

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА**

Факультет комп'ютерних наук та кібернетики
Кафедра теорії та технології програмування

Кваліфікаційна робота
на здобуття ступеня бакалавра
за спеціальністю 122 Комп'ютерні науки
на тему:

**РОЗРОБКА UX/UI МОБІЛЬНОГО ЗАСТОСУНКУ
ОДНОРАНГОВОЇ ВЗАЄМОДІЇ**

Виконала студентка 4-го курсу
Марія АНУФРІЄВА



Науковий керівник:
доцент кафедри теорії та технології програмування
Олексій ТКАЧЕНКО

Засвідчую, що в цій роботі немає
запозичень з праць інших авторів без
відповідних посилань.

Студент



(підпис)

Роботу розглянуто й допущено до
захисту на засіданні кафедри теорії та
технології програмування
«05» червня 2023 р.,
протокол № 18
Завідувач кафедри
Микола НІКІТЧЕНКО

РЕФЕРАТ

Обсяг роботи 74 сторінки, 58 ілюстрацій, 2 таблиці, 34 джерела посилань.

UX/UI ДИЗАЙН, USER PERSONA, WIREFRAMES, APPLICATION STRUCTURE, UX/UI РОЗРОБКА, АРХІТЕКТУРА ДОДАТКА, АНАЛІЗ КОНКУРЕНТІВ, USER FLOWS, COMPETITOR RESEARCH, СЦЕНАРІЇ ВЗАЄМОДІЇ, WIREFRAMING, WIREFRAMES, PROTOTYPING

Предметна область. Зручність використання мобільного додатка для поставлених цілей.

Об'єктом дослідження даної роботи є розробка мобільних застосунків. Предметом дослідження є проектування UX/UI мобільного застосунку однорангової взаємодії.

Мета роботи. Розробка мобільного додатка для людей, які потребують допомоги в особливих умовах.

Методи дослідження: загальнонаукові, метод моделювання, об'єктно-орієнтований підхід.

Інструменти розроблення: програмний продукт Figma, інтерактивний та колаборативний інструмент для візуалізації та роботи над концепціями, мапами, архітектурами, сценаріями FigJam, інтегроване середовище для розробки (IDE) Rider.

Результати роботи. Досліджено предметну область, виявлено цільову аудиторію, проаналізовано наявні мобільні застосунки, призначені для допомоги постраждалим людям через війну, створено картки типів потенційних користувачів, побудовано інформаційну архітектуру додатку, яка відображає основну навігацію, спроектовано рішення (wireframes), створено прототипи, виконано прототипування готових рішень, реалізовано сценарій додавання оголошення користувачем.

ЗМІСТ

РЕФЕРАТ	1
ЗМІСТ	2
СКРОЧЕННЯ ТА УМОВНІ ПОЗНАЧЕННЯ	3
ВСТУП	4
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	8
1.1 Принципи проектування інтерфейсу	8
1.2 Етапи проектування UX/UI мобільних додатків	12
1.3 Зручність використання обраного інтерфейсу відповідно до ЦА	19
РОЗДІЛ 2. ОГЛЯД UX/UI РІШЕНЬ АНАЛОГІВ	21
2.1 Мобільний додаток “Banderolka”	21
2.2 Мобільний додаток “ЕРАУНД”	26
2.3 Мобільний додаток “TURBOTA”	28
2.4 Висновки щодо функціональності аналогів	33
РОЗДІЛ 3. РОЗРОБКА UX/UI МОБІЛЬНОГО ЗАСТОСУНКУ	34
3.1 Проектування інформаційної архітектури додатка (information structure)	34
3.2 Аналіз користувачів та створення User Personas	34
3.3 Проектування рішень (wireframes)	35
3.4 Проектування сценаріїв взаємодії користувача із застосунком (user flows)	36
3.5 Створення прототипів із додаванням стилістики, кольору та контенту (UI елементів)	36
3.6 Прототипування	50
3.7 Дизайн-система додатка “Helper”	52
РОЗДІЛ 4. ЧАСТИНА РОЗРОБКИ МОБІЛЬНОГО ДОДАТКА	55
4.1 Структура проекту	55
4.2 Реалізація класів проекту	57
4.3 Реалізація дизайну інтерфейсу	64
ВИСНОВКИ	68
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	69
ДОДАТОК А	73
ДОДАТОК Б	74

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАЧЕННЯ

UX - User Experience

UI - User Interface

ЦА - Цільова аудиторія

ОС - Операційна система

MVVM - Model-View-ViewModel

IDE - Integrated Development Environment

HIG - Human Interface Guidelines

ВСТУП

Об'єктом дослідження даної роботи є розробка мобільних застосунків.

Предметом дослідження є проектування UX/UI мобільного застосунку однорангової взаємодії.

Оцінка сучасного стану об'єкта розробки. На сьогодні галузь розробки мобільних додатків є дуже розвиненою, адже люди витрачають значно більше часу на використання мобільних додатків для досягнення певних цілей у повсякденному житті, ніж на використання веб-сайтів, адаптованих під мобільний пристрій. Користувачі мобільних застосунків вже звикли, що можна купувати, зберігати, сплачувати, додавати щось, не витрачаючи зайвий час на повторну авторизацію чи реєстрацію на певному сайті, адже вся необхідна користувачу інформація завжди знаходиться у швидкому для них доступі.

Переваги мобільних додатків:

- 1) Доступність та зручність використання. Користувач може будь-де і в будь-який зручний для нього час виконати необхідні йому дії, наприклад, щось сплатити.
- 2) Швидкий доступ до певних функцій. Адаптовані веб-сайти під мобільний пристрій можуть мати обмежений функціонал, адже є функції, з якими може взаємодіяти користувач, які можуть бути реалізовані лише у мобільному додатку.
- 3) Використання технологій. У мобільних застосунках можуть бути використані різні технології, наприклад, штучний інтелект. Саме завдяки цій можливості продукт можна виділити серед його конкурентів та привернути увагу самих користувачів, зробивши їх досвід взаємодії з додатком більш насиченим та цікавим.

- 4) Обмін інформацією між бізнесом та клієнтом. До функціоналу мобільних додатків можна додавати push-повідомлення, чати з адміністратором та багато іншого. Це дасть можливість бізнесу комунікувати з користувачами продукту та отримувати необхідну інформацію для покращення його.
- 5) Збір даних та його аналіз. У застосунках часто зустрічаються додаткові поля при реєстрації або певні опитування, завдяки яким бізнес отримує необхідну інформацію про користувачів та має змогу проаналізувати отримані дані, щоб вдосконалити продукт та покращити користувацький досвід.
- 6) Індивідуальний підбір послуг та пропозицій. Мобільні додатки зазвичай підбирають пропозиції для користувачів, враховуючи їх інтереси та потреби.
- 7) Розвиток бізнесу та збільшення доходу. У мобільних застосунках часто розміщуються певні пропозиції по типу передплат, покупок влаштованого функціоналу або ж рекламні послуги. Завдяки такому типу контенту дохід бізнесу значно зростатиме.

Актуальність роботи та підстави для її виконання. В умовах воєнного стану, техногенних катастроф у людей виникає потреба у різноманітних речах та послугах, які допомагатимуть їм вижити у подібних умовах та полегшити побут. Мобільний додаток може закрити певні їх потреби. На даній платформі люди зможуть додавати оголошення, наголосивши на своїй потребі в чомусь, та відобразити своє місцезнаходження, щоб оточуючі люди змогли надати допомогу потребуючим якнайшвидше.

Також користувачі, які бажають допомогти, матимуть можливість опублікувати оголошення з речами, які вони готові пожертвувати. Адже на

даний момент дійсно присутня така проблема, що якщо людина хоче допомогти постраждалим не тільки грошима, а й певними речами, вона не може зробити це швидко й якісно, наприклад, через певний ресурс. А вже існуючі додатки не надають користувачам необхідний для цього функціонал.

Мета й завдання роботи. Метою роботи є розробка мобільного додатка для людей, які перебувають допомоги в особливих умовах.

Для досягнення цієї мети було поставлено такі завдання:

- 1) Дослідити предметну область, визначити ЦА, потреби та проблеми користувачів.
- 2) Проаналізувати існуючі мобільні додатки на ринку, призначені для допомоги постраждалим людям через війну. Визначити сильні та слабкі сторони.
- 3) Проаналізувати типи користувачів, їх користувацькі болі та досвід
- 4) Побудувати архітектуру мобільного додатку (Information Structure).
- 5) Спроекувати рішення, відобразивши їх у чорно-білому варіанті (Wireframes).
- 6) Спроекувати сценарії виконання певних дій користувачем на основі рішень (User Flow).
- 7) Створення прототипів. Додавання кольору, стилістики та контенту до спроектованих рішень.
- 8) Прототипування спроектованих сторінок для аналізу користувацької взаємодії.
- 9) Реалізація сценарію додавання оголошення.

Об'єкт і методи дослідження або розроблення. Для розробки UI частини мобільного застосунку однорангової взаємодії використовується

програмний продукт Figma. У Figma розробляються макети сторінок, прототипи, будуються сценарії та налаштовується прототипування.

Для розробки UX частини мобільного застосунку однорангової взаємодії використовується інтерактивний інструмент FigJam, розроблений компанією Figma. За допомогою даного інструмента було побудова навігаційну карту мобільного додатка (information structure) та картки потенційних користувачів (user personas).

Для реалізації сценарію додавання оголошення користувачем в якості інструменту розробки було обрано Rider - інтегроване середовище розробки (IDE) від компанії JetBrains. Розробка на мові програмування C#.

Можливі сфери застосування. Даний застосунок може бути застосований у сфері волонтерства, адже його основною задачею є поміч потерпілим людям.

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Принципи проектування інтерфейсу

Інтерфейс користувача (UI) та його досвід (UX) відіграють важливу роль у розробці будь-якого продукту, адже це перше, з чим зіштовхується користувач, запустивши його. Якщо користувач запустить додаток, але не зможе дійти до своєї цілі, отримати результат, на який він розраховував, то навряд він звернеться до даного додатка знову, щоб закрити свої певні потреби, адже перше враження та перший досвід взаємодії був негативним і відклався у пам'яті. У додаток це підтверджує те, що наявність ідеального коду та відсутність якісного та інтуїтивно зрозумілого інтерфейс все одно приречить продукт на провал та принесе бізнесу лише збитки. Отже, UX/UI дизайн це невід'ємна частина роботи у розробці будь-якого продукту [1].

Основні принципи розробки UX/UI дизайну мобільного додатка:

- 1) Простота взаємодії та інтуїтивність. Дизайн має бути інтуїтивно зрозумілим для користувачів, з мінімальною кількістю кроків на шляху досягнення поставлених цілей клієнта. Користувачу має бути чітко зрозуміло, як взаємодіяти з елементами та який результат він отримає після взаємодії. Елементи керування мають бути звичними.
- 2) Зручна навігація. Користувачу не потрібна маса різних функцій, адже це може лише його заплутати та збити зі шляху, який він планував проробити, встановивши даний додаток. Також необхідно пам'ятати, що основний функціонал додатку має бути майже завжди доступним для користувачів, тому його варто розміщувати у головних елементах навігації (у верхньому та нижньому барі). Найважливіші розділи додатка необхідно розміщувати у нижньому барі, так як він є головним елементом навігації, що є керованим

великим пальцем руки, тому є легкодоступним для користувача. Приклади головних елементів навігації зображені на рисунках 1.1.1 та 1.1.2.

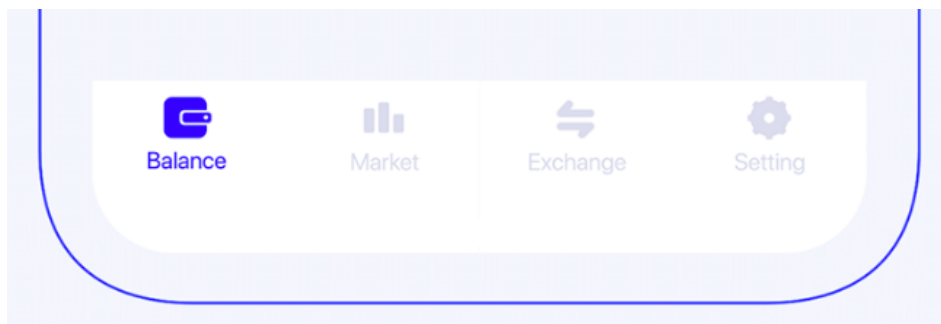


Рисунок 1.1.1 – Нижній бар (Navigation bar - Android; Tabbar - IOS) [2]

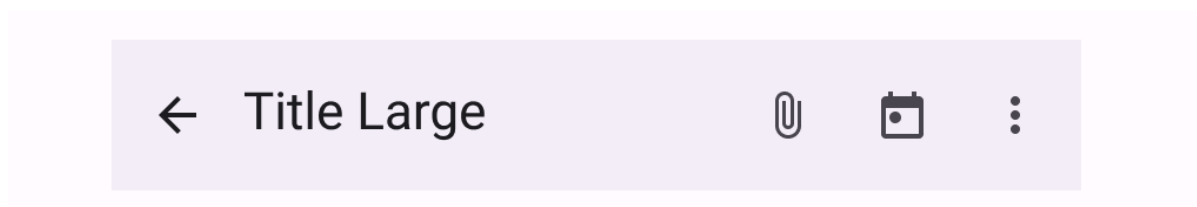


Рисунок 1.1.2 – Верхній бар (Top app bar - Android; Navigation bar - IOS) [3]

3) Специфіка інтерфейсу. Необхідно завжди пам'ятати про зону “великого пальця”, адже розміщення важливих UI елементів для взаємодії у “сліпих” зонах ускладнить користувацький досвід. Також необхідно пам'ятати, що розмір зони “великого пальця” залежить від розміру екрану смартфона, тобто чим більший екран, тим менша зона [4]. Так само необхідно враховувати рекомендовані розміри елементів інтерфейсу, наприклад, кнопок, полів вводу, текстових блоків, карток, адже, якщо вони будуть менші за рекомендований мінімальний розмір, користувачу буде важко з ними взаємодіяти. На рисунку 1.1.3 зображені зони “великого пальця”, рекомендовані для

розміщення важливих UI елементів (головних дій).

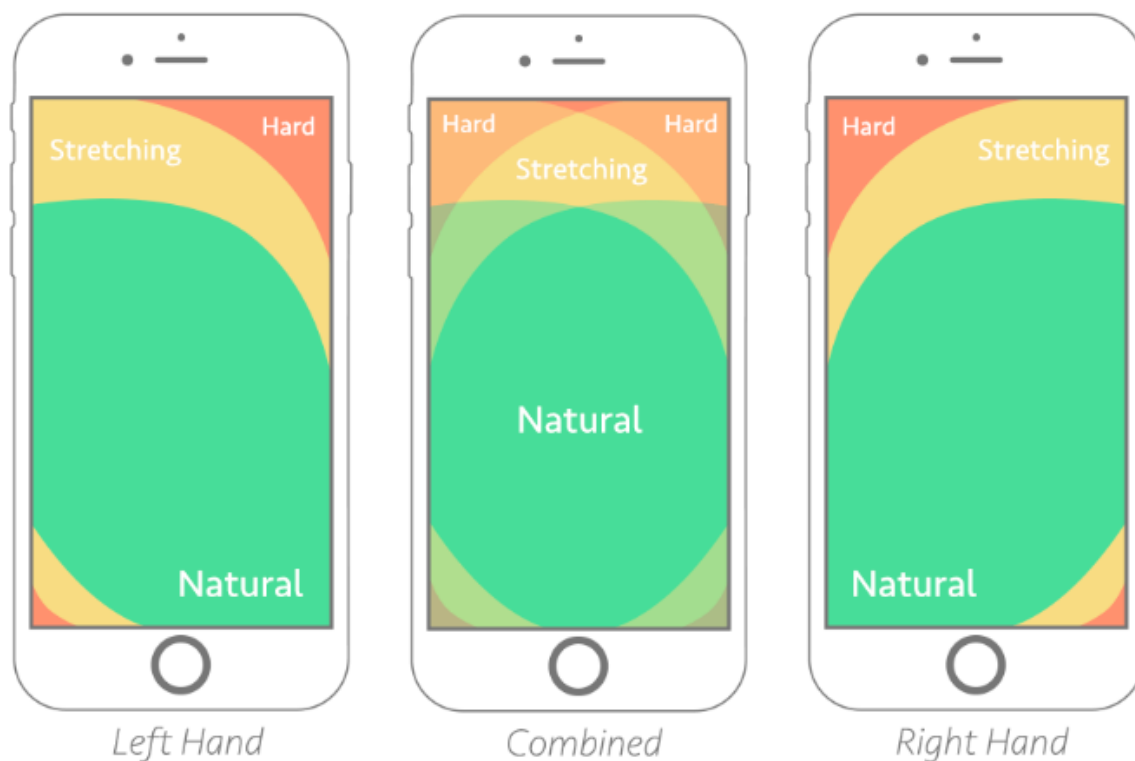


Рисунок 1.1.3 – Зона “великого пальця” для лівої, правої руки та комбінована [4]

- 4) Використання наявних можливостей пристрою та сучасних технологій. У мобільному додатку важливо надавати користувачу певні альтернативи, наприклад, надати можливість просканувати документ, замість того, щоб заповнювати купу полів. До цього також відносяться автозаповнення, автовизначення, автофокус. Так само бажано використовувати доступні сучасні технології, наприклад, камеру, розпізнавання тексту тощо. Адже наявність таких можливостей у додатку значно покращать користувацький досвід, а в додаток ще й виділить продукт серед його конкурентів.
- 5) Оптимізація платформи та використання нативних компонентів Human Interface Guidelines (дизайн-система від Apple) та Material

Design Guidelines (дизайн-система від Google). Щоб забезпечити користувачам позитивний досвід взаємодії з продуктом, необхідно враховувати те, що клієнти можуть користуватись різними операційними системами, а з цього слідує те, що не усім користувачам буде зручно користуватись додатком, який притримується певної дизайн-системи під обрану ОС, адже дизайн додатка буде зовсім інакшим та незвичним, а пошук необхідних елементів стане для користувача цілим квестом. Саме тому бажано враховувати те, якою ОС користуються клієнти та розробляти рішення, яке задовольнить усіх. На рисунку 1.1.4 зображена візуальна різниця між елементами двох ОС [3].

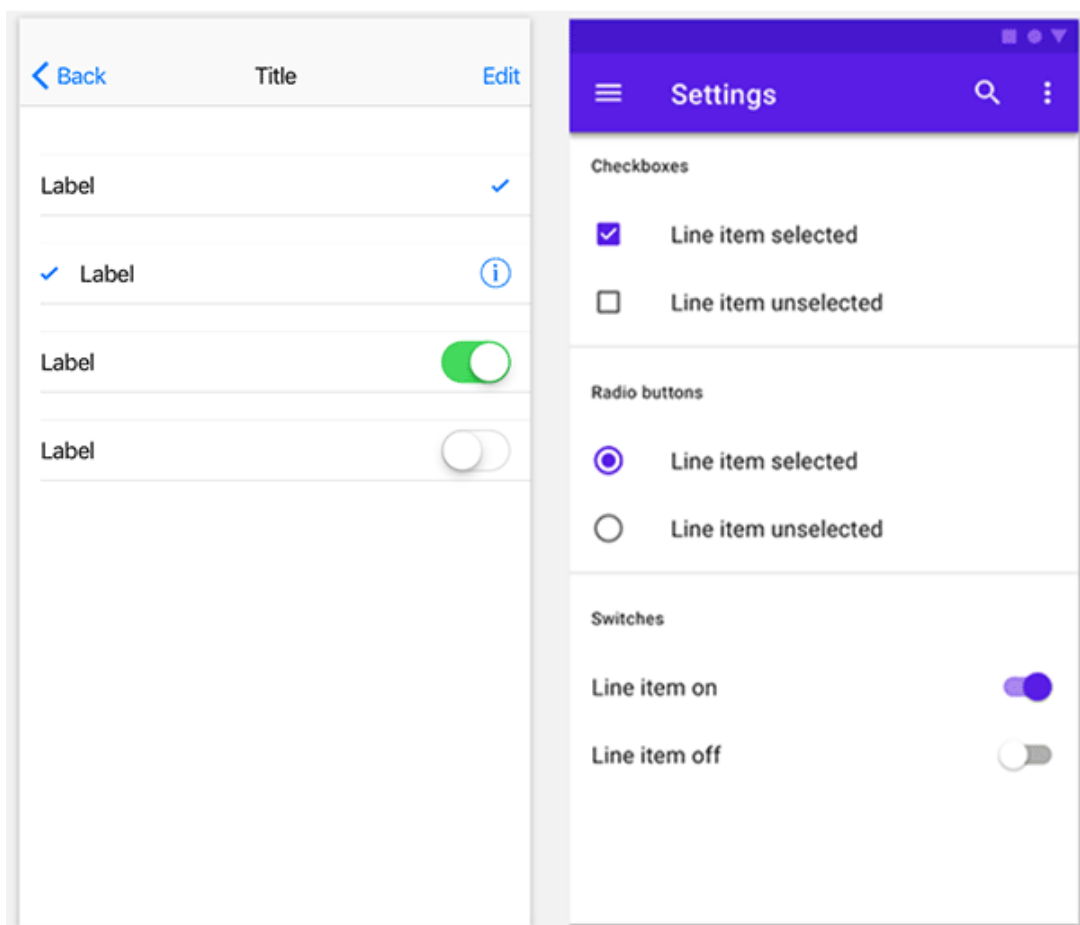


Рисунок 1.1.4 – Візуальна відмінність нативних елементів IOS та Android

[5]

- 6) Привабливість інтерфейсу. Дизайн має бути естетично привабливим для користувача, щоб у нього виникало бажання взаємодіяти з продуктом. Має бути гармонічно підібрана палітра кольорів, елементи мають бути пропорційними, а текст - зручно розташований та зрозумілий.
- 7) Консистентність. Елементи керування, стилі, шрифти та інші UI елементи мають бути однаковими, щоб користувач міг легко орієнтуватись та сприймати інформацію належним чином.
- 8) Тестування. Дизайн додатку необхідно тестувати на реальних користувачах, щоб проаналізувати їх взаємодію, позитивний та негативний досвід, зробити висновки з отриманих результатів та вдосконалити дизайн.

