯ

Бакалаврська дипломна робота складається зі 70 сторінок формату А4, на яких є 43 рисунки, 3 таблиць, список використаних джерел містить 23 найменування.

У бакалаврській дипломній роботі розглядається процес розробки та реалізації мобільного застосунка, спрямованого на задоволення потреб мешканців об'єднання співвласників багатоквартирних будинків (ОСББ). Досліджуються вимоги до функціоналу застосунку, розробляється його архітектура та інтерфейс. Реалізація базового функціоналу, тестування та виправлення помилок також розглядаються.

Особлива увага приділяється інтеграції застосунка з іншими інструментами ОСББ для підвищення ефективності та зручності його використання. Результати цієї роботи мають значення для подальшого удосконалення управління багатоквартирними будинками та поліпшення якості життя їх мешканців.

Застосунок розроблено з використанням мови програмування Dart та середовищ розробки програмного забезпечення Visual Studio Code.

Отримані в бакалаврській дипломній роботі результати можуть бути використані для покращення якості життя мешканців багатоквартирних будинків шляхом створення зручного та функціонального мобільного застосунка. Цей застосунок сприятиме ефективнішому управлінню ОСББ, полегшить доступ до необхідної інформації та послуг, а також збільшить залученість мешканців до спільного життя та рішення повсякденних питань.

Ключові слова: об'єднання співвласників багатоквартирних будинків, зручність, покращення якості життя, програмний застосунок.

ABSTRACT

The Bachelor's thesis comprises 70 pages of A4 format, including 43 figures, 3 tables, and a bibliography containing 23 references.

This Bachelor's thesis investigates the process of developing and implementing a mobile application aimed at meeting the needs of residents of housing cooperatives (HOAs). The functional requirements of the application are examined, and its architecture and interface are designed. The implementation of basic functionality, testing, and error correction are also discussed.

Special attention is paid to integrating the application with other HOA tools to enhance its efficiency and usability. The results of this work are significant for further improving the management of multi-apartment buildings and enhancing the quality of life for their residents.

The application is developed using the Dart programming language and the Visual Studio Code development environment.

The findings of this Bachelor's thesis can be utilized to enhance the quality of life for residents of multi-apartment buildings by creating a convenient and functional mobile application. This application will facilitate more efficient HOA management, improve access to necessary information and services, and increase resident engagement in communal living and decision-making.

Keywords: housing cooperatives, convenience, quality of life, software application.

ЗМІСТ

ВСТУП	3
1. ОБҐРУНТУВАННЯ ВИБОРУ МЕТОДУ РОЗРОБКИ ТА ПОСТАНОВКА ЗАДАЧ ДОСЛІДЖЕННЯ	6
1.1 Аналіз стану мобільних застосунків для мешканців ОСББ.....	6
1.2 Порівняльний аналіз аналогів.....	8
1.3 Аналіз методів розв’язання поставленої задачі	14
1.4 Висновки	17
2. РОЗРОБКА СТРУКТУРИ ТА АЛГОРИТМІВ ПРОГРАМНОГО ПРОДУКТУ	18
2.1 Аналіз вхідних даних системи.....	18
2.2 Розробка структури інтерфейсу програмного застосунку	19
2.3 Розробка моделі системи	21
2.4 Розробка алгоритмів роботи системи	25
2.5 Висновки	27
3. РОЗРОБКА ПРОГРАМНИХ МОДУЛІВ ТА РЕАЛІЗАЦІЯ В МОБІЛЬНОМУ ЗАСТОСУНКУ	28
3.1 Варіантний аналіз і обґрунтування вибору засобів для реалізації програмного забезпечення	28
3.2 Розробка інтерфейсу та навігації.....	30
3.3 Розробка модулів застосунку	49
3.4 Висновки	58
4 ТЕСТУВАННЯ ПРОГРАМИ	60
4.1 Тестування системи	60
4.2 Розробка інструкції користувача	65
4.3 Висновки	67
ВИСНОВКИ.....	68
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	69
ДОДАТКИ.....	71
ДОДАТОК А – ТЕХНІЧНЕ ЗАВДАННЯ. Помилка! Закладку не визначено.	
ДОДАТОК Б – ПРОТОКОЛ ПЕРЕВІРКА НА ПЛАГІАТ Помилка! Закладку не визначено.	
ДОДАТОК В – ЛІСТИНГ ПРОГРАМИ.....	76
ДОДАТОК Г – ГРАФІЧНА ЧАСТИНА.....	93

ВСТУП

Обґрунтування вибору теми дослідження. У сучасній цифровій епохі мобільні застосунки стали невід'ємними інструментами для покращення ефективності, зручності та зв'язку в різних сферах повсякденного життя. Однією з таких сфер, де мобільні застосунки можуть мати значний вплив, є управління об'єднаннями співвласників багатоквартирних будинків (ОСББ), зокрема в багатоквартирних будинках. У даній бакалаврській роботі досліджується розробка та впровадження мобільного застосунка, призначеного для задоволення конкретних потреб мешканців у таких житлових об'єднаннях.

Застосунок дасть можливість вирішити проблеми, з якими стикаються мешканці та управлінські комітети ОСББ, шляхом надання простого у використанні та функціонального мобільного застосунка. З використанням можливостей сучасної мобільної технології застосунок спрямований на оптимізацію комунікації, полегшення ефективних управлінських процесів та покращення загального досвіду проживання для мешканців.

Через цю роботу буде отримано важливі уявлення про те, як технології можуть бути використані для оптимізації операцій ОСББ, покращення залучення мешканців та, нарешті, сприяння поліпшенню якості життя для всіх зацікавлених сторін.

Тому тема роботи «Розробка та реалізація мобільного застосунку для мешканців ОСББ» є актуальною.

Зв'язок роботи з науковими програмами, планами, темами. Робота виконувалася згідно плану виконання наукових досліджень на кафедрі програмного забезпечення.

Мета та завдання дослідження. Основною метою цієї роботи є спрощення управління житловими питаннями, надання доступу до необхідної інформації та послуг, а також поліпшення комунікації між мешканцями та управлінцями об'єднання та вирішення проблем, що виникають у мешканців

та управлінських комітетів ОСББ. Для досягнення цієї мети передбачено такі завдання:

- визначити вимоги до програмного застосунку;
- розробка модуля для авторизації користувача;
- реалізація модуля графічного інтерфейсу;
- інтеграція різних модулів у єдину мобільну систему, придатну для використання мешканцями ОСББ;
- проведення тестування розробленого програмного забезпечення.

Об'єкт дослідження – процес розробки програмного застосунку для мешканців ОСББ.

Предмет дослідження – методи та засоби розробки та реалізація мобільного застосунку для мешканців ОСББ.

Методи дослідження. Основні методи, що використовуються у цій роботі, включають концептуальні моделі для аналізу потреб мешканців, проектування інтерфейсів користувача, розробку необхідних функцій, методи тестування, стратегії виправлення помилок та інтеграцію з існуючими системами управління ОСББ. Крім того, використовуються методи реалізації програмних застосунків на мові програмування Dart у середовищі розробки Visual Studio Code та Flatter Framework.

Новизна отриманих результатів.

1. Розроблено концептуальну модель управління ОСББ, що дозволяє оцінити вплив факторів функціональності (інтеграція з системами відеоспостереження, можливість голосування щодо важливих питань житлового комплексу, доступ до різноманітних сервісів, включаючи послуги електрика, сантехніка, прибирання, доставки) з ефективністю управління ОСББ.

2. Розроблено програмну модель управління ОСББ, що дозвляє залучати мешканців ОСББ до процесів управління житловим комплексом.

Практична цінність отриманих результатів. Основна цінність отриманих результатів полягає в розробці програмного застосунку для

покращення управління житловими об'єднаннями. Отримані результати також можуть бути корисними для управління нерухомістю загалом. Це відкриває шлях для подальших досліджень та розвитку подібних інструментів для різних типів житлових об'єктів і спільнот.

Особистий внесок здобувача. Отримані результати дослідження та розробки представлені автором особисто, включаючи аналіз існуючих рішень. Автор самостійно провів комплексне дослідження потреб мешканців житлового комплексу, розробив архітектуру застосунка, створив і впровадив ключові програмні модулі, а також здійснив тестування та оптимізацію роботи застосунка. Крім того, автор провів аналіз та порівняння існуючих рішень на ринку, визначив їх переваги та недоліки, що дозволило створити конкурентоспроможний та ефективний продукт.

Апробація результатів здійснена на міжнародній науково-практичній інтернет-конференції студентів, аспірантів та молодих науковців «Глобалізація наукових знань: міжнародна співпраця та інтеграція галузей наук», (Біла Церква, 2024)

Публікації. Тези доповідей представлені в збірнику міжнародної науково-практичної інтернет-конференції студентів, аспірантів та молодих науковців «Глобалізація наукових знань: міжнародна співпраця та інтеграція галузей наук», (Біла Церква, 2024)

<https://archive.liga.science/index.php/conference-proceedings/issue/view/inter-07.06.2024>

1. ОБҐРУНТУВАННЯ ВИБОРУ МЕТОДУ РОЗРОБКИ ТА ПОСТАНОВКА ЗАДАЧ ДОСЛІДЖЕННЯ

1.1 Аналіз стану мобільних застосунків для мешканців ОСББ.

Сучасні тенденції в галузі управління житловими будинками дедалі більше схилиються до інтеграції новітніх технологій, зокрема мобільних застосунків, що дозволяють значно покращити якість обслуговування та рівень задоволення мешканців. ОСББ (об'єднання співвласників багатоквартирних будинків) також зацікавлені в ефективних рішеннях для оптимізації управлінських процесів. Для розробки ефективного мобільного застосунка, що задовольнить потреби мешканців ОСББ, необхідно провести ретельний аналіз поточного стану ринку таких застосунків, визначити їхні переваги та недоліки, а також врахувати очікування користувачів.

На сьогоднішній день ринок мобільних застосунків для ОСББ представлений кількома основними продуктами, кожен з яких пропонує свій набір функцій та можливостей. Аналіз існуючих рішень включає порівняння функціональності, оцінку користувацького досвіду та розгляд технічних аспектів реалізації.

Розпочнемо з порівняння функціональності основних застосунків. Більшість з них пропонують базові функції, такі як реєстрація та авторизація користувачів, створення запитів та звернень, отримання сповіщень про важливі події, оплата рахунків, перегляд документів, а також участь у голосуваннях та опитуваннях. Деякі застосунки надають додаткові можливості, такі як інтеграція з системами безпеки, управління парковками, бронювання спільних приміщень тощо. Втім, варто зазначити, що не всі застосунки однаково добре справляються з реалізацією цих функцій, і користувачі нерідко стикаються з технічними проблемами або незручним інтерфейсом.

Тому можна виділити декілька наявних недоліків існуючих застосунків:

- Обмежена функціональність: Більшість існуючих застосунків для мешканців ОСББ мають обмежений функціонал, що часто не враховує потреб користувачів.
- Низька зручність використання: Деякі застосунки можуть мати складний інтерфейс або неінтуїтивні функції, що ускладнює їх використання для користувачів.
- Недостатність інтеракції з управлінням ОСББ: Багато застосунків обмежені лише до сповіщень та замовлення послуг, не забезпечуючи повноцінної взаємодії з управлінням ОСББ та іншими мешканцями.

Виділивши декілька недоліків можна перейти до оцінки користувацького досвіду (UX), що є важливим аспектом аналізу. Зручність використання застосунка, інтуїтивно зрозумілий інтерфейс, швидка навігація та відсутність помилок під час роботи – все це впливає на задоволеність користувачів. Деякі застосунки мають складну та заплутану структуру меню, що ускладнює доступ до потрібних функцій. Інші ж навпаки, відзначаються простим і зрозумілим дизайном, що дозволяє легко орієнтуватися навіть недосвідченим користувачам.

Технічні аспекти реалізації застосунків також мають велике значення. Важливою є архітектура застосунка, яка повинна забезпечувати стабільну роботу та високу продуктивність. Застосунки, що використовують сучасні технології, такі як Dart та Flutter, мають перевагу у швидкості розробки та крос-платформенності. Це дозволяє розробляти застосунок одночасно для Android та iOS, що значно знижує витрати на підтримку та розвиток. Крім того, використання цих технологій забезпечує високу продуктивність та якість графіки, що важливо для створення привабливого та функціонального інтерфейсу користувача.

Для визначення потреб користувачів було проведено інтерв'ю та опитування мешканців ОСББ. Результати показали, що основні потреби включають зручність комунікації з адміністрацією, швидке отримання

інформації про новини та події, можливість створення запитів та звернень, а також функціонал для оплати рахунків та перегляду важливих документів. Користувачі також висловили бажання мати можливість брати участь у голосуваннях та опитуваннях, отримувати сповіщення про важливі події та мати доступ до контактної інформації служб підтримки.

Проведений аналіз дозволяє зробити висновок, що існуючі рішення на ринку мобільних застосунків для ОСББ мають як свої переваги, так і недоліки. З одного боку, більшість застосунків пропонують широкий спектр функцій, необхідних для ефективного управління будинками. З іншого боку, багато з них мають проблеми з користувацьким досвідом та технічною реалізацією. Тому розробка нового мобільного застосунка повинна враховувати ці аспекти для створення зручного, функціонального та надійного продукту.

Отже, аналіз стану галузі мобільних застосунків для ОСББ показує, що існує значний потенціал для покращення існуючих рішень. Основні напрямки вдосконалення включають підвищення зручності використання, розширення функціоналу, забезпечення стабільної роботи та безпеки даних. Враховуючи результати аналізу, можна сформулювати основні задачі для розробки нового мобільного застосунка, який задовольнить потреби мешканців ОСББ та забезпечить високий рівень їхнього комфорту та задоволення.

1.2 Порівняльний аналіз аналогів

Розробка мобільного застосунка для мешканців ОСББ потребує не лише розуміння потреб користувачів, але й детального аналізу існуючих аналогів на ринку. Порівняльний аналіз дозволяє визначити сильні та слабкі сторони конкурентів, виявити ключові функції, які необхідно реалізувати, а також знайти можливості для покращення та впровадження інновацій. У цьому розділі проведено порівняльний аналіз кількох популярних мобільних застосунків для ОСББ, щоб визначити оптимальні шляхи для розробки власного продукту.

На ринку мобільних застосунків для ОСББ представлено кілька основних продуктів, кожен з яких має свої особливості, функціональні можливості та технічні рішення. Для проведення порівняльного аналізу було обрано три популярних застосунки: "Дах", "OSSB.work" та "Моє ОСББ". Кожен з цих застосунків пропонує різні підходи до вирішення задач ОСББ, що дозволяє провести комплексний аналіз їхніх характеристик.

"Дах" є одним з найбільш популярних застосунків для ОСББ. Основні функції включають реєстрацію та авторизацію користувачів, створення запитів та звернень, отримання сповіщень про новини та події, оплату рахунків та участь у голосуваннях (див. рис. 1.1)[1].

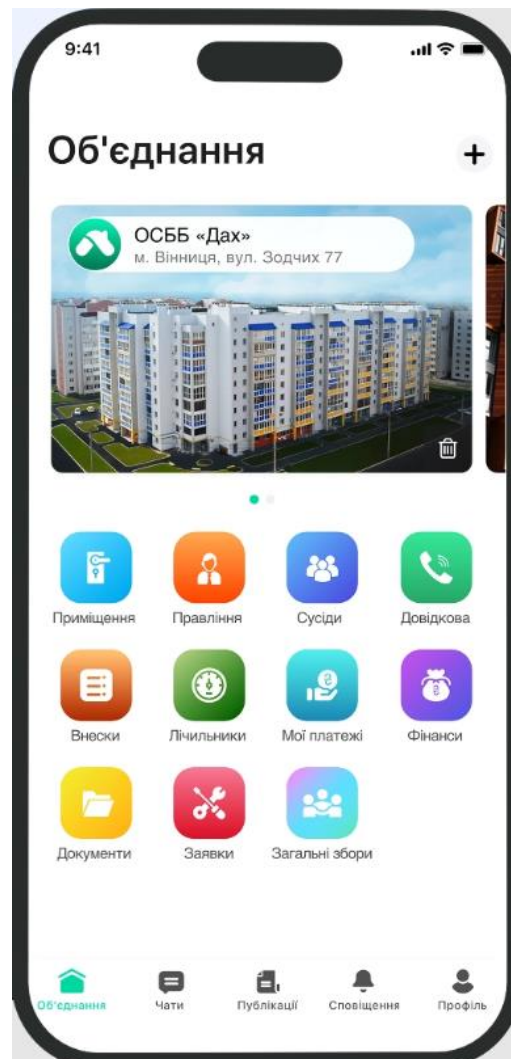


Рисунок 1.1 – Головна сторінка застосунку "Дах"

Переваги:

- Інтуїтивно зрозумілий інтерфейс користувача.
- Висока продуктивність та стабільність роботи.
- Наявність функції оплати рахунків безпосередньо через застосунок.

Недоліки:

- Обмежений набір функцій для спілкування між мешканцями.
- Відсутність інтеграції з системами безпеки та іншими зовнішніми сервісами.
- Недостатньо гнучкі налаштування профілю користувача.

"OSSB.work" пропонує більш широкий спектр функцій, зокрема управління запитами, сповіщення, оплата рахунків, перегляд документів, бронювання спільних приміщень та інтеграцію з системами безпеки (див. рис. 1.2)[2].

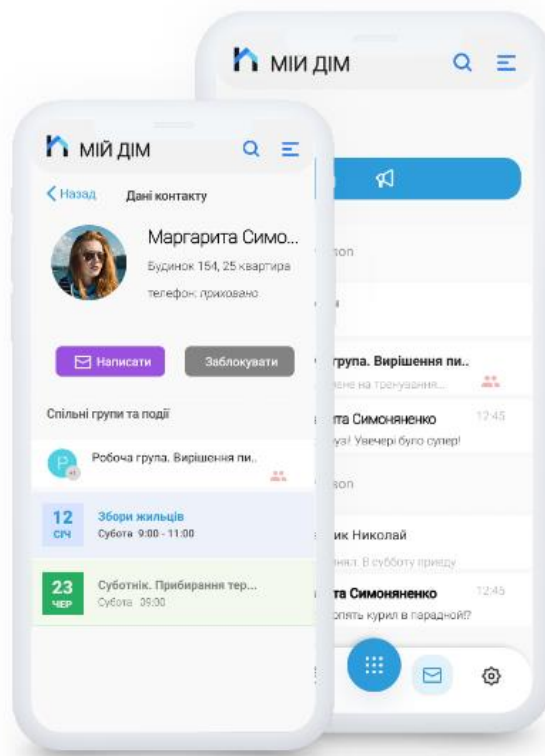


Рисунок 2.2 – Головна сторінка застосунку "OSSB.work"

Переваги:

- Широкий набір функцій, включаючи бронювання приміщень та інтеграцію з системами безпеки.
- Можливість перегляду та завантаження важливих документів.
- Динамічні сповіщення про події та новини.

Недоліки:

- Складний інтерфейс, що може бути важким для нових користувачів.
- Деякі функції потребують додаткових платних підписок.
- Проблеми зі стабільністю роботи на деяких пристроях.

"Моє ОСББ" зосереджується на забезпеченні ефективної комунікації між мешканцями та адміністрацією ОСББ. Застосунок включає функції чату, створення груп, проведення опитувань, а також стандартні функції оплати рахунків та отримання сповіщень (див. рис. 1.3)[3].

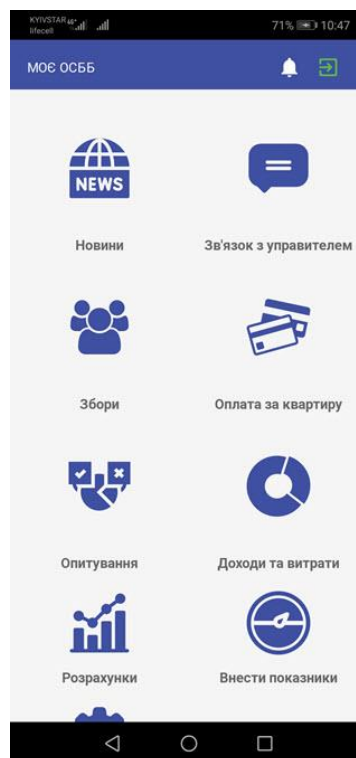


Рисунок 3.3 – Головна сторінка застосунку "Моє ОСББ"

Переваги:

- Потужні інструменти для комунікації, включаючи чати та групові дискусії.
- Можливість проведення опитувань та голосувань в реальному часі.
- Простий та зручний інтерфейс користувача.

Недоліки:

- Обмежений функціонал для управління запитами та зверненнями.
- Відсутність інтеграції з системами безпеки та зовнішніми сервісами.
- Немає можливості бронювання приміщень або перегляду документів.

Проведений аналіз показав, що кожен з розглянутих застосунків має свої унікальні переваги та недоліки. Водночас, жоден з них не забезпечує повний набір функцій, необхідний для комплексного управління ОСББ. Основні функціональні можливості, які мають бути реалізовані в розроблюваному застосунку, включають:

- Авторизація користувачів: Забезпечення безпечного та зручного доступу до застосунка.
- Створення запитів та звернень: Надання можливості мешканцям звертатися до адміністрації з різними питаннями та проблемами.
- Отримання сповіщень: Інформування мешканців про важливі події, новини та зміни.
- Оплата рахунків: Можливість швидкої та зручної оплати комунальних послуг та інших рахунків.
- Управління документами: Перегляд та завантаження важливих документів, таких як протоколи зборів, звіти тощо.
- Комунікація між мешканцями: Інструменти для спілкування, включаючи чати, групові дискусії та опитування.

