

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
УМАНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ**

Факультет економіки і підприємництва

Кафедра інформаційних технологій

Допущено до захисту

" _____ " _____ 2025 р.

Зав. кафедри, к.т.н., доцент

_____ Лішук Р.І.

**КВАЛІФІКАЦІЙНА РОБОТА
БАКАЛАВРА**

**Мобільний застосунок для забезпечення діяльності
благодійної організації**

Виконав – студент 4 курсу, групи 41-кн
освітньої програми "Комп'ютерні науки"
спеціальності 122 Комп'ютерні науки

Борисевич Богдан Віталійович

Науковий керівник – д.т.н., професор

Кучерук Володимир Юрійович

Умань – 2025

Реферат
на кваліфікаційну роботу «Мобільний застосунок для забезпечення
діяльності благодійної організації»

студента факультету економіки та підприємництва

Уманського національного університету

Борисевича Богдана Віталійовича

Структура та обсяг роботи - пояснювальна записка до кваліфікаційної роботи бакалавра складається з 3 розділів, містить 65 сторінок, 24 рисунка, 1 додаток, 22 літературних джерела.

У кваліфікаційній роботі бакалавра розроблено модулі мобільного застосунку, призначеного для підтримки діяльності благодійних організацій. Для реалізації проєкту використовувалося середовище розробки Visual Studio Code.

У процесі виконання роботи проведено аналіз сучасного стану проблеми, розглянуто основні аналоги подібних мобільних систем, виявлено їхні недоліки та здійснено порівняння з розробленим застосунком.

У ході аналізу технологій розробки було обґрунтовано вибір мови програмування JavaScript, технології React Native, фреймворку Expo та мови запитів GraphQL. У межах проєкту виконано такі основні етапи: розроблено модуль авторизації користувачів для забезпечення безпеки та доступу до персоналізованих функцій; створено модуль для перегляду інформації про волонтерські заходи та проєкти; реалізовано модуль зворотного зв'язку та функціонал надання фінансової допомоги; спроектовано зручний та адаптивний графічний інтерфейс користувача; виконано тестування програмного забезпечення відповідно до поставлених вимог.

Також розроблено навігаційні схеми застосунку та побудовано модель системи за допомогою UML-діаграм.

Результати тестування підтвердили повну працездатність мобільного застосунку та його відповідність технічному завданню. Додатково створено інструкцію користувача.

Анотація

У кваліфікаційній роботі бакалавра проведено аналіз стану мобільних систем для благодійних організацій, доведено доцільність розробки власної програмної реалізації, зважаючи на порівняльний аналіз аналогів.

У ході виконання роботи розроблено мобільний застосунок для підтримки діяльності благодійних організацій, що включає в себе модулі для оперування, перегляду, пошуку та фільтрації волонтерських проєктів. Додаток призначений для збільшення ефективності діяльності благодійних організацій.

Мобільний застосунок розроблено з використанням технології React Native та мови програмування JavaScript та середовища Visual Studio Code, для зберігання даних використано сервіс Hygraph із мовою запитів GraphQL.

Ключові слова: благодійні організації, мобільний застосунок, фреймворк.

Annotation

In the bachelor's qualification work, an analysis of the state of mobile systems for charitable organizations was conducted, the feasibility of developing our own software implementation was proven, taking into account a comparative analysis of analogues.

During the work, a mobile application was developed to support the activities of charitable organizations, which includes modules for operating, viewing, searching and filtering volunteer projects. The application is designed to increase the efficiency of charitable organizations.

The mobile application was developed using React Native technology and the JavaScript programming language and the Visual Studio Code environment, the Hygraph service with the GraphQL query language was used to store data.

Keywords: charitable organisations, mobile application, framework.

ЗМІСТ

Перелік умовних позначень, скорочень і термінів	7
ВСТУП.....	8
РОЗДІЛ 1. АНАЛІЗ РОЗВИТКУ МОБІЛЬНИХ СИСТЕМ ДЛЯ ПІДТРИМКИ БЛАГОДІЙНИХ ОРГАНІЗАЦІЙ ТА ПОСТАНОВКА ЗАДАЧ РОЗРОБКИ	10
1.1 Аналіз стану мобільних систем створених для підтримки благодійних організацій	10
1.2 Порівняльний аналіз аналогів	11
1.3 Аналіз методів розв’язання задачі	14
РОЗДІЛ 2. РОЗРОБКА МОДЕЛЕЙ ТА АЛГОРИТМІВ ПРОГРАМНОГО ПРОДУКТУ	16
2.1 Аналіз вхідних даних системи	16
2.2 Розробка моделі та структури системи	16
2.3 Розробка алгоритму захисту даних користувачів	20
РОЗДІЛ 3. РОЗРОБКА ПРОГРАМНИХ МОДУЛІВ РЕАЛІЗАЦІЇ МОБІЛЬНОГО ЗАСТОСУНКУ	22
3.1 Варіантний аналіз і обґрунтування вибору засобів для реалізації програмного забезпечення	22
3.2 Розробка модулів авторизації користувачів та перегляду інформації про волонтерські заходи	24
3.3 Тестування застосунку	39
ВИСНОВКИ	45
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	46
ДОДАТОК А Лістинг програмного коду застосунку.....	48

Перелік умовних позначень, скорочень і термінів

CMS	Content Management System
API	Application Programming Interface
SDK	Software Development Kit
URI	Uniform Resource Identifier

ВСТУП

Обґрунтування вибору теми дослідження. У сучасному світі діяльність благодійних організацій вимагає ефективних інструментів для організації, координації, комунікації та підвищення результативності виконання завдань. Мобільні застосунки відіграють у цьому ключову роль, забезпечуючи швидкий і зручний доступ до ресурсів та інформації. Водночас більшість наявних цифрових рішень не відповідають специфічним потребам благодійних структур, оскільки не враховують особливості їхньої роботи, такі як взаємодія з волонтерами, спонсорами та цільовими групами.

У відповідь на ці виклики актуальним постає завдання створення мобільного застосунку, спеціально орієнтованого на потреби благодійних організацій. Такий інструмент може значно полегшити організацію волонтерських ініціатив, автоматизувати процеси збору та обробки даних, оптимізувати комунікацію і підвищити рівень залученості учасників [1].

Розробка такого застосунку є важливим кроком на шляху до вдосконалення механізмів управління ресурсами, координації дій волонтерів і реалізації проєктів, що у підсумку сприятиме ефективнішій роботі благодійних організацій та стимулюватиме розвиток волонтерського руху. Таким чином, тема дослідження є не лише актуальною, а й соціально значущою, адже спрямована на посилення позитивного впливу благодійності на суспільство.

Мета та завдання дослідження - підвищити ефективність комунікації та координації між учасниками благодійного процесу шляхом розробки мобільного застосунку, що забезпечить зручний доступ до актуальної інформації та ресурсів.

Основні завдання дослідження:

- розробка бази даних для зберігання інформації про користувачів і благодійні проєкти;
- створення інтерфейсу користувача з орієнтацією на зручність та інтуїтивність;
- реалізація модуля авторизації для забезпечення безпеки та доступу до персоналізованих функцій;

- розробка модуля перегляду подій, проєктів та волонтерських ініціатив;
- створення модуля для надання фінансової допомоги через застосунок.

Об'єкт і предмет дослідження. Об'єкт дослідження - процес розробки мобільного застосунку, орієнтованого на потреби благодійних організацій. Предмет дослідження - методи, інструменти та технології створення мобільних застосунків для підтримки благодійної діяльності.

Методи дослідження. У процесі роботи застосовувались такі методи: розробка кросплатформного застосунку для Android та iOS з використанням React Native і Expo; побудова архітектури застосунку з підтримкою масштабованості; реалізація взаємодії з серверною частиною через CMS Hygraph для ефективного керування контентом; проектування графічного інтерфейсу за допомогою Figma для створення адаптивного та зручного UI; тестування функціональності для забезпечення стабільності та надійності застосунку.

Практична цінність отриманих результатів. Практична значущість дослідження полягає у створенні дієвого інструменту - мобільного застосунку, що може бути впроваджений у діяльність реальних благодійних організацій. Розроблене програмне забезпечення здатне оптимізувати внутрішні процеси, полегшити координацію дій, а також покращити взаємодію між організаціями, волонтерами та спонсорами.

РОЗДІЛ 1. АНАЛІЗ РОЗВИТКУ МОБІЛЬНИХ СИСТЕМ ДЛЯ ПІДТРИМКИ БЛАГОДІЙНИХ ОРГАНІЗАЦІЙ ТА ПОСТАНОВКА ЗАДАЧ РОЗРОБКИ

1.1 Аналіз стану мобільних систем створених для підтримки благодійних організацій

У сучасному світі зростає популярність мобільних застосунків, орієнтованих на покращення діяльності благодійних організацій. Це свідчить про те, що такі організації цінують зручність, доступність і швидкість у виконанні своїх завдань. Мобільні застосунки надають волонтерам можливість отримувати інформацію та працювати з будь-яких пристроїв з доступом до Інтернету, що є надзвичайно важливим у сучасних умовах.

На сьогодні існує низка програмних рішень для підтримки благодійної діяльності, але більшість із них має обмежений функціонал і не враховує всі специфічні потреби організацій. Часто застосунки мають складний або неінтуїтивний інтерфейс, що ускладнює їхнє використання. Крім того, такі програми недостатньо орієнтовані на різноманітність вимог благодійних організацій та їхніх спонсорів.

Отже, розвиток інтегрованих і функціонально насичених мобільних застосунків є важливим для ефективної підтримки діяльності благодійних організацій. Вони мають забезпечувати широкий спектр сервісів, включаючи управління проєктами, комунікацію з волонтерами та збір необхідної інформації.

Існуючі рішення демонструють значні недоліки: обмежений функціонал, складність інтерфейсу, відсутність повноцінної взаємодії між користувачами, а також недостатній рівень захисту персональних даних. Часто застосунки обмежуються лише передачею базової інформації, не забезпечуючи ефективної співпраці.

Для подолання цих проблем необхідно розробити більш комплексний і функціональний мобільний застосунок, який полегшить роботу організацій і підвищить їх ефективність. Застосунок має бути простим у користуванні,

відповідати потребам організацій та гарантувати безпеку і конфіденційність особистих даних користувачів.

Таким чином, актуальним є створення власного мобільного застосунку, що сприятиме вдосконаленню діяльності благодійних організацій, покращенню комунікації та співпраці між учасниками, а також захисту їхніх персональних даних.

1.2 Порівняльний аналіз аналогів

Під час аналізу аналогів розроблюваного програмного застосунку було встановлено, що подібних рішень наразі небагато. В якості найбільш релевантних були обрані веб-сайти «СпівДія», «VolunteerUP» та «volunter.org». «VolunteerUP» - це веб-платформа, створена для полегшення волонтерської діяльності [4]. Головна мета сайту - надати користувачам широкий доступ до різноманітних волонтерських можливостей та сприяти залученню нових волонтерів до соціальних і благодійних проєктів.

Основні функції «VolunteerUP» включають:

- Можливість швидко знаходити події, проєкти та акції, які відповідають інтересам і можливостям волонтерів.
- Реєстрацію на бажані заходи безпосередньо через платформу, що значно спрощує організацію та участь у волонтерських подіях.