1.2 Етапи проектування UX/UI мобільних додатків

- 1) Аналіз цілей та поставлених задач. Необхідно мати чітке розуміння, для чого створюється даний додаток, які проблеми користувачів він вирішує та що треба зробити для того, щоб замовник отримав бажаний результат.
- 2) Аналіз ЦА. Необхідно визначити, хто є цільовою аудиторією майбутнього продукту, який віковий діапазон, стать, вид діяльності, вповодання та інтереси вона має. Адже дизайн має бути орієнтований на своїх користувачів. Саме тоді продукт буде успішним.
- 3) Аналіз конкурентів. Необхідно детально розглянути конкурентоспроможні додатки на ринку, проаналізувати функціонал, який вони пропонують користувачів, виділити їх недоліки та

переваги та визначити, чим саме можна виділити майбутній продукт та фоні його конкурентів. Нижче зображений приклад таблиці з аналізом конкурентів, який складає команда UX/UI дизайнерів.

Таблиця 1.2 – Приклад аналізу конкурентів

Feature/Company	Our Product	Competitor 1	Competitor 2	Competitor 3	Competitor 4
Strengths	Faster performance due to new technology	Seamless onboarding experience, great design	Uses social media to their advantage	Great design, usability	Emphasizes security
Weaknesses	Not much social media presence	Not much interaction with customers	Documents hard to navigate	Sporadic social media presence	Language is formal, not user-friendly
Pricing	\$800 per month	\$800 per month	\$900 per month	\$850 a month	\$950 a month
Social media	Twitter, Instagram	Blog posts, Twitter	Blog posts, Instagram	Blog posts, Instagram, Twitter	Blog posts
Onboarding experience	Moderate number of steps to sign up	Smooth instructions	Not much support after first step	Seamless, very few steps involved	Moderate number of steps

- 4) Створення User Personas. Необхідно створити картки фіктивної ЦА, зазначивши у них потреби, користувацькі болі, певні характеристики та їх цілі, що стосуються продукту. Дані картки допомагають краще зрозуміти ЦА продукту та розробити додаток, орієнтований на неї. На рисунку 1.2.1 зображений приклад User Persona, розроблений командою UX/UI дизайнерів.

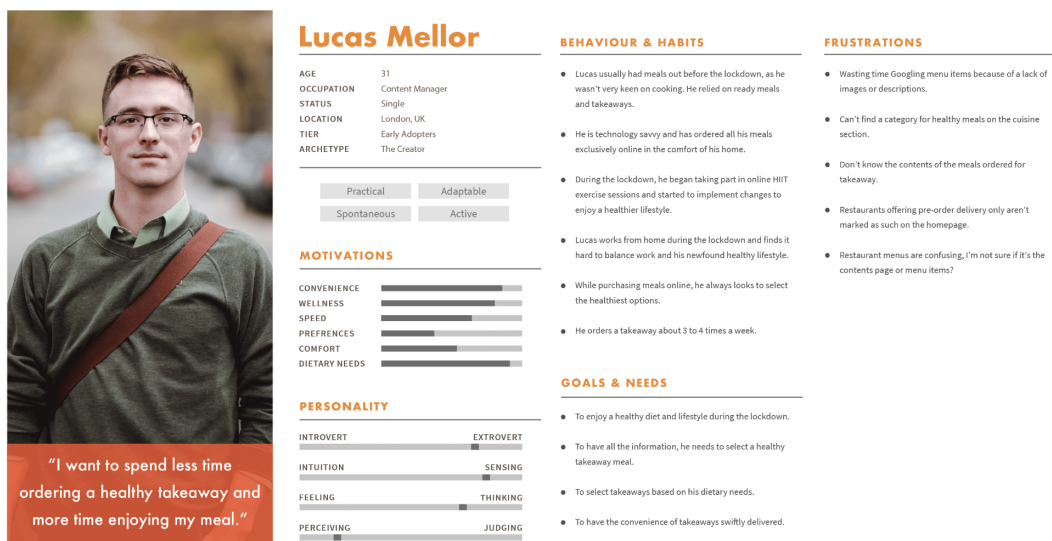


Рисунок 1.2.1 – Приклад картки уявного користувача (User Persona) [10]

5) Проектування інформаційної архітектури. Інформаційна архітектура додатка будується для того, щоб розробити зручну навігацію для користувачів, відобразити переходи між основними розділами та іншими сторінками. У ній визначаються всі навігаційні шляхи додатка, по яким буде проходити користувач. Добре спроектована інформаційна архітектура допомагає користувачам уникнути плутанини та блукання між сторінками та сприяє покращенню їх користувацького досвіду [11]. На рисунку 1.2.2 зображений приклад побудови інформаційної архітектури.

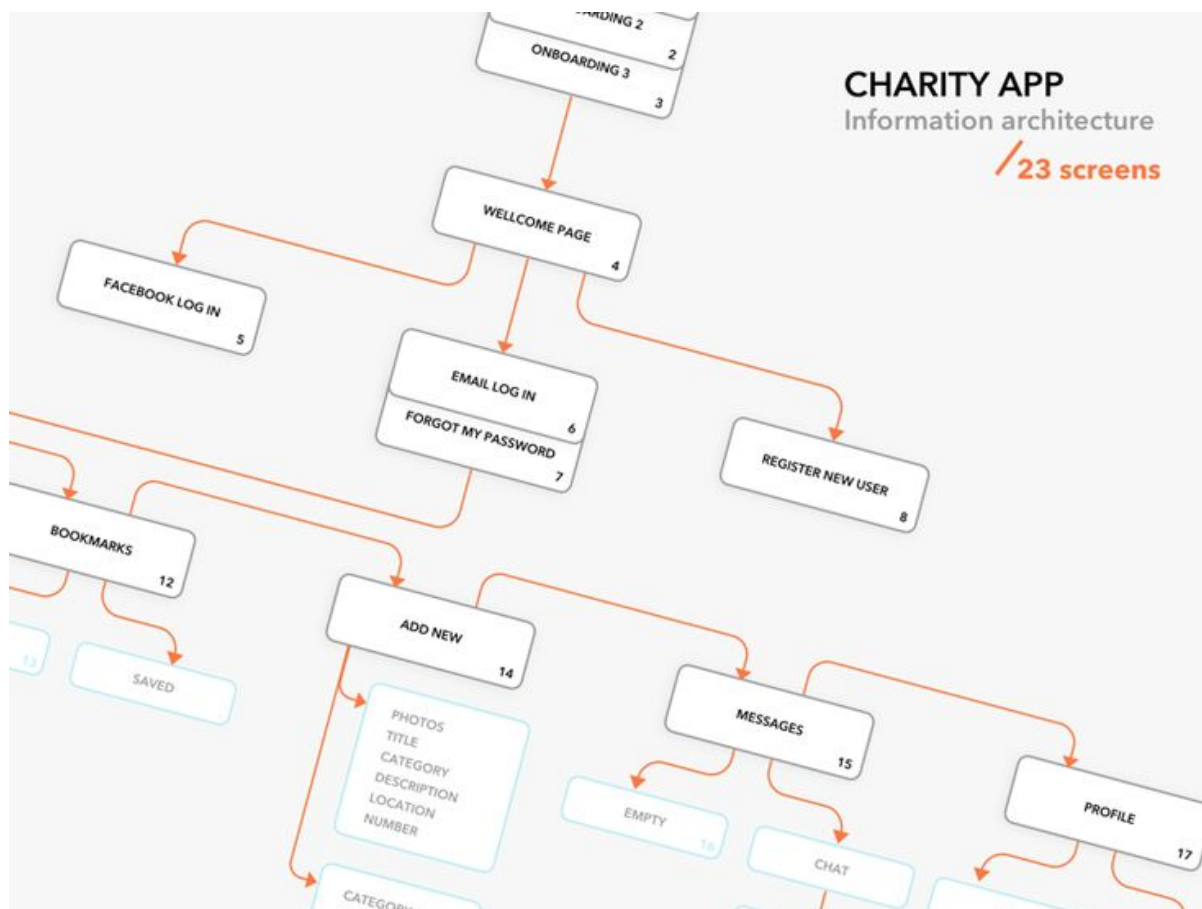


Рисунок 1.2.2 – Фрагмент інформаційної архітектури для благодійного додатка (Application Structure) [12]

6) Проектування рішень (Wireframing). Створення схематичних ескізів сторінок додатка є невід’ємною частиною UX/UI дизайну, адже саме на даному етапі починає визначатись структура, приблизне розміщення UI елементів (кнопок, полів вводу тощо) та як користувач взаємодіятиме з ними. На цьому етапі ми бачимо те, як саме ми побудувати інформаційну архітектуру, та можемо приблизно зобразити, як користувач буде переміщуватись по додатку. Також вайрфрейми допомагають виявити потенційні проблеми та недоліки, чи є навігація зручною та зрозумілою, чи виконуються поставлені задачі [13]. На рисунку 1.2.3 зображений приклад вайрфрейму однієї сторінки та її UI елементів.

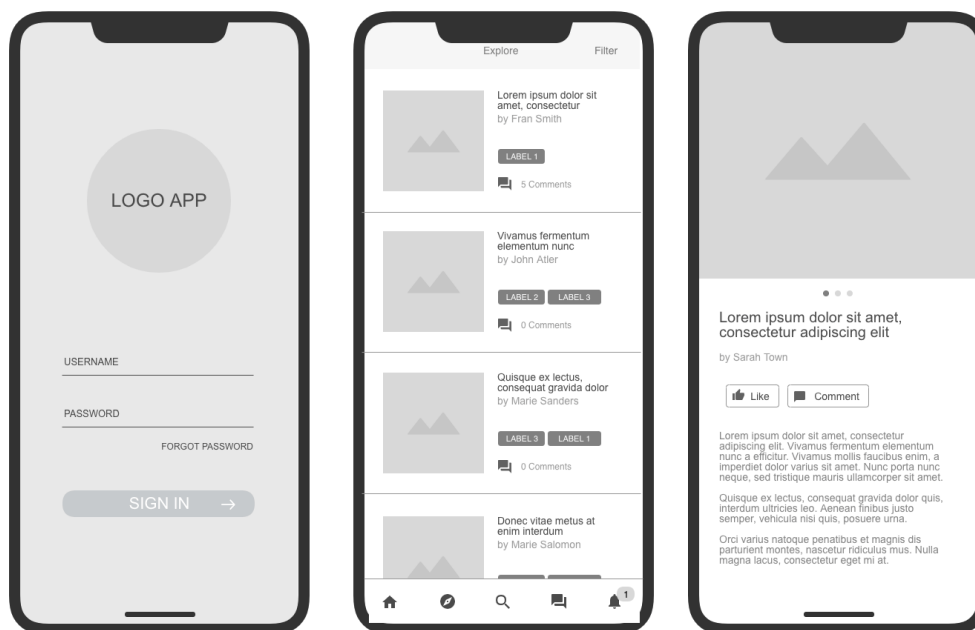
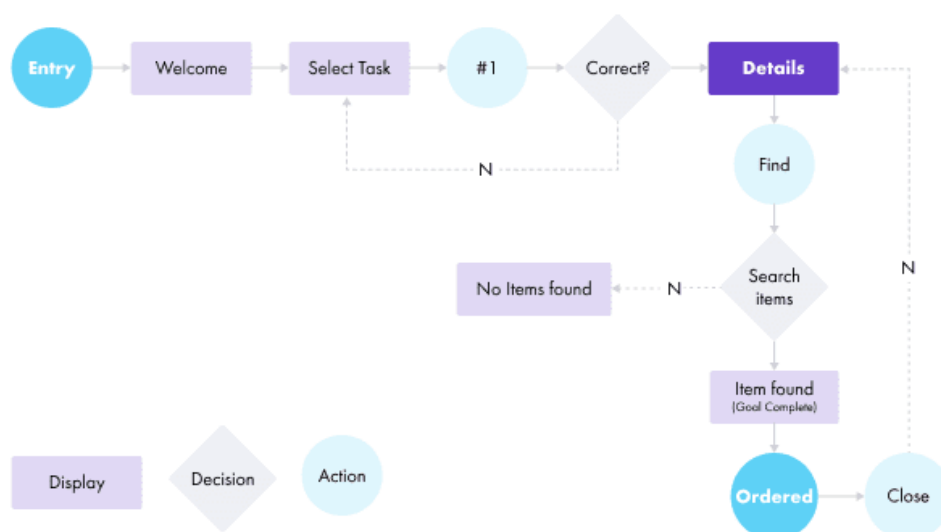
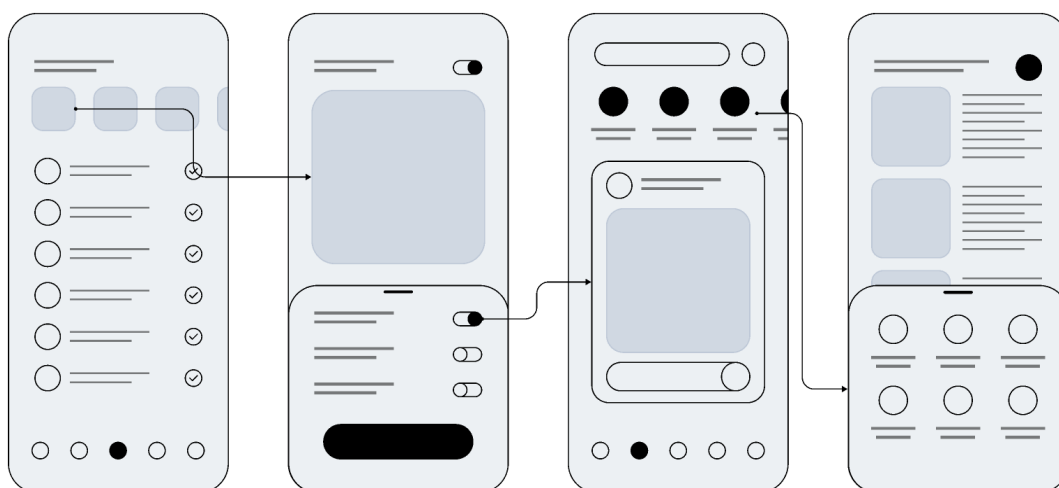


Рисунок 1.2.3 – Приклад проектування рішень для певних сторінок (Wireframes) [14]

7) Проектування сценаріїв взаємодії користувача з додатком (User flows). User flows це візуальний сценарій дій, які виконуватиме користувач на шляху до досягнення своєї цілі. Вони будуються схематично або на основі спроектованих рішень (Wireframes). Вони допомагають проаналізувати користувацьку взаємодію та можливий досвід, а в результаті - розробити покращений функціонал. Тобто, на даному етапі дизайнери аналізують поведінку користувачів. На рисунках 1.2.4 та 1.2.5 зображені User flows двох типів, а саме схематичні та з використанням спроектованих рішень.



1.2.4 – Приклад сценарію (схематичне зображення) [15]



1.2.5 – Приклад сценарію із застосуванням спроектованих рішень [16]

- 8) Створення прототипів. Розробка UI елементів. На даному етапі необхідно створити прототипи екранів майбутнього додатка на основі раніше спроектованих рішень (Wireframes). Тобто, створюється візуальний образ фінального продукту із застосуванням кольорової палітри, обраних шрифтів, стилів, із точним розміром та розміщенням UI елементів (карток, кнопок, контенту тощо).
- 9) Прототипування. Прототипування макетів надає нам можливість перевірити, як працюватиме спроектований дизайн у реальному

житті. Завдяки цьому процесу дизайнери можуть протестувати функціональність додатка, його елементи та правильність їх розміщення задля зручності користувачів. Також прототипування полегшує комунікацію UX/UI дизайнерів з розробниками та клієнтами, адже за його наявності вони можуть не тільки розказати про певний функціонал, а й продемонструвати його, щоб розробникам стало зрозуміло, про реалізацію чого йде мова, а клієнт мав можливість побачити свій майбутній продукт у роботі. На рисунку 1.2.6 зображений приклад прототипування двох макетів.

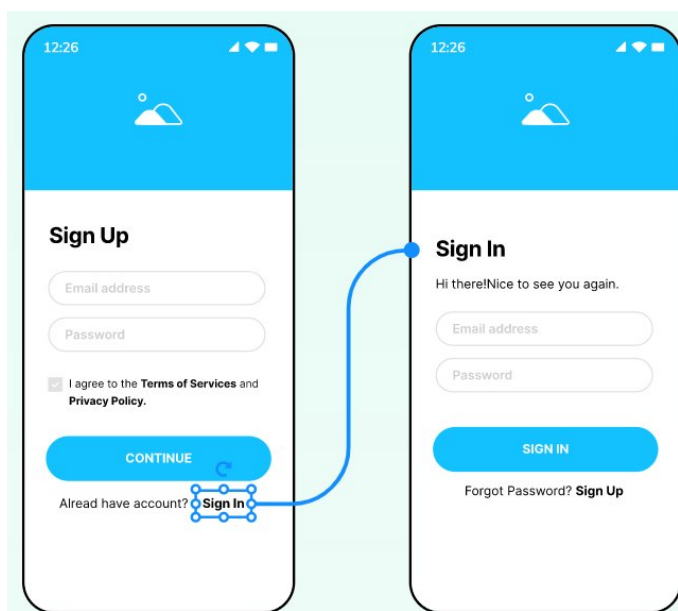


Рисунок 1.2.6 – Приклад прототипування макетів авторизації та реєстрації. Перехід між сторінками (prototyping) [17]

- 10) Тестування. Тестування прототипів на реальних користувачах = дослідження користувацького досвіду, адже з'являється можливість поспостерігати за користувачем, як він досягає певних цілей, які емоції отримує під час проходження сценаріїв, тобто дизайнери мають можливість якісно проаналізувати досвід взаємодії користувача та зробити певні висновки, щоб покращити інтерфейс.

1.3 Зручність використання обраного інтерфейсу відповідно до ЦА

За статистикою 85% українського населення надає перевагу смартфонам та мобільним додатками, а 91% власників смартфонів користуються мобільними додатками на постійній основі [18].

Розробка мобільного додатка для реалізації функціоналу, який спрямований саме на допомогу постраждалим людям через війну, має кілька переваг, а саме:

- 1) Доступність. На сьогоднішній день мобільні додатки є доступними майже на всіх більш менш нових смартфонах, а отже можна зробити висновок, що постраждала через війну людина скоріше за все матиме доступ до цього додатку, незважаючи на те, де вона зараз проживає або знаходиться. Адже багато людей на даний момент не мають можливості повернутись додому через певні обставини.
- 2) Комунікація між користувачами. У додатку може бути наявний функціонал чату, який забезпечить можливість користувачам комунікувати між собою та домовлятися за обмін речами або персональну зустріч.
- 3) Геолокація. Додаток може використовувати геолокацію, завдяки якій користувачі зможуть шукати надавати поміч потребуючим в окрузі. Також в додатку може бути наявна мапа з пунктами допомоги, на якій користувач матиме можливість переглянути, що є навколо.
- 4) Статистика. У мобільному додатку можна збирати інформацію про користувачів за допомогою опитувань, певних форм, додаткових полів під час реєстрації. Ця інформація може бути корисною як для бізнесу, так і для вдосконалення продукту.

- 5) Масштабованість. Мобільний додаток можна буде з часом вдосконалювати, враховуючи нові потреби користувачів, наприклад, додаючи певний функціонал та можливості.

РОЗДІЛ 2. ОГЛЯД UX/UI РІШЕНЬ АНАЛОГІВ

2.1 Мобільний додаток “Banderolka”

“Banderolka” - мобільний застосунок від компанії Geniusee [20], створений на об’єднання людей між собою, на взаємопоміч та взаємодопомогу. Гаслом додатка є “Взаємодопомога тут”. Даний продукт можуть використовувати не тільки потребуючі допомоги люди, але й ті, хто бажає допомогти оточуючим [19]. Використання даного застосунку є безкоштовним.

Проведемо аналіз головних сторінок та основних сценаріїв додатка.

1. Авторизація та реєстрація.

Переваги. Сценарії авторизації та реєстрації зручні, є можливість перейти до авторизації/реєстрації/відновлення паролю за допомогою кнопок.

Недоліки. Немає можливості авторизуватись або ж зареєструватись за допомогою аккаунтів Google, Facebook тощо. Даний функціонал є дуже важливим, адже це максимально швидко та зручно, саме такому способу входу в систему надають перевагу користувачі. На рисунку 2.1.1 зображені сторінки авторизації даного додатка.