- Інтеграція з зовнішніми сервісами: Підключення до систем безпеки, управління парковками та інші можливості для підвищення комфорту мешканців.
- Бронювання сервісів: Можливість бронювання сервісів, таких як прибирання квартири або доставку продуктів.

Порівняльний аналіз аналогів показав, що існуючі застосунки для ОСББ мають різні сильні та слабкі сторони. На основі цього аналізу можна зробити висновок а також створити таблицю порівняння аналогів 1.1, що розроблюваний застосунок повинен об'єднувати найкращі риси кожного з розглянутих продуктів, забезпечуючи при цьому інтуїтивно зрозумілий інтерфейс, високу продуктивність та стабільність роботи. Важливим аспектом є також гнучкість налаштувань та можливість інтеграції з різними зовнішніми сервісами, що дозволить створити дійсно корисний та зручний інструмент для мешканців ОСББ.

Таблиця 1.1 – Порівняння аналогів

Критерії оцінювання	Дах	OSSB.work	Моє ОСББ	WOW House
Авторизація користувачів	+	+	+	+
Створення запитів та звернень	+	+	-	+
Отримання сповіщень	+	+	+	+
Оплата рахунків	+	+	+	+

Продовження таблиці 1.1

Перегляд та завантаження документів	-	+	-	+
Інтеграція з системами безпеки	-	+	-	+
Інтуїтивно зрозумілий інтерфейс	+	+	-	+
Висока продуктивність та стабільність	+	-	-	+
Гнучкість налаштувань	+	+	-	+

Порівняльний аналіз показує, що розроблюваний мобільний застосунок WOW House перевершує існуючі аналоги за всіма критеріями. Завдяки повному набору необхідних функцій, інтуїтивно зрозумілому інтерфейсу, високій продуктивності та стабільності роботи, застосунок WOW House забезпечить оптимальне рішення для мешканців ОСББ, задовольняючи всі їхні потреби та очікування.

1.3 Аналіз методів розв’язання поставленої задачі

Розробка мобільного застосунка для мешканців ОСББ потребує ретельного аналізу доступних методів та технологій, які можуть бути застосовані для вирішення поставлених задач. Враховуючи специфіку проекту, необхідно обрати такі методи, які забезпечать високу продуктивність,

зручність використання, надійність та безпеку. У цьому розділі розглянуто основні методи та підходи до розробки мобільних застосунків, їх переваги та недоліки, а також обґрунтовано вибір оптимальних технологій для реалізації проекту.

Для розробки мобільного застосунка існує кілька підходів, які можна класифікувати за типом платформ: нативні застосунки, кросплатформенні застосунки та веб-застосунки.

- **Нативні застосунки:** Розробляються окремо для кожної платформи (iOS та Android) з використанням специфічних мов програмування (Swift для iOS, Kotlin або Java для Android). Основними перевагами є висока продуктивність та доступ до всіх можливостей апаратного забезпечення пристрою. Однак, цей підхід потребує значних ресурсів для розробки та підтримки двох окремих кодових баз[4].

- **Кросплатформенні застосунки:** Розробляються з використанням фреймворків, що дозволяють писати один код, який може виконуватись на обох платформах. До таких фреймворків належать Flutter, React Native, Xamarin тощо. Перевагами є скорочення часу та витрат на розробку, а також можливість легко підтримувати та оновлювати застосунок[5].

- **Веб-застосунки:** Це мобільні веб-сайти, що адаптовані для роботи на мобільних пристроях. Вони не потребують установки, але мають обмежений доступ до функцій пристрою та зазвичай поступаються у продуктивності нативним та кросплатформенним застосункам[6].

У даному проекті вирішено використовувати кросплатформенний підхід, а саме фреймворк Flutter. Це обумовлено наступними причинами:

- **Flutter:** Фреймворк від Google, що використовує мову програмування Dart[7]. Забезпечує високу продуктивність, оскільки використовує рендеринг на власному рушію. Flutter дозволяє створювати красиві, привабливі інтерфейси з високою швидкістю та гнучкістю. Завдяки єдиній кодовій базі, розробка для iOS та Android здійснюється одночасно, що суттєво знижує витрати на підтримку та розвиток застосунка.

Для забезпечення стабільності, масштабованості та легкості підтримки проекту важливо обрати правильну архітектуру застосунка.

- MVC (Model-View-Controller): Класична архітектурна патерн, що розділяє логіку застосунка на три компоненти: модель, вид і контролер[8]. Це дозволяє легко управляти та оновлювати окремі частини застосунка, проте може бути складним у масштабних проектах.
- MVVM (Model-View-ViewModel): Більш сучасна архітектура, яка добре підходить для кросплатформених застосунків[9]. Забезпечує чітке розділення відповідальностей та полегшує тестування. Flutter має вбудовану підтримку для цього патерну через бібліотеки та інструменти.
- BLoC (Business Logic Component): Особливо популярна архітектура у спільноті Flutter[10]. Вона забезпечує чітке розділення бізнес-логіки та представлення, що робить застосунок більш структурованим та легким у підтримці.

Безпека даних користувачів є ключовим аспектом при розробці застосунка для ОСББ. Основні методи забезпечення безпеки включають:

- Аутентифікація та авторизація: Використання надійних методів аутентифікації (наприклад, OAuth2, JWT) для забезпечення безпеки доступу до застосунка.
- Шифрування даних: Шифрування всіх даних, що зберігаються на пристрої та передаються через мережу, для захисту від несанкціонованого доступу.
- Безпечне зберігання даних: Використання безпечних сховищ для зберігання конфіденційної інформації, таких як Keychain для iOS та Keystore для Android.

На основі аналізу методів розв'язання задачі було прийнято рішення використовувати кросплатформенний фреймворк Flutter, який забезпечує високу продуктивність, гнучкість та швидкість розробки. Обрана архітектура BLoC дозволить чітко розділити бізнес-логіку та представлення, що полегшить підтримку та розвиток застосунка. Для забезпечення безпеки даних

користувачів буде використано надійні методи аутентифікації, шифрування та безпечного зберігання даних. Ці методи забезпечать створення функціонального, безпечного та зручного мобільного застосунка для мешканців ОСББ.

1.4 Висновки

У першому розділі було ретельно досліджено актуальний стан сфери мобільних застосунків для об'єднаних співвласників багатоквартирних будинків (ОСББ). Шляхом проведення порівняльного аналізу існуючих аналогів було зроблено висновок, що виготовлення мобільного застосунку для ОСББ є не лише справедливою, але й суттєвою потребою у сучасному світі. Вибір конкретних технологій для розробки цього застосунку — архітектури BLoC та кросплатформенного фреймворку Flutter — був обґрунтований ретельним аналізом їхніх можливостей та переваг. BLoC, з її можливістю ефективного контролю стану застосунку, і Flutter, що забезпечує швидку та однакову розробку для різних платформ, утворюють потужну комбінацію для створення динамічного та універсального мобільного застосунку для ОСББ. Ця комбінація технологій не лише відповідає сучасним вимогам ринку, але й має значний потенціал для подальшого розвитку та вдосконалення функціональності застосунку.

2. РОЗРОБКА СТРУКТУРИ ТА АЛГОРИТМІВ ПРОГРАМНОГО ПРОДУКТУ

2.1 Аналіз вхідних даних системи

Розробка мобільного застосунка для мешканців ОСББ потребує уважного аналізу доступних даних, які будуть використовуватися в системі. В даному розділі будуть розглянуті основні типи даних, характеристики та джерела їх збору, що впливає на ефективність та функціональність застосунку.

Для мобільного застосунка для мешканців ОСББ важливо врахувати різноманітні типи даних, включаючи особисту інформацію користувачів, дані про рахунки та платежі, повідомлення від управління ОСББ та інші параметри, які стосуються їхнього проживання. Необхідно розглянути формат, обсяг та частоту оновлення даних. Наприклад, інформація про рахунки може містити числа з плаваючою комою, а повідомлення від управління ОСББ можуть бути у формі текстових або мультимедійних файлів.

Дані можуть надходити з різних джерел, включаючи внутрішні бази даних ОСББ, зовнішні API платіжних систем або сервіси доставки повідомлень. Важливо встановити взаємозв'язки між різними типами даних, щоб забезпечити правильну роботу застосунку. Наприклад, дані про платежі можуть бути пов'язані з конкретними рахунками користувачів. Забезпечення безпеки даних користувачів є надзвичайно важливим аспектом розробки застосунку. Потрібно використовувати надійні методи аутентифікації, шифрування та безпечного зберігання даних для захисту інформації від несанкціонованого доступу.

Аналіз вхідних даних системи для мобільного застосунка для мешканців ОСББ дозволяє розуміти, які дані будуть потрібні для правильної роботи застосунку, як їхні характеристики впливають на функціональність та ефективність системи, а також як забезпечити безпеку даних користувачів.

2.2 Розробка структури інтерфейсу програмного застосунку

Розробка структури інтерфейсу застосунку є одним із найважливіших етапів у створенні користувацького досвіду. На цьому етапі важливо визначити потреби користувачів, логічно та лаконічно розмістити елементи інтерфейсу, забезпечуючи зручну та інтуїтивно зрозумілу навігацію. Основною метою цього етапу є організація інформації таким чином, щоб користувачі могли легко та швидко знайти необхідні дані та функції.

Проектування структури інтерфейсу застосунку потребує дотримання стандартів юзабіліті [11]. Основні правила включають:

- **Цільова аудиторія:** Інтерфейс завжди створюється для конкретної групи людей з урахуванням їхніх особливостей, знань, навичок, очікувань та потреб. Системи, орієнтовані на "середнього користувача", рідко задовольняють потреби більшості.
- **Контекст використання:** Рішення, які підходять в одному контексті, можуть бути неприйнятними в іншому. Наприклад, голосовий інтерфейс може бути ефективним для роботи в ізольованому приміщенні, але незручним для спільної роботи кількох людей в одному просторі.
- **Досягнення цілей:** Якщо інтерфейс не допомагає користувачу досягати своїх цілей, він вважається невдалим. Для успішного проектування потрібно дослідити цілі користувачів та врахувати їх у дизайні.

Хоча немає універсальних правил для розробки структури інтерфейсу, існують кілька загальноприйнятих принципів:

- **Стандартні розміщення:** Наприклад, логотип зазвичай розміщується у верхньому лівому куті, а поле пошуку – у верхньому правому.
- **Виділення елементів:** Елементи для взаємодії з користувачем, такі як кнопки, повинні бути помітними, щоб одразу було зрозуміло, що робити на сторінці.

- **Пріоритетність інформації:** Найважливіші елементи, які представляють ціль та ідею застосунку, повинні бути на видному місці, щоб користувач одразу знаходив те, що шукає.

Інтерфейс застосунку для мешканців ОСББ розроблений з урахуванням зазначених принципів і має плитковий дизайн (див. рис. 2.1)



Рисунок 2.1 – Структура інтерфейсу головної сторінки

Такий підхід забезпечує чітко структуровану робочу поверхню, розділяючи її на окремі зони взаємодії. Плитки, засновані на базових геометричних формах, динамічно адаптуються до вмісту. Зрозумілий дизайн

та послідовне розташування елементів керування створюють характерний лінійний вигляд інтерфейсу користувача.

Для створення інтерфейсу програмного застосунку було обрано досить темні кольори, оскільки вони естетично привабливі та легко сприймаються користувачем [12]. Такий вибір допомагає створити комфортну та спокійну атмосферу під час використання застосунку. Додатково, застосування оранжевого та сірого кольорів забезпечує контрастність між елементами інтерфейсу, що покращує його вигляд та зрозумілість. Приклад готового інтерфейсу можна переглянути на рисунку 2.2.

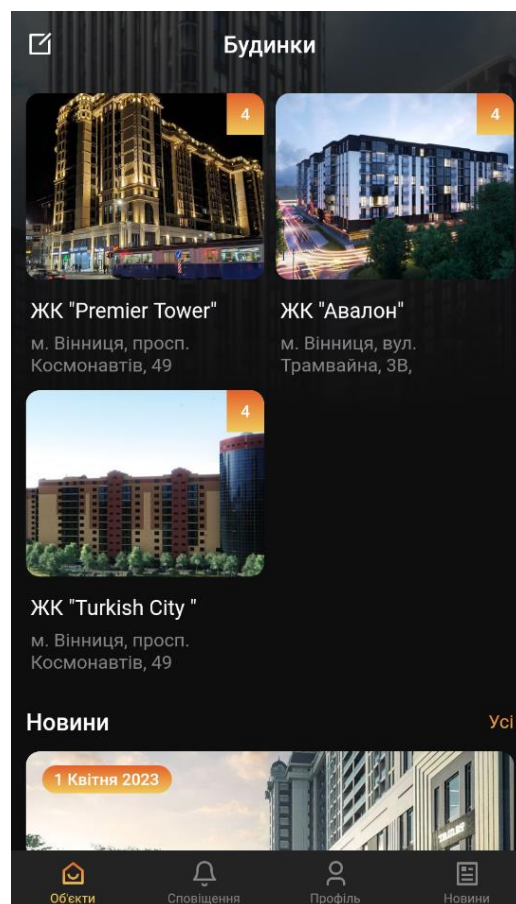


Рисунок 2.2 – Приклад головного екрану застосунку

2.3 Розробка моделі системи

Для кращого осмислення роботи програмних модулів мобільної системи для мешканців ОСББ було вирішено використовувати модель системи на

прикладі UML (Unified Modeling Language) діаграми діяльності [13]. Ця діаграма дозволить графічно зобразити послідовність операцій та взаємодію різних модулів системи під час виконання конкретної діяльності.

На UML діаграмі діяльності будуть зображені різні етапи та кроки, які виконує система для реалізації основних функцій застосунку. Це охоплює:

- Авторизацію користувача.
- Перехід до головного меню.
- Вибір квартири.
- Виконання операцій з доступними функціями (наприклад, перегляд квитанцій, участь в обговореннях та голосуваннях, звернення до диспетчерської служби тощо).

Кожна дія та перехід між ними буде чітко відображений на діаграмі діяльності, що допоможе зрозуміти логіку та послідовність взаємодій між модулями системи. Діаграма також може включати умови та розгалуження, які впливають на подальші дії системи в залежності від певних умов або подій (див. рис. 2.3).

Запропоновано модель, яка оцінює вплив таких факторів функціональності:

- інтеграція з системами відеоспостереження,
- можливість голосування щодо важливих питань житлового комплексу,
- доступ до різноманітних сервісів, включаючи послуги електрика, сантехніка, прибирання, доставки,
- передача показників лічильника,
- служба підтримки,
- перегляд та оплата квитанцій онлайн,
- подання заявок на ремонтні роботи та інші послуги,
- отримання сповіщень про важливі події та новини ОСББ,
- доступ до документів та протоколів зборів ОСББ.
- можливість зворотного зв'язку з адміністрацією ОСББ.

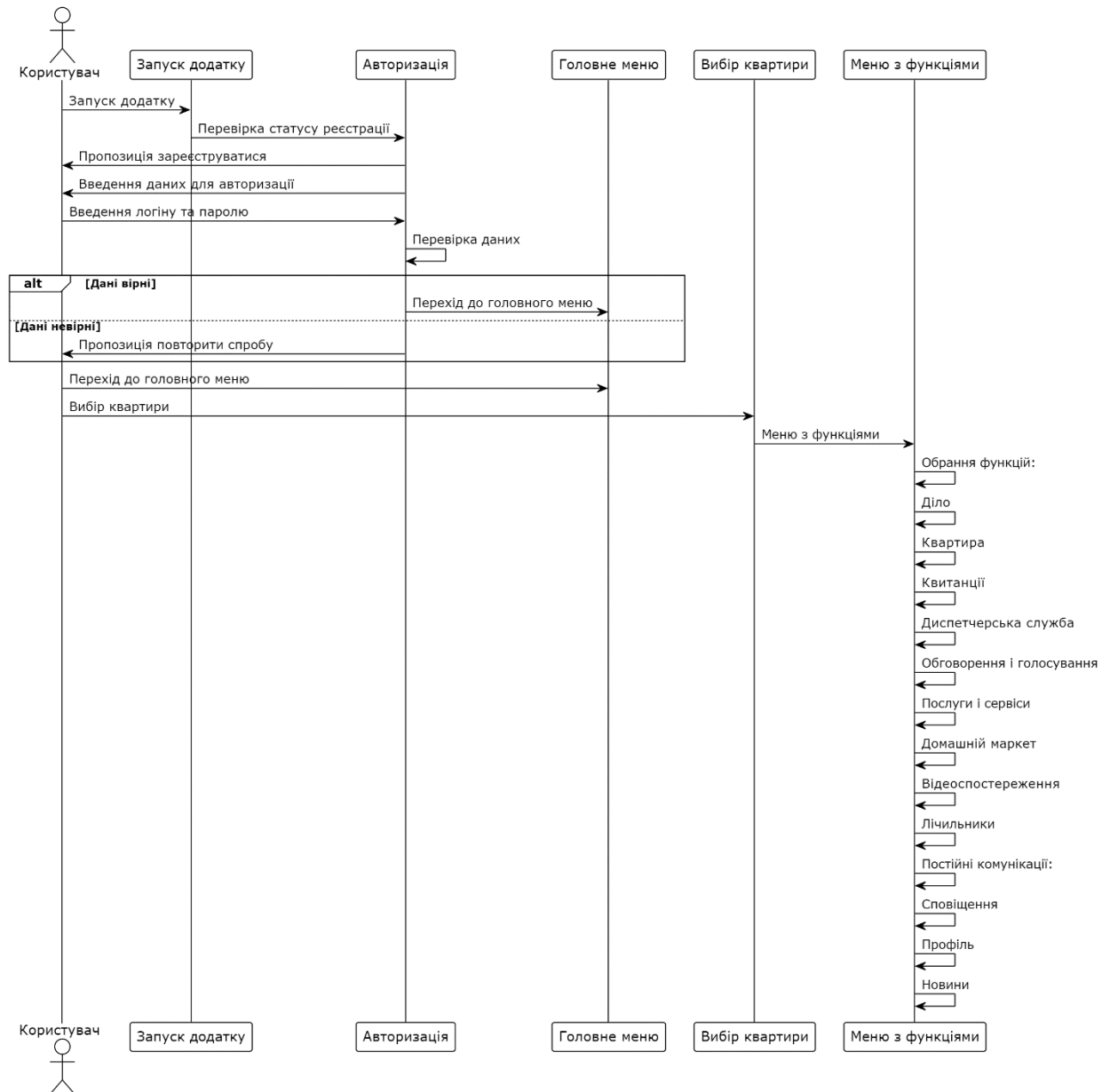


Рисунок 2.3 – Діаграма діяльності мобільної системи

Опис дій зображених на UML діаграмі:

1. Авторизація користувача

Перевірка статусу реєстрації: При запуску застосунку перевіряється, чи зареєстрований користувач.

- Якщо користувач не зареєстрований, йому пропонується зареєструватися.
- Якщо користувач зареєстрований, він переходить до авторизації.

- Введення даних для авторизації: Користувач вводить логін та пароль.
- Перевірка даних: Система перевіряє введені дані.
- Якщо дані введено вірно, користувач потрапляє в головне меню.
- Якщо дані введено невірно, користувачу пропонується повторити спробу.

2. Перехід до головного меню

Після успішної авторизації користувач бачить головне меню, де може вибрати потрібну функцію.

3. Вибір квартири

Користувач обирає свою квартиру зі списку.

Після вибору квартири відкривається меню з доступними функціями.

4. Меню з функціями

Користувач може вибрати одну з наступних функцій:

- Діло: Перегляд особистих справ та повідомлень.
 - Квартира: Інформація про квартиру, зміни в ній.
 - Квитанції: Перегляд та оплата квитанцій.
 - Диспетчерська служба: Виклик та взаємодія з диспетчерською службою.
 - Обговорення і голосування: Участь в обговореннях та голосуваннях мешканців.
 - Послуги і сервіси: Замовлення додаткових послуг та сервісів.
 - Домашній маркет: Перегляд та замовлення товарів і послуг від місцевих продавців.
 - Відеоспостереження: Перегляд камер відеоспостереження.
 - Лічильники: Перегляд та введення показників лічильників.
- ## 5. Постійні комунікації
- У нижній частині екрана присутні три основні кнопки для швидкого доступу:

- Сповіщення: Перегляд сповіщень про події в застосунку.
- Профіль: Перегляд та редагування особистого профілю користувача.
- Новини: Останні новини та оголошення для мешканців.

Розробка UML діаграми діяльності дозволить краще уявити процес роботи системи та виявити можливі питання або удосконалення. Вона слугуватиме важливим інструментом для аналізу, спілкування та вдосконалення системи перед її фактичним впровадженням та розробкою програмних модулів.

2.4 Розробка алгоритмів роботи системи

Для створення ефективного та зручного мобільного застосунка для мешканців ОСББ, необхідно розробити чіткі алгоритми [14], які визначатимуть послідовність дій користувача та відповідні реакції системи. Розглянемо основні екрани та функції застосунка, а також алгоритми їх роботи.