Інтерфейс веб-застосунку «VolunteerUP» наведено на рис. 1.1.



Рисунок 1.1. Веб-застосунок «VolunteerUP»

Організатори мають можливість публікувати оголошення про свої події, формувати команди волонтерів та координувати їхню діяльність. Волонтери, у свою чергу, можуть спілкуватися між собою та ділитися досвідом.

«СпівДія» - це онлайн-платформа, створена для залучення волонтерів до соціальних і культурних проєктів [5]. Головна мета «СпівДії» - допомогти кожному волонтеру знайти проєкт, що відповідає його інтересам і можливостям. Інтерфейс веб-застосунку наведено на рис. 1.2.

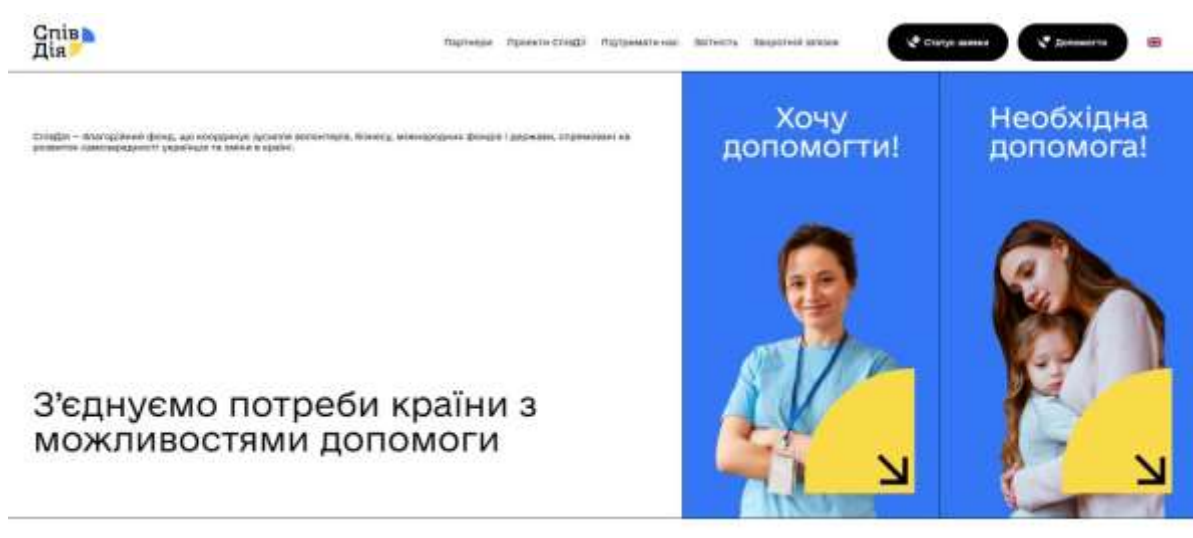


Рисунок 1.2. Веб-застосунок «СпівДія»

Основні функції «СпівДія»:

Платформа пропонує широкий вибір проєктів для участі - від благодійних акцій до екологічних заходів і культурних подій. Волонтери можуть швидко зареєструватися на обрані проєкти та отримати всю необхідну інформацію щодо участі.

Для обговорення ідей, обміну досвідом і координації діяльності передбачені спеціальні форуми та чати. Волонтери також можуть відстежувати свій внесок у різні проєкти та отримувати звіти про результати своєї роботи.

«ВолонтерОрг» - це онлайн-платформа, яка об'єднує волонтерів і організації, що потребують допомоги [6]. Головна мета платформи - сприяти залученню волонтерів до різноманітних соціальних та благодійних ініціатив.

Інтерфейс веб-застосунку наведено на рис. 1.3.

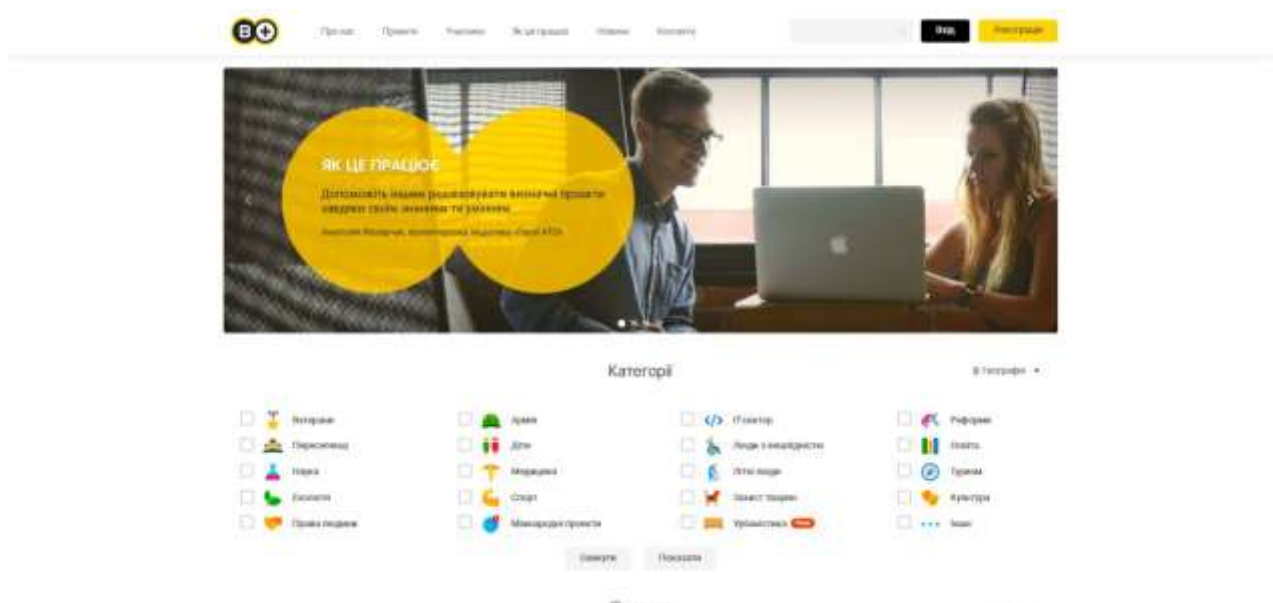


Рисунок 1.3. Веб-застосунок «ВолонтерОрг»

Основні функції платформи «Волонтер.орг»:

- Волонтери можуть шукати можливості для участі за різними критеріями, такими як тематика, регіон або тип діяльності.
- Можливість реєстрації на обрані проекти з отриманням детальної інформації про участь.
- Організатори можуть публікувати інформацію про свої проекти та запрошувати волонтерів долучатися до них.
- Волонтери мають змогу спілкуватися з організаторами та іншими учасниками через спеціальні чати та форуми, обговорювати плани й обмінюватися досвідом.

Для узагальнення порівняння аналогічних платформ наведено табл. 1.1.

Згідно з таблицею порівняльних характеристик, розробка власного мобільного застосунку для підтримки діяльності благодійних організацій є виправданою та доцільною. Створений продукт зможе усунути недоліки існуючих рішень і надати розширені можливості для користувачів. Таким чином, розробка такого застосунку відкриває широкі перспективи для покращення благодійної діяльності, зокрема волонтерства, за рахунок ефективної організації

проектної роботи та впровадження додаткових функцій.

Таблиця 1.1 – Порівняльна характеристика аналогів

Критерії	VolunteerUP	СпівДія	ВолонтерОрг	Власна реалізація
Можливість авторизації	+	+	+	+
Розбиття проектів на категорії	-	-	+	+
Відстеження прогресу	-	+	-	+
Можливість разової фінансової підтримки	-	+	+	+
Інтеграція з соц. мережами	-	-	-	+

1.3 Аналіз методів розв’язання задачі

Розробка мобільного застосунку для підтримки діяльності благодійних організацій потребує ретельного аналізу та вибору ефективних методів реалізації. Цей процес можна умовно поділити на кілька основних етапів:

1. Аналіз існуючих рішень та дослідження літератури. Вивчаються доступні інтерактивні системи для волонтерів, наукові публікації та інші джерела, що стосуються розробки подібних застосунків.

2. Дослідження потреб користувачів. Проводиться аналіз типових сценаріїв використання, визначаються ключові функціональні потреби та очікування цільової аудиторії. Для цього можуть застосовуватися опитування та фокус-групи серед представників спільнот.

3. Вибір технологічної бази. Обираються платформи розробки (Android, iOS, кросплатформні рішення), мови програмування, бази даних та інші технічні інструменти, з урахуванням можливості масштабування та підтримки.

4. Проектування архітектури застосунку. Формується структура, модулі та компоненти програми, з особливим акцентом на створенні зручного і інтуїтивного інтерфейсу, що відповідає потребам користувачів.

5. Розробка та програмування. Реалізується функціонал і дизайн застосунку відповідно до встановлених вимог.

6. Тестування. Проводиться комплексна перевірка роботи застосунку на різних пристроях і платформах для виявлення та усунення помилок.

7. Впровадження та підтримка. Застосунок запускається серед волонтерської спільноти, забезпечується підтримка користувачів, регулярні оновлення та вирішення технічних питань.

Для успішної розробки застосунку необхідно розв'язати такі завдання:

- Аналіз потреб користувачів: Вивчити очікування волонтерів і організацій; Визначити ключові функції та вимоги до застосунку.
- Визначення пріоритетів функціональності: Сформулювати основні функції; Ранжувати їх за важливістю для користувачів.
- Вибір платформи розробки: Обрати відповідну платформу з урахуванням аудиторії та технічних особливостей.
- Проектування структури застосунку: Спроекувати модулі, забезпечити зручність і оптимальність роботи.
- Реалізація функціоналу: Написати якісний програмний код, що відповідає вимогам.
- Розробка інтерфейсу: Створити інтуїтивно зрозумілий дизайн, орієнтований на користувачів.
- Тестування: Забезпечити стабільність і ефективність роботи, усунути помилки.

Виконання цих кроків дозволить створити якісний мобільний застосунок, який максимально відповідає потребам благодійних організацій і сприятиме їх ефективній діяльності.

РОЗДІЛ 2. РОЗРОБКА МОДЕЛЕЙ ТА АЛГОРИТМІВ ПРОГРАМНОГО ПРОДУКТУ

2.1 Аналіз вхідних даних системи

Аналіз вхідних даних системи є ключовим етапом у розробці мобільного застосунку для підтримки діяльності благодійних організацій. Цей процес полягає у детальному вивченні та розумінні різних типів інформації, які надходять до системи.

Особливу увагу слід приділити особистій інформації користувачів, зокрема даним їхніх «Google» акаунтів. Ця інформація не лише дозволяє ідентифікувати користувача, а й забезпечує можливість безпосереднього зворотного зв'язку між користувачем і розробником.

Також важливо враховувати дані про пропозиції та скарги користувачів. Ця інформація має велике значення, оскільки допомагає оперативно виявляти проблеми та недоліки застосунку. Ретельний аналіз таких відгуків дозволяє не тільки швидко реагувати на виникаючі питання, а й вдосконалювати продукт, підвищуючи рівень задоволеності користувачів.