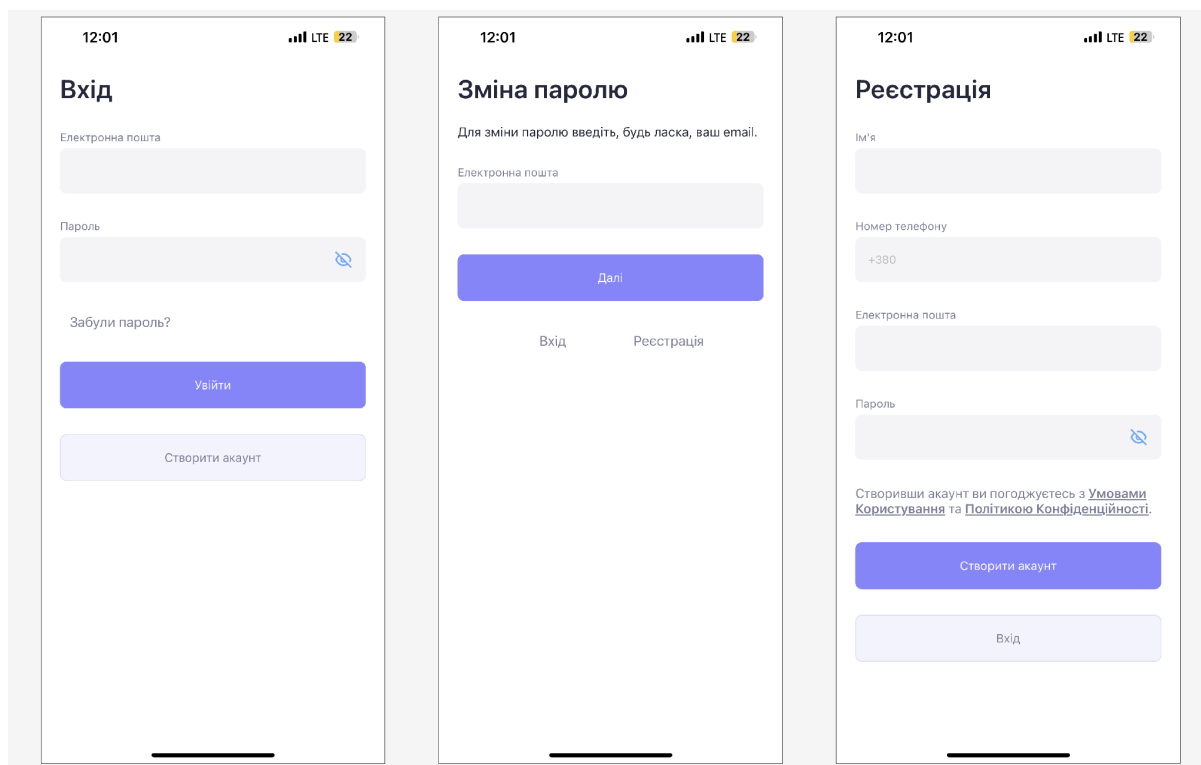


Рисунок 2.1.1 – Сторінки авторизації додатка “Banderolka”

2. Сторінка запитів.

Переваги. Наявність фільтрації запитів та їх збереження.

Недоліки. Фільтрація не є зручною для користувача, адже кожна характеристика грає роль окремого фільтру, тобто користувачу прийдеться натискати на кожну по черзі та налаштовувати окремо. Якби усі пункти фільтрації були б реалізовані на одній сторінці, користувач витрачав би менше часу та дій на їх заповнення. Кожен фільтр необхідно відкривати окремо та зберігати зміни.

Недолік фільтрації по регіону. Області указані звичайним списком, відсутній пошук, необхідно гортати та шукати очима, що ускладнює вибір користувачам.

Під час збереження оголошення спочатку не зрозуміло, куди саме вони зберігаються, тільки з часом помічається те, що збережені

оголошення, оголошення в обробці та усі разом це все окремі розділи у нижньому барі. Користувачу було б зручніше працювати з оголошеннями, знаходячись в одному розділі, не виконуючи додаткових дій.

На рисунку 2.1.2 зображені сторінки оголошень та фільтрації додатка.

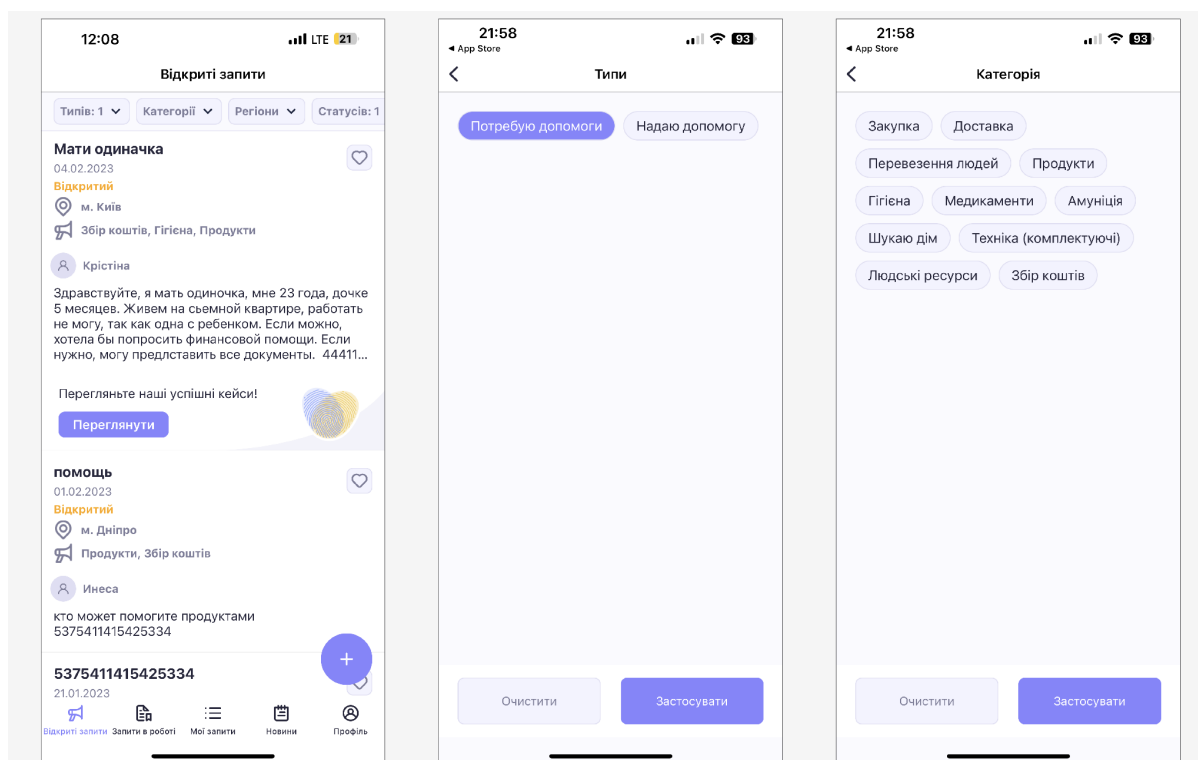


Рисунок 2.1.2 – Сторінки оголошень та фільтрації додатка “Banderolka”

3. Сторінка оголошення.

Переваги. Наявний акцент на назві оголошень, також є можливість написати повідомлення власнику оголошення. Також при натисканні на номер телефону з’являється можливість зателефонувати. Наявна можливість поділитись обраним оголошенням.

Недоліки. Замало акцентів на сторінці, хотілось би бачити акцент на категорії оголошення та місці знаходження. Також дуже важко

зрозуміти, що номер телефону є клікабельним, оскільки він ніяк не виділений та немає відповідної кнопки, номер нічим не відрізняється від вищевказаного імені власника, яке не є активним. Випадково натрапила на даний функціонал, натиснувши на нього.

На рисунку 2.1.3 зображена сторінка оголошення та спливаюче вікно зв'язку з власником у додатку “Banderolka”.

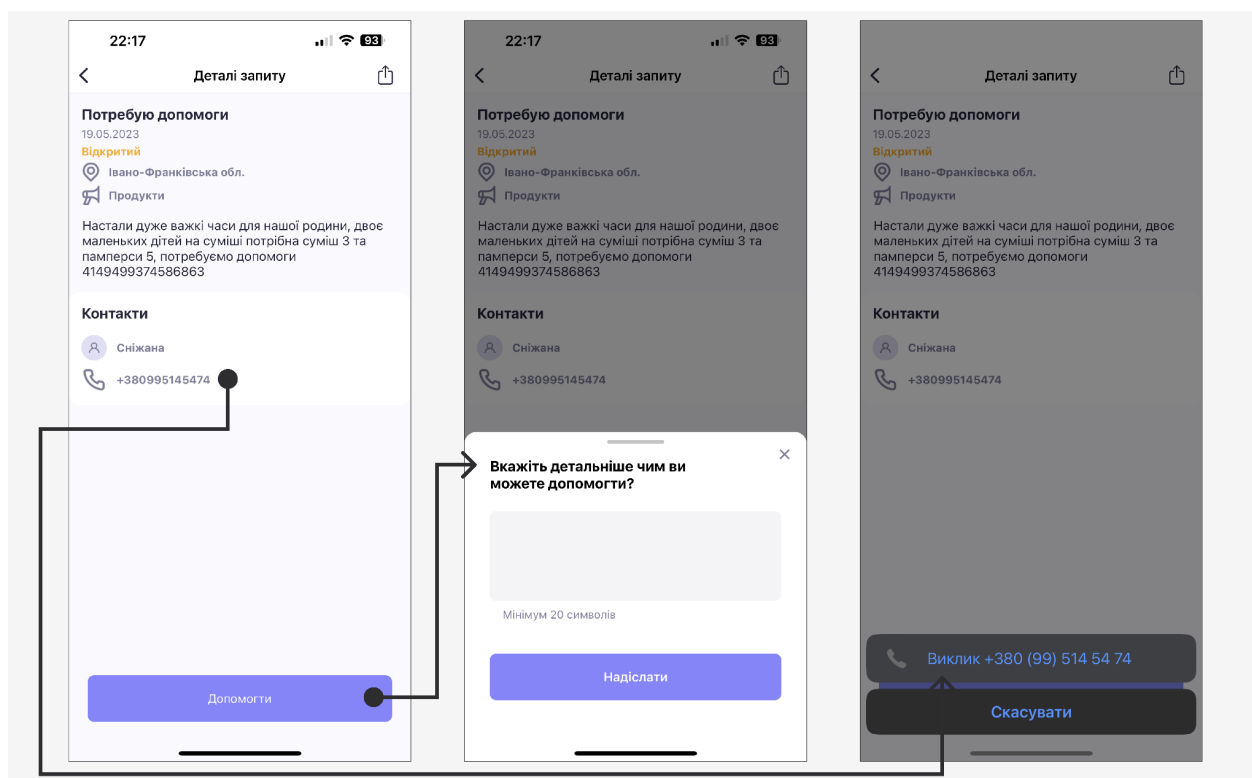


Рисунок 2.1.3 – Сторінка з оголошеннями та рор-ups зв'язку та виклику у додатку “Banderolka”

4. Сценарій створення оголошення.

Переваги. Наявний функціонал вибору типу категорії оголошення. Можливий вибір регіону та наявність поля для додаткової інформації. Наявна можливість завантаження файлів.

Недоліки. Відсутні будь-які обмеження для полів вводу, немає мінімальної довжини тощо, тобто користувачі можуть вказувати будь-що у назві або у полі для додаткової інформації.

На рисунку 2.1.4 зображена сторінка додавання оголошення та наявні додаткові функції.

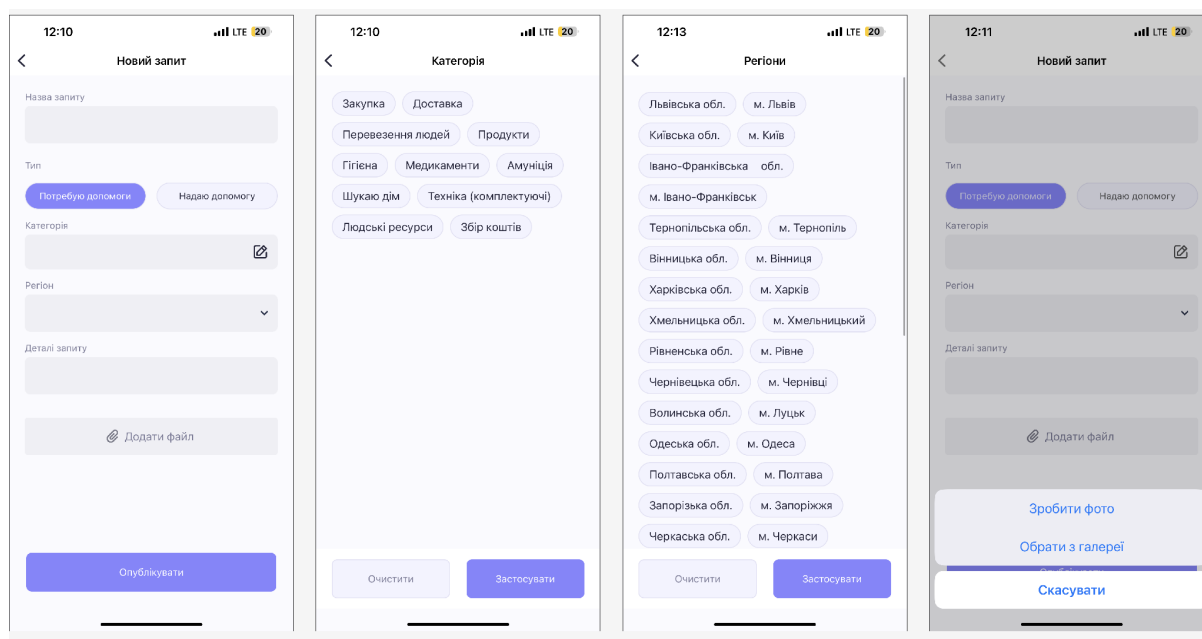


Рисунок 2.1.4 – Сторінка створення оголошення додатка “Banderolka” та її додаткові

5. Профіль.

Переваги. Наявна можливість заміни персональних даних користувача та додавання фотографії. Є можливість вибрати певні категорії, що є цікавими. Наявна зміна регіону. Аккаунт можна видалити у будь-який момент, без зайвих складнощів.

Недоліки. Розміщення пунктів “Тип користувача/Категорія/Регіон” у розділі “Сповідання”, оскільки вони не відносяться до них.

На рисунку 2.1.5 зображені сторінка профілю, його редагування та розділ сповіщень мобільного додатка “Banderolka”.

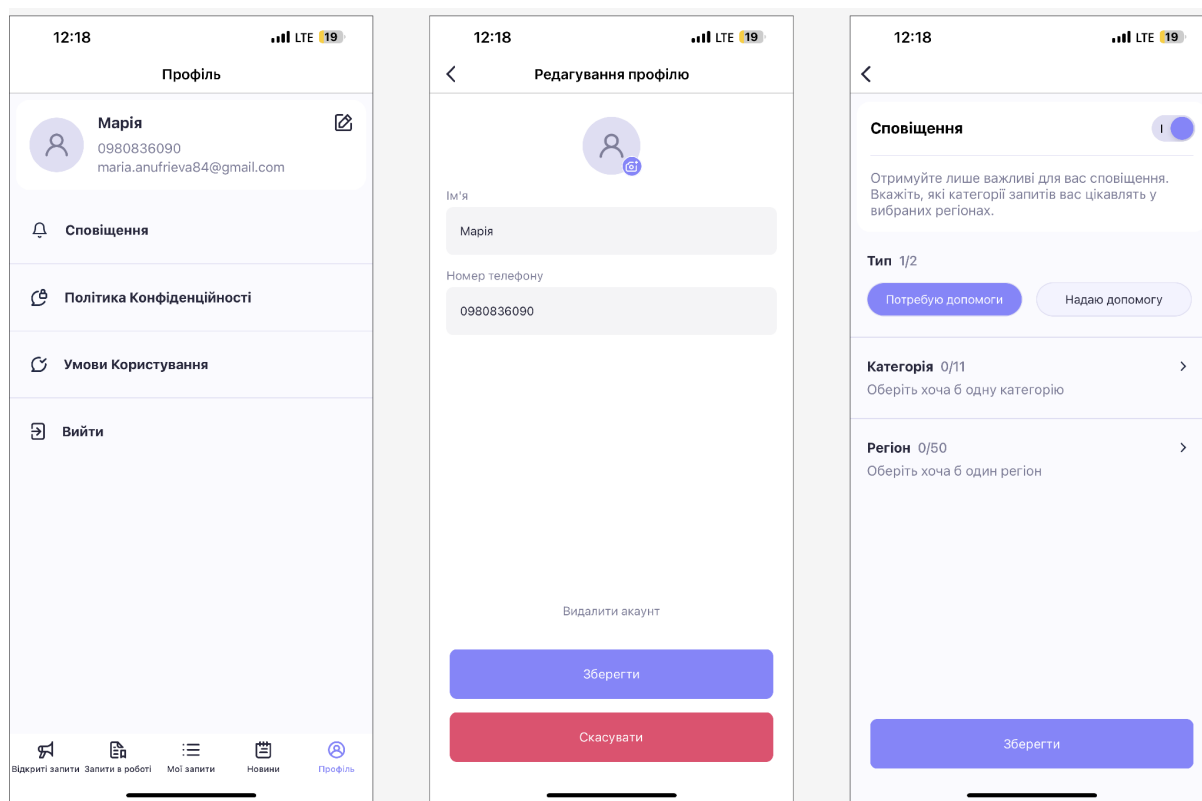


Рисунок 2.1.5 – Сторінка профілю, його редагування та розділ сповіщень мобільного додатка “Banderolka”

Мої висновки щодо реалізації функціоналу даного продукту. Додаток “Banderolka” не є досконалим зі сторони UX/UI дизайну, але є найкращим рішенням на ринку на даний момент часу.

2.2 Мобільний додаток “ЕРАУНД”

“ЕРАУНД” - це інтерактивна мапа з оголошеннями від людей, які потребують або надають допомогу [21]. Використання даного застосунку є безкоштовним.

Проведемо аналіз головних сторінок та основних сценаріїв додатка.

1. Мапа та встановлення місця розташування користувача.

Переваги. Наявність функціоналу автоматичного визначення місцеположення.

Недоліки. Відсутність можливості введення місця розташування. Визначення місцезнаходження по карті є не дуже зручним та навіть складуватим в орієнтуванні.

На рисунку 2.2.1 зображена сторінка мапи додатка “ЕРАУНД”.

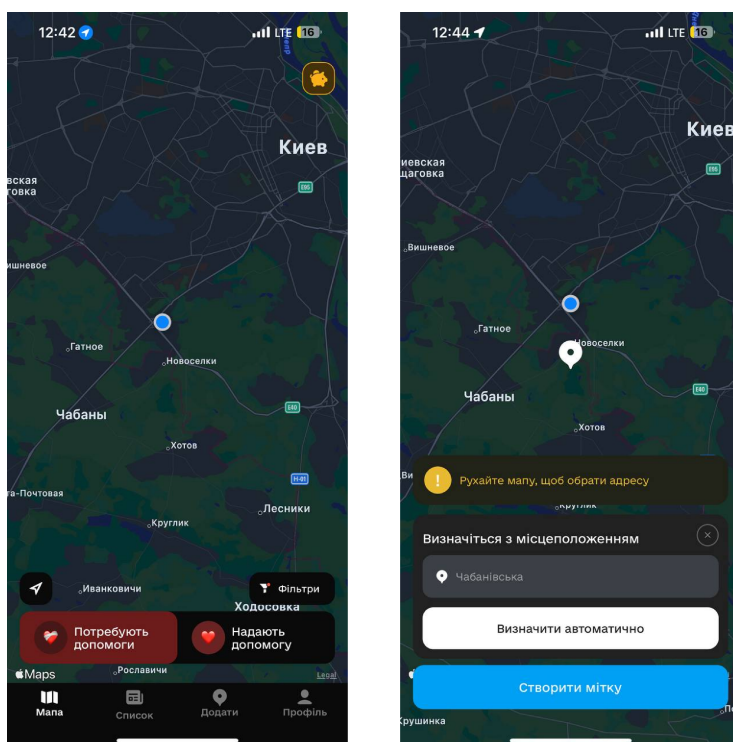


Рисунок 2.2.1 – Сторінка мапи додатка “ЕРАУНД”

2. Створення оголошення та його картка у профілі.

Переваги. Гарний дизайн карток вибору категорії. Наявність вибору типу оголошення. Можливість додавання опису. Карткою оголошення можна поділитись через соціальні мережі. Наявне прокладання маршруту. Оголошення на мапі відображається з іконкою категорії.

Недоліки. Усі контактні поля є опціональними, що дуже дивно, оскільки користувач може нічого не заповнювати, а в результаті інші не матимуть способу зв'язатись з власником оголошення. Відсутня сторінка оголошення. Також не побачила жодного оголошення на

мапі, крім свого, можливо додатком не користуються. Не мала можливості переглянути та проаналізувати їх картки.

На рисунку 2.2.2 зображена сторінка мапи профілю зі створеним оголошенням у додатку “ЕРАУНД”.

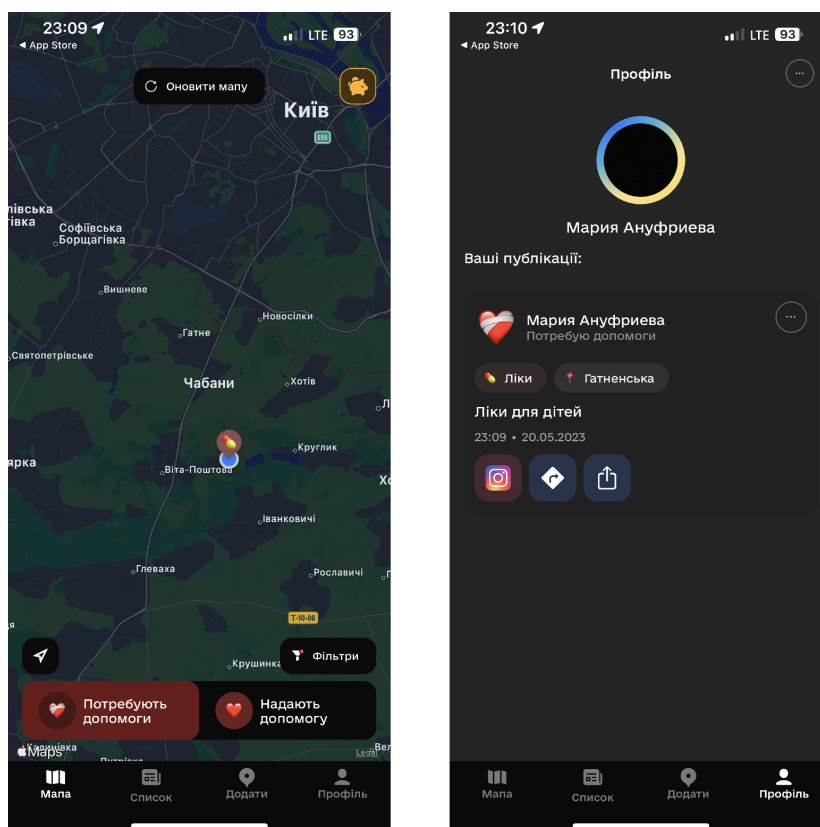


Рисунок 2.2.2 – Сторінка мапи та профілю зі створеним оголошенням у додатку “ЕРАУНД”

3. Профіль.