Під час авторизації користувач відкриває застосунок та потрапляє на екран авторизації, де він вводить свій логін та пароль. Система перевіряє ці дані: якщо вони вірні, користувач автоматично переходить до головного меню. У випадку невірних даних, система відображає повідомлення про помилку та пропонує повторити спробу. Важливою частиною цього процесу є забезпечення безпеки та захисту особистих даних користувачів (див. рис. 2.4).

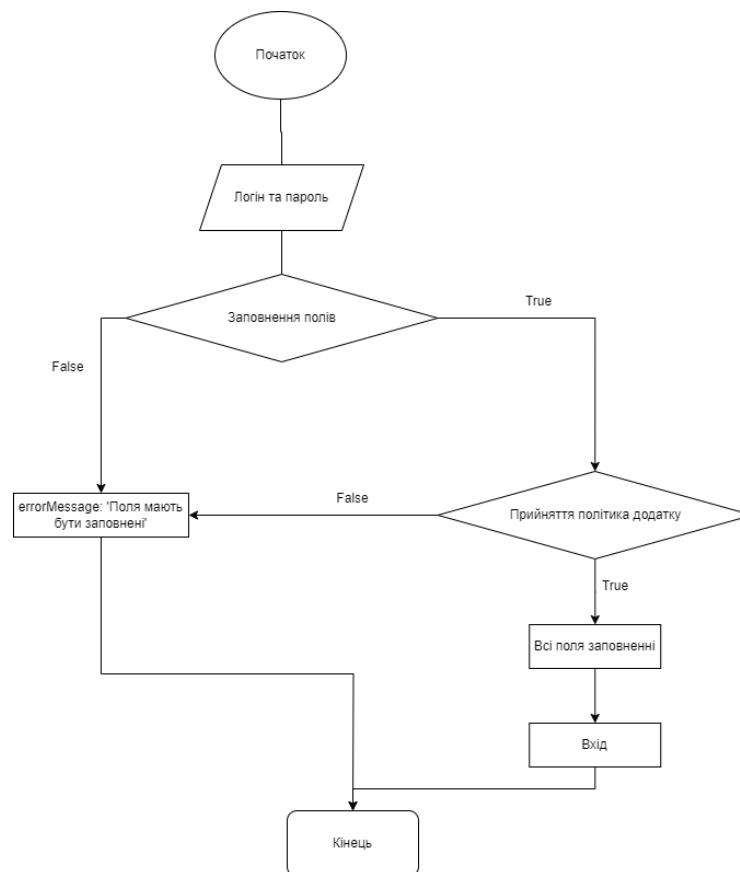


Рисунок 2.4 – Блок-схема алгоритму авторизації користувача

Після успішної авторизації користувача, система перенаправляє його до головного меню, де він може обрати бажану функцію застосунка, наприклад "Квартира". Після вибору функції, відбувається відкриття відповідного екрану з підменю або необхідною інформацією, що дозволяє користувачеві легко навігуватися та використовувати функціонал застосунку (див. рис. 2.5). Це забезпечує зручний і ефективний досвід використання застосунку для користувача, оскільки він може швидко отримати доступ до необхідної інформації та функцій.

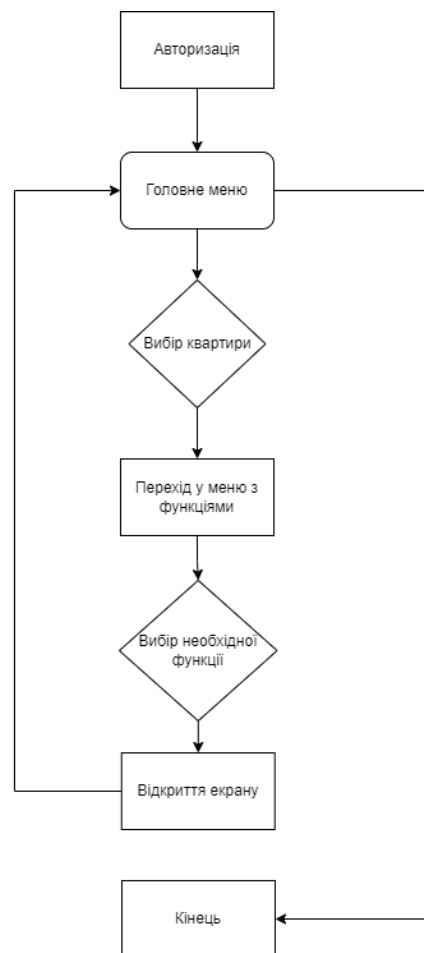


Рисунок 2.5 – Блок-схема алгоритму роботи головного екрану

2.5 Висновки

У другому розділі проведено аналіз вхідних та вихідних даних програмного продукту з метою зрозуміти, які саме дані будуть оброблятися та генеруватися застосунком. Досліджено потреби користувачів та вимоги до функціоналу, які дані є найбільш важливими для включення в продукт.

Основний алгоритм роботи мобільної платформи було розроблено з урахуванням потреб користувачів та особливостей функціоналу програмного продукту. Цей алгоритм визначає послідовність дій, які будуть виконуватися програмою в різних ситуаціях.

Також була розроблена модель роботи програмного продукту за допомогою UML діаграм та блок схем для кращого розуміння роботи застосунку.

3. РОЗРОБКА ПРОГРАМНИХ МОДУЛІВ ТА РЕАЛІЗАЦІЯ В МОБІЛЬНОМУ ЗАСТОСУНКУ

3.1 Варіантний аналіз і обґрунтування вибору засобів для реалізації програмного забезпечення

Вибір мови програмування та технологій для розробки мобільного застосунка для мешканців ОСББ залежить від ряду факторів, включаючи функціональні вимоги, продуктивність, можливість кросплатформенного розвитку, легкість підтримки, наявність ресурсів та знання розробників. Для реалізації даного проекту було обрано мову програмування Dart та фреймворк Flutter [15]. Далі наводиться обґрунтування цього вибору.

1. Кросплатформенність

Однією з основних переваг використання Dart та Flutter є можливість розробки кросплатформенних застосунків. Це означає, що один і той же код можна використовувати для створення застосунків, які працюють на iOS, Android та інших платформах. Це значно скорочує час розробки та знижує витрати на підтримку окремих кодових баз для різних платформ.

2. Продуктивність

Flutter використовує власний механізм рендерингу, що дозволяє створювати високопродуктивні застосунки з плавною анімацією та швидким відгуком інтерфейсу користувача. Dart компілюється в машинний код, що забезпечує високу продуктивність застосунків на рівні нативних рішень.

3. Розширені можливості для користувацького інтерфейсу

Flutter дозволяє створювати багаті та складні користувацькі інтерфейси завдяки вбудованим віджетам та широким можливостям кастомізації. Це особливо важливо для створення зручного та інтуїтивно зрозумілого інтерфейсу для мешканців ОСББ.

4. Активна спільнота та підтримка

Flutter має активну та зростаючу спільноту розробників. Існує багато

ресурсів, документації та прикладів, які допомагають швидко освоїти технологію та вирішувати виникаючі проблеми. Крім того, Google активно підтримує та розвиває Flutter, що гарантує стабільність та подальший розвиток фреймворка.

5. Простота навчання

Dart є простою у вивченні мовою програмування з чітким та зрозумілим синтаксисом. Розробники, які мають досвід роботи з іншими мовами програмування, такими як JavaScript, Java чи C#, можуть швидко освоїти Dart. Це дозволяє залучати до проекту нових розробників без значних витрат часу на навчання.

6. Інструменти розробки

Flutter інтегрується з популярними середовищами розробки, такими як Visual Studio Code та Android Studio, що забезпечує зручність розробки, налагодження та тестування застосунків. Також Flutter підтримує гаряче перезавантаження (hot reload), що дозволяє розробникам миттєво бачити зміни в коді без необхідності перезапускати застосунок.

Вибір Dart та Flutter для розробки мобільного застосунка для мешканців ОСББ є обґрунтованим рішенням, враховуючи наведені переваги. Ці технології дозволяють створити високоякісний, продуктивний та зручний у використанні застосунок з мінімальними витратами на розробку та підтримку.

У підсумку вибір засобів для розробки програмного забезпечення завжди є компромісом між різними вимогами та можливостями. Dart та Flutter пропонують кросплатформенність, високу продуктивність, розширені можливості для створення користувацького інтерфейсу, активну підтримку спільноти, простоту навчання та потужні інструменти для розробки. У зв'язку з цим, вибір Dart та Flutter для розробки мобільного застосунка для мешканців ОСББ є оптимальним варіантом, що дозволить створити якісний продукт з мінімальними витратами ресурсів та часу.

3.2 Розробка інтерфейсу та навігації

Розробка інтерфейсу програмного застосунку є критично важливим етапом у створенні мобільного застосунку для мешканців ОСББ [16]. Інтерфейс визначає, як користувачі взаємодіють з застосунком, тому він повинен бути зручним, естетично привабливим і функціональним.

Процес розробки інтерфейсу починається зі збору вимог та розуміння потреб користувачів. Це включає проведення досліджень, спілкування з майбутніми користувачами і збір відгуків. Розробники мають глибоко зрозуміти, які завдання користувачі планують виконувати, як вони хочуть взаємодіяти з програмою та як можна забезпечити їм найкращий досвід використання.

Після збору вимог розробники переходять до проектування інтерфейсу. Це включає створення прототипів, дизайну та організації елементів інтерфейсу. Прототипи дозволяють перевірити функціональність та взаємодію елементів перед реалізацією. Дизайн інтерфейсу повинен бути зручним, естетичним та відповідати стилю продукту. Розробники також враховують принципи взаємодії, такі як простота, зрозумілість та послідовність.

Після проектування настає етап реалізації інтерфейсу, де розробники перетворюють дизайн у функціональний інтерфейс. Це включає програмування фронтенду, розміщення елементів інтерфейсу на сторінці та забезпечення взаємодії зі зв'язками з базою даних або сервером. Важливо забезпечити, щоб інтерфейс був гнучким, адаптивним та сумісним з різними пристроями і браузерами.

Останній етап – тестування та оптимізація інтерфейсу. Розробники проводять різні тести, щоб переконатися, що інтерфейс працює належним чином, не має помилок і задовольняє вимоги користувачів. Це може включати тестування функціональності, тестування взаємодії, тестування швидкодії та тестування на різних пристроях. Після виявлення помилок або проблем розробники вносять виправлення та оптимізують інтерфейс для досягнення

найкращих результатів.

Правильно розроблений інтерфейс може значно покращити взаємодію з застосунком, сприяти легкості використання та підвищити задоволення користувачів. Початковий екран завантаження застосунку, який був створений на початковому етапі, можна побачити на рисунку 3.1. Цей екран задає тон для подальшої взаємодії користувача з застосунком, забезпечуючи перше враження та налаштовуючи користувача на позитивний досвід [17].



Рисунок 3.1 – Екран завантаження

Для забезпечення безпеки та ідентифікації користувачів був розроблений екран аутентифікації, який є першим кроком у вході до застосунку. На цьому екрані користувачам пропонується ввести свої особисті дані, такі як логін та пароль. Крім того, користувачам необхідно погодитися з Політикою конфіденційності та Умовами користування перед успішним входом до застосунку. Завдяки цьому процесу аутентифікації забезпечується

збереження конфіденційності та безпеки особистої інформації користувачів. Екран аутентифікації відображений на рисунку 3.2 і є першим кроком для доступу до функціоналу застосунку, забезпечуючи захищений та безпечний доступ для кожного користувача.

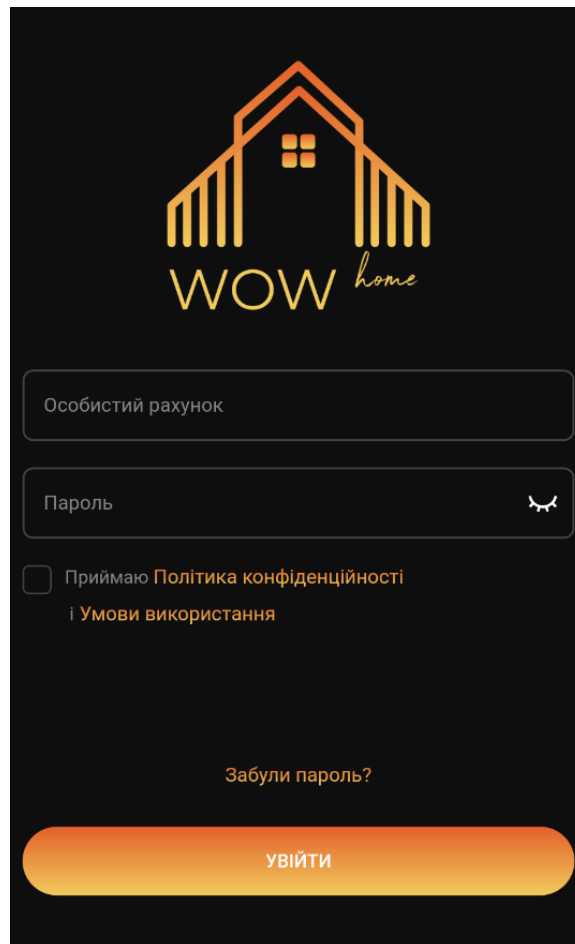


Рисунок 3.2 – Екран аутентифікації

Після успішної авторизації користувач автоматично перенаправляється на головний екран застосунку, який відображений на рисунку 3.3. Головний екран є центральним вузлом навігації і містить основні елементи керування та інформаційні блоки, необхідні для зручного користування застосунком.

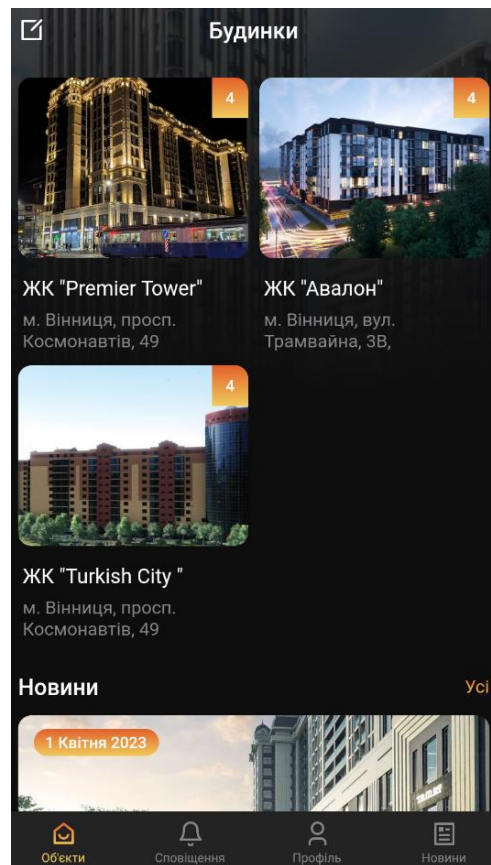


Рисунок 3.3 – Реалізація навігації

На головному екрані користувача вітає список доступних квартир та загальний вигляд житлового комплексу. У нижній частині екрану розташовані вкладки, що дозволяють швидко переходити до різних розділів застосунку, як відображений на рисунку 3.4. Ці вкладки включають: "Об'єкти" для перегляду інформації про квартири та будівлю, "Сповіщення" для важливих повідомлень, "Профіль" для доступу до особистої інформації та налаштувань облікового запису, а також "Новини" для отримання актуальної інформації про житловий комплекс та події, які стосуються мешканців. Така структура забезпечує зручний та ефективний доступ до всієї необхідної інформації та функціоналу застосунку.

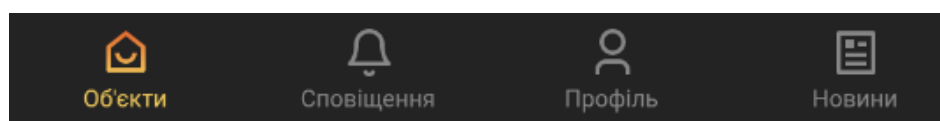


Рисунок 3.4 – Вкладки швидкого доступу

Після вибору квартири користувач буде перенаправлений до меню, де зможе отримати доступ до основних функцій застосунку та запропонованих послуг. Серед них:

- Діло: Розділ, що надає можливість користувачеві здійснювати різноманітні дії та переглядати інформацію щодо житлового комплексу.
- Квартира: Секція, де можна переглянути детальну інформацію про власну квартиру, таку як площа, кількість кімнат, план розташування тощо.
- Квитанції: Меню для перегляду та оплати рахунків і комунальних послуг онлайн.
- Диспетчерська служба: Розділ для звернення до служб технічної підтримки та вирішення питань, пов'язаних з побутом у житловому комплексі або роботою мобільного застосунку.
- Обговорення і голосування: Секція для обговорення та прийняття рішень щодо важливих питань спільноти мешканців.
- Послуги і сервіси: Розділ для замовлення різноманітних послуг, таких як ремонт, прибирання, обслуговування тощо.
- Домашній маркет: Меню для здійснення покупок товарів і послуг у межах житлового комплексу.
- Відеоспостереження: Секція для перегляду живих або записаних відеозаписів з камер, встановлених у зонах загального користування.
- Лічильники: Розділ, що надає інформацію про споживання комунальних послуг та витрати ресурсів, а також дозволяє самостійно додавати показники.

Це меню, представлене на рисунку 3.5, створено для зручного та ефективного керування житловим простором та надання доступу до всіх необхідних сервісів і інформації.

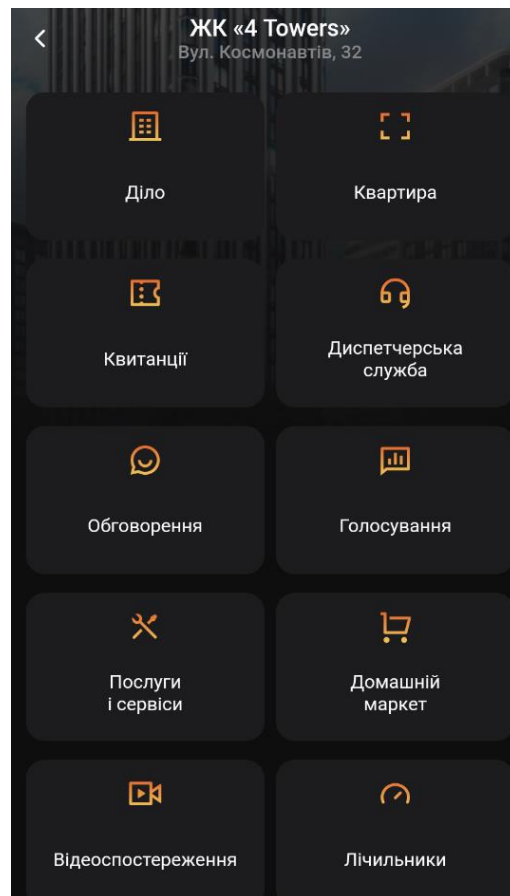


Рисунок 3.5 – Меню головних функцій застосунку

Перейшовши до вкладки "Діло", користувач отримує можливість виконувати різноманітні дії та переглядати інформацію, пов'язану з житловим комплексом. Це включає перегляд секцій і поверхів будівлі, інформацію про прибудинкові території, актуальні тарифи на послуги та оголошення про події, що відбуваються в комплексі. Вкладка "Діло" надає зручний доступ до всієї необхідної інформації, що дозволяє мешканцям бути в курсі всіх аспектів, що стосуються їхнього житлового комплексу. Цей інтерфейс зображений на рисунку 3.6.

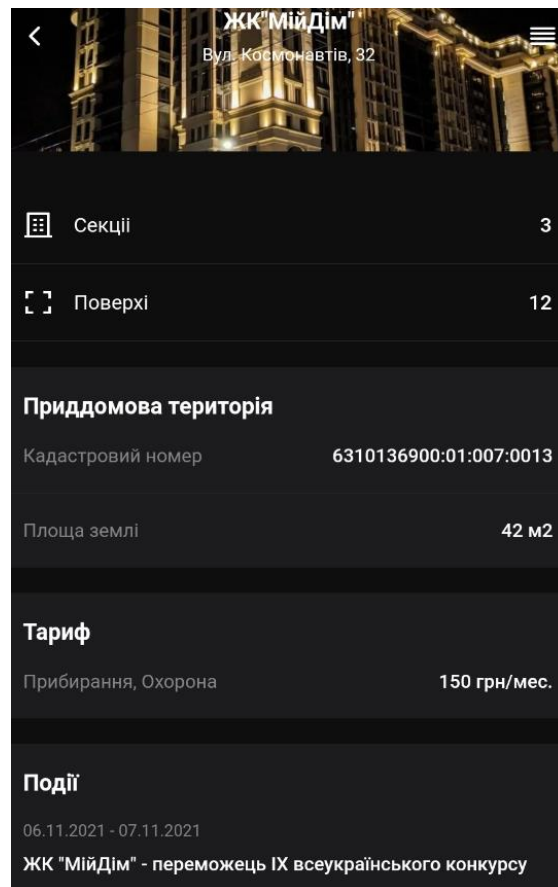


Рисунок 3.6 – Зображення вкладки "Діло"

Перейшовши до вкладки "Квартира", користувач отримує доступ до детальної інформації про свою квартиру. Ця вкладка надає можливість переглядати різноманітні параметри, що стосуються конкретної квартири в багатоквартирному будинку. Зокрема, користувач може ознайомитися з такими даними, як секція та поверх, де розташована квартира, особистий рахунок, номер квартири, площа, тарифи на комунальні послуги, а також інформація про власника. Інтерфейс цієї вкладки зображений на рисунку 3.7.