Таким чином, аналіз вхідних даних сприяє кращому розумінню потреб користувачів, оптимізації використання ресурсів та підвищенню якості застосунку. Цей етап є надзвичайно важливим для забезпечення успішності та ефективності роботи програмного забезпечення.

2.2 Розробка моделі та структури системи

Для кращого розуміння роботи модулів мобільного застосунку, призначеного для підтримки діяльності благодійних організацій, було прийнято рішення розробити модель системи за допомогою UML-діаграм [7].

Діаграма діяльності - це графічне зображення послідовності дій або процесів у системі [8]. Вона показує, як об'єкти взаємодіють у межах певного процесу або діяльності. Кожна дія представлена у вигляді прямокутника з назвою операції або функції, що виконується.

Головна перевага діаграм діяльності полягає у їхній зрозумілості та простоті сприйняття, що дає змогу швидко відстежити послідовність подій. Такі діаграми широко застосовуються під час аналізу та проєктування бізнес-процесів, системного аналізу та управління проєктами.

Використання діаграм діяльності допомагає виявити можливі проблеми у процесах та покращити їхню організацію.

Діаграма діяльності мобільного застосунку для підтримки благодійних організацій наведена на рис. 2.1.

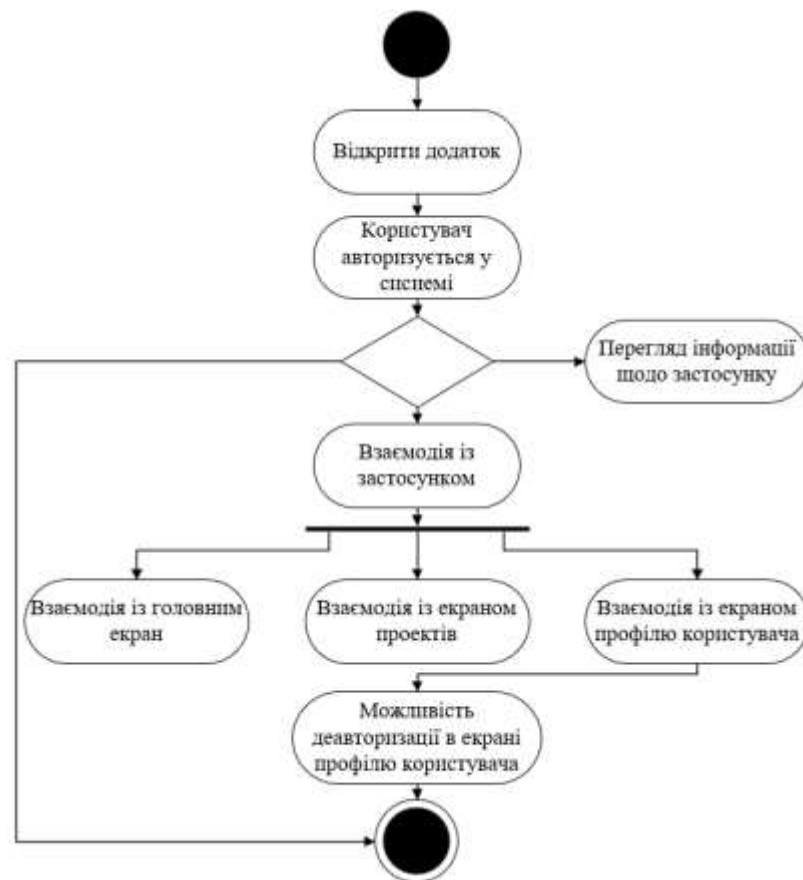


Рисунок 2.1. Діаграма діяльності мобільного застосунку

Відповідно до наведеної діаграми, користувач для початку взаємодії з мобільним застосунком повинен пройти авторизацію в системі. У разі відсутності авторизації користувач може лише переглянути інформацію про застосунок або вийти з нього.

Після успішної авторизації користувач переходить на головний екран застосунку, де йому пропонується навігаційне меню для зручного ознайомлення з основними розділами додатку.

Далі користувач має можливість вибрати потрібний екран: «Головний екран», «Екран проектів» або «Профіль користувача». Після завершення роботи з функціями застосунку користувач може деавторизуватися на екрані профілю та завершити сеанс роботи з додатком.

Діаграма діяльності була створена за допомогою програмного забезпечення Microsoft Visio [9].

Для розробки мобільного застосунку сформовано його структуру, поділену на загальні модулі, що визначають послідовність дій користувача в системі. Особливістю розробки є використання CMS-сервісу Hygraph для завантаження контенту та інтеграція мобільного застосунку для комунікації зі спонсорами, що підвищує рівень взаємодії між благодійними організаціями та їхніми партнерами.

Отже, користувач спочатку проходить авторизацію або переглядає інформацію про застосунок, після чого потрапляє на головний екран. На головному екрані доступні рекламні інтеграції, категорії проектів, список останніх проектів, а також функція пошуку проектів за назвою.

Для зручності користувача передбачено навігаційне меню, яке складається з трьох основних сторінок: «Головна», «Проекти» та «Профіль». Якщо користувач перебуває на головному екрані, навігаційне меню завжди залишається доступним.

На головному екрані також розміщено корисні функції - слайдер з рекламними інтеграціями, категорії та останні проекти, що дозволяє швидко знаходити потрібну інформацію та комфортно користуватися застосунком.

Переходячи на екран «Проекти», користувач отримує доступ до всіх створених волонтерських ініціатив та відповідного функціоналу для їх перегляду.

На екрані «Профіль» користувач може переглядати свої дані та фото, користуватися навігаційним меню, надсилати відгуки та побажання для покращення роботи застосунку, а також деавторизуватися в системі.

Також створено діаграму, що відображає структуру мобільного застосунку для підтримки діяльності благодійних організацій. Цю структуру наведено на рис. 2.2.

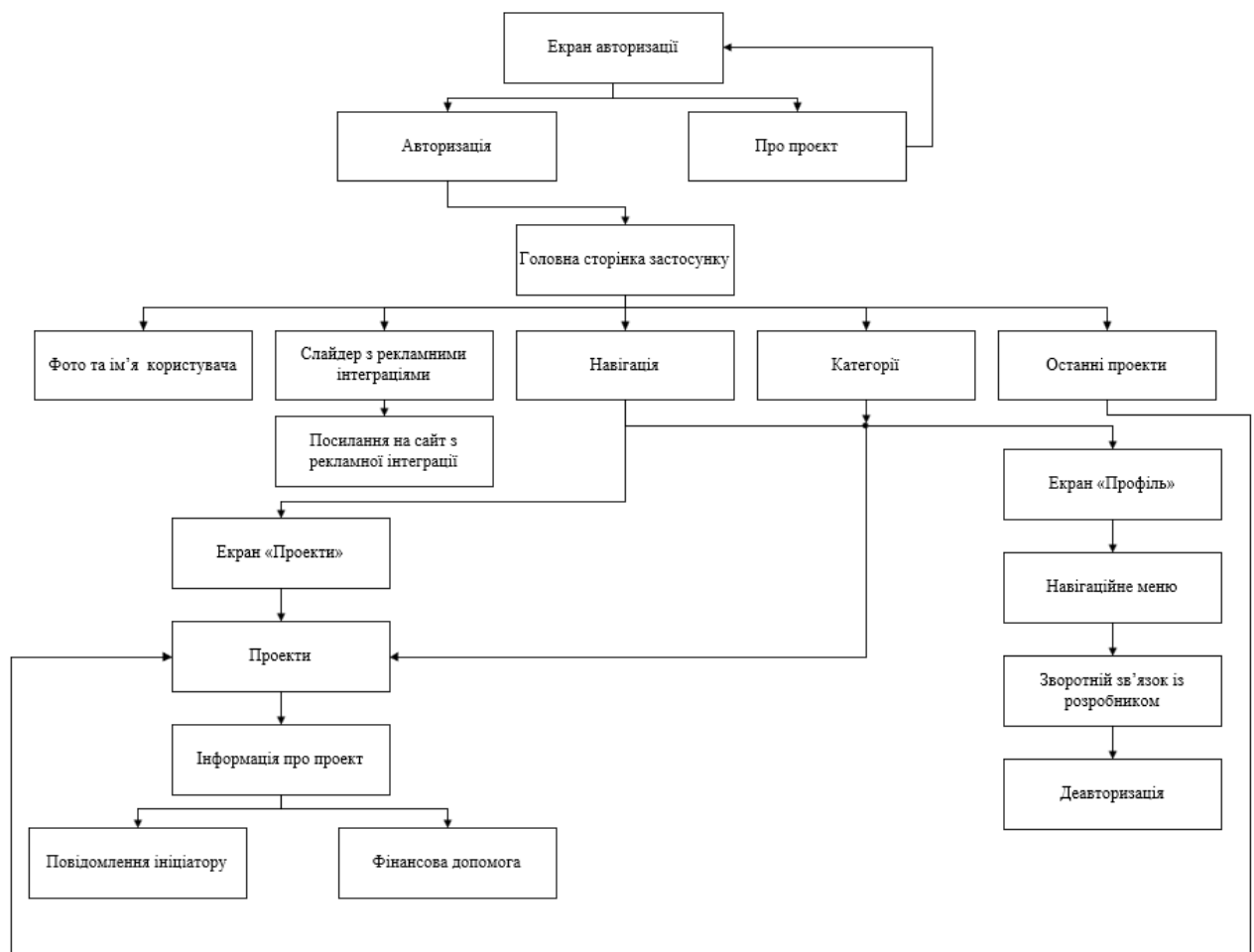


Рисунок 2.2. Структура мобільного застосунку

Процес створення архітектури мобільного застосунку включає визначення його структури та послідовності переходів між екранами. Формування чіткої структури є ключовим етапом розробки, який забезпечує зручну та інтуїтивно зрозумілу навігацію для користувачів.

2.3 Розробка алгоритму захисту даних користувачів

Для забезпечення конфіденційності та безпеки даних користувачів у мобільних додатках був розроблений алгоритм захисту на основі бібліотеки Exposed-secure-store. Він складається з двох ключових компонентів: SecureStore та Crypto, які спільно відповідають за шифрування та безпечне зберігання конфіденційної інформації.

SecureStore забезпечує зберігання даних у форматі пар ключ-значення безпосередньо на мобільному пристрої, організовуючи зручний доступ до них у вигляді рядків. Компонент Crypto реалізує шифрування та розшифрування цих даних за допомогою асиметричного алгоритму RSA, що гарантує високий рівень безпеки.

Алгоритм роботи має наступний порядок:

1. Під час запуску додатку генерується пара ключів RSA - публічний і приватний. Публічний ключ використовується для шифрування, а приватний - для розшифрування даних.
2. Конфіденційні дані перед збереженням у SecureStore шифруються публічним ключем.
3. Зашифровані дані зберігаються у SecureStore у вигляді пар ключ-значення, що забезпечує їх безпеку на пристрої.
4. При запиті даних вони зчитуються із SecureStore у зашифрованому вигляді.
5. Зчитані дані розшифровуються за допомогою приватного ключа.
6. Розшифровані дані передаються додатку для подальшого використання.

Блок-схема алгоритму захисту даних користувачів наведена на рис. 2.3.