Переваги. Відсутні. У профілі можна переглянути власні оголошення та змінити ім'я.

Недоліки. Відсутня персональна інформація користувача. Дуже примітивний функціонал.

2.3 Мобільний додаток “TURBOTA”

“TURBOTA” -платформа взаємопомічі для тих, хто потребує допомоги або бажає допомогти.

Проведемо аналіз головних сторінок та основних сценаріїв додатка.

1. Авторизація.

Переваги.Наявний вибір ролі користувачем на першому етапі авторизації. Авторизація основана на введенні номера мобільного телефону користувача, який перевіряється надсиланням повідомлення з кодом на вказаний номер.

Недоліки. Додаток не збирає ніяких даних про користувача (ім'я, стать, електронна адреса, місце розташування тощо), окрім мобільного телефону, тобто на даному етапі незрозуміло, яким чином ці дані будуть потрапляти в оголошення, як проводиться аналіз користувачів даного продукту.

На рисунку 2.3.1 зображені сторінки авторизації додатка “TURBOTA”.

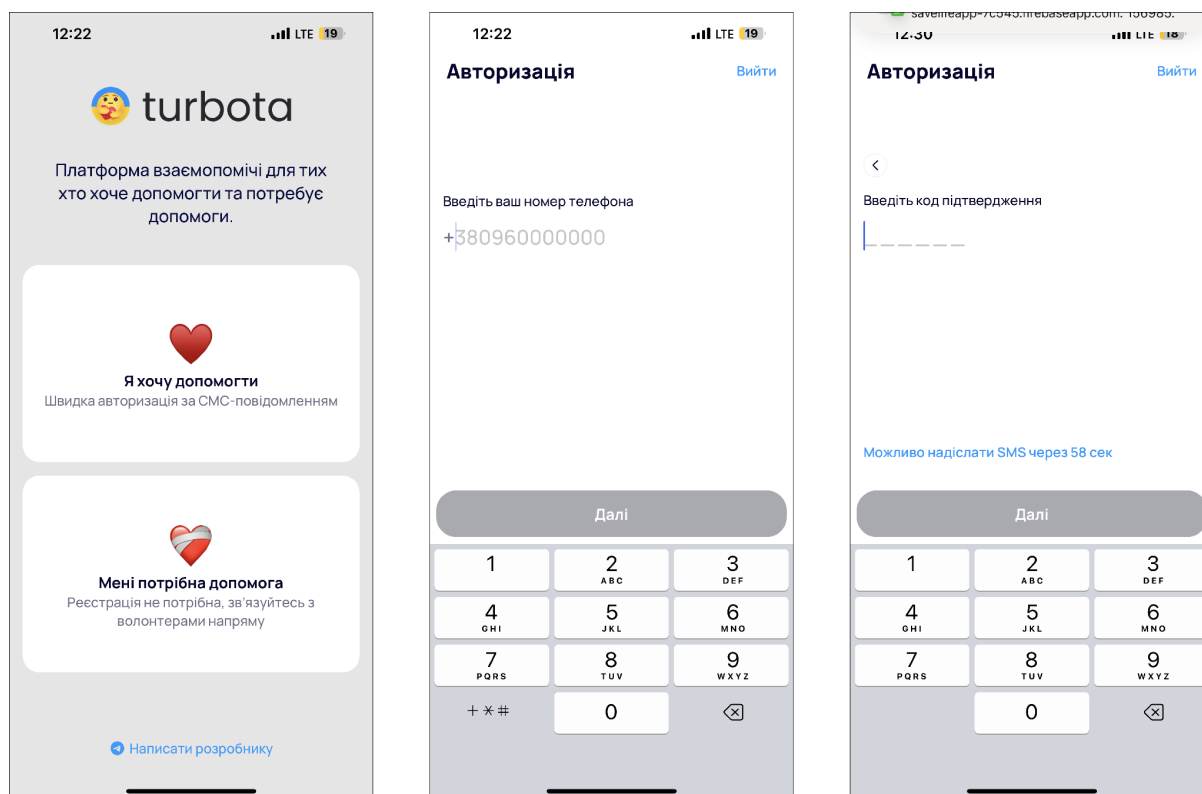


Рисунок 2.3.1 – Сторінки авторизації додатка “TURBOTA”

2. Сторінка з власними оголошеннями та сценарій додавання нового.

Переваги. Сторінка з власними оголошеннями має дизайн стану, коли оголошення відсутні. Наявний зручний вибір категорії та варіанту зв'язку з власником оголошення. Є можливість визначити місце розташування автоматично. У полях вводу є підказки, що саме потрібно вводити користувачу.

Недоліки. Поля, які містять інформацію, потрібно заповнювати кожен раз, коли користувач бажає створити оголошення, адже відсутній профіль, де можна було їх вказати та використовувати по замовчуванню. Коли поле вводу адреси знаходиться в активному стані, відсутня можливість відмінити ввід, користувач може лише визначити місцезнаходження автоматично або ж взагалі вийти, перервавши сценарій.

На рисунку 2.3.2 зображений сценарій додавання нового оголошення у додатку “TURBOTA”.

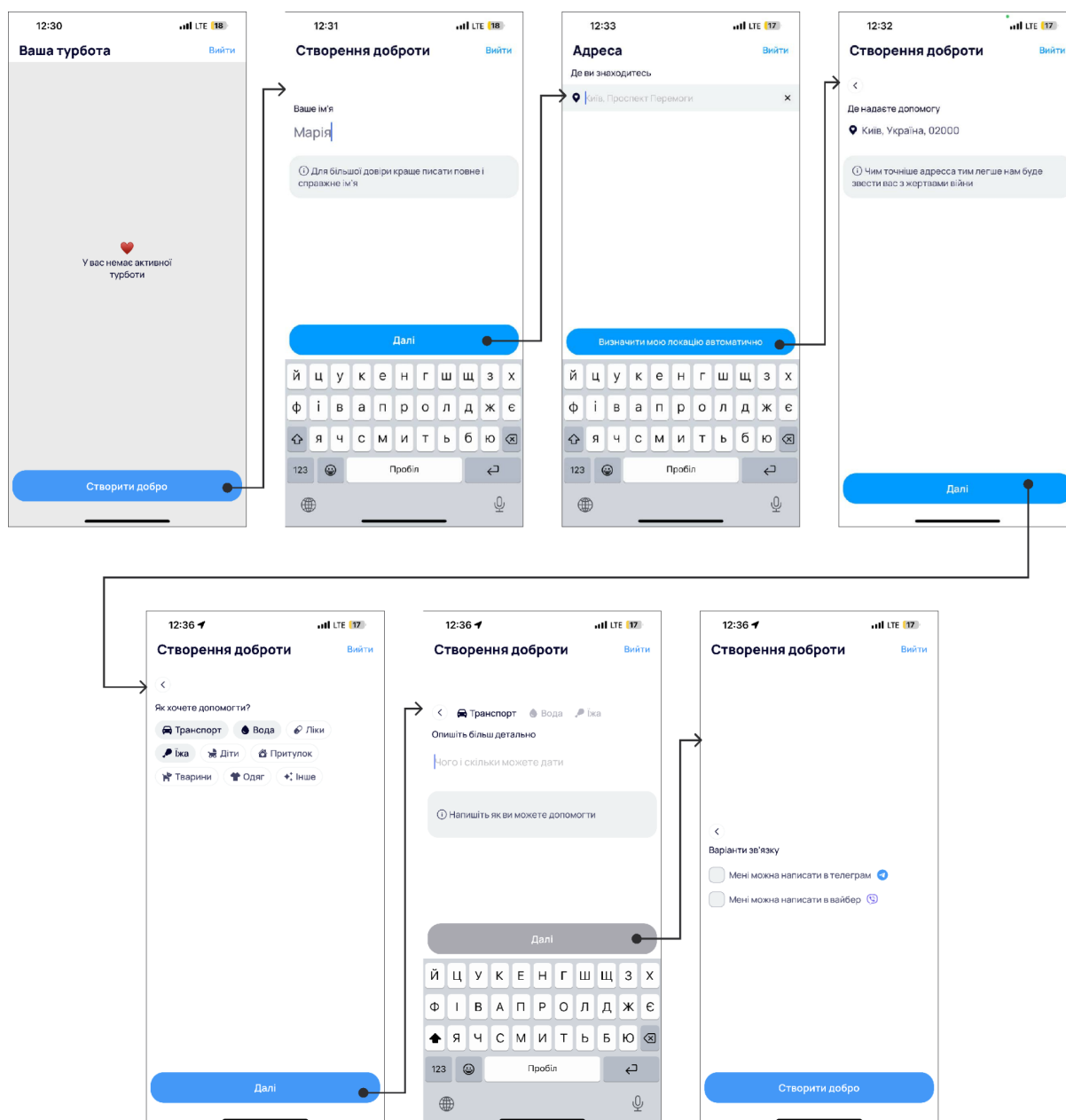


Рисунок 2.3.2 – Сценарій додавання нового оголошення у додатку “TURBOTA”

3. Головна сторінка та фільтрація. Сторінка оголошення

Переваги.

Головна сторінка: наявна фільтрація; роблений акцент на категорії оголошення; якщо вказана адреса у полі фільтрації, відображається приблизна відстань до місця розташування власників оголошень.

Сторінка оголошення: є декілька варіантів зв'язку з власником оголошення; активний номер телефону.

Недоліки.

Головна сторінка: на картках пріоритетний акцент зроблений на імені власника оголошення, хоча воно є не першою по важливості інформацією, на відміну від назви оголошення.

Сторінка оголошення: відсутня можливість зв'язку з іншими користувача у самому додатку; дивно зроблений акцент на місці знаходження, який розміщений у верхньому барі на даній сторінці, адже користувач шукає очима адресу у тілі сторінки та не очікує побачити її у барі, де зазвичай розміщується назва сторінки та певні контекстні інструменти.

На рисунку 2.3.3 зображені головна сторінка з фільтрацією та сторінка оголошення додатка “TURBOTA”.

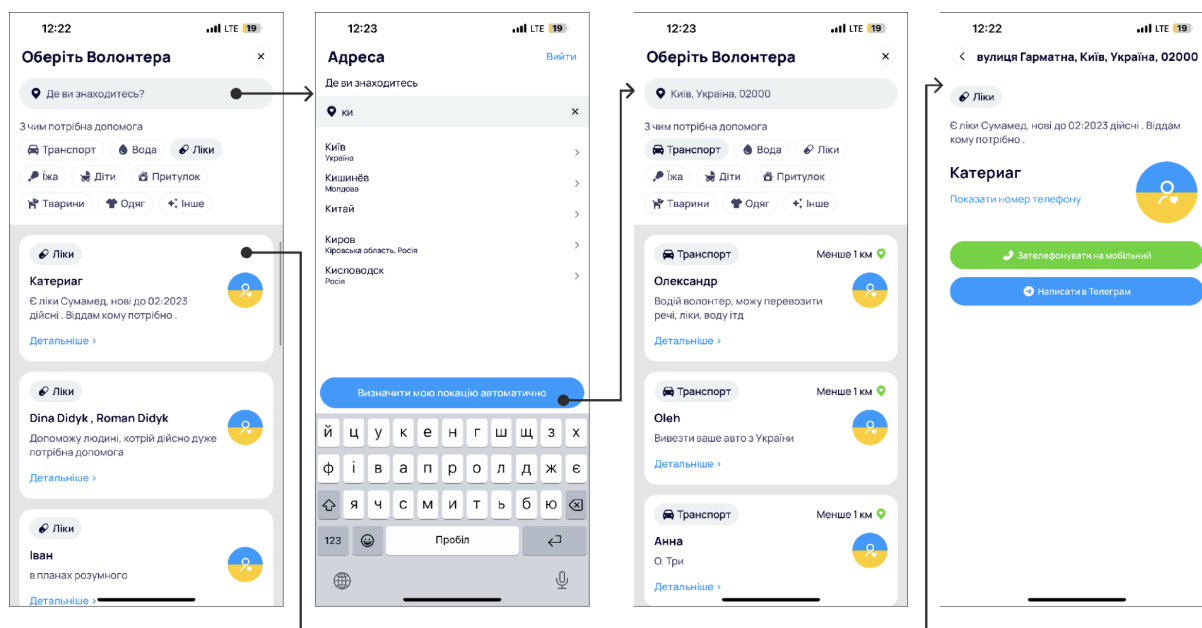


Рисунок 2.3.3 – Головна сторінка з фільтрацією та сторінка оголошення додатка “TURBOTA”

2.4 Висновки щодо функціональності аналогів

Проаналізувавши функціональні можливості конкурентних додатків, була сформована така порівняльна таблиця (рис. 2.4):

Таблиця 2.4 – Порівняльна таблиця функціональності серед конкурентів

Функціональності додатку	Banderolka	Turbota	ЕРАУНД	Notes
Онбординг				-
Авторизація/Реєстрація	●	●		-
Авторизація через соціальні мережі				Відсутність функціоналу робить авторизацію менш зручною для користувачів.
Відновлення паролю	●			-
Запити (пости) допомоги	●	●	●	Категорії, фотографії, коротка інформація, контакти власника, дата публікації.
Створення запиту допомоги	●	●	●	-
Збереження запитів допомоги	●			Важлива частина. Користувач може втратити запити, які його зацікавили.
Комунікація з користувачем (дзвінок та інше)		●		Це може бути не обов'язково, якщо користувач хоче швидко зателефонувати, не заходячи в чат.
Чат				Багато користувачів вважають за краще писати текстові повідомлення, а не дзвонити незнайомцям.
Фільтрація оголошень	●	●		Зручний і швидко доступний функціонал.
Пошук місцезнаходження. Вибір регіону	●	●		Це простий спосіб знайти те, що потрібно користувачеві. Ми можемо пропонувати варіанти, поки користувач набирає текст на клавіатурі.
Наявність карти. Вибір регіону	●		●	Незручний у використанні

РОЗДІЛ 3. РОЗРОБКА UX/UI МОБІЛЬНОГО ЗАСТОСУНКУ

3.1 Проектування інформаційної архітектури додатка (information structure)

Для мобільного додатка “Helper” була спроектована інформаційна архітектура. Вона вміщує у собі всі сторінки, з якими взаємодіятиме користувач, та переходи між ними. На рисунку 3.1 зображений фрагмент інформаційної архітектури мобільного додатка “Helper”, оскільки розміщення її у повному вигляді є неможливим через значні розміри.

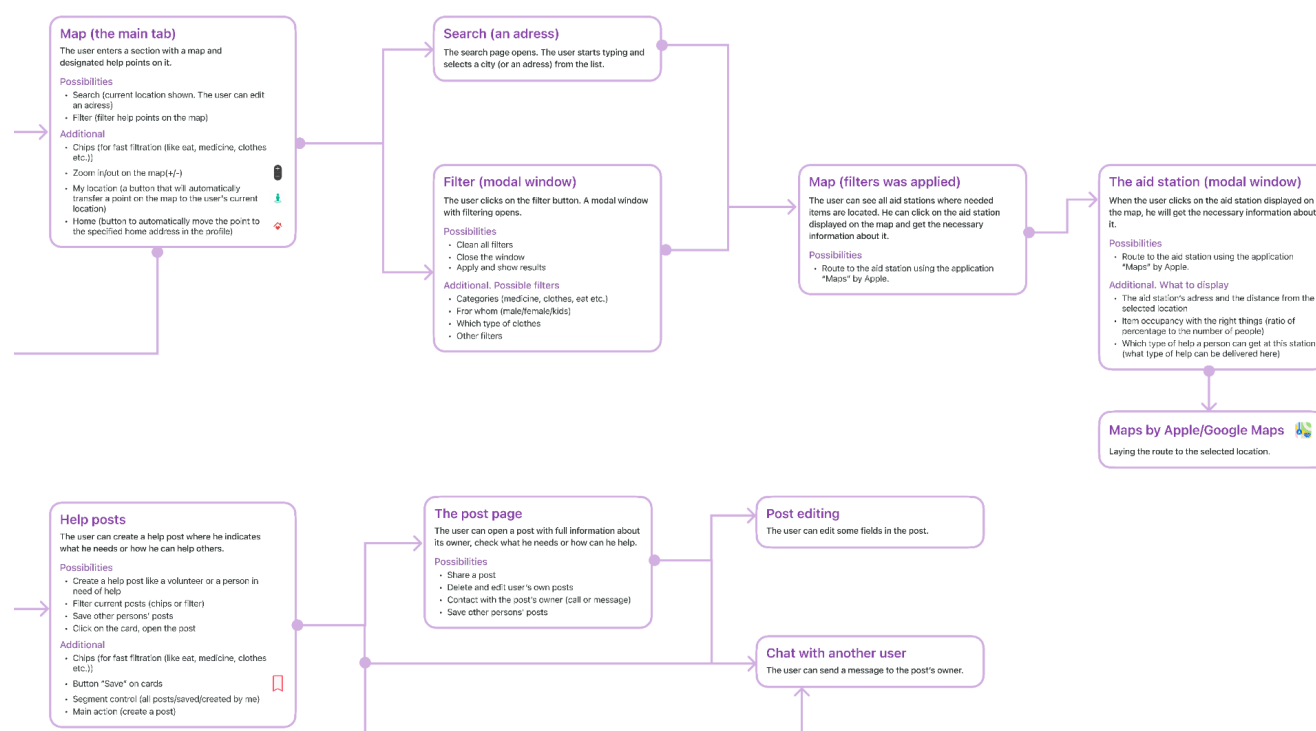


Рисунок 3.1 – Інформаційна архітектури мобільного додатка “Helper” [24]

3.2 Аналіз користувачів та створення User Personas

Було проаналізовано типи майбутніх користувачів та створено дві User Personas. Перший тип - людина, яка потребує допомоги. Другий тип - людина, яка має бажання допомогти оточуючим. Картки User Personas розміщені у додатку А.

3.3 Проектування рішень (wireframes)

Було спроектовано більше 50 рішень у чорно-білому кольорі, зі схематичним зображенням функціоналу та контенту. На рисунку 3.3 зображено приклад проектування рішень для певних сторінок мобільного додатка “Helper”.

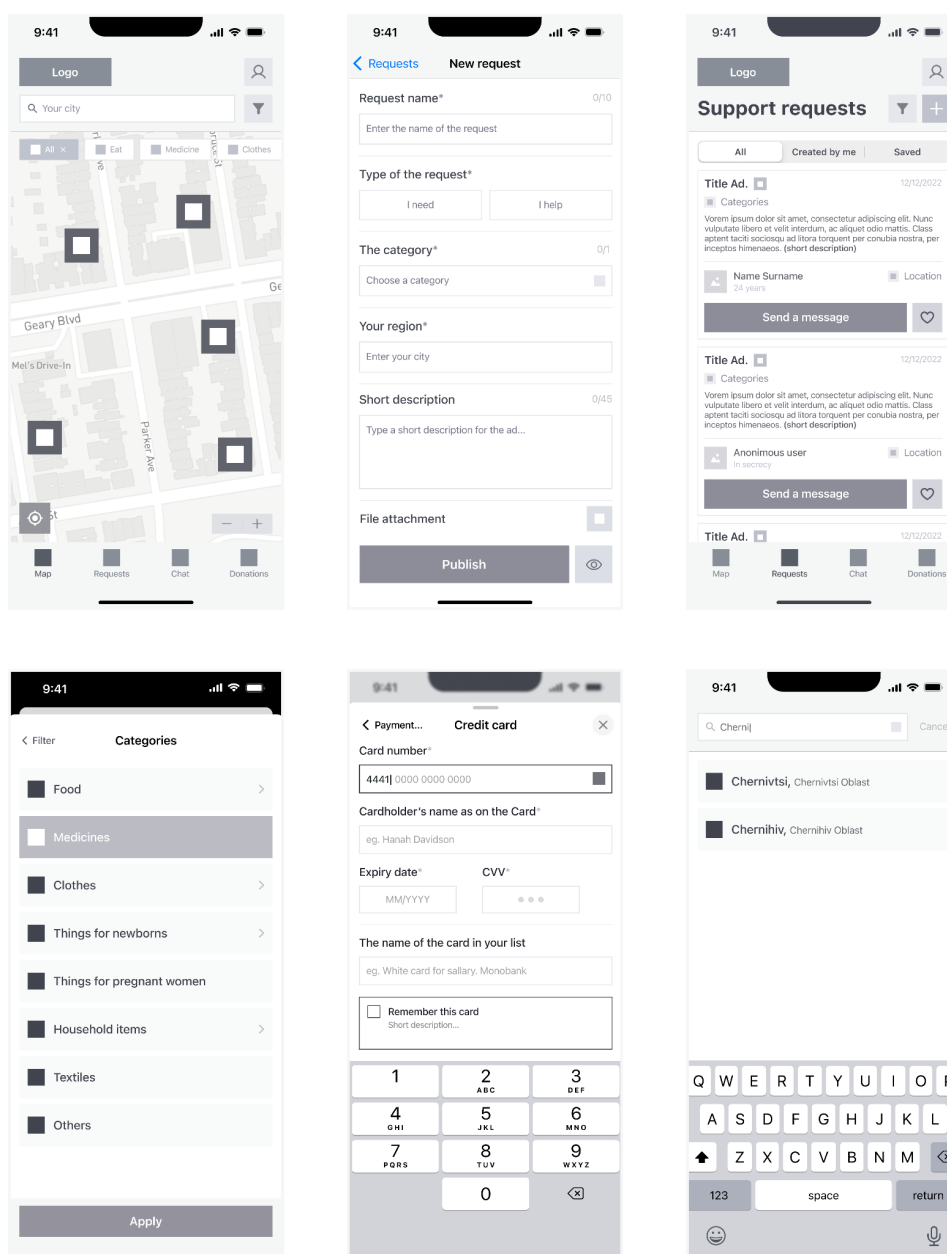


Рисунок 3.3 – Приклад проектування рішень для певних сторінок мобільного додатка “Helper”

3.4 Проектування сценаріїв взаємодії користувача із застосунком (user flows)

Було розроблено сценарії взаємодії на основі спроектованих рішень, щоб проаналізувати користувацький досвід та якість навігації. Приклад сценарію взаємодії розміщений у додатку Б.