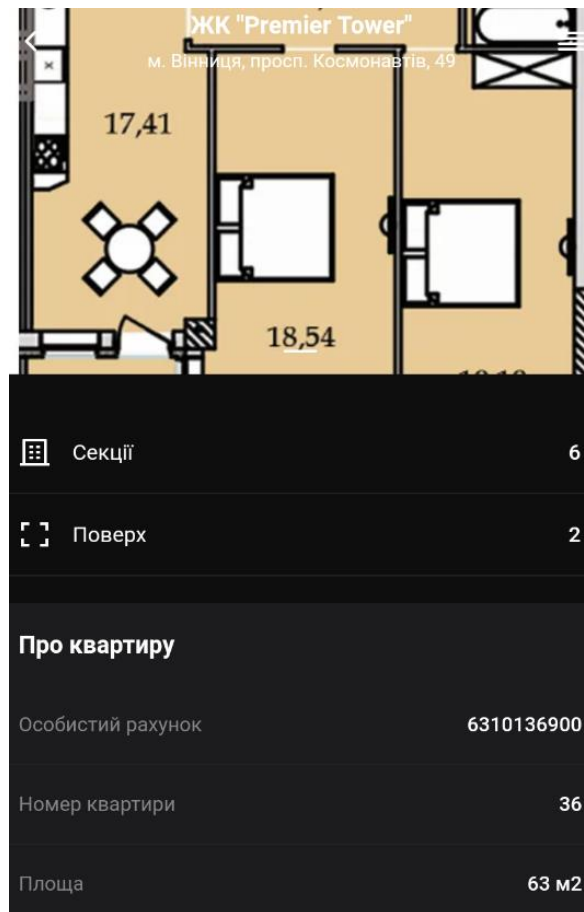


Рисунок 3.7 – Інтерфейс інформації про квартиру

Обравши вкладку "Квитанції", користувач переходить на екран, де відображаються всі квитанції, надіслані йому за комунальні послуги. На цьому екрані користувач може переглядати деталі кожної квитанції, включаючи суми, дати та опис послуг це все зображено на рисунках 3.8 – 3.9. Також користувач може здійснити оплату комунальних послуг безпосередньо через застосунок, використовуючи підключену платіжну картку або інші доступні методи оплати. Це забезпечує зручність і швидкість у керуванні фінансовими операціями, пов'язаними з комунальними платежами.

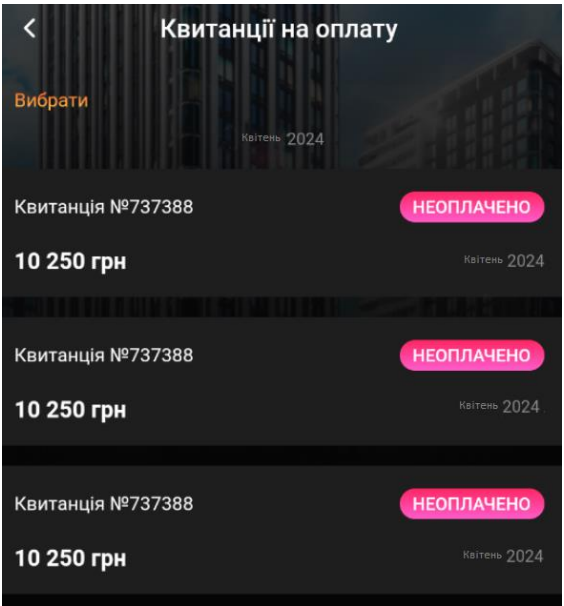


Рисунок 3.8 – Інтерфейс з квитанціями

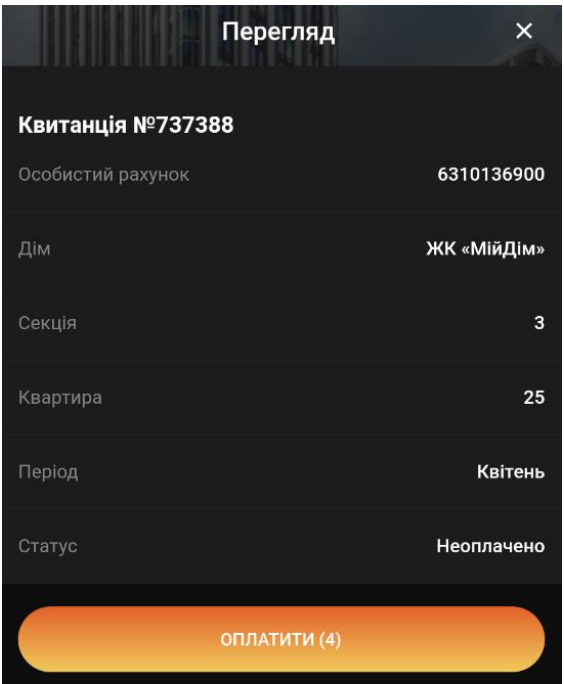


Рисунок 3.9 – Екран з відкритою квитанцією

Відкривши меню "Диспетчерська служба", користувач отримує доступ до функцій, призначених для взаємодії з технічною підтримкою житлового комплексу (див. рис. 3.10). На цьому екрані користувач може подати заявки на ремонт або обслуговування, повідомити про неполадки, а також стежити за статусом своїх заявок. Це меню створене для забезпечення ефективної та

зручної комунікації між мешканцями та адміністрацією житлового комплексу, сприяючи швидкому вирішенню технічних проблем.

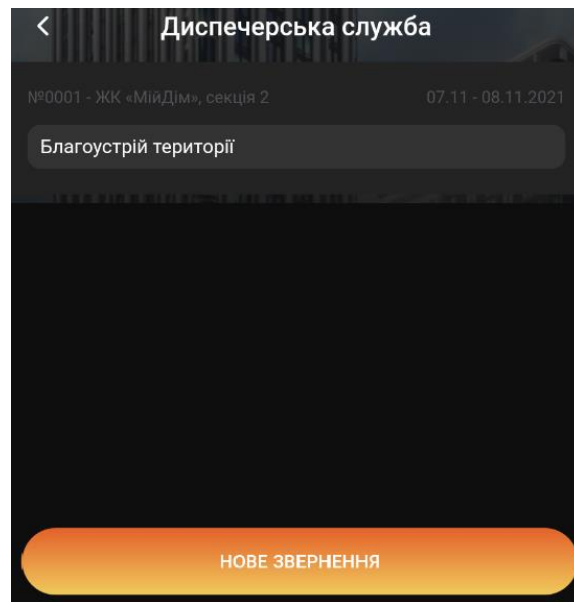


Рисунок 3.10 – Меню "Диспетчерська служба"

Також на рисунку 3.11 зображено приклад звернення служби підтримки до користувача.

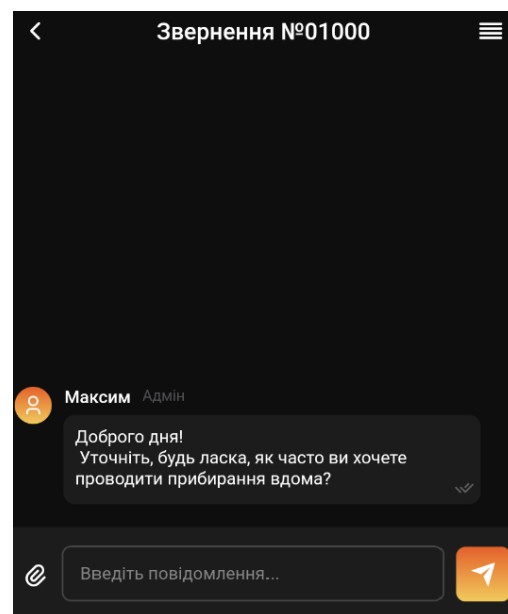


Рисунок 3.11 – Звернення служби підтримки до користувача

Надалі було створено екран "Обговорення", призначений для спілкування мешканців житлового комплексу (див. рис. 3.12). На цьому екрані користувачі можуть брати участь у дискусіях, обмінюватися думками та пропозиціями щодо покращення умов проживання. Екран "Обговорення" включає функції створення нових тем, коментування, голосування за найактуальніші питання, а також можливість відстежувати активні обговорення. Це сприяє розвитку спільноти, взаємодії мешканців та прийняттю колективних рішень з важливих питань.

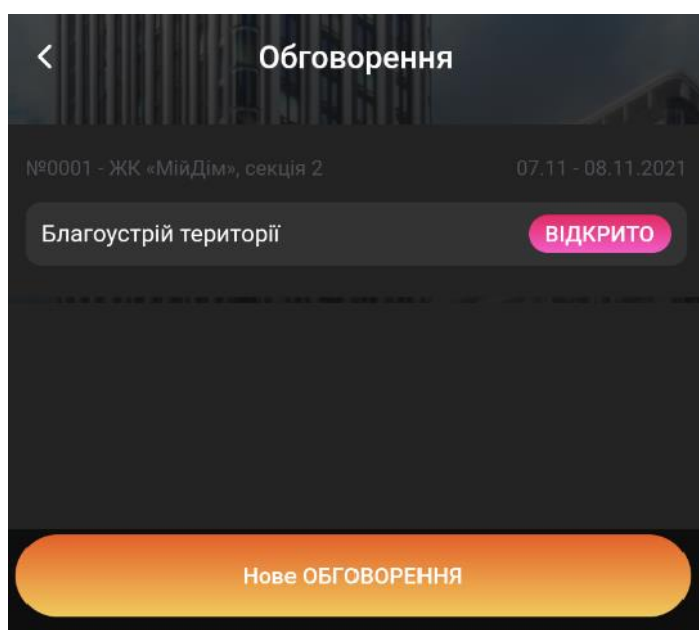


Рисунок 3.12 – Екран "Обговорення"

Було створено екран "Голосування", який дозволяє мешканцям брати участь у прийнятті рішень стосовно важливих питань житлового комплексу. Цей екран надає можливість організовувати та проводити голосування, відстежувати їхні результати та аналізувати думки спільноти. Користувачі можуть пропонувати нові питання для голосування, переглядати активні та минулі голосування, а також залишати свої голоси. Функціонал "Голосування" сприяє прозорості та залученню мешканців до управління спільними ресурсами і ухвалення рішень щодо покращення житлових умов. Це все зображено на рисунках 3.13 – 3.14.

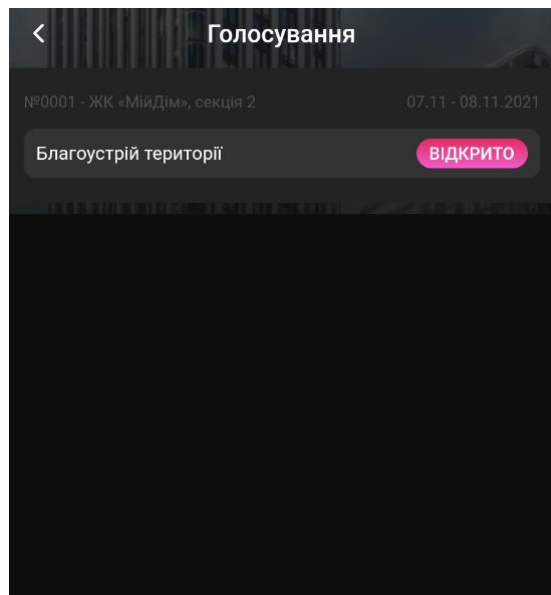


Рисунок 3.13 – Екран "Голосування"

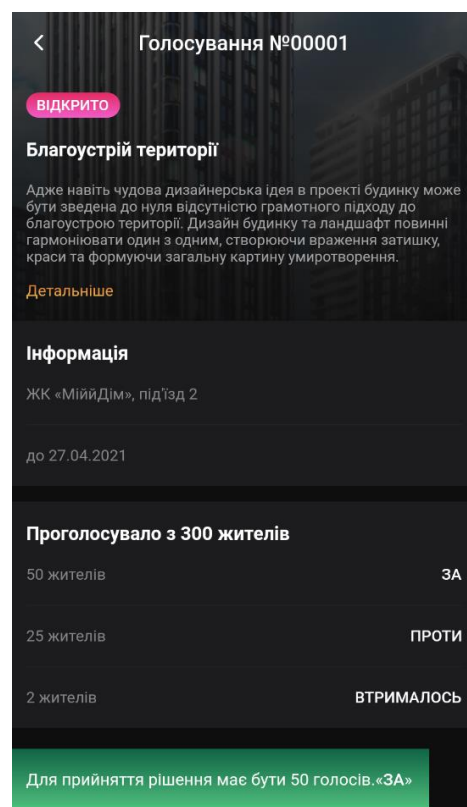


Рисунок 3.14 – Приклад відкритого голосування

Для зручності користувачів та мешканців було створено розділ "Послуги та сервіси", який містить наступні пункти:

- Послуги електрика

- Послуги Сантехніка
- Прибирання
- Дезінфекція
- Безпека
- Аварійне відкриття замків
- Ремонтні роботи
- Домашній майстер
- Перепланування
- Переїзд
- Продаж та оренда квартири
- Хімчистка одягу
- Хімчистка меблів та килимів
- Догляд за тваринами
- Догляд за рослинами
- Таксі

Цей розділ розроблено з метою забезпечення максимальної зручності для мешканців, надаючи їм широкий спектр послуг для вирішення побутових питань та покращення якості життя у житловому комплексі. Завдяки інтеграції цих сервісів в один застосунок, мешканці можуть швидко та легко замовляти необхідні послуги, не витрачаючи час на пошуки різних підрядників. Це дозволяє їм зосередитися на важливіших справах та насолоджуватися комфортом свого житла. Кожен пункт меню надає детальну інформацію про доступні послуги та дає можливість зв'язатися з відповідними фахівцями або службами. Користувачі можуть легко замовити послуги через застосунок, відстежувати статус замовлення та отримувати необхідну підтримку.

Додатково, розділ "Послуги та сервіси" допомагає створити спільноту мешканців, які можуть ділитися відгуками та рекомендаціями про якість отриманих послуг. Це сприяє підвищенню загального рівня обслуговування та задоволеності мешканців. Цей функціонал детально продемонстрований на

рисунках 3.15 – 3.16, де можна побачити інтерфейс користувача, доступні опції та приклади взаємодії з послугами.

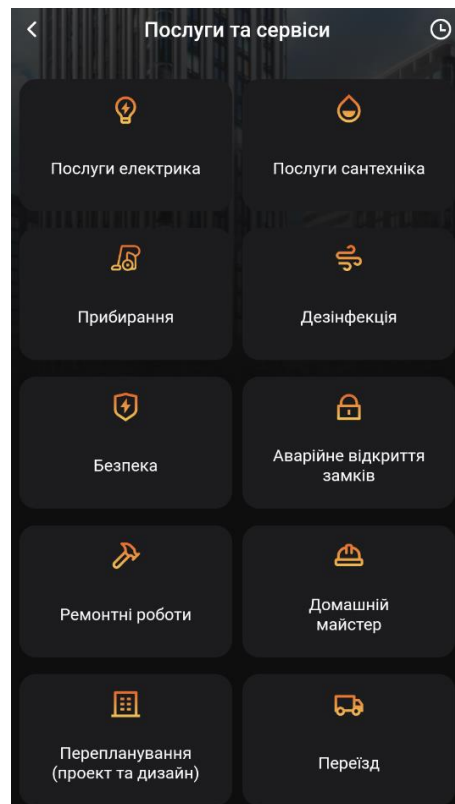


Рисунок 3.15 – Функціонал з розділу "Послуги та сервіси"

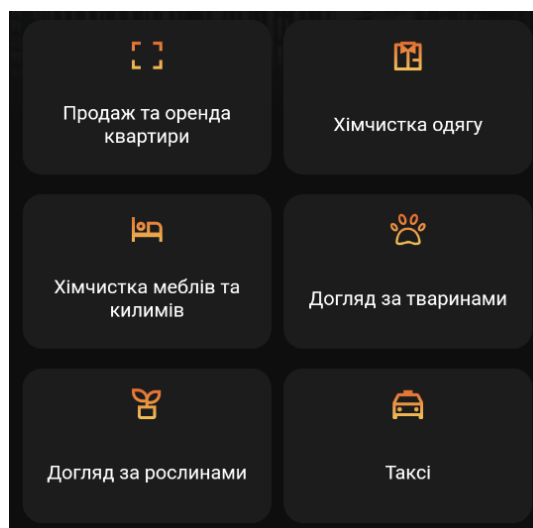


Рисунок 3.16 – Функціонал з розділу "Послуги та сервіси"

Після цього було розроблено екран "Домашній маркет", який містить такі підрозділи: Доставка їжі, Доставка кави, Доставка води, Квіти і

подарунки. Цей екран дозволяє мешканцям легко замовляти необхідні товари та послуги безпосередньо до своїх дверей, забезпечуючи максимальний комфорт та економію часу.

Розділ "Домашній маркет" розроблений з урахуванням різноманітних потреб мешканців. Наприклад, підрозділ "Доставка їжі" дає можливість обирати з широкого асортименту страв від різних ресторанів та кафе. Підрозділ "Доставка кави" спеціально створений для любителів цього напою, пропонуючи швидку доставку свіжої кави прямо додому. Підрозділ "Доставка води" допомагає забезпечити постійний запас чистої питної води, а підрозділ "Квіти і подарунки" дозволяє замовляти квіти та подарункові набори для різних святкових подій. Цей функціонал, разом із його інтерфейсом детально продемонстрований на рисунку 3.17.

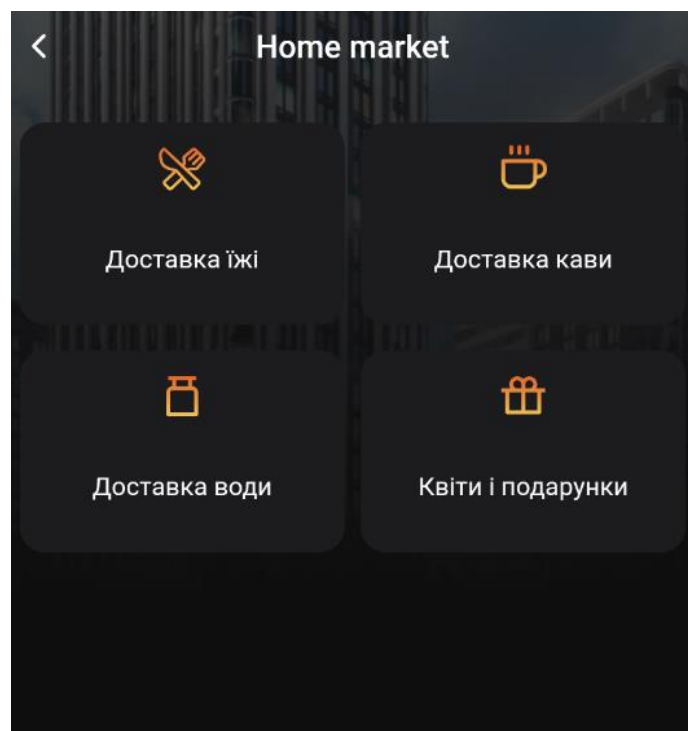


Рисунок 3.17 – Екран "Домашній маркет"

Було також розроблено інтерфейс для вкладки "Відеоспостереження". Ця функція дозволяє мешканцям переглядати відеозаписи з камер спостереження, розташованих на території житлового комплексу. Завдяки

цьому інтерфейсу користувачі можуть в режимі реального часу спостерігати за безпекою та подіями, що відбуваються навколо їхніх осель. Рисунок 3.18 ілюструє, як виглядає цей інтерфейс.

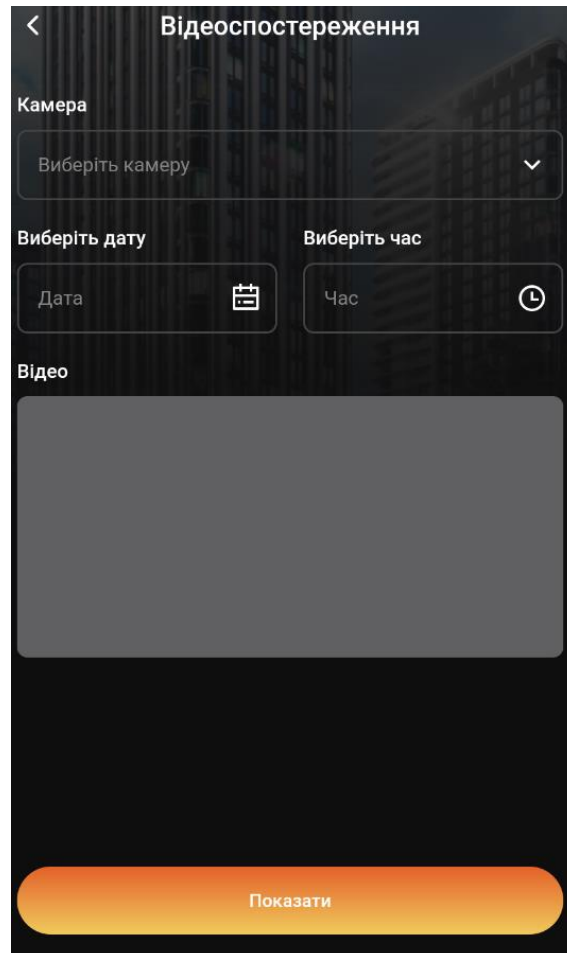


Рисунок 3.18 – Інтерфейс вкладки "Відеоспостереження"

Останнім розділом головного меню управління квартирою є "Лічильники". Цей розділ надає користувачам можливість переглядати та керувати показниками лічильників води, газу та електроенергії, що допомагає їм ефективніше контролювати споживання ресурсів і відповідні витрати (див. рис. 3.19).