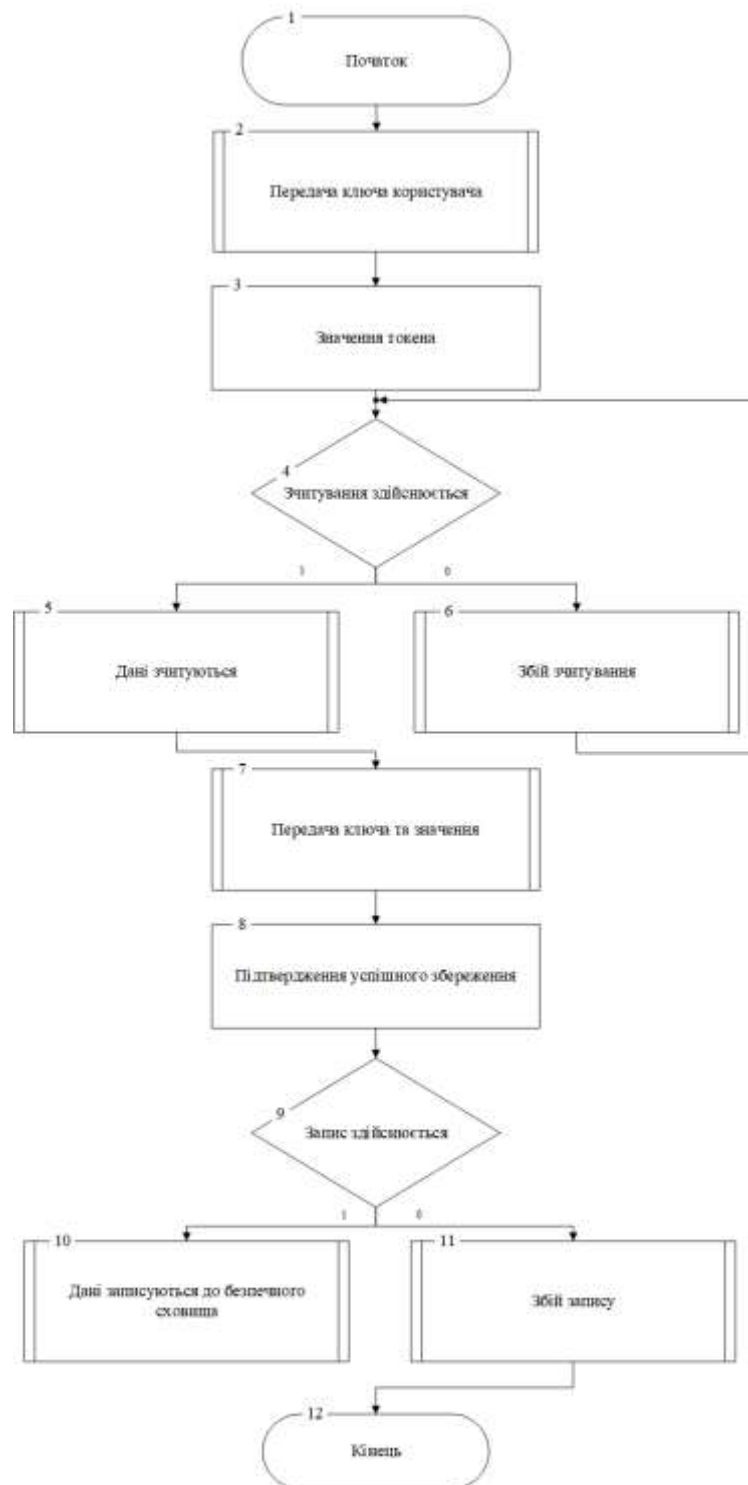


Рисунок 2.3. Блок-схема алгоритму захисту даних користувачів

Таким чином, алгоритм забезпечує надійний захист конфіденційних даних користувачів, поєднуючи шифрування з безпечним зберіганням на мобільному пристрої. Це робить його оптимальним рішенням для використання у мобільних додатках.

РОЗДІЛ 3. РОЗРОБКА ПРОГРАМНИХ МОДУЛІВ РЕАЛІЗАЦІЇ МОБІЛЬНОГО ЗАСТОСУНКУ

3.1 Варіантний аналіз і обґрунтування вибору засобів для реалізації програмного забезпечення

При розробці мобільного застосунку для підтримки діяльності благодійних організацій важливо обрати оптимальні технології для реалізації кожного модуля системи.

Сучасний підхід передбачає використання технології React Native, фреймворку Expo та мови програмування JavaScript. На відміну від традиційних рішень, це забезпечує спрощену розробку та кросплатформність застосунку.

React Native - це фреймворк, який дозволяє створювати мобільні додатки для iOS та Android на базі JavaScript та React [10]. Завдяки використанню єдиної кодової бази, розробники можуть одночасно підтримувати обидві платформи. Використання компонентної архітектури спрощує побудову інтерфейсів, а функція гарячого перезавантаження дозволяє миттєво бачити зміни в коді без необхідності перекомпіляції [11].

Expo є надбудовою над React Native, що значно спрощує та пришвидшує процес розробки [12]. Він автоматично конфігурує проект, надає доступ до численних корисних бібліотек і інструментів, а також дозволяє розробляти додатки без складного налаштування середовища під кожную платформу окремо [13]. Expo містить готові компоненти для швидкого створення складних інтерфейсів, а зручний командний інтерфейс полегшує запуск додатків на емуляторах та реальних пристроях. Крім того, Expo забезпечує просту взаємодію з нативними API пристроїв, що розширює функціональність додатку, а також дозволяє швидко розгорнути додаток для тестування і відладки через інтернет, спрощуючи командну роботу.

Головні переваги React Native та Expo полягають у наступному:

- Використання однієї кодової бази для iOS і Android, що суттєво скорочує час і витрати на розробку.

- Використання популярної мови JavaScript, що полегшує формування команди розробників.
- Активна спільнота і велика кількість документації, що допомагає швидко знаходити рішення та поради.
- Функція гарячого перезавантаження для оперативної розробки та відладки.
- Широкий вибір готових компонентів і бібліотек, які прискорюють створення функціоналу.
- Простота налаштування середовища розробки, особливо для новачків.
- Можливість роботи з нативними функціями пристрою без складного вивчення платформозалежних API.

Для серверної частини застосунку використано CMS-сервіс Hygraph і мову запитів GraphQL.

GraphQL - це мова запитів та середовище виконання, що забезпечує гнучку взаємодію з сервером даних [14]. Основна перевага GraphQL полягає у можливості точно вказати, які саме дані потрібні, без зайвих запитів і отримання непотрібної інформації [15]. На відміну від REST API, де кожен ендпоінт відповідає за конкретний ресурс, GraphQL дозволяє формувати структуру й обсяг даних у запиті.

Крім того, GraphQL підтримує запити, які одночасно отримують пов'язані дані (глибокі запити), що знижує кількість необхідних звернень до сервера. Вбудована система типів забезпечує опис структури даних і автоматичну валідацію запитів, що допомагає виявляти помилки на ранніх етапах розробки. Директиви дозволяють динамічно фільтрувати та модифікувати дані у відповідь на умови. Важливою особливістю є підтримка підписок, що дає змогу клієнтам отримувати оновлення в реальному часі, що критично для реал-тайм застосунків, таких як мобільний застосунок для підтримки благодійних організацій.

Таким чином, GraphQL є потужним інструментом для створення ефективних, гнучких API, що оптимізують роботу з даними.

Отже, вибір React Native, Expo та GraphQL для розробки кросплатформного мобільного застосунку забезпечує комплексний підхід, що поєднує простоту, продуктивність та надійність. Це дозволяє створити функціональний, зручний і ефективний інструмент для підтримки діяльності благодійних організацій.

3.2 Розробка модулів авторизації користувачів та перегляду інформації про волонтерські заходи

При розробці модуля авторизації користувачів слід враховувати низку технічних аспектів, зокрема забезпечення високого рівня безпеки даних. Для реалізації авторизації у застосунку було обрано технологію Clerk. Використання Clerk у React Native Expo додатку дозволяє поєднати надійність захисту з зручністю для користувачів [16].

Першим кроком є встановлення необхідних пакетів, після чого налаштовується Clerk SDK, що забезпечує інтеграцію застосунку з сервісом Clerk. Ця технологія надає готові компоненти для входу, реєстрації та інших функцій авторизації.

Інформація про авторизованих користувачів відображається безпосередньо у системі Clerk, як показано на рис. 3.1.

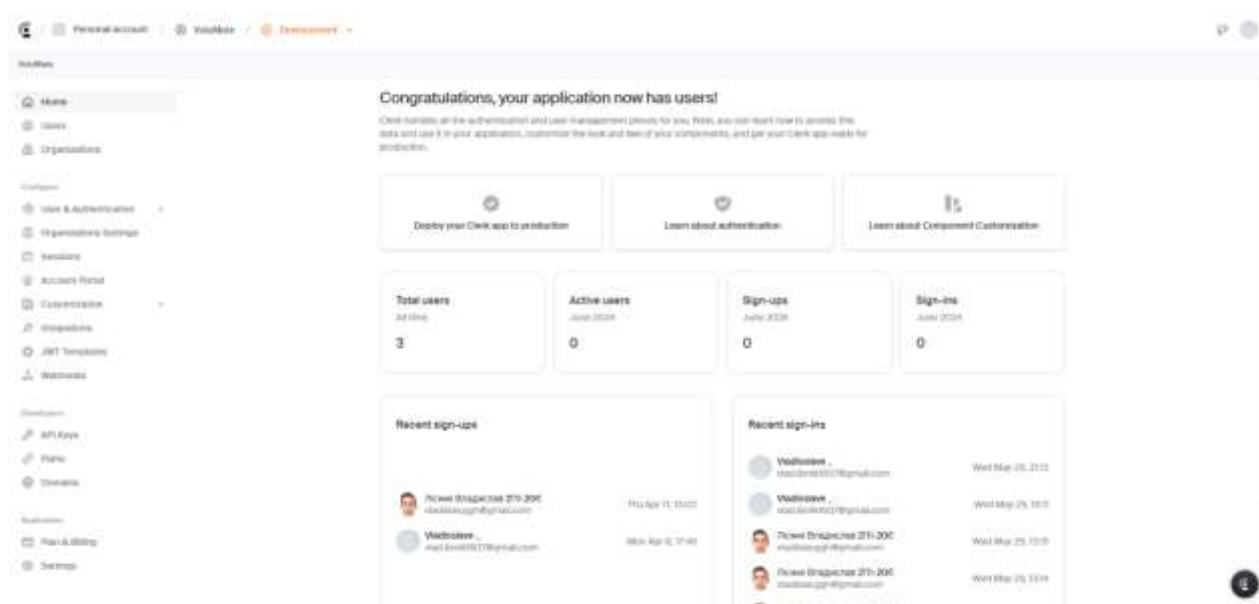


Рисунок 3.1. Система Clerk для обліку користувачів застосунку

Спочатку виконується перевірка, чи користувач увійшов у систему. В залежності від результату відображається або навігаційний контейнер із основним інтерфейсом додатку, або екран входу.

Якщо користувач авторизований, він бачить головний інтерфейс застосунку; у протилежному випадку - екран для входу.