3.5 Створення прототипів із додаванням стилістики, кольору та контенту (UI елементів)

- 1) Сторінки онбордингу (Onboarding). Це сторінки, що бачить новий користувач, запустивши додаток. Було прийнято рішення продемонструвати головний функціонал додатка, таким чином ознайомивши його з можливостями. Користувач завжди матиме можливість пропустити даний етап, натиснувши на кнопку “Skip”.

Він складатиметься з 3 етапів:

- демонстрація мапи з пунктами допомоги;
- демонстрація карток оголошень;
- демонстрація карток зборів.

На рисунку 3.3.1 зображені сторінки онбордингу додатка “Helper”.

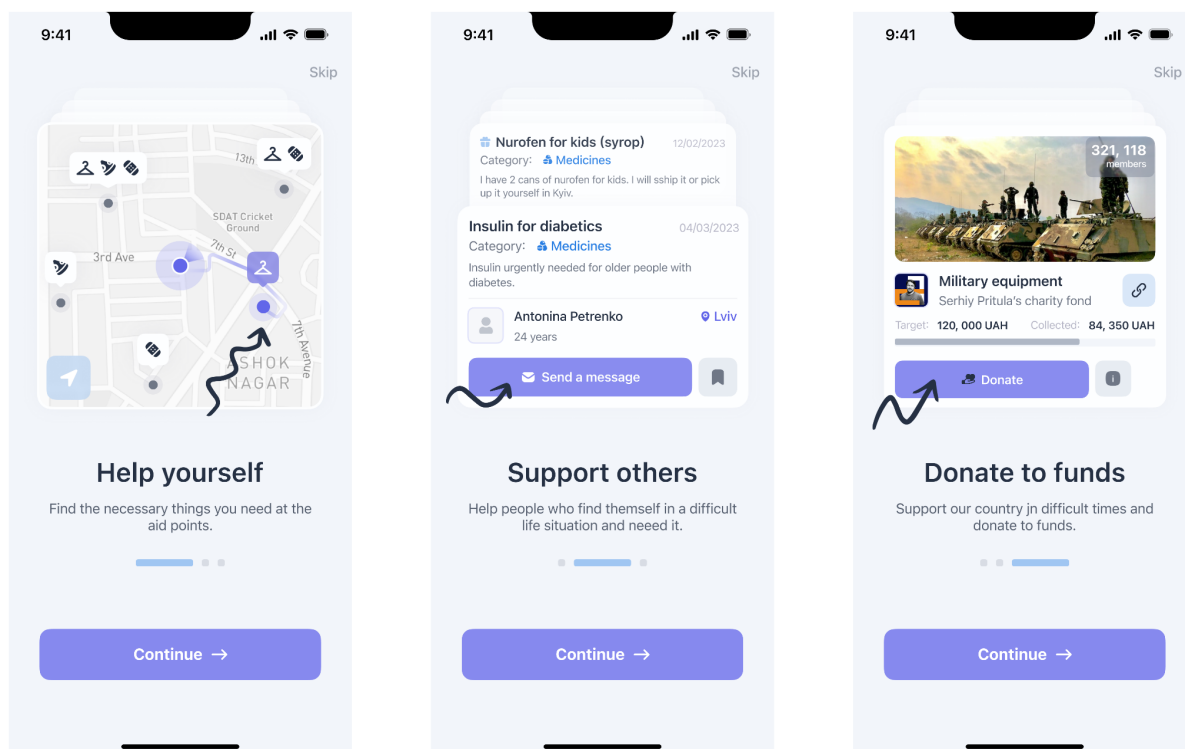


Рисунок 3.5.1 – Сторінки онбордингу додатка “Helper”

2) Сторінка “Welcome” з варіантами авторизації

Дана сторінка зустрічає нового користувача після онбордингу, пропонуючи йому декілька варіантів входу у систему.

А саме таких, як:

- авторизація;
- реєстрація;
- вхід за допомогою соц.мереж (facebook, google тощо).

На рисунку 3.3.2 зображена сторінка “Welcome” додатка “Helper”.

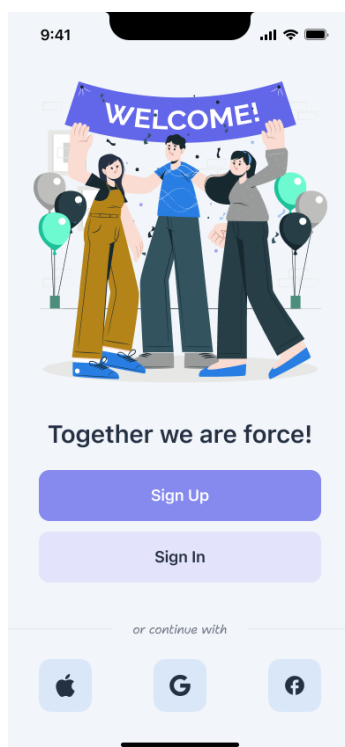


Рисунок 3.5.2 – Сторінка “Welcome” додатка “Helper”

3) Сторінки завантаження додатка та підгрузки контенту.

Дані сторінки були створені для того, щоб користувач завжди розумів, що відбувається на даний момент, що розпочався певний процес і необхідно почекати.

На рисунку 3.5.3 зображені сторінка завантаження додатка “Helper” та сторінки підгрузки контенту.

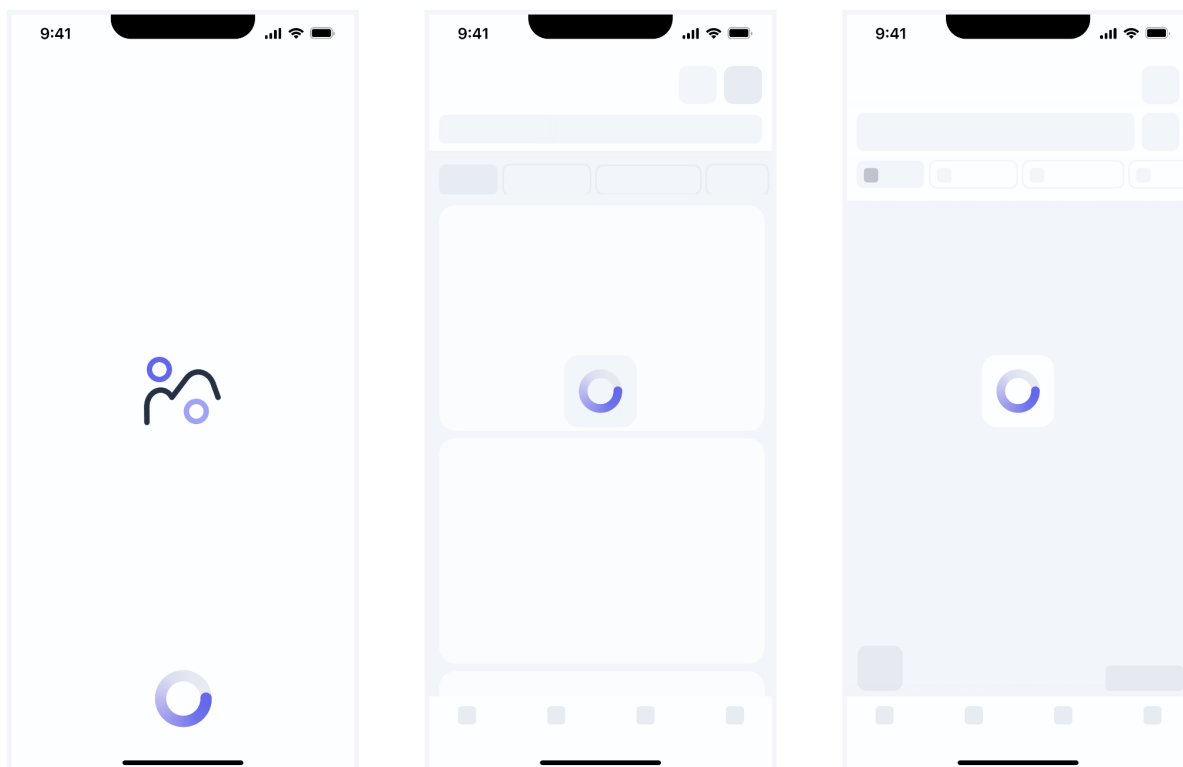


Рисунок 3.5.3 – Сторінка завантаження додатка “Helper” та сторінки підгрузки контенту

4) Авторизація

Вхід у систему відбувається за рахунок вводу електронної пошти/номера телефону користувача та паролю. Також наявна авторизація через соціальні мережі. На сторінці є можливість повернутись на попередню сторінку за допомогою стрілки у верхньому барі або перейти до реєстрації за допомогою кнопки. Наявна можливість відновлення паролю, якщо користувач не може авторизуватись в системі.

Поля вводу мають декілька станів: неактивний, активний та в процесі заповнення. Пароль можна приховати, за бажанням.

На рисунку 3.5.4 зображена сторінка авторизації додатка “Helper” та стани полів вводу.

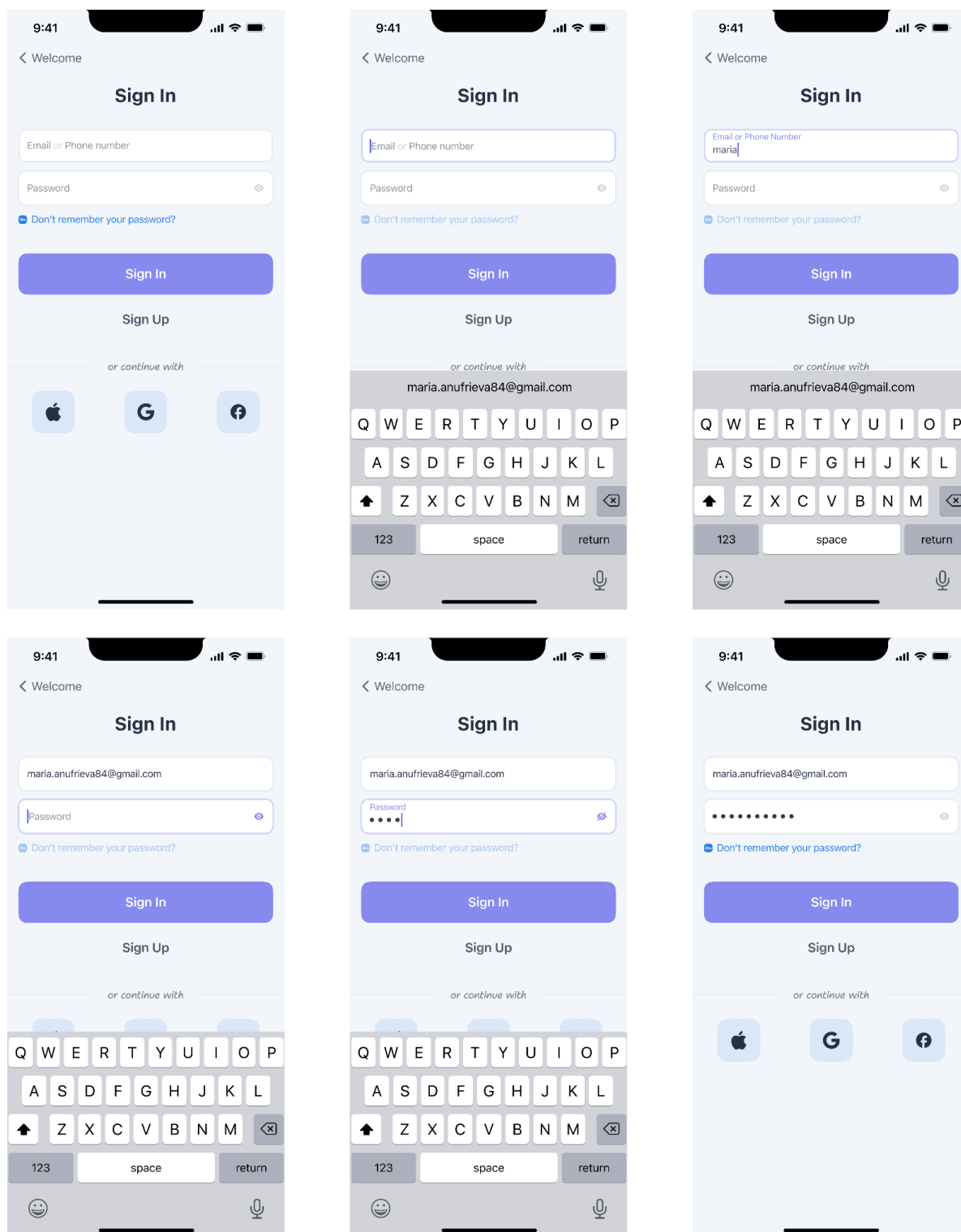


Рисунок 3.5.4 – Сторінка авторизації додатка “Негр” та стани полів вводу
Також наявна валідація полів (рис. 3.5.5)

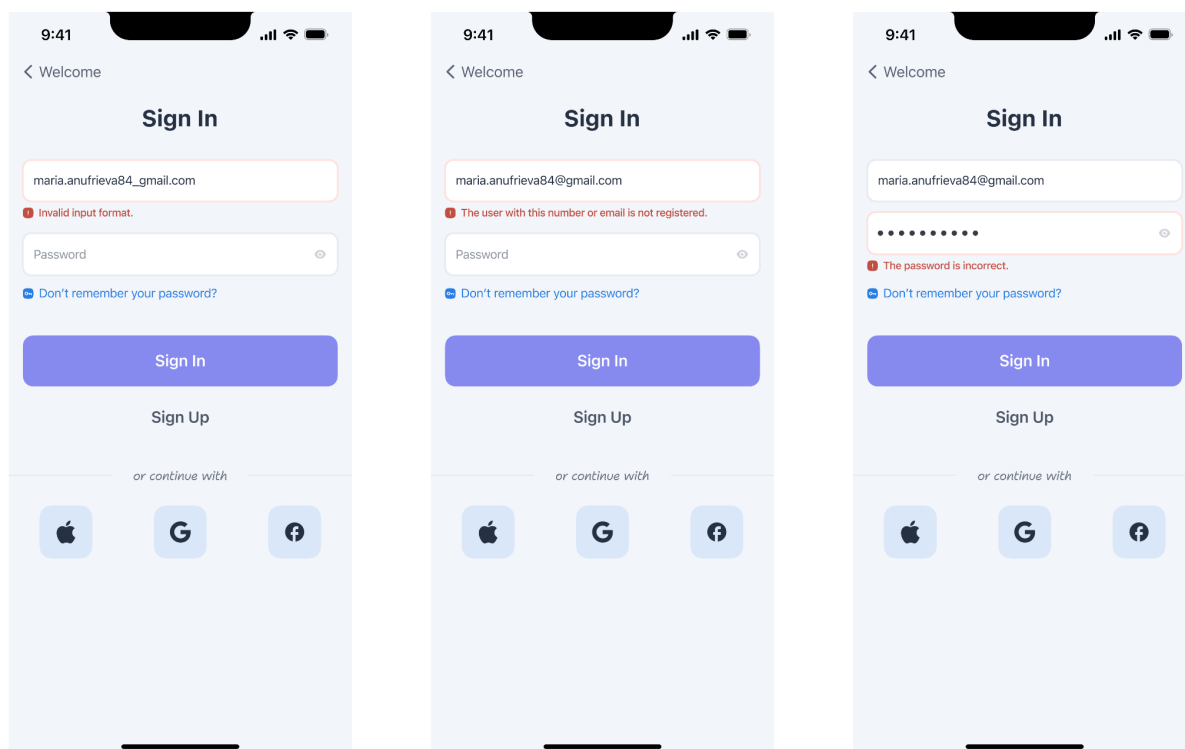


Рисунок 3.5.5 – Валідація полів вводу

5) Сторінка з оголошеннями “Support Requests”

На даній сторінці наявний вибір сегменту серед усіх оголошень, власних та збережених. Також наявна швидка фільтрація по категоріях (chips). На сторінці наявна кнопка створення оголошення та відкриття фільтрів. Власні оголошення можна видаляти та редагувати, оголошення інших користувачів - зберігати та ділитися ними. На рисунку 3.5.6 зображена сторінка з оголошеннями додатка “Helper” відповідно до обраного сегменту.

Також був розроблений дизайн пустих станів даної сторінки, оскільки користувач може не мати збережених або створених оголошень. На рисунку 3.5.7 зображений пустий стан сторінки з оголошеннями у додатку “Helper”.

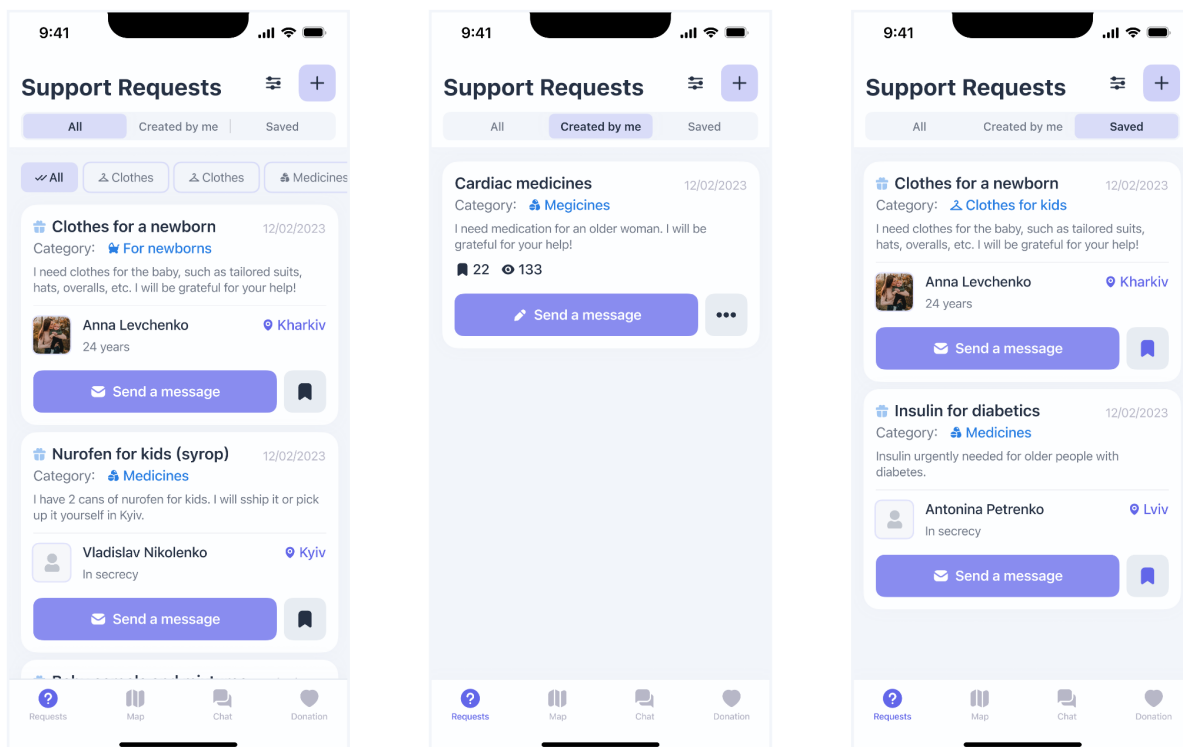


Рисунок 3.5.6 – Сторінка з оголошеннями додатка “Helper” відповідно до обраного сегменту

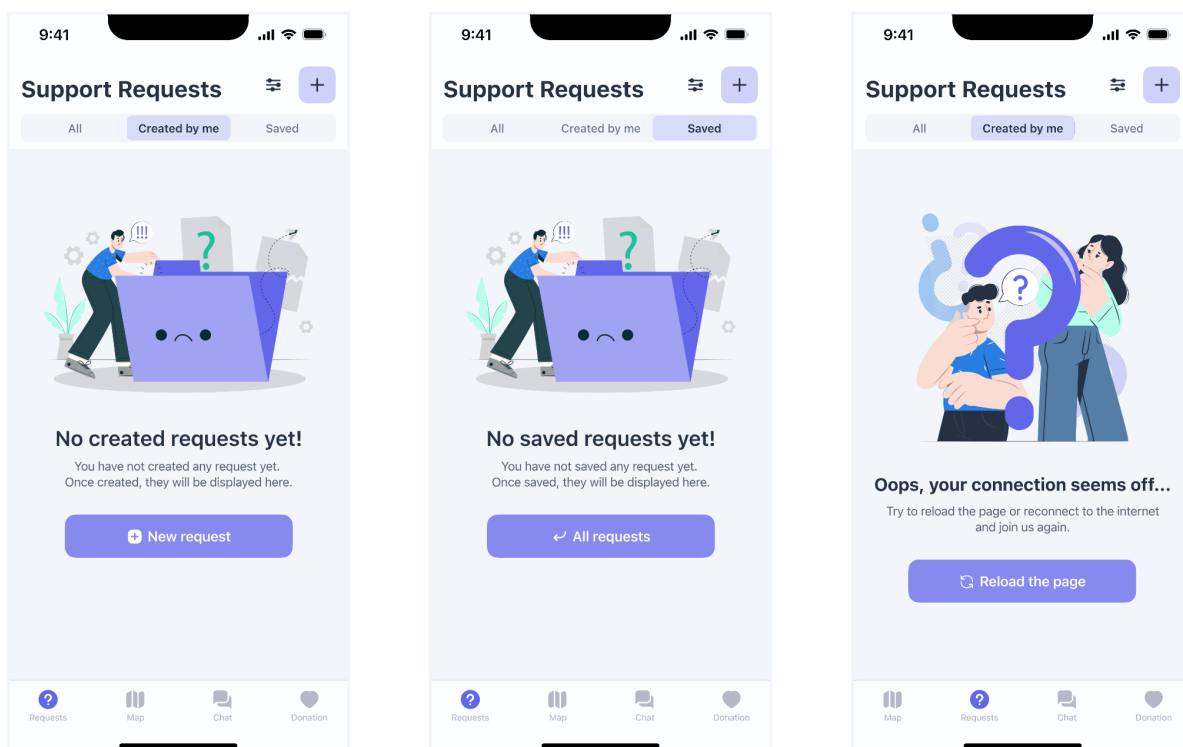


Рисунок 3.5.7 – Дизайн пустих станів та відсутності з’єднання

б) Створення оголошення

Для того, щоб створити нове оголошення, необхідно вказати його назву, обрати тип, категорію та вказати місце розташування даної речі. Також наявні такі опціональні поля, як додавання опису та прикріплення фотокартки. На рисунку 3.5.8 зображена сторінка створення оголошення у додатка “Helper”.