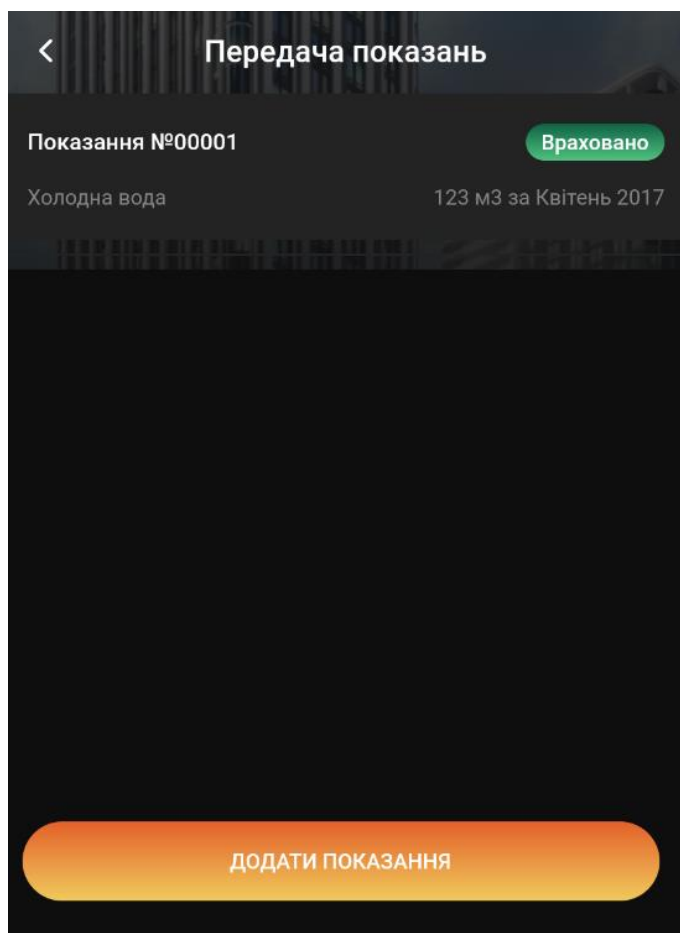


Рисунок 3.19 – Розділ "Лічильники"

Було створено інтерфейсні розділи для пунктів "Сповіщення", "Профіль" та "Новини", які знаходяться у вкладці швидкого доступу на нижній панелі головного меню застосунка. Ці розділи надають користувачам можливість зручно керувати налаштуваннями, переглядати актуальні новини та отримувати важливі сповіщення безпосередньо з головного екрана застосунка.

Екран "Сповіщення" розроблено з метою забезпечення мешканців житлового комплексу своєчасною та зручною інформацією про важливі події, новини та зміни, що відбуваються в комплексі (див. рис. 3.20). Цей розділ надає можливість користувачам отримувати та переглядати всі сповіщення, які можуть стосуватися як загальних питань житлового комплексу, так і індивідуальних повідомлень для кожного мешканця.

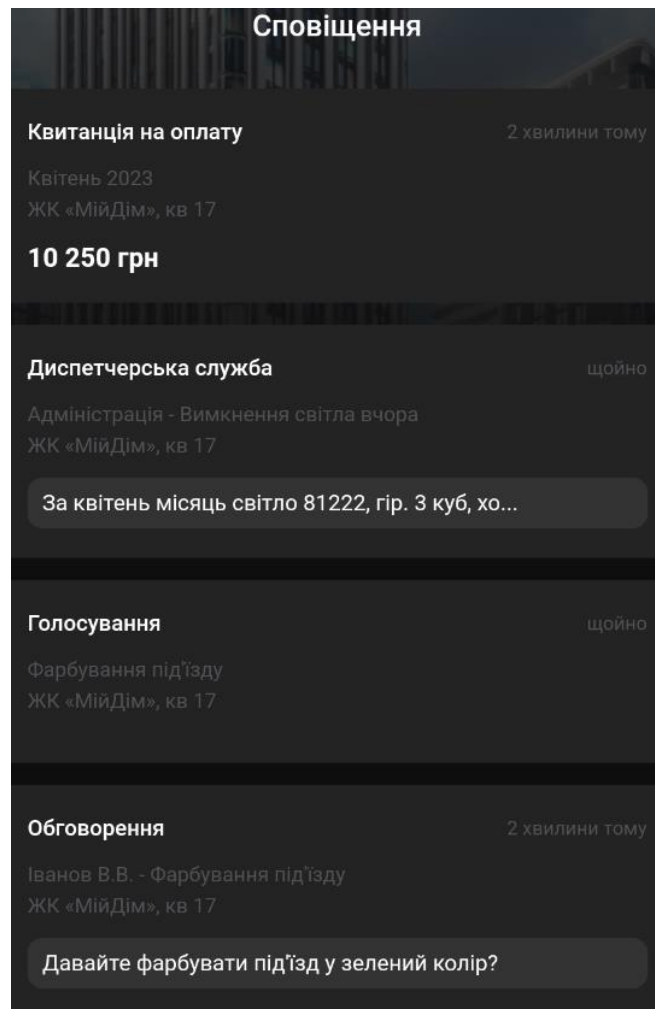


Рисунок 3.20 – Розділ "Сповіщення"

Екран "Профіль" розроблений для забезпечення користувачів можливістю керувати своєю особистою інформацією та налаштуваннями в мобільному застосунку (див. рис. 3.21). Цей розділ дозволяє мешканцям житлового комплексу оновлювати свої дані, налаштовувати сповіщення, керувати платіжною інформацією та змінювати мову застосунка. Екран "Профіль" є ключовим елементом, що забезпечує персоналізацію та зручність використання застосунка.

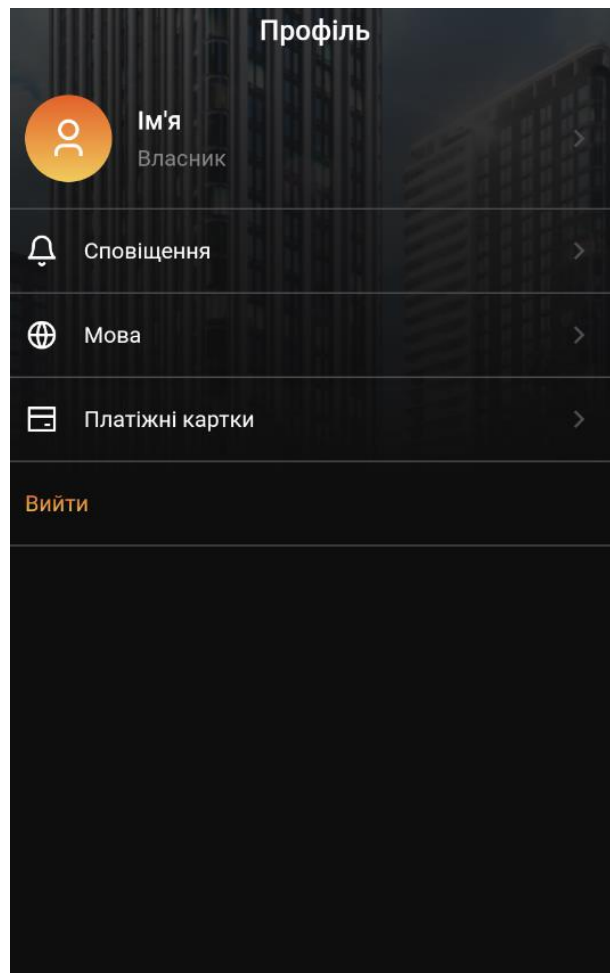


Рисунок 3.21 – Розділ "Профіль"

Екран "Новини" у мобільному застосунку розроблено для надання мешканцям житлового комплексу актуальної інформації про події, оголошення, новини та важливі оновлення. Цей екран забезпечує зручний доступ до всіх новинних повідомлень та дозволяє користувачам бути в курсі всіх подій, що відбуваються у їхньому житловому комплексі. Завдяки інтерактивному та інтуїтивно зрозумілому інтерфейсу, мешканці можуть легко переглядати останні новини, дізнаватися про важливі анонси та бути в курсі будь-яких змін або заходів.

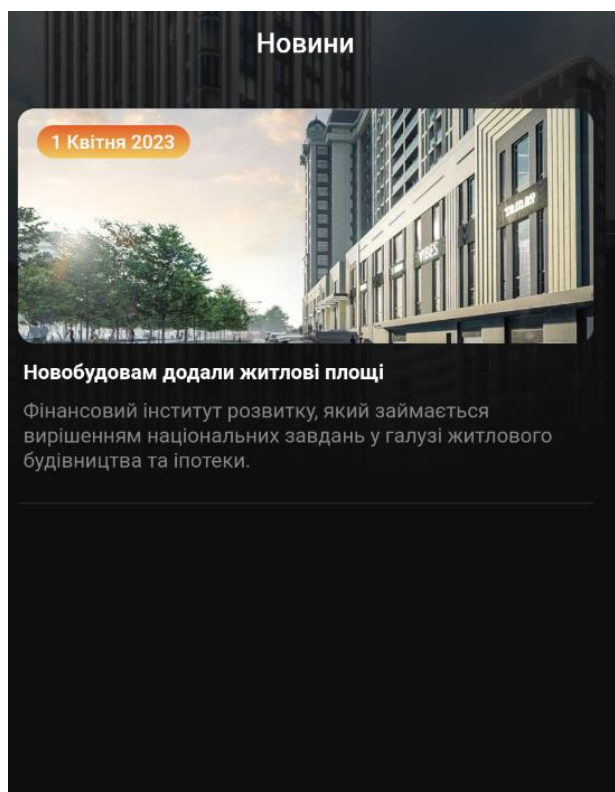


Рисунок 3.21 – Розділ "Новини"

Розробка графічного інтерфейсу успішно завершилася, і в процесі було проведено оптимізацію для досягнення максимальної зручності та ефективності його використання.

3.3 Розробка модулів застосунку

Розробка модулів мобільного застосунку для мешканців ОСББ є важливим етапом, який включає створення функціональних компонентів, що забезпечують виконання всіх необхідних завдань користувачами [18]. Кожен модуль відповідає за окрему частину функціоналу застосунку і дозволяє мешканцям ефективно взаємодіяти з системою.

Головні модулі як потрібно розробити:

1. Модуль авторизації
2. Модуль вибору квартири
3. Модуль головного меню квартири, яке містить наступні пункти:

- Діло.
 - Квартира.
 - Квитанції.
 - Диспетчерська служба.
 - Обговорення і голосування.
 - Послуги і сервіси.
 - Домашній маркет.
 - Відеоспостереження.
 - Лічильники.
4. Модуль налаштування профілю
 5. Модуль сповіщень
 6. Модуль новин

Кожен з цих модулів розробляється з урахуванням принципів зручності, простоти використання та безпеки, що дозволяє створити інтуїтивно зрозумілий та функціональний застосунок для мешканців ОСББ.

Для зручності користувачів у модулі авторизації додана функція "Забули пароль", яка відображена на рисунку 3.22. Ця функція дозволяє користувачам швидко відновити доступ до свого облікового запису без необхідності звертатися до служби підтримки. Крім того, розробка модулів передбачає використання сучасних методів шифрування даних та автентифікації для забезпечення максимального рівня безпеки інформації користувачів.

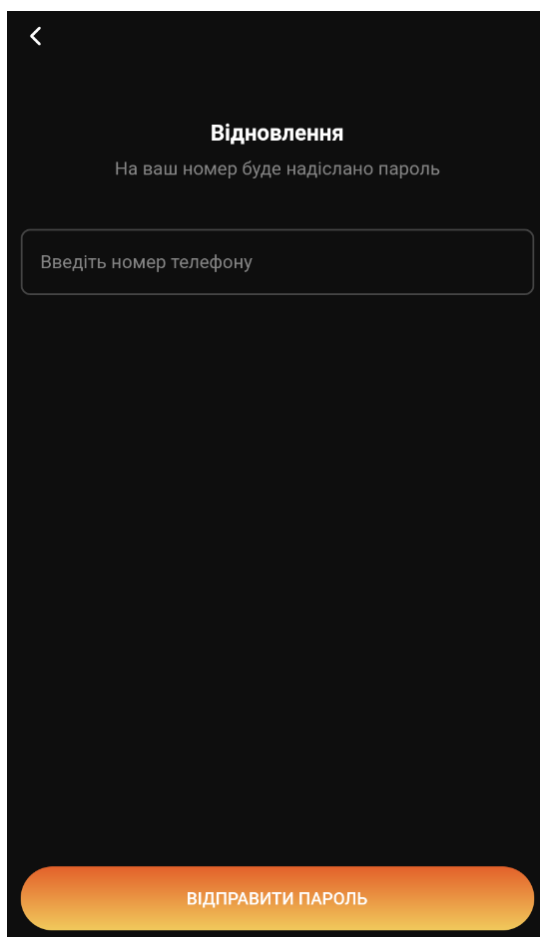


Рисунок 3.22 – Екран відновлення доступу до облікового запису

Коли користувач натискає на посилання "Забули пароль", система автоматично перенаправляє його на екран відновлення пароля. На цьому екрані користувач повинен ввести свій номер телефону, пов'язаний з обліковим записом. Після введення номеру, система надсилає інструкції для відновлення пароля на вказаний номер телефону.

Процес відновлення пароля включає такі кроки:

- Введення номеру телефону: Користувач вводить свій номер телефону, пов'язаний з обліковим записом.
- Отримання коду: На вказаний номер надсилається код для відновлення пароля, який потрібно ввести в застосунку.
- Скидання пароля: Користувач переходить на екран, де може встановити новий пароль для свого облікового запису.

- Підтвердження нового пароля: Після успішного скидання пароля користувач повертається на екран авторизації та входить у систему, використовуючи новий пароль.

Функція "Забули пароль" забезпечує простий та безпечний спосіб відновлення доступу до облікового запису, що підвищує зручність використання застосунку та допомагає уникнути неприємних ситуацій, пов'язаних з втратою доступу до особистих даних.

Наступний код реалізує екран авторизації для мобільного застосунку. Він забезпечує функціонал введення облікових даних користувача та входу в систему. Приклад коду зображено на рисунку 3.23.

```
class _SignInPageState extends State<SignInPage> {
  final FocusScopeNode _focusScopeNode = FocusScopeNode();
  final TextEditingController _accountController = TextEditingController();
  final TextEditingController _passwordController = TextEditingController();

  bool _isObscureText = true;
  bool _policyAccept = false;

  bool get _isValidFields {
    final String account = _accountController.text.trim();
    final String password = _passwordController.text.trim();

    if (account.isEmpty && password.isEmpty) {
      showDialog(
        context: context,
        builder: (_) => const ErrorModal(
          errorMessage: 'Поля мають бути заповнені',
        ),
      );

      return false;
    }

    if (!_policyAccept) {
      return false;
    }

    return true;
  }

  void _signIn() {
    FocusScope.of(context).unfocus();

    if (_isValidFields) {
      LocalDB.instance.saveUser(
        const UserModel(
          userId: 1,
        ),
      );

      Navigator.of(context, rootNavigator: true).pushNamedAndRemoveUntil(
        MainPage.routeName,
        (route) => false,
      );
    }
  }
}
```

Рисунок 3.23 – Реалізація екрану авторизації

Цей код створює екран входу в мобільний застосунок. Він включає два текстових поля для введення особистого рахунку та пароля, а також різні компоненти користувацького інтерфейсу, такі як кнопки, чекбокси та

повідомлення про помилки. Код перевіряє, чи заповнені поля та чи прийнято політику конфіденційності. Якщо все в порядку, користувач зберігається в базі даних і перенаправляється на головну сторінку застосунка.

Для реалізації функції "Забули пароль" було створено код зображений на рисунку 3.24. Код перевіряє, чи правильно введений номер телефону (повинен складатися з 14 символів) і чи не залишене поле порожнім. Якщо дані введені правильно, користувач перенаправляється на сторінку для введення ОТР (одноразовий пароль).

```
class PasswordRecoveryPage extends StatefulWidget {
  const PasswordRecoveryPage({Key? key}) : super(key: key);

  static const routeName = '/password_recovery';

  @override
  _PasswordRecoveryPageState createState() => _PasswordRecoveryPageState();
}

class _PasswordRecoveryPageState extends State<PasswordRecoveryPage> {
  final TextEditingController _phoneController = TextEditingController();

  String? _errorMessage;

  bool get _isValidFields {
    final String phone = _phoneController.text.trim();
    String? errorMessage;

    if (phone.isEmpty) {
      errorMessage = 'Заповніть поле';
    }

    if (phone.length != 14) {
      errorMessage = 'Неправильний номер телефону';
    }

    setState(() {
      _errorMessage = errorMessage;
    });

    return errorMessage == null;
  }

  void _validateAndSend() {
    FocusScope.of(context).unfocus();
    if (_isValidFields) {
      Navigator.pushNamed(
        context,
        OtpPage.routeName,
      );
    }
  }
}
```

Рисунок 3.24 – Реалізація функції "Забули пароль"

Було створено код який реалізує головний екран мобільного застосунку, він забезпечує навігацію між основними розділами застосунку, такими як "Об'єкти", "Сповідання", "Профіль" та "Новини", з використанням нижньої панелі навігації (див. рис. 3.25). Він використовує BlocProvider і

BlocListener для управління станом навігації через NavigationBloc. Головна сторінка (MainPage) містить нижню навігаційну панель, яка дозволяє користувачам перемикатися між різними вкладками: об'єкти (ObjectsPage), повідомлення (NotificationsPage), профіль (ProfilePage) і новини (NewsPage). Перемикання вкладок виконується за допомогою методу `_onSelectTab`, який оновлює поточну вкладку і завантажує відповідну сторінку, використовуючи глобальний ключ навігатора.

```
class MainPage extends StatefulWidget {
  const MainPage({Key? key}) : super(key: key);

  static const routeName = '/main';

  @override
  State<MainPage> createState() => _MainPageState();
}

class _MainPageState extends State<MainPage> {
  static const List<String> _pages = [
    ObjectsPage.routeName,
    NotificationsPage.routeName,
    ProfilePage.routeName,
    NewsPage.routeName,
  ];

  static final GlobalKey<NavigatorState> _globalKey =
    GlobalKey<NavigatorState>();

  int _currentTab = 0;

  void _onSelectTab(int index) {
    if (_globalKey.currentState != null) {
      if (_currentTab != index) {
        setState(() => _currentTab = index);

        _globalKey.currentState!.pushNamedAndRemoveUntil(
          _pages[index],
          (route) => false,
        );
      } else {
        _globalKey.currentState!.popUntil((route) => route.isFirst);
      }
    }
  }
}
```

Рисунок 3.25 – Реалізація головного екрану застосунка

Надалі було створено код реалізації головного меню для керування квартирою в мобільному застосунку (див. рис. 3.25). Меню надає користувачам доступ до різних розділів, пов'язаних з управлінням квартирою та послугами. На сторінці відображаються різні розділи у вигляді плиток, організованих у сітці за допомогою StaggeredGrid. Кожна плитка представляє певний розділ, наприклад, "Квартира", "Квитанції", "Диспетчерська служба", "Обговорення" тощо. Користувач може натискати на плитки для переходу до відповідних сторінок.

```

class ApartmentSectionsPage extends StatelessWidget {
  const ApartmentSectionsPage({Key? key}) : super(key: key);

  static const routeName = '/apartment_sections';
  @override
  Widget build(BuildContext context) {
    return ImageScaffold(
      appBar: CustomAppBar(
        title: const Column(
          children: [
            Text(
              'ЖК «4 Towers»',
            ), // Text
            Text(
              'Бул. Космонавтів, 32',
              style: TextStyle(
                color: AppColors.textSecondary,
                fontSize: 16.0,
              ), // TextStyle
            ), // Text
          ],
        ), // Column
        onLeading: () => Navigator.pop(context),
      ), // CustomAppBar
      body: SingleChildScrollView(
        padding: const EdgeInsets.symmetric(
          horizontal: 16.0,
        ).copyWith( // EdgeInsets.symmetric
          top: 18.0,
          bottom: 32.0,
        ),
        child: StaggeredGrid.count(
          crossAxisCount: 6,
          mainAxisSpacing: 16.0,
          crossAxisSpacing: 10.0,
          children: [
            StaggeredGridTile.count(
              crossAxisCellCount: 3,
              mainAxisCellCount: 2,
              child: SectionItem(
                icon: AppIcons.build,
                title: 'Діло',
                onTap: () => RouteNavigator.pushNamed(
                  context,
                  ApartmentCasePage.routeName,
                ),
              ),
            ),
          ],
        ),
      ),
    );
  }
}

```

Рисунок 3.25 – Реалізації головного меню для керування квартирою

Після цього було розроблено відповідні модулі, такі як: Діло, Квартира, Квитанції, Диспетчерська служба, Обговорення і голосування, Послуги і сервіси, Домашній маркет, Відеоспостереження та Лічильники. Кожен з цих модулів забезпечує окремий функціональний блок застосунку, що дозволяє користувачам керувати різними аспектами житлового комплексу та взаємодіяти з доступними сервісами. Детальний код для реалізації цих пунктів продемонстровано у Додатку В.