Для реалізації авторизації через обліковий запис Google застосовується створення відповідного запиту у браузері за допомогою React Hook.

Цей React Hook використовує бібліотеку expo-web-browser, що дозволяє попередньо завантажувати (нагріти) та закривати (охолодити) веб-браузер для більш плавної роботи. Під час імпорту React і expo-web-browser створюється користувацький хук useWarmUpBrowser.

Процес завантаження браузера з вибором облікового запису для авторизації наведено на рис. 3.2.

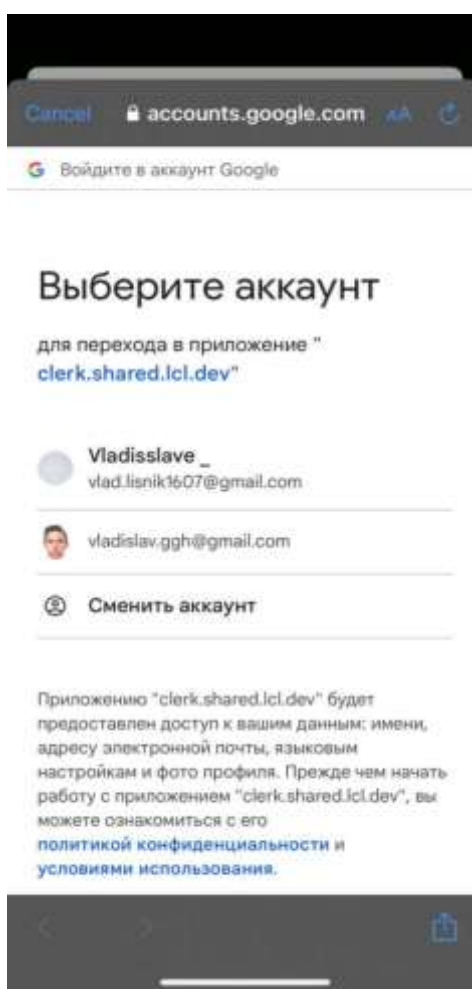


Рисунок 3.2. Завантаження браузера із вибором облікового запису

У цьому модулі використовується React-ефект, який автоматично виконується після рендерингу компонента. В межах ефекту викликається функція `WebBrowser.warmUpAsync()`, що попередньо завантажує браузер для швидшого відкриття веб-сторінок.

Після демонтажу компонента виконується функція `WebBrowser.coolDownAsync()`, яка «охолоджує» браузер, допомагаючи економити ресурси пристрою. Загалом, цей хук оптимізує роботу з веб-браузером у застосунку.

Для безперебійної авторизації користувачів у застосунку реалізовані спеціалізовані функції.

В основі розробки лежить компонент `Login` у `React Native`, який використовує `OAuth` для авторизації через `Google`. Для завершення `OAuth`-сесії застосовується функція `WebBrowser.maybeCompleteAuthSession()`. Також використовується хук `useWarmUpBrowser()` для попереднього «нагріву» браузера.

Далі застосовується хук `useOAuth` з параметром `oauth_google` для ініціалізації `OAuth`-потoku. При натисканні кнопки виконується функція `onPress`, що запускає процес авторизації.

Якщо сесія створена успішно, вона стає активною. У разі невдачі можна викликати `signIn` або `signUp` для подальших дій, наприклад багатофакторної автентифікації. Помилки виводяться у консоль для діагностики.

Головною метою авторизації є захист даних користувачів. `Clerk` забезпечує не лише безпеку, а й зручність використання, адже користувачам не потрібно запам'ятовувати окремі облікові дані для кожного застосунку - все зберігається централізовано.

При авторизації дані користувачів одразу ж зберігаються у безпечному сховищі у вигляді токенів [17].

Деавторизація відбувається шляхом завершення сесії, що дозволяє повторно увійти до системи. Таким чином, система авторизації працює автономно, що значно підвищує зручність використання сервісу.

Головною задачею модуля перегляду інформації про волонтерські заходи та проєкти є організація взаємодії користувачів з волонтерськими ініціативами, їх фінансуванням та розвитком.

Для цього створено файл-навігатор та низку файлів для роботи з проєктами.

Зокрема, файл `ProjectListItemSmall` - це React Native компонент, який відображає інформацію про окремий проєкт: зображення, назву, контактну особу та категорію.

При натисканні на цей компонент користувач переходить на сторінку з деталями проєкту. Стилі компонента визначені в об'єкті `styles`, створеному за допомогою `StyleSheet.create()`.

Область відображення останніх проєктів на головному екрані - це інтерактивна зона з підтримкою дотику, що дозволяє швидко перейти до докладної інформації.

Всі дані для відображення проєктів отримуються з бази даних за допомогою запиту, що вибирає необхідну користувачу інформацію.

Для кращого візуального сприйняття застосовано численні стилі.

Для отримання списку проєктів із бази створено файл `ProjectList.jsx`, який через запит вибирає усі проєкти з їх наповненням.

Дані зберігаються у базі даних CMS-сервісу Hygraph; інтерфейс із завантаженими проєктами наведено на рис. 3.3.

Екран із проєктами містить список усіх створених ініціатив, які мають статус «Розпочатий» або «Активний». Завершені проєкти автоматично виключаються зі списку та не відображаються користувачеві.

Для реалізації функції надсилання повідомлень ініціатору проєкту було використано бібліотеку `Linking` у середовищі React Native. Відправлення повідомлень здійснюється за допомогою URI-схеми `mailto:`, яка відкриває стандартний поштовий клієнт на пристрої з автоматичним формуванням нового листа. За допомогою методу `Linking.openURL()` користувач може швидко

перейти до поштового клієнта з уже заповненим полем електронної адреси одержувача.

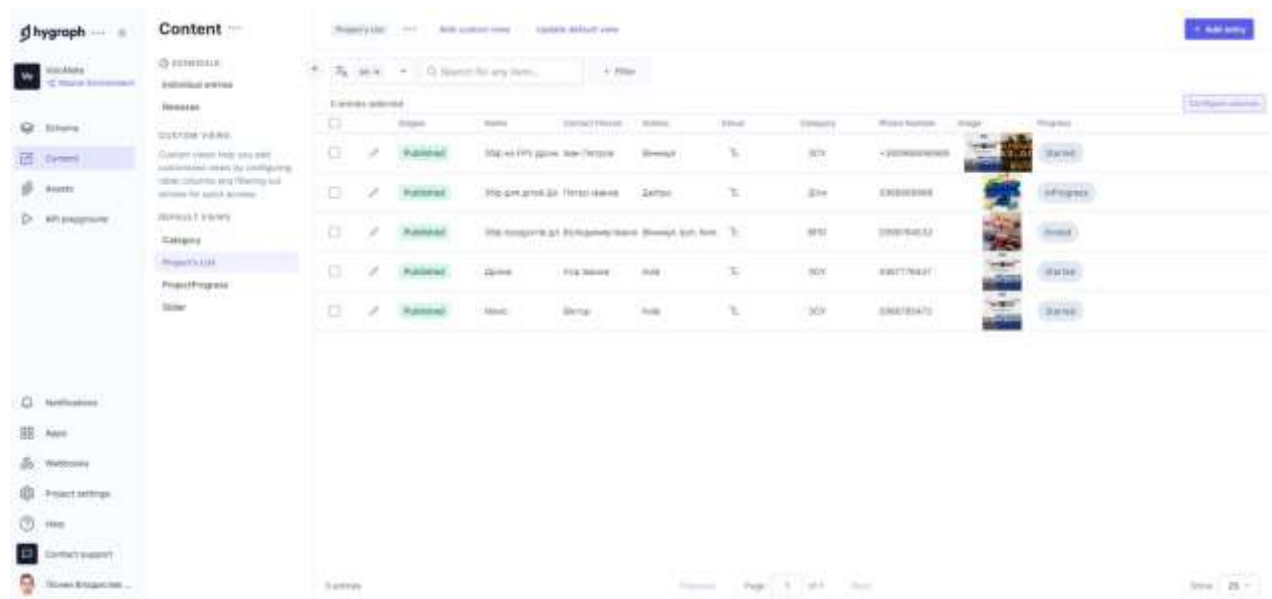


Рисунок 3.3. Інтерфейс сервісу для управління контентом із завантаженими проєктами

Для реалізації механізму фінансової допомоги інтегровано платіжну систему Stripe з використанням бібліотеки Stripe React Native. Stripe є популярною платформою для прийому онлайн-платежів, яка забезпечує безпечну обробку фінансових транзакцій.

Інтерфейс платіжної системи з налаштованим проєктом представлено на рис. 3.4.

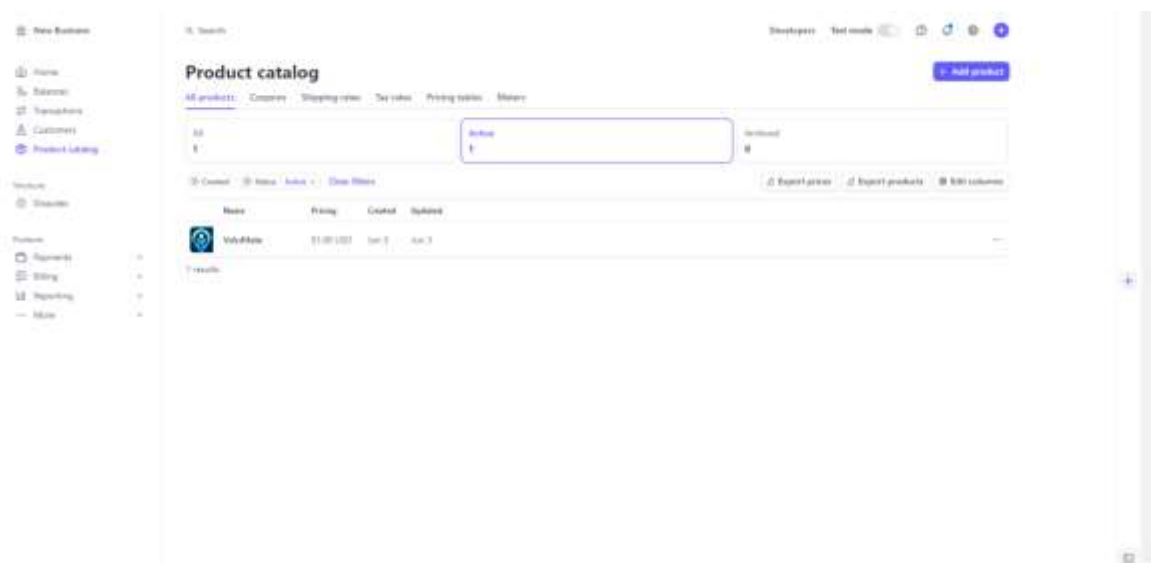


Рисунок 3.4. Інтерфейс платіжної платформи з інтеграцією проєкту

У React Native використання Stripe дає змогу розробникам інтегрувати платіжні функції безпосередньо в мобільні додатки. Ця бібліотека забезпечує прийом кредитних карток, обробку платежів і безпечну роботу на бекенді Stripe. Завдяки цьому користувачі можуть зручно і безпечно здійснювати покупки або оплачувати послуги безпосередньо в додатку.