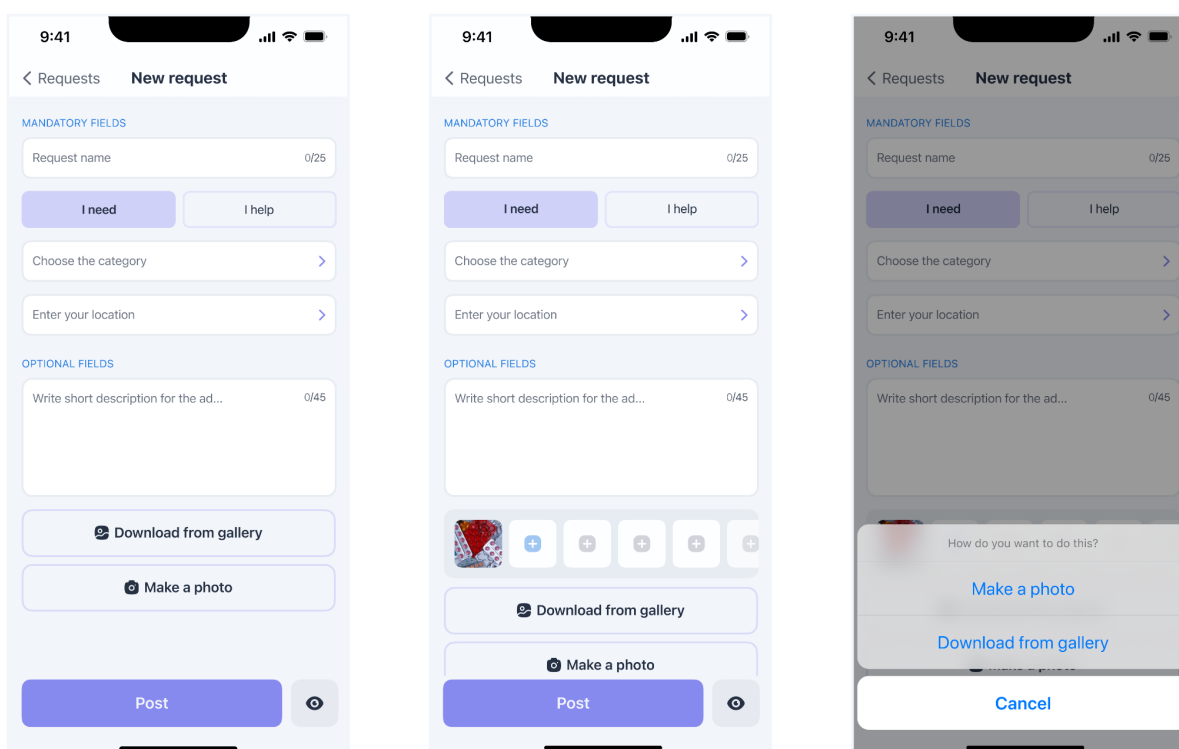


Рисунок 3.5.8 – Сторінка створення оголошення у додатку “Helper”

7) Пошук місця розташування та вибір категорії

Пошук місця розташування реалізований окремою сторінкою. Під час вводу назви міста “розумний” пошук пропонуватиме користувачу варіанти, щоб покращити його користувацький досвід.

Вибір категорії реалізований на основі модального вікна, щоб у користувача була можливість згорнути його, якщо він змінив

рішення. На рисунку 3.5.9 зображені сторінки пошуку та вибору категорії у додатку “Helper”.

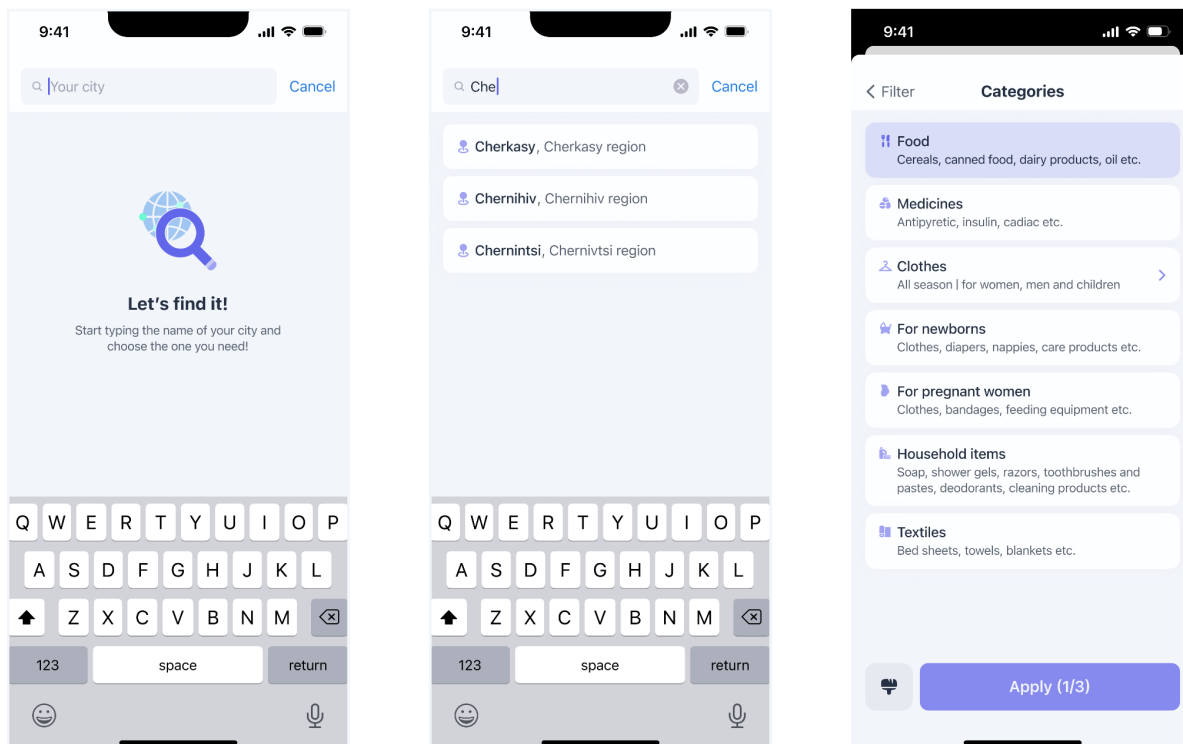


Рисунок 3.5.9 – Сторінки пошуку та вибору категорії у додатку “Helper”

8) Мапа з пунктами помічі

Даний функціонал був реалізований для того, щоб користувачі, які були вимушені покинути свій рідний дім через війну, могли знайти пункти помічі поряд зі своїм місцем розташування та отримати необхідну допомогу.

На даній сторінці наявні два види фільтрації:

- швидка фільтрація (по категоріях);
- розгорнута фільтрація (із заповненням полів).

Перебуваючи у даному розділі, контекстною дією є профіль, де можна змінити персональні дані тощо.

На сторінці наявне поле для введення місця розташування та кнопка автоматичного визначення, що знаходиться у нижньому лівому куті.

На рисунку 3.5.10 зображені варіації сторінки мапи у додатку “Helper”.

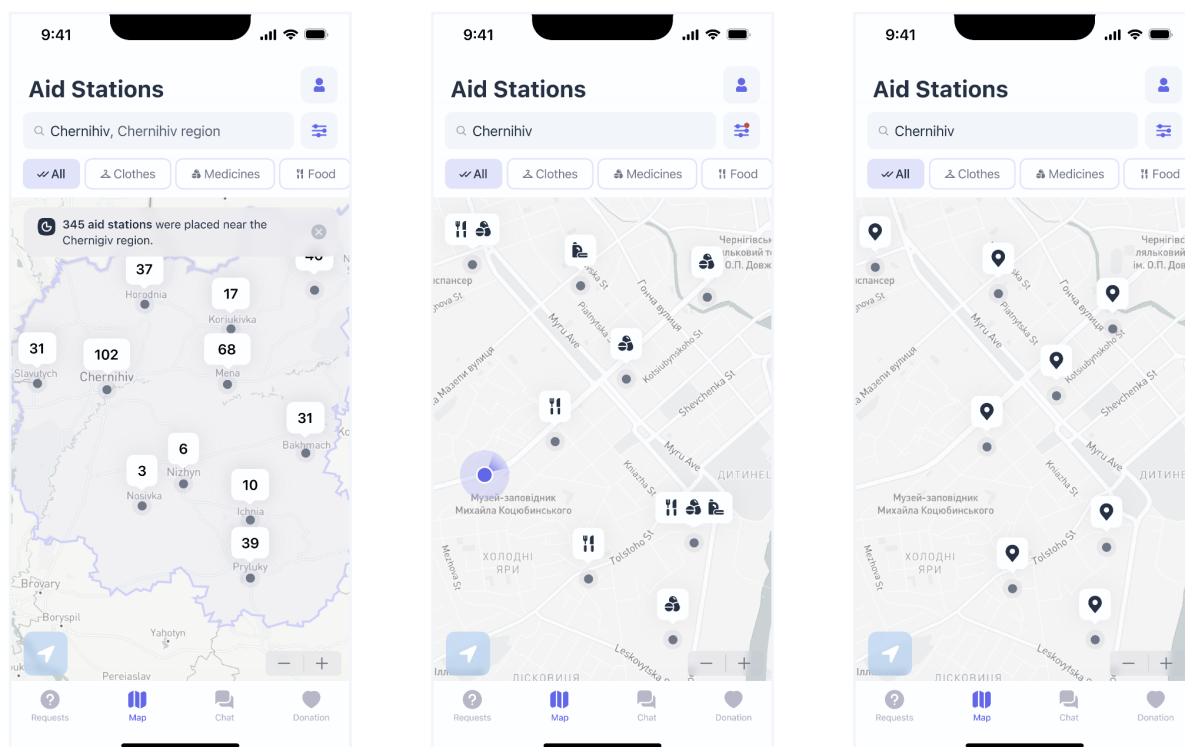


Рисунок 3.5.10 – Варіації сторінки мапи у додатку “Helper”

9) Сторінки доступу

Для того, щоб додаток міг скористатись функціоналом автоматичного визначення місця знаходження користувача, додаванням фотографії, камерою тощо, необхідно надати доступ додатку. На рисунку 3.5.11 зображені сторінки доступу до персональних даних та інформації користувача у додатку “Helper”.

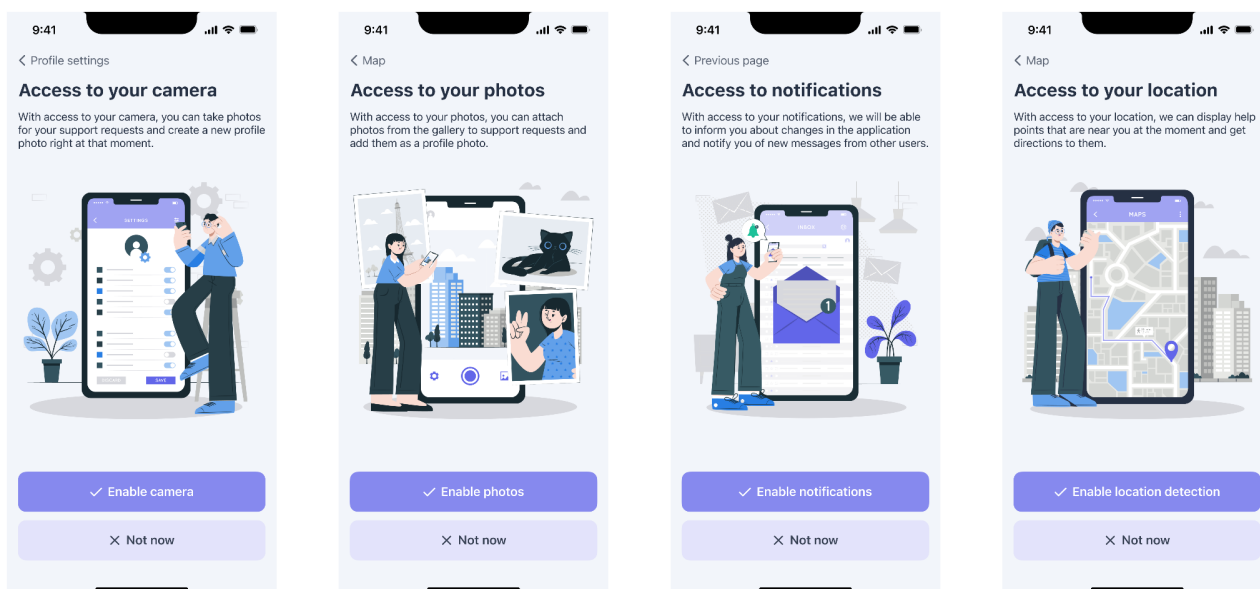


Рисунок 3.5.11 – Сторінки доступу до персональних даних та інформації користувача у додатку “Helper”

10) Фільтрація пунктів допомоги на мапі

Фільтрація пунктів допомоги реалізована у вигляді нативного компонента Anchored Sheet з дизайн-системи HIG від Apple.

Користувачу необхідно обрати категорію (до 3-х) та вказати місце розташування. Також наявна можливість очистити фільтр або ж просто відмінити чи згорнути. На рисунку 3.5.12 зображена сторінка фільтрації пунктів допомоги у додатку “Helper”.

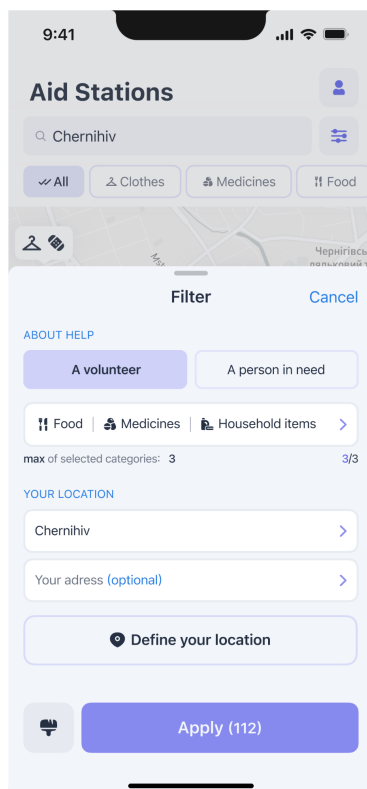


Рисунок 3.5.12 – Сторінка фільтрації пунктів допомоги у додатку “Helper”

11) Пункт помічі та способи прокладання маршруту

Пункт допомоги реалізований також у вигляді нативного компоненту Anchored Sheet з дизайн-системи HIG від Apple, який можна згорнути, на потреби. Також наявна можливість поділитись інформацією про даний пункт допомоги за межами застосунку.

Пункт допомоги містить інформацію про місце розташування, що саме на ньому розміщено, який графік роботи та його заповненість речами.

Натиснувши на кнопку прокладання маршруту, відкриється новий Anchored Sheet із запропонованими варіантами додатків, яке реалізують даний функціонал.

На рисунку 3.5.13 зображена сторінка пункту помічі у додатку “Helper”.

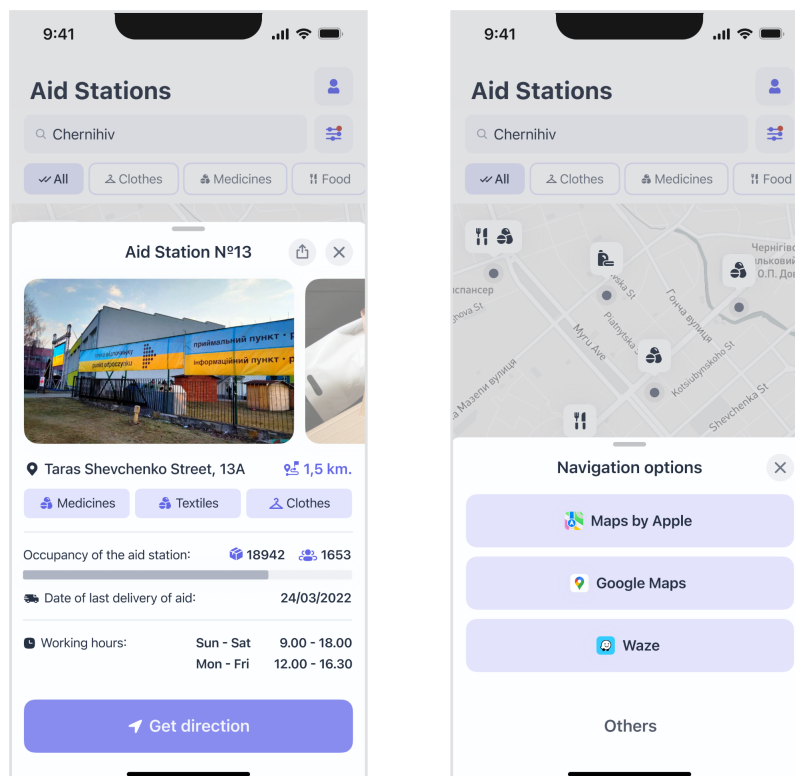


Рисунок 3.5.13 – Anchored Sheets. Пункт помічі та способи прокладання маршруту за межами додатка “Helper”

12) Сторінка зборів

У даному розділі розміщені картки популярних зборів та фондів. У користувача наявна можливість переслати кошти на збір у додатку та дізнатись певну інформацію про нього або про фонд, що його створив.

На рисунку 3.5.14 зображені сторінки фондів та зборів у додатку “Helper”.

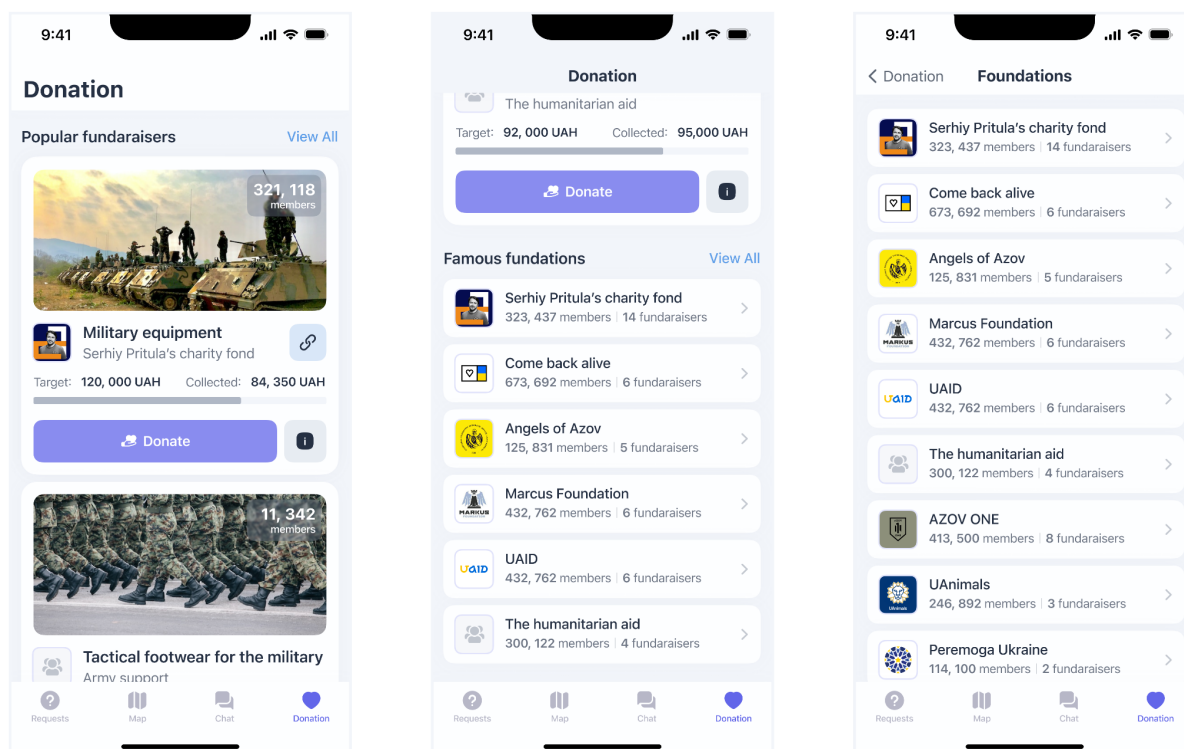


Рисунок 3.5.14 – Сторінки фондів та зборів у додатку “Helper”

Також були розроблені Anchored Sheets для способу оплати та додавання картки (рис. 3.5.15).