Також було розроблено модулі навігації "Сповіщення", "Профіль" та "Новини", розміщені у нижній панелі навігації. Спочатку було створено модуль "Профіль", де користувач може оновлювати інформацію свого

профілю, налаштовувати сповіщення, змінювати мову інтерфейсу, а також змінювати або додавати платіжні картки для обробки платежів (див. рис. 3.26).

```
class ProfilePage extends StatelessWidget {
  const ProfilePage({Key? key}) : super(key: key);

  static const routeName = '/profile';

  @override
  Widget build(BuildContext context) {
    return ImageScaffold(
      resizeToAvoidBottomInset: false,
      appBar: const CustomAppBar(
        title: Text('Профіль'),
      ), // CustomAppBar
      body: ListView(
        children: [
          SettingsItem(
            leadingIcon: Container(
              height: 62.0,
              width: 62.0,
              decoration: const BoxDecoration(
                shape: BoxShape.circle,
                gradient: kAccentGradient,
              ), // BoxDecoration
            child: ClipOval(
              child: imageUrl == null
                ? Padding(
                    padding: const EdgeInsets.all(16.0),
                    child: SvgPicture.asset(
                      AppIcons.profile,
                      color: AppColors.textPrimary,
                    ), // SvgPicture.asset
                  ) // Padding
                : CachedNetworkImage(
                    imageUrl: imageUrl!,
                    fit: BoxFit.cover,
                    progressIndicatorBuilder: (_, __, progress) {
                      return ImageLoader(
                        progress: progress,
                      ); // ImageLoader
                    },
                  ),
            ),
          ),
        ],
      ),
    );
  }
}
```

Рисунок 3.26 – Реалізація модулю "Профіль"

В подальшому було реалізовано модуль "Сповіщення", який дозволяє користувачам отримувати важливі повідомлення та оновлення (див. рис. 3.27). Цей модуль забезпечує швидкий доступ до всіх сповіщень, включаючи інформацію про нові події, важливі новини, а також статус платежів і технічного обслуговування. Користувачі можуть легко переглядати, фільтрувати та управляти своїми сповіщеннями, що сприяє оперативному реагуванню на важливу інформацію.

```

class NotificationsPage extends StatelessWidget {
  const NotificationsPage({Key? key}) : super(key: key);

  static const routeName = '/notifications';

  @override
  Widget build(BuildContext context) {
    return ImageScaffold(
      appBar: const CustomAppBar(
        title: Text('Сповіщення'),
      ), // CustomAppBar
      body: ListView.separated(
        padding: const EdgeInsets.only(
          top: 15.0,
        ), // EdgeInsets.only
        itemCount: _notificationsItems.length,
        separatorBuilder: (_, __) {
          return const SizedBox(
            height: 15.0,
          ); // SizedBox
        },
        itemBuilder: (_, int i) {
          return _notificationsItems[i];
        },
      ), // ListView.separated
    ); // ImageScaffold
  }
}

```

Рисунок 3.27 – Реалізація модулю "Сповіщення"

Після завершення розробки модуля "Сповіщення" було прийнято рішення створити також модуль "Новини" (див. рис. 3.28). Цей модуль надає користувачам доступ до актуальних новин і подій, що стосуються житлового комплексу. Користувачі можуть переглядати новини про важливі зміни, заходи, оновлення та інші події, що відбуваються в ОСББ. Це допомагає підтримувати інформованість мешканців про всі важливі аспекти життя в житловому комплексі.

Таким чином, модуль "Новини" не лише інформує мешканців, але й сприяє створенню більш згуртованої та активної спільноти в ОСББ.

```

class NewsPage extends StatelessWidget {
  const NewsPage({Key? key}) : super(key: key);

  static const routeName = '/news';

  @override
  Widget build(BuildContext context) {
    return ImageScaffold(
      appBar: const CustomAppBar(
        title: Text('Новини'),
      ), // CustomAppBar
      body: ListView.separated(
        padding: const EdgeInsets.symmetric(
          vertical: 20.0,
          horizontal: 16.0,
        ), // EdgeInsets.symmetric
        itemCount: 10,
        separatorBuilder: (_, __) {
          return const Padding(
            padding: EdgeInsets.only(
              top: 10.0,
              bottom: 5.0,
            ), // EdgeInsets.only
            child: Divider(
              color: AppColors.border,
            ), // Divider
          ); // Padding
        },
        itemBuilder: (_, int i) {
          return NewsItem(
            onTap: () => Navigator.pushNamed(
              context,
              NewsDetailPage.routeName,
              arguments: i,
            ),
          ); // NewsItem
        },
      ), // ListView.separated
    ); // ImageScaffold
  }
}

```

Рисунок 3.28 – Реалізація модулю "Новини"

3.4 Висновки

У цьому розділі було детально розглянуто процес розробки та реалізації програмних модулів для мобільного застосунку, призначеного для мешканців ОСББ. Основною метою даного розділу було створення функціональних компонентів застосунку, які забезпечують зручність та ефективність користування. В рамках розділу було реалізовано наступні ключові модулі:

- Головний екран - основний інтерфейс застосунку, що забезпечує швидкий доступ до всіх основних функцій через інтуїтивно зрозумілу навігаційну панель.
- Модулі квартир - включають функції управління інформацією про квартиру, перегляду квитанцій, звернення до диспетчерської служби, доступ до відеоспостереження, обговорень та голосувань, послуг і сервісів, а також домашнього маркету.

- Модулі навігації - такі як "Сповіщення", "Профіль" та "Новини", що розташовані на нижній панелі навігації і дозволяють користувачам легко переходити між основними розділами застосунку.

Зокрема, було розроблено модулі "Діло", "Квартира", "Квитанції", "Диспетчерська служба", "Обговорення", "Голосування", "Послуги і сервіси", "Домашній маркет", "Відеоспостереження", "Лічильники", а також забезпечено їхню інтеграцію в загальну структуру застосунку.

Також було приділено увагу розробці модуля "Профіль", який надає користувачам можливість оновлювати свої дані, налаштовувати сповіщення, змінювати мову інтерфейсу та керувати платіжною інформацією. Після цього були реалізовані модулі "Сповіщення" та "Новини", що забезпечують актуальне інформування користувачів.

Розробка кожного модуля включала в себе проектування інтерфейсу, реалізацію бізнес-логіки та інтеграції. В результаті було створено повнофункціональний мобільний застосунок, що відповідає вимогам сучасних користувачів і значно полегшує взаємодію мешканців ОСББ з адміністрацією та один з одним.

Таким чином, виконана робота з розробки та реалізації програмних модулів мобільного застосунку є важливим етапом у створенні ефективного інструменту для управління житлом та підвищення комфорту мешканців.

4 ТЕСТУВАННЯ ПРОГРАМИ

4.1 Тестування системи

В розробці програмного застосунку велику увагу зосереджують на тестуванні, що є ключовим етапом. Основна мета тестування полягає у виявленні можливих дефектів або помилок у програмному забезпеченні перед його випуском для користувачів, забезпечуючи таким чином його надійну та коректну роботу.

У тестуванні Flutter використовуються різні методи для перевірки правильності та ефективності програмного продукту. Серед них можна виділити наступні:

- Інтеграційне тестування: цей вид тестів спрямований на перевірку взаємодії та інтеграції різних компонентів або модулів у програмі. Вони перевіряють поведінку програми в цілому, включаючи навігацію, потік даних та взаємодію між різними екранами або функціями. Інтеграційні тести допомагають забезпечити правильну роботу всієї системи як єдиної сутності. Для їх написання Flutter надає пакет "flutter_driver", який дозволяє взаємодіяти з програмою через окремий процес [19].
- Енд-то-енд (E2E) тести: це тести, які перевіряють повну функціональність програми від початку до кінця, включаючи всі інтеграції та залежності. E2E тести імітують реальну поведінку користувача, перевіряючи взаємодію з інтерфейсом, сервером, базою даних та іншими компонентами системи. Такі тести допомагають гарантувати, що всі частини програми працюють разом як очікується, і що кінцевий користувач отримає очікуваний досвід. E2E тести можуть виявляти проблеми, які можуть не бути видимими на рівні модульних або інтеграційних тестів, тим самим забезпечуючи високу якість та надійність програмного забезпечення [20].
- Мокінг і ін'єкція залежностей: ці методи допомагають ізолювати код програми від зовнішніх залежностей, таких як API або бази даних, що

дозволяє забезпечити передбачувану роботу системи та спростити процес тестування. Фреймворки впровадження залежностей, такі як "get_it" або "provider", можна використовувати для керування залежностями та полегшення моделювання в тестах [21-22].

Тестування продуктивності також є важливим елементом, що дозволяє перевірити ефективність та надійність мобільного застосунку при реальному навантаженні користувачів [23]. Результати тестування наведено на рисунку 4.1.

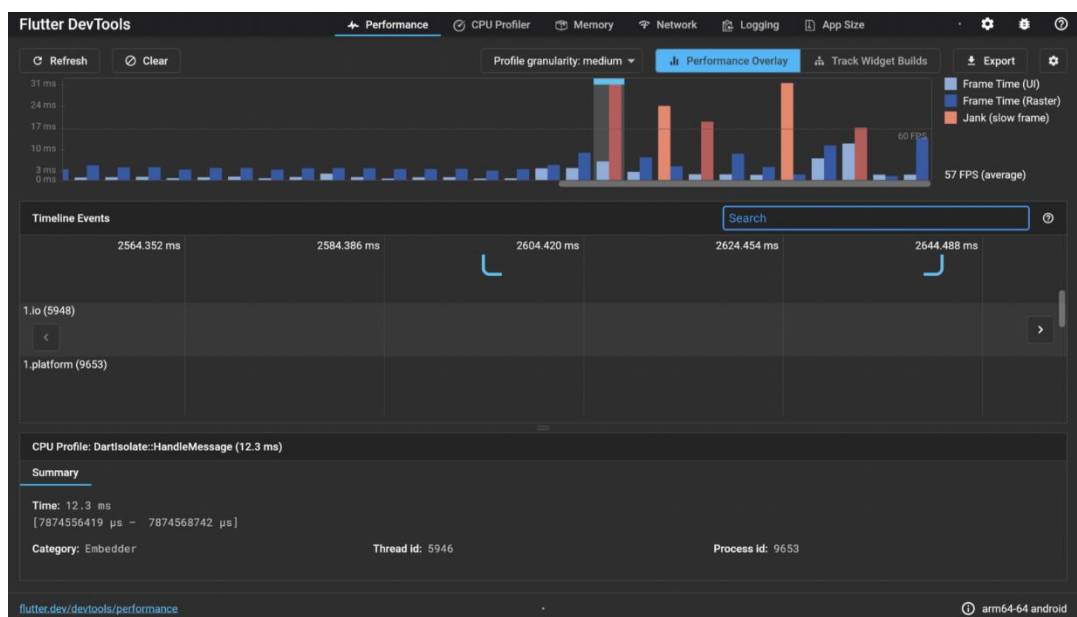


Рисунок 4.1 – Тестування продуктивності

Після цього ми проводимо перевірку розділу "Забули пароль", заповнюючи поля порожніми даними. Якщо користувач спробує залишити поле порожнім, система автоматично виведе повідомлення про помилку, щоб проінформувати його про необхідність введення всіх обов'язкових даних. Це допомагає забезпечити коректну роботу функціоналу та покращити користувацький досвід. Результат тестування зображено на рисунку 4.2.

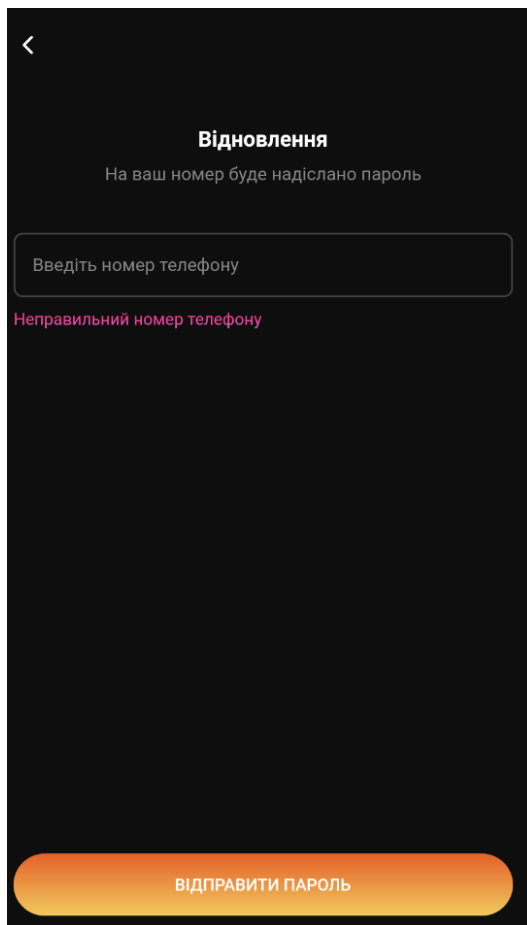


Рисунок 4.2 – Тестування спроби відновити пароль з пустими даними

Під час редагування або додавання платіжної інформації, іноді користувач може випадково ввести дані у неправильному форматі. Для запобігання можливим помилкам, система автоматично перевіряє введені дані на відповідність стандартному формату. Якщо формат введених даних не відповідає очікуваному, система негайно видає відповідне повідомлення про помилку та запрошує користувача ввести коректну інформацію. Це дозволяє забезпечити точність та надійність введених даних у профілі користувача. Результати тестування на правильність введення платіжної інформації представлено на рисунку 4.3.

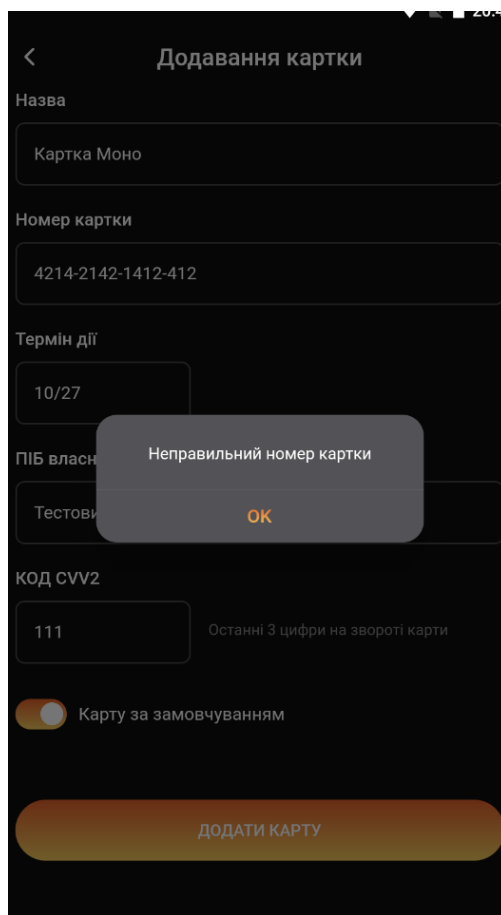


Рисунок 4.3 – Тестування правильності платіжної інформації

Під час замовлення послуг можливі випадки, коли користувач вводить невірну або неповну інформацію. У таких ситуаціях система повинна надати зрозуміле та точне повідомлення про помилку, що чітко вказує на характер неправильності. Це дозволить користувачеві зрозуміти, де саме виникла помилка та як її виправити, вводячи достовірні дані. Результати тестування відображення помилки під час введення невірної інформації під час замовлення послуги представлено на рисунку 4.4.

Рисунок 4.4 – Тестування неправильно веденої інформації при замовленні послуг

Отже, тестування системи включає оцінку функціональності, надійності та продуктивності програмного забезпечення. Виконані тести охоплювали інтеграційне тестування для перевірки взаємодії між різними компонентами системи, золоті тести для випробування ключових функцій та сценаріїв, а також мокінг і ін'єкцію залежностей для імітації різних умов та динамічних взаємодій.

Також при поданні показників користувач може забути заповнити поле або не додати необхідну інформацію. Це може призвести до подачі неправильних або неповних даних. В таких випадках система також має надати відповідне повідомлення про помилку, яке інформує користувача про те, що деякі поля залишилися незаповненими або містять неправильні дані. Це сприяє покращенню точності та повноти поданих даних. Результати

тестування виведення повідомлення про помилку щодо незаповнених полів під час подання показників представлено на рисунку 4.5.

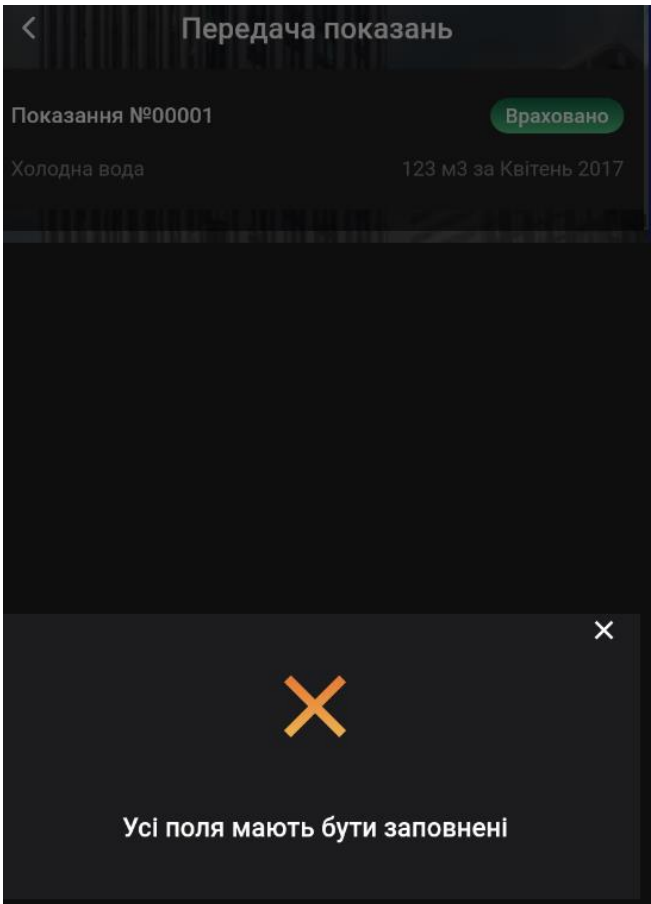


Рисунок 4.5 – Тестування ведення не повних показників

4.2 Розробка інструкції користувача

Інструкція користувача пропонує послідовні кроки, необхідні для досягнення певних цілей або виконання операцій. Основна мета цієї інструкції - забезпечити зручність та ефективність у використанні продукту. Деталі щодо мінімальної конфігурації телефону користувача наведено в таблиці 4.1.

Таблиця 4.1 – Мінімальна вимоги:

Операційна система	Android
Тип процесора	2-ядерний з тактовою частотою не менше 1.5 ГГц

Продовження таблиці 4.1

Об'єм оперативної пам'яті	2 ГБ
Місце на жорсткому диску	300 Мб
Версія операційної системи	Android 7.0
Пропускна здатність мережі	підтримка 3G/4G

Деталі щодо рекомендованих вимог до телефону наведено в таблиці

4.2.

	Для Android
Тип процесора	4-ядерний з тактовою частотою 2.0 ГГц або вище
Об'єм оперативної пам'яті	3 ГБ
Місце на жорсткому диску	500 ГБ
Версія операційної системи	Остання версія Android
Пропускна здатність мережі	підтримка 4G/LTE

Під час процедури авторизації в системі будь-які помилки негайно повідомляються користувачу. Після успішної авторизації система перевіряє наявність облікового запису. У випадку, якщо користувач забув свій пароль та намагається скинути його, не вказавши номер телефону, або якщо такого облікового запису не існує в системі, він отримає повідомлення про помилку та система запросить правильні дані від користувача. Крім того, якщо

користувач введе неправильну платіжну інформацію, система негайно попередить його про помилку та точно вкаже, де саме поля містять невірні дані. У випадку замовлення будь-яких послуг через застосунок, якщо користувач не заповнить всю необхідну інформацію у відповідних полях, застосунок покаже попередження про помилку та вказівку, де саме відбулася помилка та вимагати заповнити всі поля. Якщо в усіх сценаріях надана інформація є достовірною і відповідає системним стандартам, користувач не отримає повідомлення про помилку

Отже, інструкція користувача виступає ключовим документом, який сприяє успішному використанню продукту чи послуги, надаючи користувачам необхідну інформацію та вказівки для ефективної взаємодії з ними. Вона допомагає уникнути непорозумінь і помилок, що дозволяє заощадити час користувача, оскільки містить чітку та структуровану інформацію.

4.3 Висновки

У четвертому розділі було проведено ретельне тестування функціональних та графічних елементів мобільного застосунка для перевірки їх працездатності. На основі результатів тестування було складено керівництво користувача для системи, яке містить вичерпну інформацію про конфігурацію пристрою користувача з операційною системою Android.

ВИСНОВКИ

У цій бакалаврській дипломній роботі було розглянуто та досліджено ряд важливих аспектів розробки мобільного застосунку для управління житловим комплексом. Основною метою дослідження було створення зручного та функціонального інструменту для мешканців житлового комплексу з можливістю управління різними аспектами їхнього життя.

У результаті аналізу сучасних тенденцій та практик в розробці мобільних застосунків було обрано та застосовано найбільш відповідні методи та інструменти для створення застосунку. Зокрема, використання Flutter дозволило швидко та ефективно реалізувати багатофункціональний застосунок з кросплатформовою підтримкою.

Під час розробки було надано особливу увагу до забезпечення безпеки та зручності користувачів. Різноманітні модулі та функціонал застосунку дозволяють мешканцям легко та ефективно керувати різними аспектами їхнього побуту.