Для перегляду інформації про волонтерські заходи та проєкти створено відповідний модуль з усіма супутніми екранами та функціями. Додаткові можливості, такі як зворотній зв'язок з ініціатором проєкту та підтримка фінансування, покращують функціональність і взаємодію з користувачем.

При розробці інтерфейсу додатку було приділено особливу увагу його простоті та зручності використання [18]. Особливо важливо було створити візуально привабливий дизайн, що відповідає цілям і потребам цільової аудиторії.

Інтерфейс застосунку розроблявся за допомогою Figma - популярного веб-інструменту для дизайну та прототипування інтерфейсів. Основною перевагою Figma є її доступність онлайн, що дозволяє працювати над проєктом з будь-якого пристрою без потреби встановлювати додаткове ПЗ.

Інтерфейс Figma представлений на рис. 3.5.

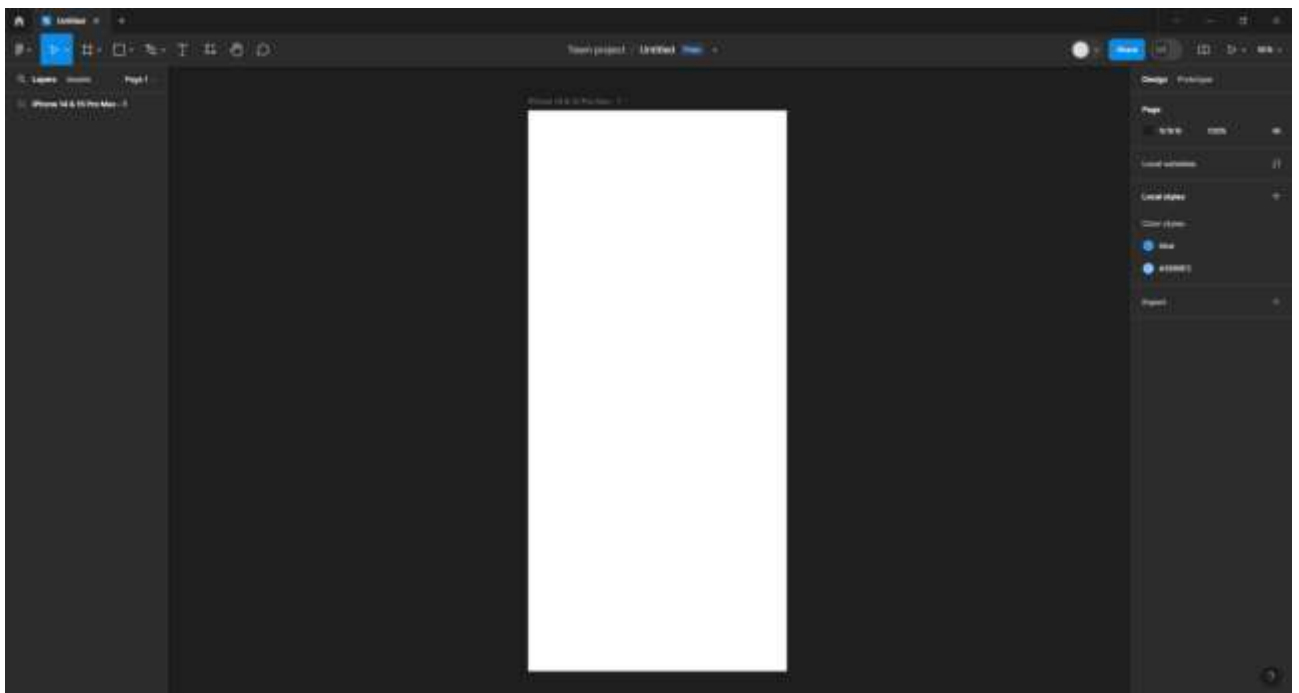


Рисунок 3.5. Інтерфейс інструменту Figma

Figma вирізняється серед інших інструментів своєю високою колаборативністю. Дизайнери, розробники та інші учасники проєкту можуть одночасно працювати над дизайном, спостерігати зміни в реальному часі та легко взаємодіяти між собою. Це значно спрощує процес командної роботи і прискорює досягнення результатів.

Інтерфейс Figma зручний і інтуїтивно зрозумілий, що дозволяє легко створювати макети, компоувати елементи та розробляти інтерактивні прототипи. Завдяки широкому набору інструментів та можливості імпорту ресурсів, таких як зображення і іконки, Figma надає багато можливостей для творчої роботи.

Однією з важливих особливостей є підтримка створення компонентів і стилів, які можна багаторазово використовувати. Це допомагає забезпечити єдиний стиль дизайну та ефективно керувати змінами у проєкті.

Ще одна ключова перевага Figma - робота з векторною графікою, що робить створені інтерфейси масштабованими і гнучкими. Це дозволяє розробляти дизайни, які зберігають якість при адаптації до різних розмірів екранів мобільних пристроїв - критично важливий фактор для мобільних застосунків.

Також Figma дає змогу створювати інтерактивні прототипи, які допомагають тестувати функціональність і навігацію додатку ще до початку розробки. Така інтерактивність дозволяє команді і клієнтам краще зрозуміти, як користувач буде взаємодіяти з додатком у реальних умовах, що сприяє виявленню та усуненню помилок на ранніх етапах.

Figma користується популярністю серед веб-дизайнерів, UI/UX-дизайнерів, графічних дизайнерів, продукт-менеджерів, розробників та інших фахівців у сфері цифрового дизайну. Особливо цінують її онлайн-команди, що працюють віддалено, адже Figma суттєво спрощує спільну роботу і обмін ідеями.

Крім того, інструмент має вбудовані можливості для взаємодії з розробниками, зокрема зручний експорт CSS-коду та генерацію документації

для мобільних застосунків. Це допомагає підтримувати єдність дизайну та полегшує інтеграцію між дизайном і розробкою.

Приклад плагіна для експорту CSS-коду дизайну застосунку наведено на рис. 3.6.

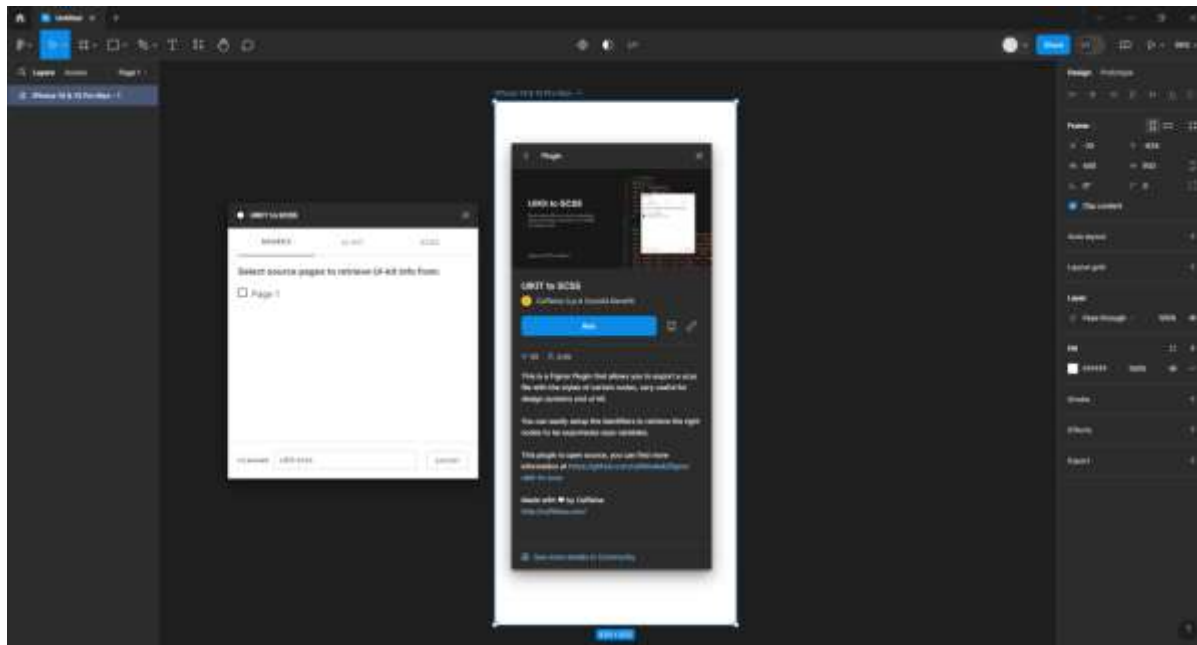


Рисунок 3.6. Використання плагінів для експортування стилів

Загальна доступність та потужний функціонал Figma роблять цей інструмент ідеальним для розробки інтерфейсів мобільних застосунків. Він дозволяє командам швидко і ефективно працювати над проєктами, спільно втілюючи ідеї та вдосконалюючи дизайн.

Для інтерфейсу було обрано основний блакитний колір, який добре пасує до тематики та легко сприймається користувачами. Другорядними кольорами стали світлі відтінки блакитного та сірого. Як допоміжні використані білий і чорний кольори, що створюють чіткий контраст і підвищують читабельність.

Отже, Figma пропонує безліч переваг: підтримку роботи в режимі реального часу, можливість спільної роботи, високу доступність та зручність використання. Інструмент дозволяє створювати компоненти, що спрощує процес дизайну та повторне використання елементів інтерфейсу. Крім того, Figma дає змогу розробляти інтерактивні прототипи і проводити презентації проєктів.

Figma працює не лише з векторною графікою, але й підтримує растрові зображення, що особливо корисно для дизайнерів, які хочуть створювати реалістичні інтерактивні прототипи.

Початковий екран мобільного застосунку, що зустрічає користувача, наведено на рис. 3.7.



Рисунок 3.7. Початковий екран

Після запуску застосунку користувача зустрічає початковий екран із можливістю авторизації через обліковий запис Google. При натисканні кнопки «Про проект» відкривається екран з інформацією про застосунок. Якщо користувач обирає кнопку «Авторизація», його буде перенаправлено на екран вибору облікового запису для входу. Цей екран наведено на рис. 3.8. Після успішної авторизації користувача вітає головний екран застосунку - ключовий елемент, який формує його ідентичність та створює перше враження про додаток. Головний екран показано на рис. 3.9.