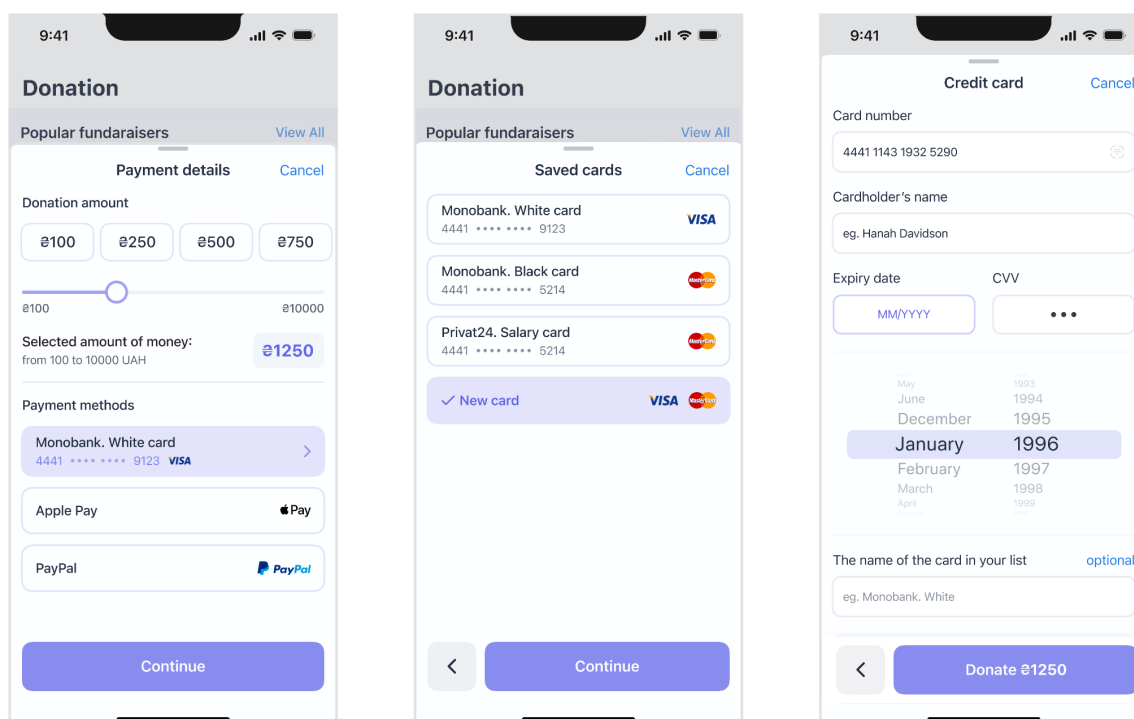


Рисунок 3.5.15 – Anchored Sheets. Способи оплати та додавання картки

3.6 Прототипування

Для проведення аналізу користувацького досвіду було запрототиповано розроблені макети. Приклади прототипування зображені на рисунках 3.6.1 – 3.6.3.

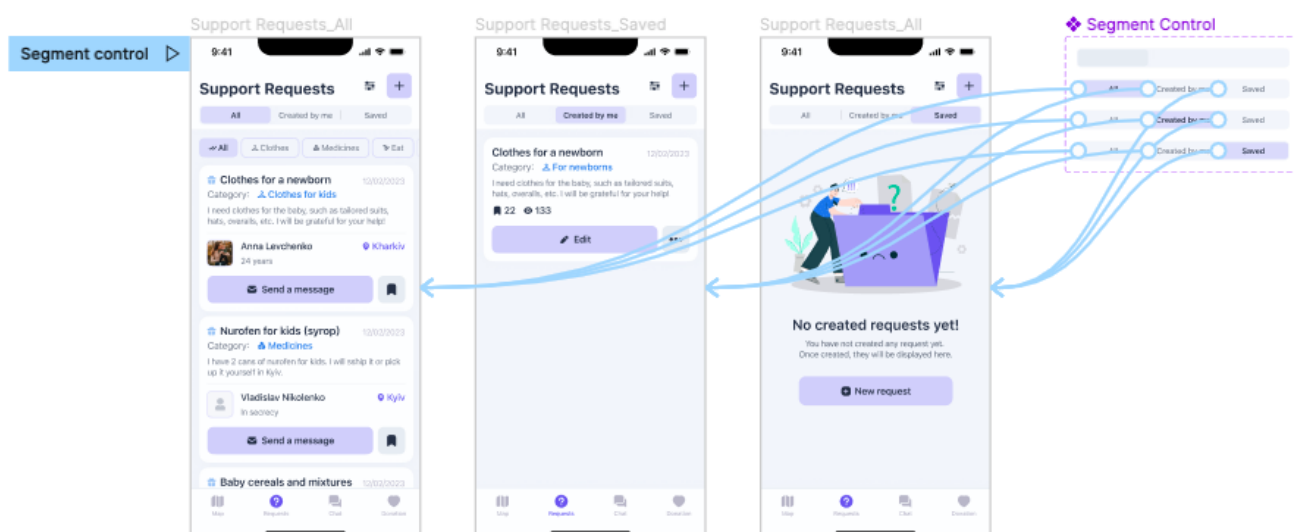


Рисунок 3.6.1 – Приклад прототипування елемента Segment Control

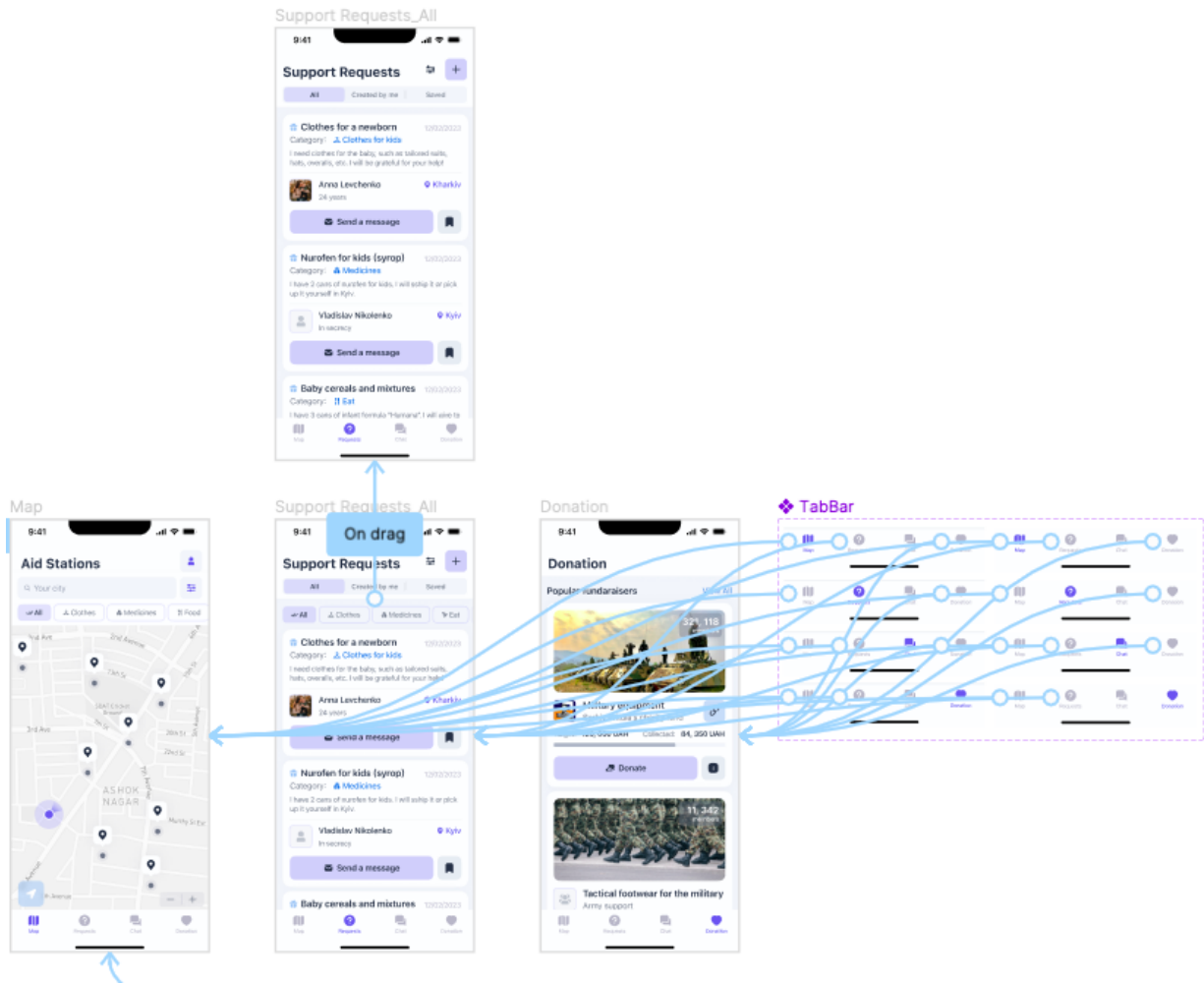


Рисунок 3.6.2 – Приклад прототипування елементів нижнього бару

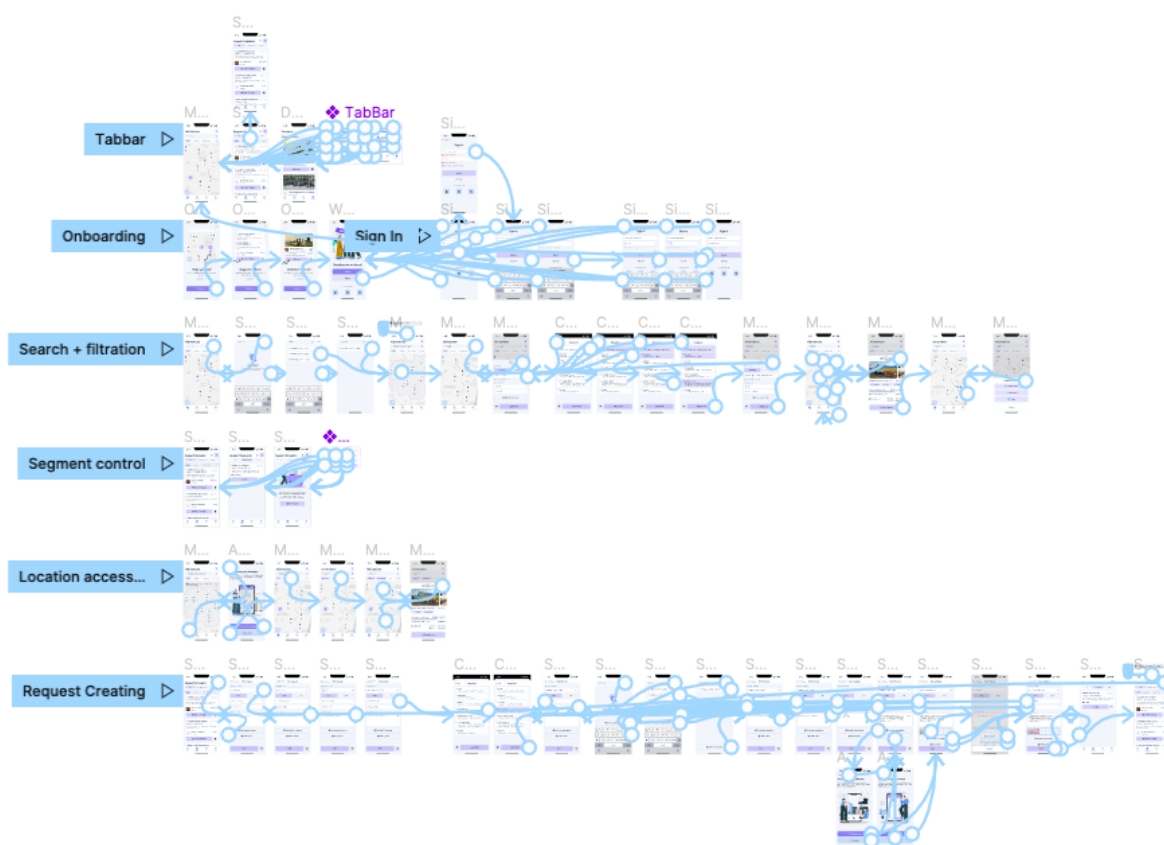


Рисунок 3.6.3 – Запрототиповані сценарії додатка “Helper”

3.7 Дизайн-система додатка “Helper”

Так як більша частина UI компонентів є кастомною, була розроблена дизайн-система для мобільного додатка “Helper” (рис. 3.7.1 – 3.7.4).

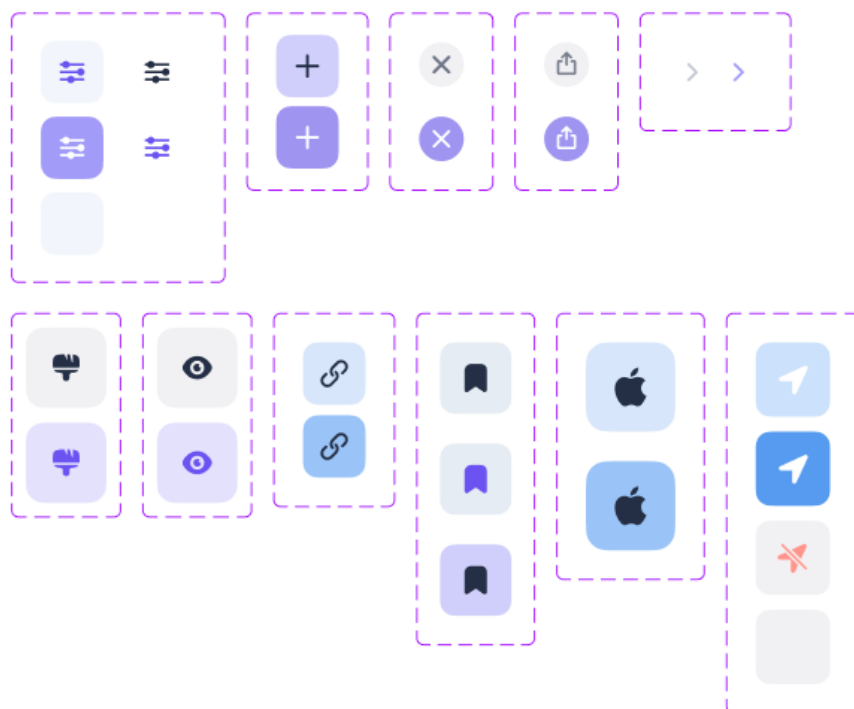


Рисунок 3.7.1 – Компоненты кнопок без текста

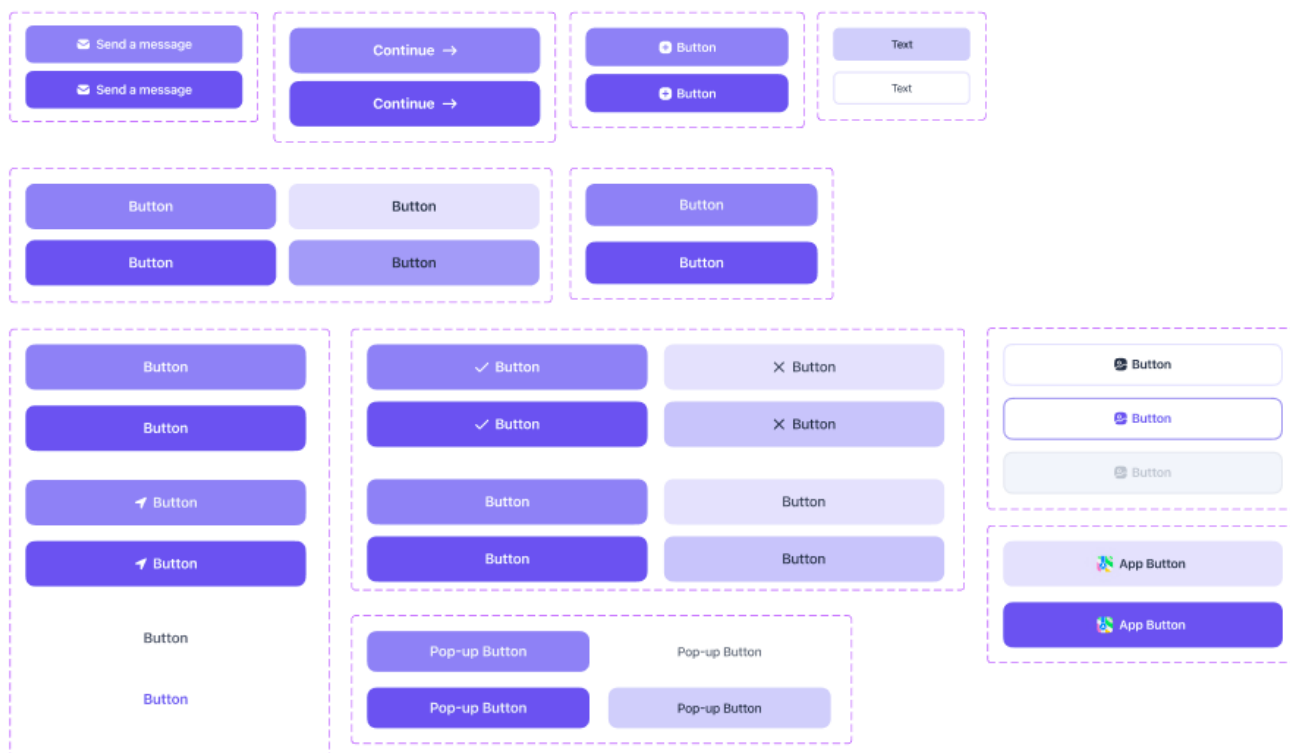


Рисунок 3.7.2 – Компоненти кнопок

Payment Sheet Elements

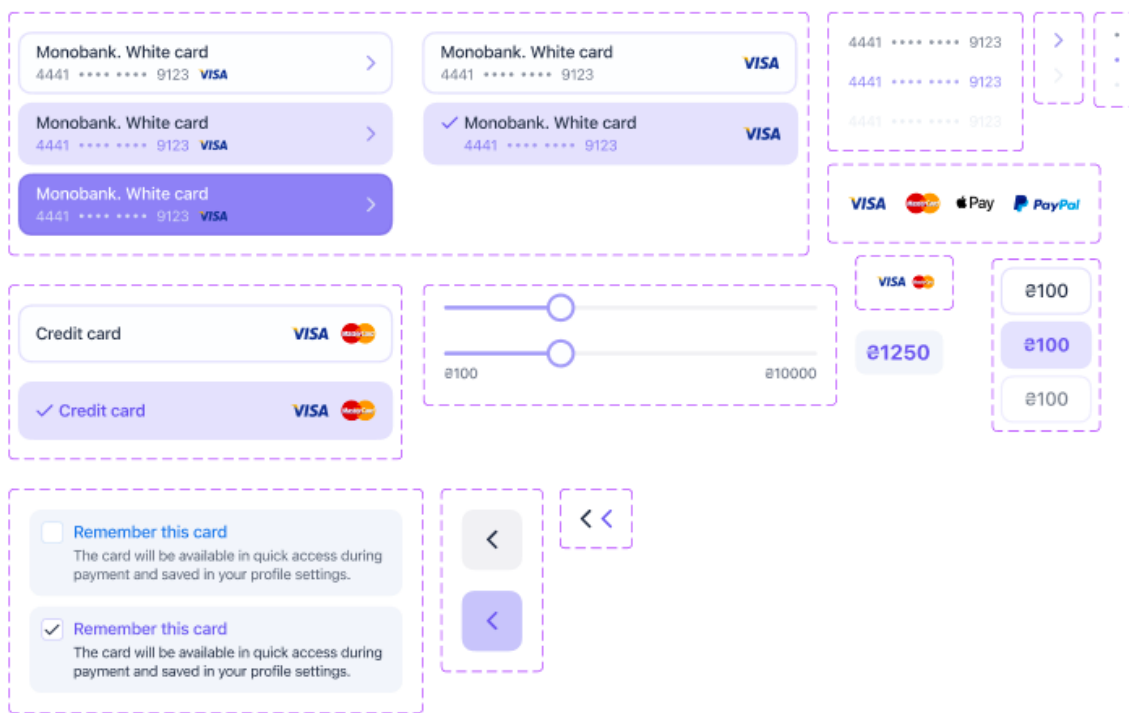


Рисунок 3.7.3 – Компоненти, використані у формі оплати

Inputs, lists and others

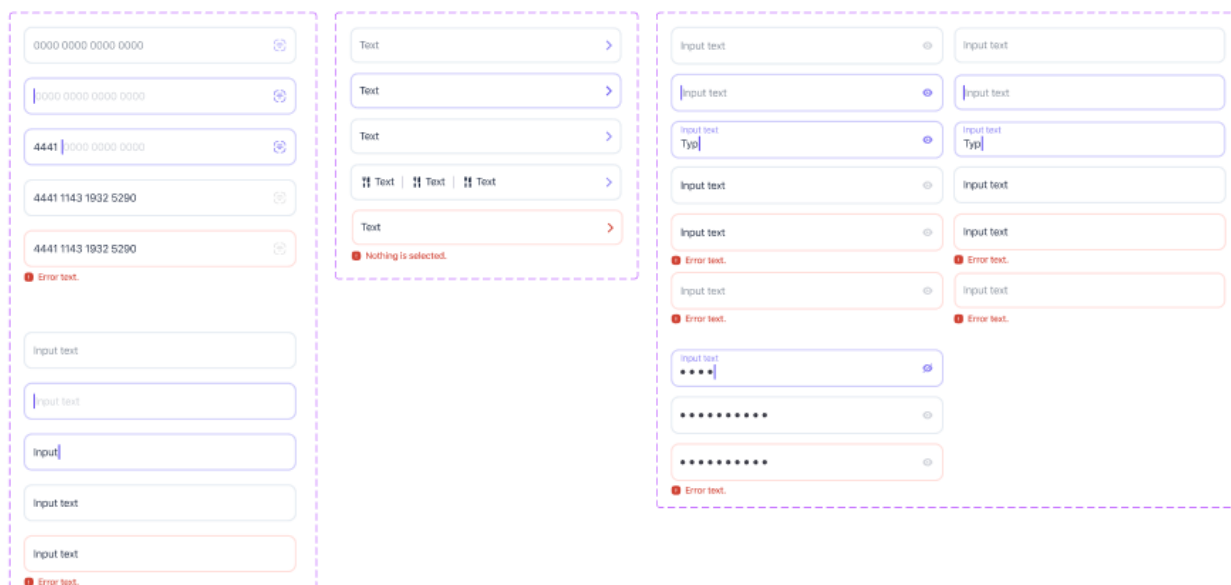


Рисунок 3.7.4 – Компоненти полів вводу

РОЗДІЛ 4. ЧАСТИНА РОЗРОБКИ МОБІЛЬНОГО ДОДАТКА

Для розробки мобільного застосунку було обрано мову програмування C# та фреймворк .NET з Xamarin. Xamarin - інструмент для розробки мобільних додатків, що дозволяє писати кросплатформний код, який працюватиме на iOS [25], адже дизайн був розроблений під дану операційну систему з використанням нативних елементів з дизайн-системи Human Interface Guidelines [22]. Для розробки було використано інтегроване середовище Rider.