У результаті виконаної роботи було розроблено та протестовано функціональний прототип мобільного застосунку для управління житловим комплексом. Отримані результати показали високу ефективність та зручність використання застосунку для користувачів.

Загалом, робота виявилася успішною та відкрила широкі можливості для подальшого вдосконалення та розвитку в цьому напрямку.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

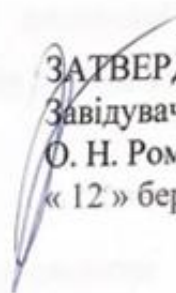
1. Дах [Електронний ресурс]. – Режим доступу: <https://dah-online.com/>
2. OSSB.work [Електронний ресурс]. – Режим доступу: <https://osbb.work/>
3. Моє ОСББ [Електронний ресурс]. – Режим доступу: <https://moeosbb.com/>
4. Розробка мобільних застосунків [Електронний ресурс]. – <https://webcase.com.ua/uk/razrabotka-mobilnyh-prilozhenij/>
5. Розробка кроссплатформових застосунків [Електронний ресурс]. – <https://webcase.com.ua/uk/krossplatformennaya-razrabotka/>
6. Веб-додаток: що це таке, типи, переваги, принцип роботи [Електронний ресурс]. – <https://megasite.ua/ua/veb-prilogenie-hto-eto-takoe-tipi-preimushchestva-printsip-raboti>
7. Розробка застосунків на фреймворку Flutter [Електронний ресурс]. – <https://avada-media.ua/ua/services/flutter/>
8. Ознайомлення з патерном MVC (Model-View-Controller) [Електронний ресурс]. – <https://javarush.com/ua/groups/posts/uk.2536.chastina-7-oznayomlennja-z-paternom-mvc-model-view-controller>
9. Model-View-ViewModel [Електронний ресурс]. – <https://www.wikidata.uk-ua.nina.az/Model-View-ViewModel.html>
10. Архітектура Flutter: як обрати і чи підійде BLoC [Електронний ресурс]. – <https://highload.today/uk/blogs/navit-dosvidcheni-rozrobniki-stikayutsya-z-neochevidnimi-rechami-yak-obrati-arhitekturu-dodatku-na-flutter/>
11. Методи оцінки юзабіліті для мобільних застосунків [Електронний ресурс]. – <https://artjoker.ua/blog/metody-otsenki-yuzabiliti-dlya-mobilnykh-prilozheniy/>
12. Color in UX design. [Електронний ресурс]. – <https://medium.com/@mariya.lytv/color-in-ux-design-519e85091b29>

13. Unified Modeling Language [Електронний ресурс]. – https://uk.wikipedia.org/wiki/Unified_Modeling_Language
14. Етапи створення мобільного застосунку [Електронний ресурс]. – <https://webcase.com.ua/uk/blog/etapi-stvorennja-mobilnogo-dodatku/>
15. Dart [Електронний ресурс]. – <https://uk.wikipedia.org/wiki/Dart>
16. Створення інтерфейсів (UI), дизайн, проектування [Електронний ресурс]. – <https://design-masha.com/uk/stvorennja-interfejsiv-ui-dyzajn-proektuvannya/>
17. Дизайн - DIZZ - Dizz Agency [Електронний ресурс]. – <https://dizz.in.ua/uk/d0-b4-d0-b8-d0-b7-d0-b0-d0-b9-d0-bd-uk/>
18. Розробка мобільних застосунків від А до Я: повний гайд [Електронний ресурс]. – <https://dan-it.com.ua/uk/blog/rozrobka-mobilnih-dodatkov-vid-a-do-ja-povnij-gajd/>
19. Інтеграційне тестування [Електронний ресурс]. – <https://qalight.ua/baza-znaniy/integratsijne-testuvannya>
20. Тестування End-to-end [Електронний ресурс]. – <https://dou.ua/forums/topic/46158/>
21. Чим корисні моки в тестуванні ПЗ? [Електронний ресурс]. – <https://foxminded.ua/shcho-take-moky/>
22. Що таке “впровадження залежностей”? [Електронний ресурс]. – <https://doc.nette.org/uk/dependency-injection/introduction>
23. Тестування продуктивності [Електронний ресурс]. – <https://qalight.ua/baza-znaniy/testuvannya-produktivnosti/>

ДОДАТКИ

ДОДАТОК А – ТЕХНІЧНЕ ЗАВДАННЯ

Міністерство освіти і науки України
Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії

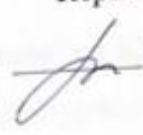
ЗАТВЕРДЖУЮ
Завідувач кафедри ПЗ
О. Н. Романюк
« 12 » березня 2024 р.

ТЕХНІЧНЕ ЗАВДАННЯ

на бакалаврську дипломну роботу

«Розробка програмного мобільного застосунку для мешканців ОСББ»
за спеціальністю 121 – Інженерія програмного забезпечення

Керівник бакалаврської дипломної роботи:

к.т.н., доц. каф. ПЗ Г.Б. Ракитянська
" 12 " березня 2024р.

Виконав:

студент гр. ЗПІ-20Б М.В. Ковальчук
" 12 " березня 2024р.

1. Найменування та галузь застосування

Бакалаврська дипломна робота: «Розробка програмного мобільного застосунку для мешканців ОСББ».

Галузь застосування – для мешканців та управлінських комітетів ОСББ.

2. Підстава для розробки.

Підставою для виконання бакалаврської дипломної роботи (БДР) є індивідуальне завдання на БДР та наказ № 80 від 11 березня 2024 р. ректора ВНТУ про закріплення тем БДР.

3. Мета та призначення розробки.

Метою роботи є розробка моделей управління ОСББ шляхом впровадження мобільного застосунку, спрямованого на вирішення проблем, що виникають у мешканців та управлінських комітетів ОСББ.

Основними задачами є:

- визначити вимоги до програмного застосунку;
- розробка модуля для авторизації користувача;
- реалізація модуля графічного інтерфейсу;
- інтеграція різних модулів у єдину мобільну систему, придатну для використання мешканцями ОСББ;
- проведення тестування розробленого програмного забезпечення

4. Вихідні дані для проведення БДР

Перелік основних літературних джерел, на основі яких буде виконуватись БДР.

5. Технічні вимоги

Середовище розробки – Visual Studio Code, мови розробки – Dart, фреймворк – Flutter засоби організації даних програми – база даних Firebase, операційна система – Android.

6. Конструктивні вимоги

Графічна та текстова документація повинна відповідати діючим

стандартам України.

7. Перелік технічної документації, що пред'являється по закінченню робіт:

1. Пояснювальна записка до БДР;
2. Технічне завдання;
3. Лістинги програми.

8. Вимоги до рівня уніфікації та стандартизації

При розробці програмних засобів слід дотримуватися уніфікації і ДСТУ.

9. Стадії та етапи розробки:

№ з/п	Назва етапів бакалаврської дипломної роботи	Строк виконання етапів роботи
1	Обґрунтування вибору методу розробки та постановка задач дослідження	14.03.23 – 02.04.23
2	Розробка структури та алгоритмів програмного продукту	03.04.23 – 10.04.23
3	Розробка інтерфейсу	10.04.23 – 15.04.23
4	Розробка модулів в мобільному застосунку	15.04.23 – 05.05.23
5	Тестування програми	05.05.23 – 10.05.23
6	Оформлення матеріалів до захисту БДР	11.05.23 – 30.05.23

10. Порядок контролю та прийняття

Порядок контролю і приймання роботи регламентується відповідними документами ВНТУ і державними стандартами.

ДОДАТОК Б – ПРОТОКОЛ ПЕРЕВІРКА НА ПЛАГІАТ

ПРОТОКОЛ ПЕРЕВІРКИ БАКАЛАВРСЬКОЇ ДИПЛОМНОЇ РОБОТИ НА НАЯВНІСТЬ ТЕКСТОВИХ ЗАПОЗИЧЕНЬ

Назва роботи: Розробка методів і програмних засобів антропометричних вимірювань з використанням тривимірного моделювання

Тип роботи: БДР

Підрозділ: кафедра програмного забезпечення, ФІТКІ

Науковий керівник: Ракитянська Г.Б.

Оригінальність	98,7
Схожість	1,3

Аналіз звіту подібності

▪ Запозичення, виявлені у роботі, оформленні коректно і не містять ознак плагіату.

☐ Виявлені у роботі запозичення не мають ознак плагіату, але їх надмірна кількість викликає сумніви щодо цілісності роботи і відсутності самостійності її автора. Роботу направити на доопрацювання.

☐ Виявлені у роботі запозичення є недобросовісними і мають ознаки плагіату та/або в ній містяться навмисні спотворення тексту, що вказують на спроби приховання недобросовісних запозичень.

Особа, відповідальна за перевірку



Черноволик Г.О.

Ознайомлені з повним звітом подібності, який був згенерований системою Unichек

Автор роботи

Керівник роботи



Ковальчук М.В

Ракитянська Г.Б.

ДОДАТОК В – ЛІСТИНГ ПРОГРАМИ

main_page.dart

```
import 'package:flutter/material.dart';
import 'package:flutter_bloc/flutter_bloc.dart';
import 'package:wowhome_flutter/blocs/navigation_bloc/navigation_bloc.dart';
import 'package:wowhome_flutter/pages/news_pages/news_page/news_page.dart';
import
'package:wowhome_flutter/pages/notifications_pages/notifications_page/notifications_page.dart'
;
import 'package:wowhome_flutter/pages/objects_pages/objects_page/objects_page.dart';
import 'package:wowhome_flutter/pages/profile_pages/profile_page/profile_page.dart';
import 'package:wowhome_flutter/routes/app_router.dart';
import 'package:wowhome_flutter/widgets/navigation/custom_bottom_navigation_bar.dart';

class MainPage extends StatefulWidget {
  const MainPage({Key? key}) : super(key: key);

  static const routeName = '/main';

  @override
  State<MainPage> createState() => _MainPageState();
}

class _MainPageState extends State<MainPage> {
  static const List<String> _pages = [
    ObjectsPage.routeName,
    NotificationsPage.routeName,
    ProfilePage.routeName,
    NewsPage.routeName,
  ];

  static final GlobalKey<NavigatorState> _globalKey =
    GlobalKey<NavigatorState>();

  int _currentTab = 0;

  void _onSelectTab(int index) {
    if (_globalKey.currentState != null) {
      if (_currentTab != index) {
        setState(() => _currentTab = index);

        _globalKey.currentState!.pushNamedAndRemoveUntil(
          _pages[index],
          (route) => false,
        );
      } else {
        _globalKey.currentState!.popUntil((route) => route.isFirst);
      }
    }
  }
}
```

```

    }
  }
}

@override
Widget build(BuildContext context) {
  return BlocProvider<NavigationBloc>(
    create: (_) => NavigationBloc(),
    child: WillPopScope(
      onWillPop: () async {
        final bool maybePop = await _globalKey.currentState!.maybePop();
        return !maybePop;
      },
      child: BlocListener<NavigationBloc, NavigationState>(
        listener: (_, state) {
          if (state.status == NavigationStateStatus.tab) {
            _onSelectTab(state.tabIndex);
          }
        },
        child: Scaffold(
          body: Navigator(
            key: _globalKey,
            initialRoute: ObjectsPage.routeName,
            onGenerateRoute: AppRouter.generateRoute,
          ),
          bottomNavigationBar: CustomBottomNavigationBar(
            currentTab: _currentTab,
            onSelect: _onSelectTab,
          ),
        ),
      ),
    ),
  );
}
}

```

notifications_page.dart

```

import 'package:flutter/material.dart';
import 'package:wowhome_flutter/widgets/navigation/custom_app_bar.dart';
import 'package:wowhome_flutter/widgets/uncategorized/image_scaffold.dart';

import 'widgets/notification_item.dart';

class NotificationsPage extends StatelessWidget {
  const NotificationsPage({Key? key}) : super(key: key);

```

```

static const routeName = '/notifications';

static final List<NotificationItem> _notificationsItems = [
  NotificationItem(
    title: 'Квитанція на оплату',
    dispatchTime: '2 хвилини тому',
    details: 'Квітень 2023 ',
    houseName: 'ЖК «МійДім», кв 17',
    price: '10 250 грн',
    onTap: () {
      //
    },
  ),
  NotificationItem(
    title: 'Диспетчерська служба',
    dispatchTime: 'щойно',
    details: 'Адміністрація - Вимкнення світла вчора',
    houseName: 'ЖК «МійДім», кв 17',
    message: 'За квітень місяць світло 81222, гір. 3 куб, хо...',
    onTap: () {
      //
    },
  ),
  NotificationItem(
    title: 'Голосування',
    dispatchTime: 'щойно',
    details: "Фарбування під'їзду",
    houseName: 'ЖК «МійДім», кв 17',
    onTap: () {
      //
    },
  ),
  NotificationItem(
    title: 'Обговорення',
    dispatchTime: '2 хвилини тому',
    details: "Іванов В.В. - Фарбування під'їзду",
    houseName: 'ЖК «МійДім», кв 17',
    message: "Давайте фарбувати під'їзд у зелений колір?",
    onTap: () {
      //
    },
  ),
  NotificationItem(
    title: 'Додаткова послуга',
    dispatchTime: '2 хвилини тому',
    details: 'Няня — Прогулянка парком',
    houseName: 'ЖК «МійДім», кв 17',
    message: 'Так, ми можемо гуляти парком і будь-яким ...',
    onTap: () {
      //
    },
  ),
];

```

```

    },
  ),
];

@override
Widget build(BuildContext context) {
  return ImageScaffold(
    appBar: const CustomAppBar(
      title: Text('Сповідання'),
    ),
    body: ListView.separated(
      padding: const EdgeInsets.only(
        top: 15.0,
      ),
      itemCount: _notificationsItems.length,
      separatorBuilder: (_, __) {
        return const SizedBox(
          height: 15.0,
        );
      },
      itemBuilder: (_, int i) {
        return _notificationsItems[i];
      },
    ),
  );
}
}

```

profile_page.dart

```

import 'package:cached_network_image/cached_network_image.dart';
import 'package:flutter/material.dart';
import 'package:flutter_svg/flutter_svg.dart';
import 'package:wowhome_flutter/database/local_database.dart';
import 'package:wowhome_flutter/models/users/user_model.dart';
import 'package:wowhome_flutter/pages/auth_pages/sign_in_page/sign_in_page.dart';
import 'package:wowhome_flutter/pages/profile_pages/edit_profile_page/edit_profile_page.dart';
import
'package:wowhome_flutter/pages/profile_pages/payment_cards_page/payment_cards_page/pay
ment_cards_page.dart';
import
'package:wowhome_flutter/pages/profile_pages/profile_notifications_page/profile_notifications_
page.dart';
import 'package:wowhome_flutter/pages/profile_pages/profile_page/widgets/lang_items.dart';
import 'package:wowhome_flutter/resources/app_colors.dart';
import 'package:wowhome_flutter/resources/app_icons.dart';
import 'package:wowhome_flutter/utils/constants.dart';
import 'package:wowhome_flutter/utils/route_navigator.dart';

```

```

import 'package:wowhome_flutter/widgets/loaders/image_loader.dart';
import 'package:wowhome_flutter/widgets/modals/error_modal.dart';
import 'package:wowhome_flutter/widgets/modals/modal_bottom_sheet.dart';
import 'package:wowhome_flutter/widgets/navigation/custom_app_bar.dart';
import 'package:wowhome_flutter/widgets/uncategorized/gradient_box.dart';
import 'package:wowhome_flutter/widgets/uncategorized/image_scaffold.dart';

import 'widgets/setting_item.dart';

class ProfilePage extends StatelessWidget {
  const ProfilePage({Key? key}) : super(key: key);

  static const routeName = '/profile';

  //TODO: replace variable with user.imageUrl
  final String? imageUrl = null;

  void _signOut(BuildContext context) {
    //TODO: implement logout

    showDialog(
      context: context,
      builder: (_) => ErrorModal(
        errorMessage: 'Ви впевнені, що хочете залишити програму?',
        cancelText: 'Відміна',
        applyText: 'Вийти',
        onApply: () {
          LocalDB.instance.saveUser(
            const UserModel(),
          );

          Navigator.of(context, rootNavigator: true).pushNamedAndRemoveUntil(
            SignInPage.routeName,
            (route) => false,
          );
        },
      ),
    );
  }

  @override
  Widget build(BuildContext context) {
    return ImageScaffold(
      resizeToAvoidBottomInset: false,
      appBar: const CustomAppBar(
        title: Text('Профіль'),
      ),
      body: ListView(
        children: [
          SettingsItem(

```

```

leadingIcon: Container(
  height: 62.0,
  width: 62.0,
  decoration: const BoxDecoration(
    shape: BoxShape.circle,
    gradient: kAccentGradient,
  ),
  child: ClipOval(
    child: imageUrl == null
      ? Padding(
        padding: const EdgeInsets.all(16.0),
        child: SvgPicture.asset(
          AppIcons.profile,
          color: AppColors.textPrimary,
        ),
      )
      : CachedNetworkImage(
        imageUrl: imageUrl!,
        fit: BoxFit.cover,
        progressIndicatorBuilder: (_, __, progress) {
          return ImageLoader(
            progress: progress,
          );
        },
      ),
    ),
  ),
  ),
  titleStyle: const TextStyle(
    fontSize: 18.0,
    fontWeight: FontWeight.w500,
  ),
  title: "Ім'я",
  caption: 'Власник',
  onTap: () => RouteNavigator.pushNamed(
    context,
    EditProfilePage.routeName,
    rootNavigator: true,
  ),
),
SettingsItem(
  leadingIcon: SvgPicture.asset(
    AppIcons.notifications,
    color: AppColors.textPrimary,
  ),
  title: 'Сповідання',
  onTap: () => Navigator.pushNamed(
    context,
    ProfileNotificationsPage.routeName,
  ),
),
),

```

```

SettingsItem(
  leadingIcon: SvgPicture.asset(
    AppIcons.lang,
    color: AppColors.textPrimary,
  ),
  title: 'Мова',
  onTap: () => ModalBottomSheet.show(
    context: context,
    rootNavigation: true,
    title: 'Мова',
    child: const LangItems(),
  ),
),
SettingsItem(
  leadingIcon: SvgPicture.asset(
    AppIcons.card,
    color: AppColors.textPrimary,
  ),
  title: 'Платіжні картки',
  onTap: () => RouteNavigator.pushNamed(
    context,
    PaymentCardsPage.routeName,
    rootNavigator: true,
  ),
),
InkWell(
  onTap: () => _signOut(context),
  child: Container(
    padding: const EdgeInsets.symmetric(
      vertical: 20.0,
    ).copyWith(
      left: 18.0,
      right: 25.0,
    ),
    decoration: const BoxDecoration(
      border: Border(
        bottom: BorderSide(
          color: AppColors.divide,
        ),
      ),
    ),
  ),
  child: const GradientBox(
    child: Text(
      'Вийти',
      style: TextStyle(
        fontSize: 16.0,
        fontWeight: FontWeight.w400,
      ),
    ),
  ),
),
),

```

```

    ),
  )
],
),
);
}
}

```

profile_notifications_page.dart

```

import 'package:flutter/material.dart';
import 'package:flutter_svg/flutter_svg.dart';
import 'package:wowhome_flutter/resources/app_colors.dart';
import 'package:wowhome_flutter/resources/app_icons.dart';
import 'package:wowhome_flutter/widgets/navigation/custom_app_bar.dart';
import 'package:wowhome_flutter/widgets/uncategorized/gradient_box.dart';

import 'widgets/notification_tab.dart';

class ProfileNotificationsPage extends StatefulWidget {
  const ProfileNotificationsPage({Key? key}) : super(key: key);

  static const routeName = '/profile/notifications';

  @override
  State<ProfileNotificationsPage> createState() =>
    _ProfileNotificationsPageState();
}

class _ProfileNotificationsPageState extends State<ProfileNotificationsPage> {
  bool _suspendAll = false;
  bool _paymentReceipt = true;
  bool _dispatchingService = true;
  bool _discussions = true;
  bool _polls = true;
  bool _additionalServices = true;

  //TODO: implement notification change logic with request

  @override
  Widget build(BuildContext context) {
    return SafeArea(
      child: Scaffold(
        appBar: CustomAppBar(
          title: const Text('Повідомлення'),
          onLeading: () => Navigator.pop(context),
        ),

```



```

body: ListView(
  children: [
    const SizedBox(
      height: 50.0,
    ),
    GradientBox(
      child: SvgPicture.asset(
        AppIcons.notifications,
        width: 82.0,
        height: 82.0,
      ),
    ),
    const SizedBox(
      height: 20.0,
    ),
    const Text(
      'Налаштування повідомлень',
      textAlign: TextAlign.center,
      style: TextStyle(
        fontSize: 18.0,
        fontWeight: FontWeight.w500,
      ),
    ),
    const SizedBox(
      height: 40.0,
    ),
    NotificationTab(
      value: _suspendAll,
      title: 'Зупинити все',
      onChange: (bool value) => setState(() {
        _suspendAll = value;
        if (value) {
          _paymentReceipt = !value;
          _dispatchingService = !value;
          _discussions = !value;
          _polls = !value;
          _additionalServices = !value;
        }
      })),
    Container(
      color: AppColors.message,
      height: 26.0,
    ),
    NotificationTab(
      value: _paymentReceipt,
      title: 'Квитанції про оплату',
      bottomBorder: false,
      onChange: (bool value) => setState(() {
        if (value) {