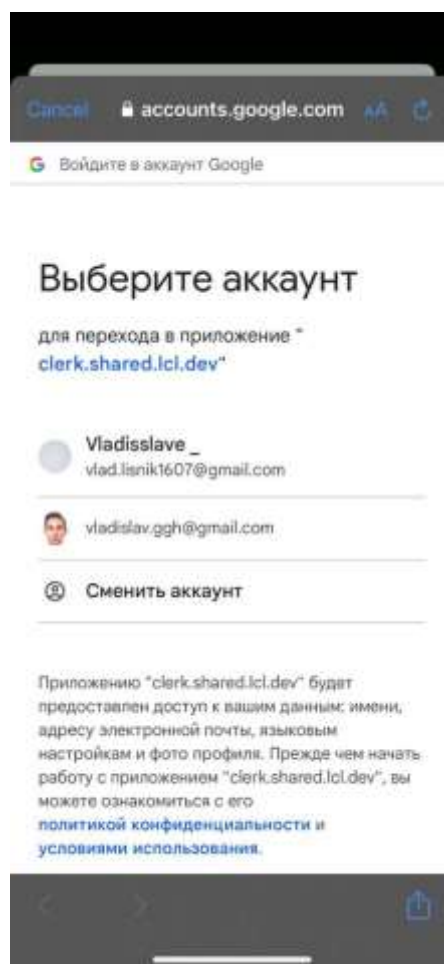


Рисунок 3.8. Экран wyboru облікового запису

Головний екран застосунку пропонує користувачу насичений інтерфейс, який інтуїтивно відображає основний функціонал, вітає користувача, демонструючи його фото та ім'я. Загалом застосунок складається з трьох основних розділів: «Головна», «Проекти» та «Профіль». На екрані «Головна» зосереджено найбільше контенту - тут доступний пошук за назвою проекту, слайдер з рекламними інтеграціями, список категорій і останні волонтерські проекти. При натисканні на зображення у слайдері користувач буде перенаправлений на відповідний веб-сайт рекламної інтеграції.

Після вибору певної категорії користувачу відображається список волонтерських проектів, що належать до цієї категорії, як показано на рис. 3.10. У списку представлені фото проектів, контактна особа та локація. При переході на конкретний волонтерський проект користувач бачить повну інформацію про нього. Екран з деталями проекту наведено на рис. 3.11. Після ознайомлення з

інформацією про проект користувач може задати запитання. Натиснувши кнопку «Повідомлення», відкривається вікно електронної пошти для уточнення деталей або звернення з питаннями, що відображено на рис. 3.12.



Рисунок 3.9. Головний екран застосунку

Користувач також може надати фінансову підтримку, натиснувши кнопку «Приєднатися» і долучившись до відповідного збору. Після натискання користувача перенаправляють на сторінку, де потрібно вказати електронну адресу, платіжні дані та додати коментар до внеску, як показано на рис. 3.13. Головний екран завершується відображенням списку останніх волонтерських проектів.

Повідомлення буде надіслано на електронну скриньку контактної особи зазначеної у проекті.

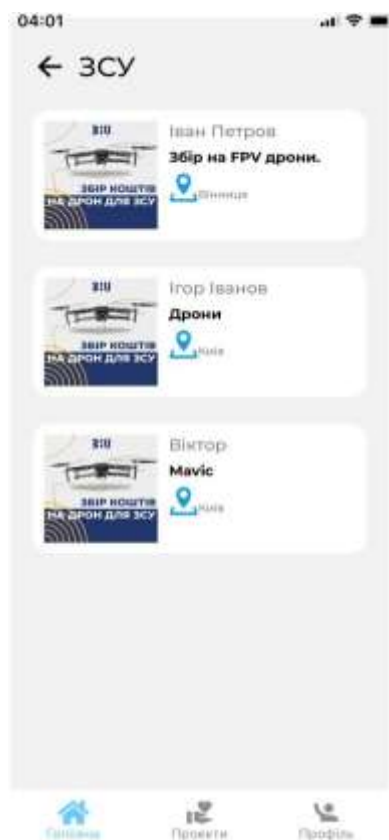


Рисунок 3.10. Список проектів відносно категорії



Рисунок 3.11. Екран інформації про проект

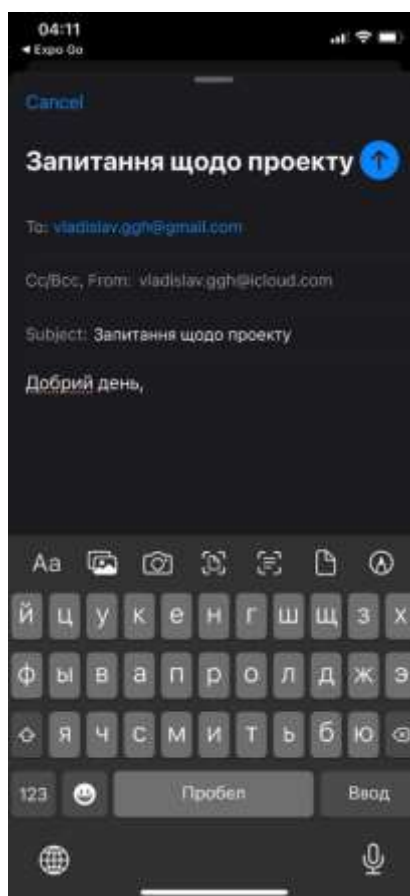


Рисунок 3.12. Відправка користувачем повідомлення при виникненні запитань щодо проекту

Цей список представлено на рис. 3.14.

Наступним у навігації застосунку є екран «Проекти», де представлено повний список проектів із назвами, контактними особами, позначками категорій та індикаторами їхнього прогресу. Екран «Проекти» мобільного застосунку наведено на рис. 3.15.

Третім, також ключовим, є екран «Профіль», на якому відображається інформація про користувача та навігаційне меню застосунку. Крім того, тут передбачено можливість зв'язку з розробником і виходу з облікового запису. Екран профілю користувача наведено на рис. 3.16. Меню профілю дозволяє переходити на екрани «Головна» та «Проекти», надіслати листа розробнику або вийти з застосунку.

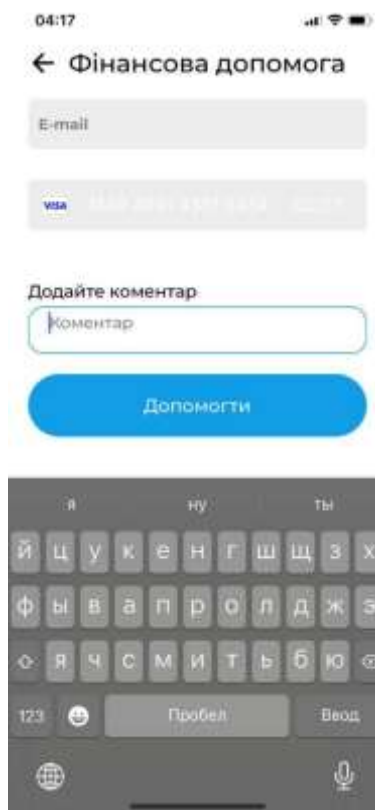


Рисунок 3.13. Екран фінансової допомоги



Рисунок 3.14. Відображення списку останніх проектів



Рисунок 3.15. Екран «Проекти»

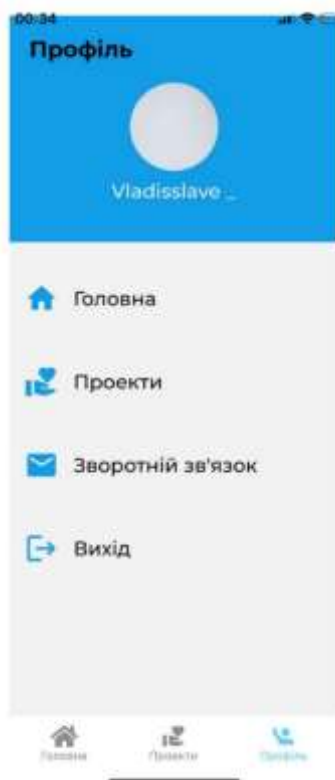


Рисунок 3.16. Екран профілю користувача

Зворотній зв'язок із розробником відображено на рис. 3.17.

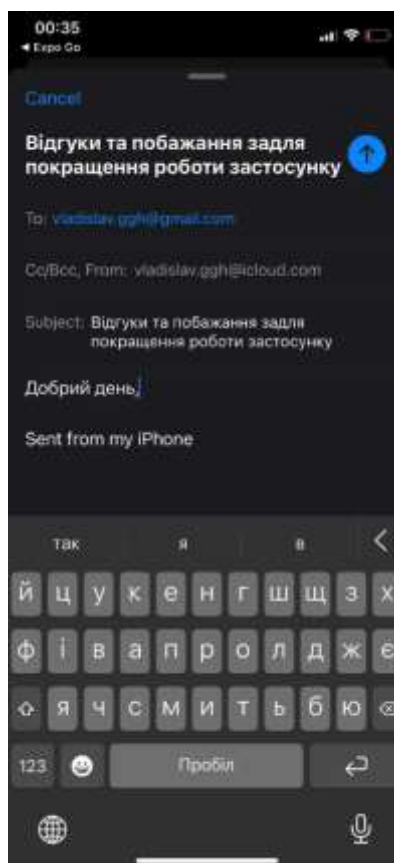


Рисунок 3.17. Зв'язок із розробником

Попереднє проектування інтерфейсу застосунку здійснювалося у дизайн-інструменті Figma, тоді як безпосередня розробка графічного інтерфейсу виконувалась у середовищі Visual Studio Code із використанням React Native та мови програмування JavaScript.

3.3 Тестування застосунку

Тестування програмного забезпечення - це процес перевірки програмного продукту з метою виявлення помилок, дефектів чи збоїв, а також перевірки відповідності вимогам замовника [19]. Головна мета тестування - забезпечити високу якість та надійність програмного продукту перед його випуском або впровадженням у експлуатацію.

Перед початком тестування на пристроях з операційними системами Android або iOS необхідно встановити тестову версію додатку. Важливо також підготувати супровідну документацію, яка включає вимоги та план тестування.

Процес тестування складається з кількох основних етапів. Перший - функціональне тестування, яке перевіряє коректність роботи програми, включно з обробкою введених і виведених даних та відповідністю бізнес-логіці [20]. Другий етап - тестування відмов, що спрямоване на виявлення дефектів і оцінку здатності програми відновлюватися після помилок [21]. Далі проводиться тестування продуктивності, яке включає вимірювання швидкості виконання операцій.

Важливим етапом є також тестування сумісності, що перевіряє коректну роботу додатку на різних операційних системах і апаратних платформах [22].

Під час функціонального тестування додатково оцінюється швидкість реагування мобільного застосунку на різні запити. Такий підхід допомагає виявити й усунути потенційні проблеми з продуктивністю, які можуть впливати на користувацький досвід. Забезпечується швидка обробка вхідних даних та оптимізація роботи додатку для досягнення найкращої продуктивності.

Ілюстрація тестування швидкості завантаження компонентів та обробки запитів наведена на рис. 3.18.

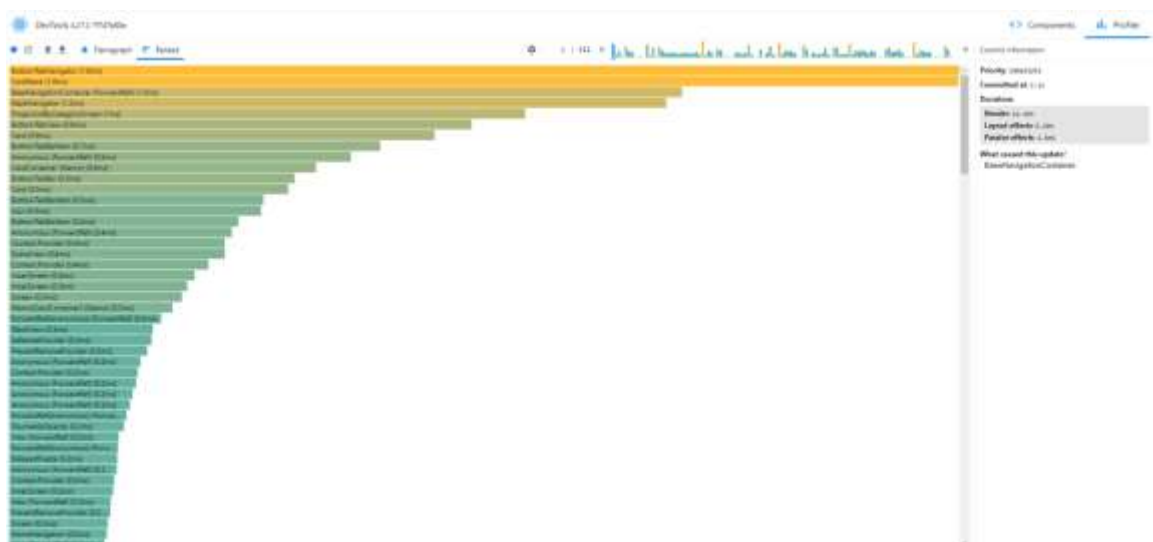


Рисунок 3.18. Тестування швидкості завантаження компонентів та обробки запитів

Після завершення тестування швидкості завантаження компонентів і обробки запитів було прийнято рішення провести додаткове тестування, спрямоване на перевірку реакції програмного забезпечення на ситуації, коли відсутні вхідні дані. Метою такого тестування стало з'ясування, як додаток поводить себе в подібних випадках, чи надає користувачеві відповідні повідомлення та інструкції щодо подальших дій.

У результаті тестування було встановлено, що програма коректно обробляє сценарій відсутності вхідних даних: вона інформує користувача про проблему та пропонує можливі шляхи її вирішення.

Ключовим завданням цього етапу було оцінити ефективність реакції додатку на нестандартні ситуації та перевірити, наскільки чітко і зрозуміло він надає користувачеві інструкції. Це забезпечує зручність використання навіть у непередбачуваних умовах, сприяючи позитивному користувацькому досвіду.

Ілюстрація тестування реакції застосунку на відсутність вхідних даних наведена на рис. 3.19.

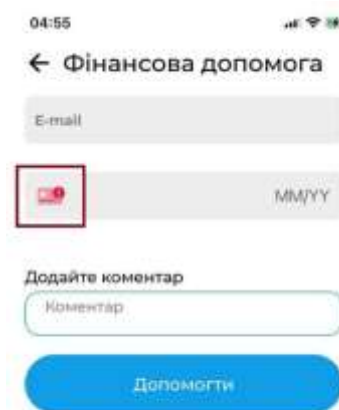


Рисунок 3.19 – Реакція застосунку на відсутність вхідних даних

У випадку введення некоректних даних застосунки автоматично змінює

вигляд іконки картки - вона стає червоною зі знаком оклику, що вказує на помилку введення. Крім того, користувач отримує відповідне повідомлення з поясненням помилки, як показано на рис. 3.20.

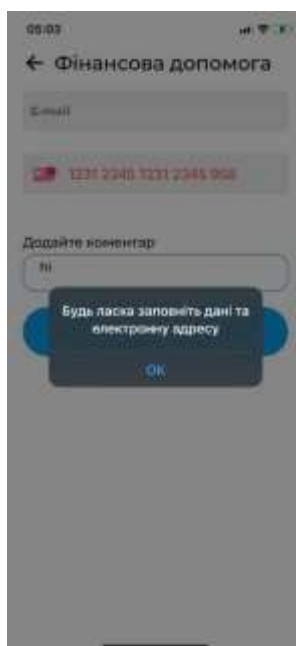


Рисунок 3.20. Реакція застосунку на відсутність вхідних даних

Результати тестування підтвердили, що програмне забезпечення успішно розпізнає відсутність вхідних даних і надає користувачеві відповідні повідомлення або підказки. Це дозволяє користувачам впевнено орієнтуватися в ситуації та знижує ймовірність помилок під час взаємодії з додатком, забезпечуючи комфортний і передбачуваний досвід користування.

У межах тестування також було проведено перевірку сумісності програмного продукту з різними версіями операційних систем і мінімальними характеристиками апаратного забезпечення. Зокрема, додаток тестувався на сумісність з операційними системами Android та iOS. Для цього використовувалися як сучасні, високопродуктивні пристрої на базі iOS, так і застарілі Android-девайси.

Результати підтвердили стабільну роботу програмного забезпечення на широкому спектрі пристроїв, що свідчить про добру оптимізацію та високу ефективність його роботи. У користувацькій інструкції також визначено технічні

вимоги для запуску додатку. Детальну інформацію щодо мінімальної та рекомендованої конфігурації наведено в табл. 3.1.

Таблиця 3.1. Вимоги до апаратного забезпечення

Вимоги до конфігурування	Мінімальні	Рекомендовані
Процесор	3 тактовою частотою 1,5- 2ГГц	3 тактовою частотою 2-2,5 ГГц
Оперативна	2 ГБ RAM	4 ГБ RAM
ий простір у	1 ГБ	2 ГБ
Батарея	Не менше 1500 мАг	Більше 2000 мАг
Операційна	Android 8.0, iOS 12	Android 12, iOS 18
Дисплей	Роздільна здатність 720p, діагональ 5-6 дюймів	Роздільна здатність 1080p, діагональ 5,5-6,5 дюймів

Після запуску мобільного застосунку та проходження авторизації користувача зустрічає головний екран, який відіграє важливу роль у формуванні першого враження та ідентичності продукту. Інтерфейс головного екрану є насиченим та інтуїтивно зрозумілим: він демонструє основний функціонал додатку, вітає користувача, відображаючи його ім'я та фотографію.

Застосунок складається з трьох основних екранів: «Головна», «Проекти» та «Профіль».

Головний екран містить найбільший обсяг інформації. Тут розміщено: поле пошуку проєктів за назвою; слайдер з рекламними інтеграціями, при натисканні на зображення якого користувач перенаправляється на відповідний вебсайт; список категорій; перелік останніх волонтерських проєктів. При виборі категорії користувач бачить список проєктів, що до неї належать.

На екрані «Проекти» представлено всі незавершені волонтерські ініціативи незалежно від їхньої категорії. Кожен проєкт має індикатор прогресу виконання. Натиснувши на обраний проєкт, користувач переходить до детальної сторінки, де може ознайомитися з повною інформацією, поставити запитання ініціатору через електронну пошту або здійснити фінансову підтримку.

Екран «Профіль» відображає персональну інформацію користувача та містить навігаційне меню додатку. З цього розділу можна перейти на інші екрани застосунку, зв'язатися з розробником або вийти зі свого облікового запису.

Головний екран застосунку наведено на рис. 3.21.



Рисунок 3.21. Головний екран мобільного застосунку

ВИСНОВКИ

У кваліфікаційній роботі бакалавра було розроблено модулі мобільного застосунку, спрямованого на підтримку діяльності благодійних організацій. Розробку здійснено у середовищі Visual Studio Code із використанням сучасних технологій.

У процесі виконання роботи проведено аналіз актуального стану проблематики. Розглянуто основні існуючі аналоги мобільних систем, виявлено їхні недоліки та проведено порівняння з власною реалізацією. На основі цього сформульовано завдання кваліфікаційної роботи.

Під час аналізу технологій розробки обґрунтовано вибір мови програмування JavaScript, фреймворку React Native, платформи Expo та мови запитів GraphQL.

У рамках роботи реалізовано наступне:

- розроблено модуль авторизації користувачів для забезпечення безпеки доступу та розширення функціоналу;
- створено модуль для перегляду інформації про волонтерські ініціативи та проєкти;
- впроваджено функціонал зворотного зв'язку та можливості надання фінансової допомоги;
- розроблено адаптивний та зручний графічний інтерфейс;
- проведено тестування програмного забезпечення відповідно до визначених технічних вимог.

Також розроблено схеми навігації мобільного застосунку та побудовано модель системи з використанням UML-діаграм.

Результати тестування підтвердили повну працездатність програмного продукту та його відповідність поставленому технічному завданню. Окрім того, було підготовлено інструкцію користувача.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Актуальність розробки застосунків для волонтерства [Електронний ресурс] – Режим доступу до ресурсу: <http://surl.li/uegwh>.
2. Лісник В.І. Мобільний застосунок для автоматизації процесів діяльності волонтерських організацій/ В.П. Майданюк, В.І. Лісник // Матеріали конференції «LIII Науково-технічна конференція підрозділів Вінницького національного технічного університету (2024)», Вінниця, 2024. [Електронний ресурс] – Режим доступу до ресурсу: <http://surl.li/uegwr>.
3. Лісник В. І. Актуальність розробки серверної частини мобільних застосунків за допомогою sms-орієнтованих сервісів/ В.П. Майданюк, В.І. Лісник // Матеріали міжнародної науково-практичної інтернет конференції «Молодь в науці: дослідження, проблеми, перспективи (МН-2024)», Вінниця, 2024. [Електронний ресурс] – Режим доступу до ресурсу: <http://surl.li/uegwu>.
4. VolunteerUP [Електронний ресурс] – Режим доступу до ресурсу: <http://surl.li/uegxh>.
5. СпівДія [Електронний ресурс] – Режим доступу до ресурсу: <http://surl.li/uegxq>.
6. ВолонтерОрг [Електронний ресурс] – Режим доступу до ресурсу: <http://surl.li/uegxt>.
7. UML diagram [Електронний ресурс] – Режим доступу до ресурсу: <http://surl.li/uegxx>.
8. UML activity diagram [Електронний ресурс] – Режим доступу до ресурсу: <http://surl.li/uegyf>.
9. Visio [Електронний ресурс] – Режим доступу до ресурсу: <http://surl.li/uegyl>.
10. React Native [Електронний ресурс] – Режим доступу до ресурсу: <http://surl.li/uegyv>.
11. Benefits React Native [Електронний ресурс] – Режим доступу до ресурсу: <http://surl.li/uegze>.

12. Expo framework [Электронный ресурс] – Режим доступа до ресурсу: <http://surl.li/uegzi>.
13. Advantages of the Expo [Электронный ресурс] – Режим доступа до ресурсу: <http://surl.li/uegzl>.
14. GraphQL [Электронный ресурс] – Режим доступа до ресурсу: <http://surl.li/uegzp>.
15. Benefits of GraphQL [Электронный ресурс] – Режим доступа до ресурсу: <http://surl.li/uegzs>.
16. Clerk [Электронный ресурс] – Режим доступа до ресурсу: <http://surl.li/uegzv>.
17. Expo Secure Store [Электронный ресурс] – Режим доступа до ресурсу: <http://surl.li/uegzy>.
18. Robert C. Martin Series. Functional Design: Principles, Patterns, and Practices. 1st Edition, 2023. 384 p.
19. Software testing [Электронный ресурс] – Режим доступа до ресурсу: <http://surl.li/uehab>.
20. Functional testing [Электронный ресурс] – Режим доступа до ресурсу: <http://surl.li/uehag>.
21. Failure testing [Электронный ресурс] – Режим доступа до ресурсу: <http://surl.li/tkely>.
22. Compatibility testing [Электронный ресурс] – Режим доступа до ресурсу: <http://surl.li/ueham>.