```

```

        _suspendAll = false;
      }
      _paymentReceipt = value;
    })),
  ),
  NotificationTab(
    value: _dispatchingService,
    title: 'Диспетчерська служба',
    topBorder: false,
    onChange: (bool value) => setState(() {
      if (value) {
        _suspendAll = false;
      }
      _dispatchingService = value;
    })),
  ),
  NotificationTab(
    value: _discussions,
    title: 'Обговорення',
    onChange: (bool value) => setState(() {
      _discussions = value;
    })),
  ),
  NotificationTab(
    value: _polls,
    title: 'Толосування',
    onChange: (bool value) => setState(() {
      if (value) {
        _suspendAll = false;
      }
      _polls = value;
    })),
  ),
  NotificationTab(
    value: _additionalServices,
    title: 'Додаткові послуги',
    onChange: (bool value) => setState(() {
      if (value) {
        _suspendAll = false;
      }
      _additionalServices = value;
    })),
  ),
  const SizedBox(
    height: 80.0,
  ),
],
),
),
);

```

```
}
}
```

apartment_case_page.dart

```
import 'package:flutter/material.dart';
import
'package:wowhome_flutter/pages/objects_pages/apartment_sections_page/apartment_sections_p
age.dart';
import 'package:wowhome_flutter/resources/app_colors.dart';
import 'package:wowhome_flutter/resources/app_icons.dart';
import 'package:wowhome_flutter/widgets/items/apartment_info_item.dart';
import 'package:wowhome_flutter/widgets/items/info_row_item.dart';
import 'package:wowhome_flutter/widgets/modals/modal_bottom_sheet.dart';
import 'package:wowhome_flutter/widgets/navigation/custom_sliver_app_bar.dart';
```

```
class ApartmentCasePage extends StatelessWidget {
  const ApartmentCasePage({Key? key}) : super(key: key);

  static const routeName = '/apartment_sections/apartment_case';
```

```
void _showInfo({
  required BuildContext context,
}) {
  ModalBottomSheet.show(
    context: context,
    rootNavigation: true,
    child: const SizedBox(
      height: 350.0,
      child: SingleChildScrollView(
        child: Column(
          crossAxisAlignment: CrossAxisAlignment.start,
          children: [
            Text(
              '06.11.2021 - 07.11.2021',
              style: TextStyle(
                color: AppColors.textSecondary,
                fontWeight: FontWeight.w400,
                fontSize: 13.0,
              ),
            ),
            SizedBox(
              height: 10.0,
            ),
            Text(
```

'Щербинський ліфтобудівний завод (ЩЛЗ) представив три системи розумного ліфта, які дозволяють зробити його зручнішим та безпечнішим: SMART Air Clean,

SMART Air Key та SMART Mirror. Презентація систем відбулася на Міжнародному будівельному форумі 100+ TechnoBuild. SMART Air Clean є вбудованою системою дезінфекції ліфта на основі ультрафіолетових світлодіодів. Вона забезпечує очищення повітря та поверхонь кабіни від бактерій та вірусів на 98%. SMART Air Key — система безконтактного керування ліфтом за допомогою програми на смартфоні.',

```

        style: TextStyle(
          fontWeight: FontWeight.w400,
          fontSize: 15.0,
        ),
      ),
    ],
  ),
),
);
}

@override
Widget build(BuildContext context) {
  return Scaffold(
    body: NestedScrollView(
      headerSliverBuilder: (_, __) {
        return [
          CustomSliverAppBar(
            onLeading: () => Navigator.pop(context),
            onAction: () async {
              Navigator.pop(context);

              Navigator.of(context, rootNavigator: true).pushNamed(
                ApartmentSectionsPage.routeName,
              );
            },
          ),
        ],
      ),
    body: ListView(
      children: [
        const ApartmentInfoItem(
          margin: EdgeInsets.only(
            top: 4.0,
          ),
          icon: AppIcons.build,
          leadingText: 'Секції ',
          tailText: '3',
        ),
        const ApartmentInfoItem(
          icon: AppIcons.apartment,
          leadingText: 'Поверхі',
          tailText: '12',
        ),
      ],
    ),
  );
}

```

```

const SizedBox(
  height: 20.0,
),
Container(
  color: AppColors.message,
  child: Column(
    crossAxisAlignment: CrossAxisAlignment.start,
    children: const [
      Padding(
        padding: EdgeInsets.only(
          left: 16.0,
          top: 20.0,
        ),
        child: Text(
          'Приддомува територія',
          style: TextStyle(
            fontWeight: FontWeight.w700,
            fontSize: 18.0,
          ),
        ),
      ),
      InfoRowItem(
        leadingText: 'Кадастровий номер',
        tailText: '6310136900:01:007:0013',
      ),
      InfoRowItem(
        leadingText: 'Площа землі',
        tailText: '42 м2',
        bottomBorder: false,
      ),
    ],
  ),
),
const SizedBox(
  height: 20.0,
),
Container(
  color: AppColors.message,
  child: Column(
    crossAxisAlignment: CrossAxisAlignment.start,
    children: const [
      Padding(
        padding: EdgeInsets.only(
          left: 16.0,
          top: 20.0,
        ),
        child: Text(
          'Тариф',
          style: TextStyle(
            fontWeight: FontWeight.w700,

```

```

        fontSize: 18.0,
      ),
    ),
  ),
  InfoRowItem(
    leadingText: 'Прибирання, Охорона',
    tailText: '150 грн/мес.',
    bottomBorder: false,
  ),
],
),
),
const SizedBox(
  height: 20.0,
),
Container(
  color: AppColors.message,
  child: Column(
    crossAxisAlignment: CrossAxisAlignment.start,
    children: [
      const Padding(
        padding: EdgeInsets.only(
          left: 16.0,
          top: 20.0,
        ),
        child: Text(
          'Події',
          style: TextStyle(
            fontWeight: FontWeight.w700,
            fontSize: 18.0,
          ),
        ),
      ),
    ],
  ),
  _EventItem(
    date: '06.11.2021 - 07.11.2021',
    title:
      'ЖК "МійДім" - переможець IX всеукраїнського конкурсу',
    onTap: () => _showInfo(
      context: context,
    ),
  ),
  context: context,
),
),
const SizedBox(
  height: 40.0,
),

```

```

    ],
  ),
),
);
}
}

```

```

class _EventItem extends StatelessWidget {
  const _EventItem({
    Key? key,
    this.bottomBorder = true,
    required this.date,
    required this.title,
    required this.onTap,
  }) : super(key: key);

  final bool bottomBorder;
  final String date;
  final String title;
  final void Function() onTap;

  @override
  Widget build(BuildContext context) {
    return Material(
      color: AppColors.message,
      child: InkWell(
        onTap: onTap,
        child: Container(
          height: 90.0,
          padding: const EdgeInsets.symmetric(
            vertical: 20.0,
            horizontal: 16.0,
          ),
          decoration: BoxDecoration(
            border: Border(
              bottom: bottomBorder
                ? const BorderSide(
                    color: AppColors.primary,
                )
                : BorderSide.none,
            ),
          ),
          child: Column(
            crossAxisAlignment: CrossAxisAlignment.start,
            children: [
              Text(
                date,
                style: const TextStyle(
                  color: AppColors.textSecondary,
                  fontWeight: FontWeight.w400,

```

```

        fontSize: 13.0,
      ),
    ),
    const SizedBox(
      height: 10.0,
    ),
    Expanded(
      child: Text(
        title,
        style: const TextStyle(
          fontWeight: FontWeight.w500,
          fontSize: 15.0,
        ),
        maxLines: 1,
        overflow: TextOverflow.ellipsis,
      ),
    ),
  ],
),
),
);
}
}

```

news_page.dart

```

import 'package:flutter/material.dart';
import
'package:wowhome_flutter/pages/news_pages/news_details_page/news_details_page.dart';
import 'package:wowhome_flutter/resources/app_colors.dart';
import 'package:wowhome_flutter/widgets/items/news_item.dart';
import 'package:wowhome_flutter/widgets/navigation/custom_app_bar.dart';
import 'package:wowhome_flutter/widgets/uncategorized/image_scaffold.dart';

class NewsPage extends StatelessWidget {
  const NewsPage({Key? key}) : super(key: key);

  static const routeName = '/news';

  @override
  Widget build(BuildContext context) {
    return ImageScaffold(
      appBar: const CustomAppBar(
        title: Text('Новини'),
      ),
      body: ListView.separated(

```



```

padding: const EdgeInsets.symmetric(
  vertical: 20.0,
  horizontal: 16.0,
),
itemCount: 10,
separatorBuilder: (_, __) {
  return const Padding(
    padding: EdgeInsets.only(
      top: 10.0,
      bottom: 5.0,
    ),
    child: Divider(
      color: AppColors.border,
    ),
  );
},
itemBuilder: (_, int i) {
  return NewsItem(
    onTap: () => Navigator.pushNamed(
      context,
      NewsDetailPage.routeName,
      arguments: 1,
    ),
  );
},
),
);
}
}

```

ДОДАТОК Г – ГРАФІЧНА ЧАСТИНА

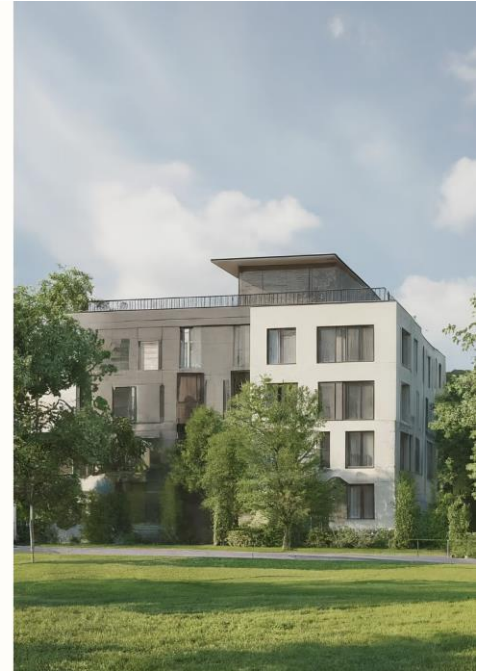
Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра програмного забезпечення

БАКАЛАВРСЬКА ДИПЛОМНА РОБОТА

3 теми: Розробка програмного мобільного застосунку для мешканців ОСББ

Виконав:
Студент групи ЗПІ-20Б Ковальчук М.В.

Науковий керівник:
к.т.н., доц. каф. ПЗ Ракитянська Г.Б.



Мета, дослідження та предмет роботи

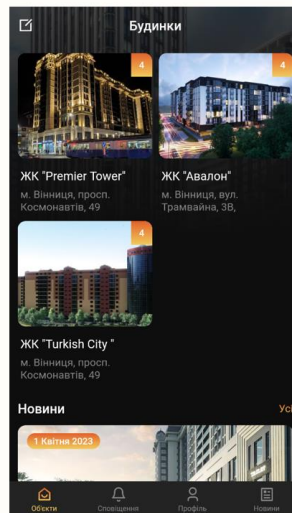
Мета бакалаврської дипломної роботи – Основною метою цієї роботи є розробка моделей управління ОСББ шляхом впровадження мобільного застосунка, спрямованого на вирішення проблем, що виникають у мешканців та управлінських комітетів ОСББ.

Об'єкт дослідження – процес розробки програмного застосунку для мешканці ОСББ.

Предмет дослідження – методи і засоби розробки та реалізація мобільного застосунку для мешканців ОСББ.



Розробка інтерфейсу



Інтерфейс мобільного застосунку для ОСББ ретельно розроблений з урахуванням принципів простоти, зручності та естетики. Плитковий дизайн, заснований на базових геометричних формах, забезпечує чітку та динамічну структуру, що динамічно адаптується до вмісту. Зрозумілий дизайн та послідовне розташування елементів керування створюють інтуїтивно зрозумілий інтерфейс, який легко використовувати навіть не досвідченим користувачам.

Для підкреслення елегантності та сучасного вигляду застосунку обрана темна палітра кольорів. Цей вибір робить інтерфейс естетично привабливим, не перевантажуючи візуально, та забезпечує комфортний перегляд інформації навіть у темних умовах.

Концептуальна модель системи

Запропоновано модель, яка оцінює вплив таких факторів функціональності:

- інтеграція з системами відеоспостереження,
- можливість голосування щодо важливих питань житлового комплексу,
- доступ до різноманітних сервісів, включаючи послуги електрика, сантехніка, прибирання, доставки,
- передача показників лічильника,
- служба підтримки,
- перегляд та оплата квитанцій онлайн,
- подання заявок на ремонтні роботи та інші послуги,
- отримання сповіщень про важливі події та новини ОСББ,
- доступ до документів та протоколів зборів ОСББ.
- можливість зворотного зв'язку з адміністрацією ОСББ.

Ефективність управління ОСББ

Вартість послуг

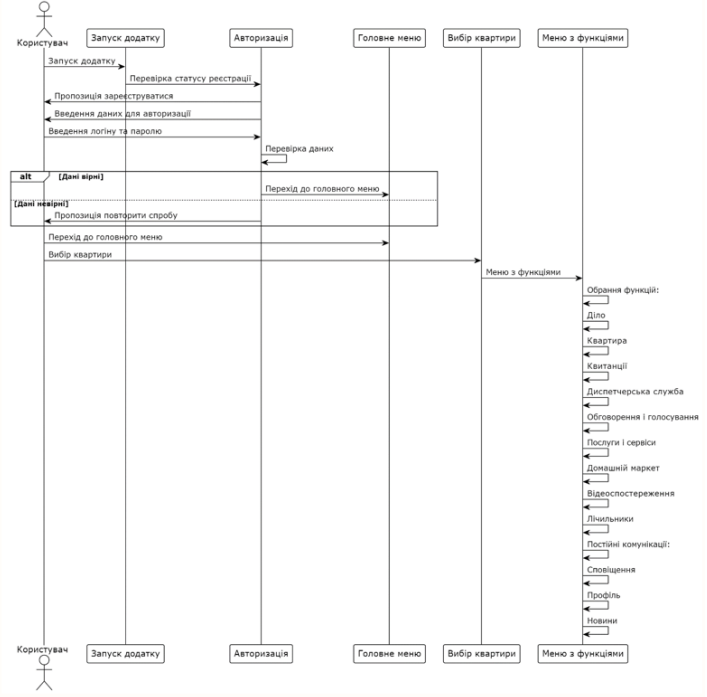
Розробка моделі системи

При запуску застосунку система перевіряє, чи зареєстрований користувач. Зареєстровані користувачі переходять до авторизації, де вводять логін та пароль. Система перевіряє дані: вірні дані ведуть до головного меню, невірні - до повторної спроби. Після успішної авторизації користувач бачить головне меню з доступними функціями. Він обирає свою квартиру зі списку, після чого відкривається меню з функціями, доступними для обраної квартири. Користувач може вибрати:

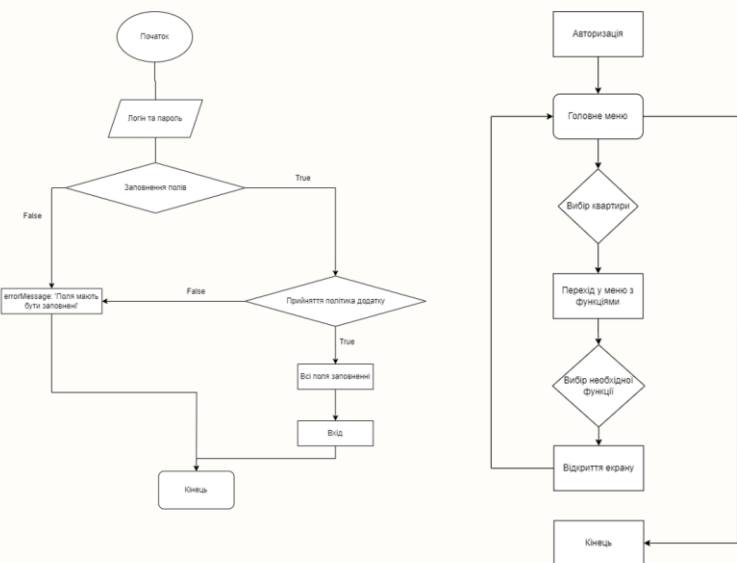
- Діло.
- Квартира.
- Квитанції.
- Диспетчерська служба.
- Обговорення і голосування.
- Послуги і сервіси.
- Домашній маркет.
- Відеоспостереження.
- Лічильники.

У нижній частині екрана знаходяться кнопки для швидкого доступу:

- Сповіщення
- Профіль
- Новини



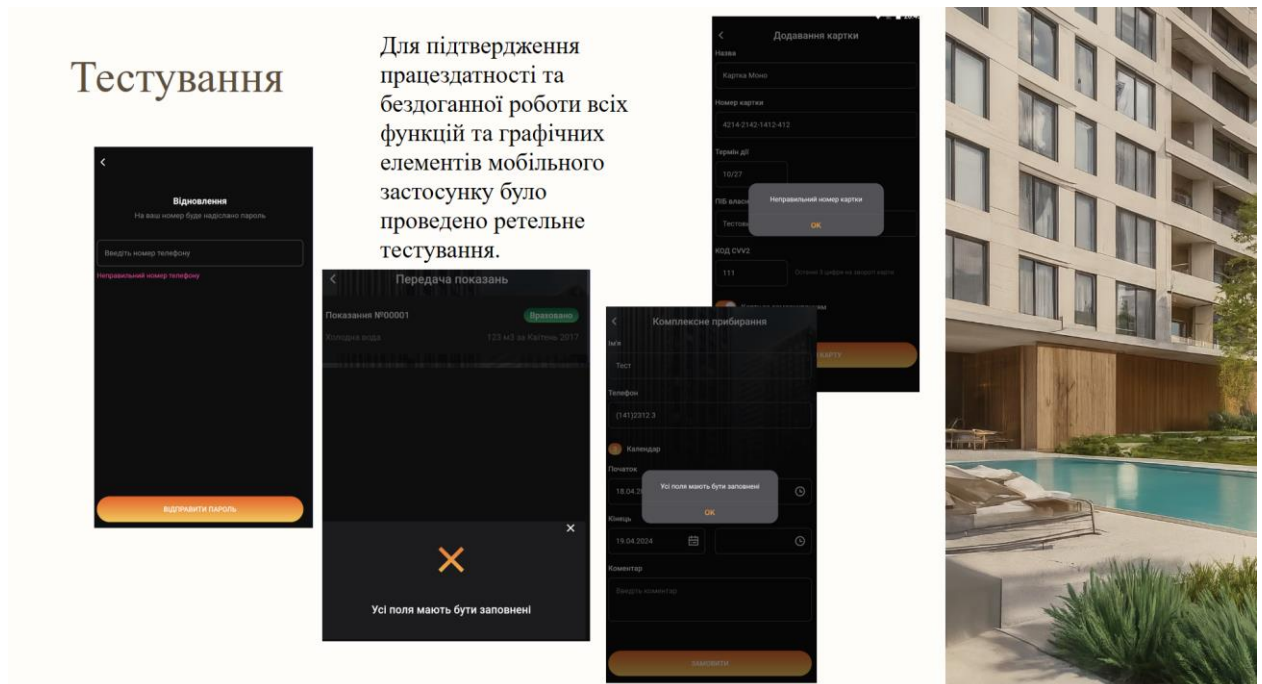
Створення алгоритму роботи застосунку



Аналіз наданих блок-схем та алгоритмів дає чітке уявлення про те, як користувач взаємодіє з мобільним застосунком. Ці алгоритми розроблені з акцентом на максимальну зручність користувача та забезпечують інтуїтивно зрозумілий інтерфейс.

Основні переваги мови програмування

1. **Кросплатформенність:** Можливість розробки одного мобільного застосунку для iOS та Android без необхідності писати окремий код для кожної платформи.
2. **Продуктивність:** Використання власного механізму рендерингу Flutter забезпечує плавну анімацію та швидкий відгук інтерфейсу користувача.
Компіляція Dart в машинний код гарантує високу продуктивність на рівні нативних застосунків.
3. **Розширені можливості для користувацького інтерфейсу:** Вбудовані віджети та широкі можливості кастомізації Flutter дозволяють створювати багаті та складні інтерфейси.
Це робить мобільний застосунок зручним та інтуїтивно зрозумілим для користувачів.
4. **Активна спільнота та підтримка:** Зростаюча спільнота розробників Flutter пропонує багато ресурсів, документації та прикладів для полегшення розробки.
Google активно підтримує та розвиває Flutter, гарантуючи його стабільність та майбутнє.
5. **Простота навчання:** Dart має чіткий та зрозумілий синтаксис, що робить його простим для вивчення.
Розробники з досвідом роботи з іншими мовами програмування можуть швидко освоїти Dart.
Це дозволяє залучати до проекту нових розробників без значних витрат часу на навчання.
6. **Інструменти розробки:** Інтеграція Flutter з популярними середовищами розробки (Visual Studio Code, Android Studio) забезпечує зручність розробки, налагодження та тестування.
Гаряче перезавантаження (hot reload) дозволяє миттєво бачити зміни в коді без перезапуску застосунку.



Висновки

Мобільний застосунок для мешканців ОСББ – це не просто зручний інструмент, а й потужний двигун для підвищення ефективності управління та зміцнення зв'язків всередині житлової спільноти. Завдяки цьому застосунку мешканці отримують доступ до актуальної інформації, маси послуг та можуть брати участь в голосуваннях та спільно вирішувати важливі питання. Всі поставлені цілі роботи були успішно досягнуті.