4.1 Структура проєкту

Проєкт складається з двох модулів: Core та iOS. На рисунку 4.1.1 зображена структура проєкту.

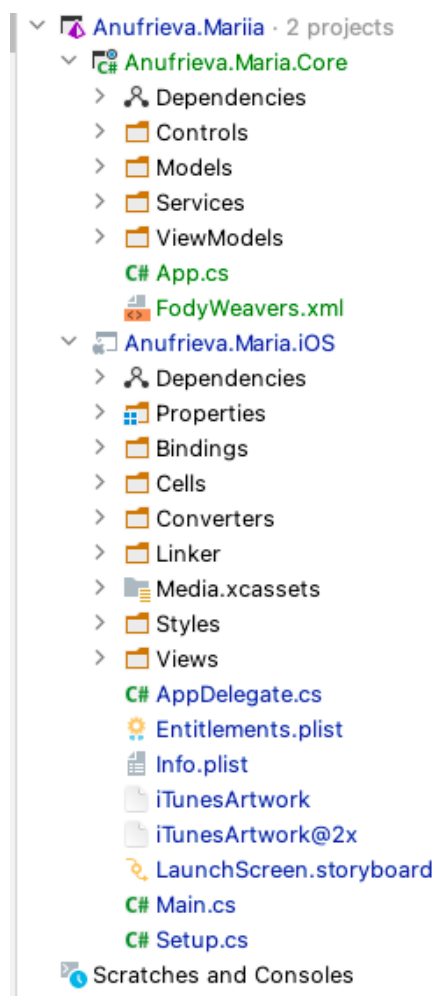


Рисунок 4.1.1 – Структура проєкту

Для реалізації додатка був використаний паттерн проєктування MVVM. Даний паттерн дозволяє розмежовувати бізнес-логіку додатка від його представлення, що робить код більш модульним, тестованим, та зручним в управлінні [26].

Його основними компонентами є:

- Модель (Model). Вона являє собою дані та бізнес-логіку (рис. 4.1.2).
- Представлення (View). Відображає певну інформацію та передає отримане введення від користувача до ViewModel (рис. 4.1.3).
- Модель представлення. Пов'язує дані моделі з представленням (рис. 4.1.4).

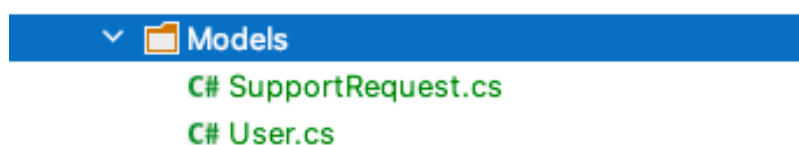


Рисунок 4.1.2 – Моделі

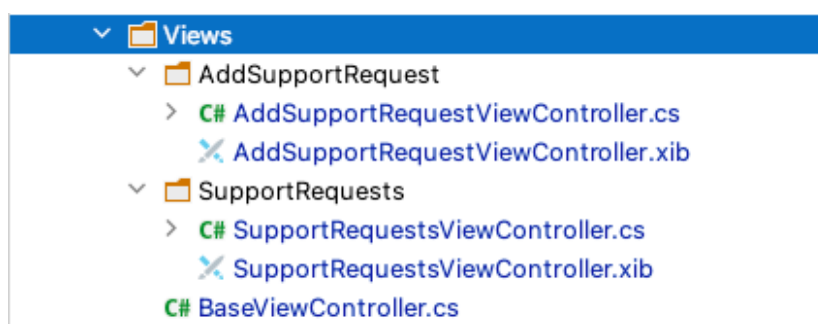


Рисунок 4.1.3 – Представлення

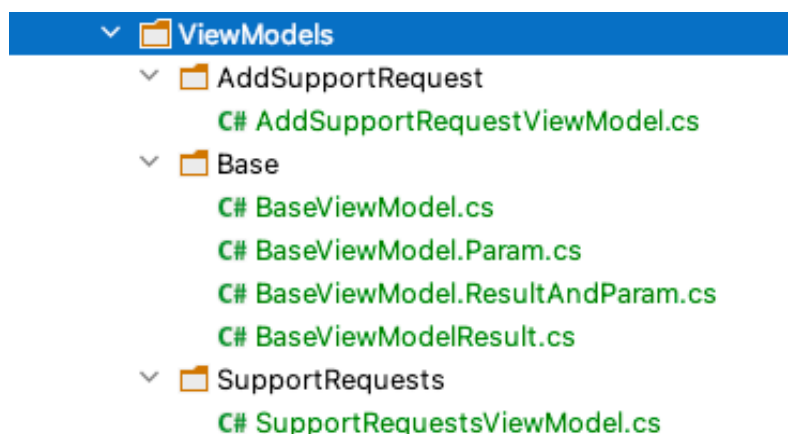


Рисунок 4.1.4 – Моделі представлення

4.2 Реалізація класів проєкту

1) SupportRequestProvider

Даний клас відповідає за надання та управління запитам користувачів.

У конструкторі `SupportRequestProvider` ініціалізуються об'єкти типу `SupportRequest` з різними значеннями полів, такими як `Title` (назва оголошення), `Date` (дата публікації), `Description` (короткий опис до оголошення) та `User` (інформація про власника оголошення). Об'єкт `User` містить інформацію про користувача, включаючи ім'я (`Name`), вік (`Age`) та місто (`City`).

Метод `Get()` повертає перелік (`IEnumerable`) всіх оголошень зі списку `_supportRequests`.

Метод `Add(SupportRequest request)` додає нове оголошення до списку `_supportRequests`.

На рисунку 4.2.1 зображений фрагмент коду з класу `SupportRequestProvider`.

```

1 usage
public sealed class SupportRequestProvider : ISupportRequestProvider
{
    private readonly List<SupportRequest> _supportRequests = new List<SupportRequest>
    {
        new SupportRequest
        {
            Title = "Clothers for a newborn",
            Date = DateTime.Now,
            Description = "I have a new family member, so would be appreciate"
                + " if you could provide me some stuff,"
                + "so I can support this little baby.",
            User = new User
            {
                Name = "Hubert Hend",
                Age = "15 years",
                City = "New Mexico"
            }
        },
        new SupportRequest
        {
            Title = "Nurofen for kids (syrup)",
            Date = DateTime.Now.Subtract(TimeSpan.FromDays(1)),
            Description = "I have 2 cans of nurofen for kids."
                + " I will sship it or pick up it yourself in Kyiv.",
            User = new User
            {
                Name = "Vladislav Nikolenko",
                Age = "In secrecy",
                City = "Kyiv"
            }
        },
    }
}

```

Рисунок 4.2.1 – Фрагмент коду з класу SupportRequestProvider

2) SupportRequest

Даний клас містить такі властивості:

- Title (Назва), що представляє собою назву оголошення.
- Description (Опис), що представляє собою опис до оголошення.
- Date (Дата) - поле типу DateTime, що представляє собою дату створення оголошення.

- Saved (Збережено) - поле, що показує, чи було оголошення збережене у список. Значення по замовчуванню - false.
- User (Користувач) - об'єкт типу User, що містить інформацію про власника оголошення.

Даний клас включає у себе декілька команд та методів:

- SendMessageCommand - команда типу IMvxCommand, яка виконується при натисканні на кнопку. Вона пов'язана з відправленням повідомлення.
- ChangeSavedCommand - команда типу IMvxCommand, яка виконується при натисканні на кнопку. Вона пов'язана зі зміною стану збереження.
- Метод ChangeSavedCommandExecute змінює значення властивості Saved на протилежне значення.

На рисунку 4.2.2 зображений фрагмент коду з класу SupportRequest.

```

17 usages 1 exposing API
public sealed class SupportRequest
{
    6 usages
    public string Title { get; set; }
    6 usages
    public string Description { get; set; } = "Write short description for the ad...";
    5 usages
    public DateTime Date { get; set; } = DateTime.Now;
    4 usages
    public bool Saved { get; set; }
    8 usages
    public User User { get; set; } = new User();

    private IMvxCommand _sendMessageCommand;
    1 usage
    public IMvxCommand SendMessageCommand => _sendMessageCommand
        ??= new MvxAsyncCommand(SendMessageCommandExecute);

    private IMvxCommand _changeSavedCommand;
    1 usage
    public IMvxCommand ChangeSavedCommand => _changeSavedCommand
        ??= new MvxCommand(ChangeSavedCommandExecute);

    1 usage
    private Task SendMessageCommandExecute() =>
        NotImplemented();

    1 usage
    private void ChangeSavedCommandExecute() =>
        Saved = !Saved;

    1 usage
    private Task NotImplemented() =>
        UserDialogs.Instance.AlertAsync(message: "This feature is not implemented :(");
}

```

Рисунок 4.2.2 – Фрагмент коду з класу SupportRequest

3) AddSupportRequestViewModel

Даний клас відповідає за логіку додавання нових оголошень.

SupportRequest NewRequest - властивість, що представляє собою нове оголошення, що має тип SupportRequest. За замовчуванням створюється новий екземпляр SupportRequest.

IMvxCommand PostCommand - команда, яка виконується при натисканні на кнопку. Вона пов'язана з додаванням нового оголошення. Ця команда викликає метод PostCommandExecute.

На рисунку 4.2.3 зображений фрагмент коду з класу AddSupportRequestViewModel.

```

4 usages
public sealed class AddSupportRequestViewModel : BaseViewModel
{
    private readonly ISupportRequestProvider _requestProvider;

    3 usages
    public SupportRequest NewRequest { get; set; } = new SupportRequest();

    private IMvxCommand _postCommand;

    1 usage
    public IMvxCommand PostCommand => _postCommand
        ??= new MvxCommand(PostCommandExecute);

    public AddSupportRequestViewModel(ILoggerFactory logFactory, IMvxNavigationService navigationService,
        ISupportRequestProvider requestProvider)
        : base(logFactory, navigationService)
    {
        _requestProvider = requestProvider;
    }

    1 usage
    private void PostCommandExecute()
    {
        _requestProvider.Add(NewRequest);
        NavigationService.Close( viewModel: this);
    }
}

```

Рисунок 4.2.3 – Фрагмент коду з класу AddSupportRequestViewModel

4) SupportRequestViewModel

Даний клас відповідає за управління списком оголошень та їх фільтрацію за обраним сегментом (segment control).

Конструктор класу SupportRequestsViewModel ініціалізує масив Segments з трьома сегментами: "All" (Всі), "Created by me" (Створені) і "Saved" (Збережені). Кожен сегмент має властивості Title (Назва) і Command (Команда), які виконуються при виборі певного сегменту (tab).

В конструкторі також викликається метод `_requestProvider.Get()`, щоб отримати всі доступні оголошення, і вони зберігаються у приватному полі `_allSupportRequests`.

На рисунку 4.2.4 зображено фрагмент коду з класу `SupportRequestViewModel`.

```
public class SupportRequestsViewModel : BaseViewModel
{
    private readonly ISupportRequestProvider _requestProvider;
    private List<SupportRequest> _allSupportRequests;
    2 usages
    public int SelectedSegmentIndex { get; set; }

    2 usages
    public Segment[] Segments { get; }

    7 usages
    public MvxObservableCollection<SupportRequest> SupportRequests { get; set; }

    private IMvxCommand _addRequestCommand;
    1 usage
    public IMvxCommand AddRequestCommandCommand => _addRequestCommand
        ??= new MvxAsyncCommand(AddRequestCommandCommandExecute);

    public SupportRequestsViewModel(ILoggerFactory logFactory, IMvxNavigationService navigationService,
        ISupportRequestProvider requestProvider)
        : base(logFactory, navigationService)
    {
        _requestProvider = requestProvider;

        Segments = new[]
        {
            new Segment { Title = "All", Command = new MvxCommand(AllCommandExecute) },
            new Segment { Title = "Created by me", Command = new MvxCommand(CreatedByMeCommandExecute) },
            new Segment { Title = "Saved", Command = new MvxCommand(SavedCommandExecute) }
        };

        _allSupportRequests = requestProvider.Get().ToList();
        SupportRequests = new MvxObservableCollection<SupportRequest>(_allSupportRequests);
    }
}
```

Рисунок 4.2.4 – Фрагмент коду з класу `SupportRequestViewModel`

5) AddSupportRequestViewController

Даний клас створений для відображення сторінки додавання нового оголошення.

Цей клас використовує бібліотеку MvvmCross для зв'язування даних між елементами екрану і моделлю даних AddSupportRequestViewModel. За допомогою методу SetupBindings, встановлюються зв'язування для елементів екрану, що дозволяють автоматично оновлювати дані на екрані при зміні моделі даних та виконувати команди при натисканні кнопок.

На рисунку 4.2.5 зображений фрагмент коду з класу AddSupportRequestViewController.

```
public partial class AddSupportRequestViewController : BaseViewController<AddSupportRequestViewModel>
{
    Maxim Chihrov
    public AddSupportRequestViewController() : base(nameof(AddSupportRequestViewController))
    { }

    0+2 usages Maxim Chihrov
    public override void ViewDidLoad()
    {
        base.ViewDidLoad();

        Title = "New request";

        SetupBindings(CreateBindingSet());
    }

    1+2 usages Maxim Chihrov
    protected override void SetupBindings(MvxFluentBindingDescriptionSet<IMvxIOSView<AddSupportRequestViewModel>,
        AddSupportRequestViewModel> set)
    {
        set.Bind(TitleLabel)
            .To(vm : AddSupportRequestViewModel => vm.NewRequest.Title);

        set.Bind(DescriptionLabel)
            .For(v : UITextView => v.BindText())
            .To(vm : AddSupportRequestViewModel => vm.NewRequest.Description);

        set.Bind(PostButton)
            .To(vm : AddSupportRequestViewModel => vm.PostCommand);

        base.SetupBindings(set);
    }
}
```

Рисунок 4.2.5 – Фрагмент коду з класу
AddSupportRequestViewController

6) SupportRequestsViewController

Даний клас був створений для відображення списку оголошень.

Даний клас відповідає за відображення списку оголошень, управління обраним сегментом (tab в segment control), додавання нового оголошення і взаємодію з елементами користувацького інтерфейсу для виконання дій, таких як зміна статусу збереження оголошення.

На рисунку 4.2.6 зображений фрагмент коду з класу SupportRequestsViewController.

```

public partial class SupportRequestsViewController : BaseViewController<SupportRequestsViewModel>
{
    private UISegmentedControl _segmentedControl;
    private MvxSimpleTableViewSource _tableSource;
    private UIButton _addButton;

    Maxim Chihirov
    public SupportRequestsViewController() : base(nameof(SupportRequestsViewController))
    { }

    0+2 usages Maxim Chihirov
    public override void ViewDidLoad()
    {
        base.ViewDidLoad();

        InitializeTable();

        SetupBindings(CreateBindingSet());

        void InitializeTable()
        {
            _tableSource = new MvxSimpleTableViewSource(Table, nibName: SupportRequestCell.Key, SupportRequestCell.Key);
            Table.RowHeight = 254;
            Table.Source = _tableSource;
            Table.ReloadData();
        }
    }

    1+2 usages Maxim Chihirov
    protected override void SetupBindings(MvxFluentBindingDescriptionSet<IMvxIosView<SupportRequestsViewModel>,
        SupportRequestsViewModel> set)
    {
    }
}

```

Рисунок 4.2.6 – Фрагмент коду з класу SupportRequestsViewController

4.3 Реалізація дизайну інтерфейсу

1) AddSupportRequestViewController.xib

Даний файл містить інформацію про розміри, властивості та зв'язки елементів інтерфейсу. У ньому описано вигляд користувацького

інтерфейсу, а саме сторінки додавання оголошення, він також включає таблицю для відображення даних.

На рисунку 4.3.1 зображений фрагмент коду з файлу `AddSupportRequestViewController.xib`.

```
<?xml version="1.0" encoding="UTF-8"?>
<document type="com.apple.InterfaceBuilder3.CocoaTouch.XIB" version="3.0" toolsVersion="21507"
    targetRuntime="iOS.CocoaTouch" propertyAccessControl="none" useAutolayout="YES" useTraitCollections="YES"
    useSafeAreas="YES" colorMatched="YES">
    <device id="retina6_12" orientation="portrait" appearance="light"/>
    <dependencies>
        <deployment identifier="iOS"/>
        <plugIn identifier="com.apple.InterfaceBuilder.IBCocoaTouchPlugin" version="21505"/>
        <capability name="Named colors" minToolsVersion="9.0"/>
        <capability name="Safe area layout guides" minToolsVersion="9.0"/>
        <capability name="System colors in document resources" minToolsVersion="11.0"/>
        <capability name="documents saved in the Xcode 8 format" minToolsVersion="8.0"/>
    </dependencies>
    <objects>
        <placeholder placeholderIdentifier="IBFilesOwner" id="-1" userLabel="File's Owner"
            customClass="AddSupportRequestViewController">
            <connections>
                <outlet property="DescriptionLabel" destination="q2G-Eu-01c" id="Mvb-t7-qQj"/>
                <outlet property="PostButton" destination="VID-ff-VhH" id="YvB-1y-exm"/>
                <outlet property="titleLabel" destination="LCy-VL-uWL" id="IH2-yx-Jcg"/>
                <outlet property="view" destination="vZ4-a4-McR" id="4sU-du-kwn"/>
            </connections>
        </placeholder>
        <placeholder placeholderIdentifier="IBFirstResponder" id="-2" customClass="UIResponder"/>
        <view contentMode="scaleToFill" id="vZ4-a4-McR">
            <rect key="frame" x="0.0" y="0.0" width="393" height="852"/>
        </view>
    </objects>
</document>
```

Рисунок 4.3.1 – Фрагмент коду з файлу
`AddSupportRequestViewController.xib`

2) `SupportRequestsViewController.xib`

У даному файлі описано вигляд користувацького інтерфейсу, а саме сторінки з картками оголошень, він також включає таблицю для відображення даних.

На рисунку 4.3.2 зображений фрагмент коду з файлу `SupportRequestsViewController.xib`.

```

<?xml version="1.0" encoding="UTF-8"?>
<document type="com.apple.InterfaceBuilder3.CocoaTouch.XIB" version="3.0" toolsVersion="21507"
    targetRuntime="iOS.CocoaTouch" propertyAccessControl="none" useAutolayout="YES"
    useTraitCollections="YES" useSafeAreas="YES" colorMatched="YES">
  <device id="retina6_12" orientation="portrait" appearance="light"/>
  <dependencies>
    <deployment identifier="iOS"/>
    <plugIn identifier="com.apple.InterfaceBuilder.IBCocoaTouchPlugin" version="21505"/>
    <capability name="Named colors" minToolsVersion="9.0"/>
    <capability name="Safe area layout guides" minToolsVersion="9.0"/>
    <capability name="System colors in document resources" minToolsVersion="11.0"/>
    <capability name="documents saved in the Xcode 8 format" minToolsVersion="8.0"/>
  </dependencies>
  <objects>
    <placeholder placeholderIdentifier="IBFilesOwner" id="-1" userLabel="File's Owner"
      customClass="SupportRequestsViewController">
      <connections>
        <outlet property="Table" destination="tyR-a9-jPL" id="4em-zQ-oyu"/>
        <outlet property="view" destination="3jl-Vh-3rQ" id="b3a-uL-0yD"/>
      </connections>
    </placeholder>
    <placeholder placeholderIdentifier="IBFirstResponder" id="-2" customClass="UIResponder"/>
    <view contentMode="scaleToFill" id="3jl-Vh-3rQ">
      <rect key="frame" x="0.0" y="0.0" width="393" height="852"/>
      <autoresizingMask key="autoresizingMask" widthSizable="YES" heightSizable="YES"/>
    </view>
  </objects>
</document>

```

Рисунок 4.3.2 – Фрагмент коду з файлу SupportRequestsViewController.xib

3) Клас ColorPalette

Даний клас надає доступ до різних кольорових палітр, які використовуються при розробці мобільних додатків під iOS.

Кожна властивість даного класу представляє собою кольоровий об'єкт UIColor, який може бути використаний для встановлення кольору фону, тексту тощо.

На рисунку 4.3.3 зображений фрагмент коду з класу ColorPalette.

```
public static class ColorPalette
{
    private const string NamespaceSeparator = "/";

    2 usages Maxim Chihirov
    public static class Action
    {
        private const string ActionNamespaceTag = nameof(Action) + NamespaceSeparator;

        2 usages Maxim Chihirov
        public static class Purple
        {
            private const string PurpleNamespaceTag = ActionNamespaceTag + nameof(Purple) + NamespaceSeparator;

            2 usages Maxim Chihirov
            public static UIColor Dark =>
                UIColor.FromName(PurpleNamespaceTag + nameof(Dark));
        }
    }

    4 usages Maxim Chihirov
    public static class Background
    {
        private const string BackgroundNamespaceTag = nameof(Background) + NamespaceSeparator;

        3 usages Maxim Chihirov
        public static UIColor Light =>
            UIColor.FromName(BackgroundNamespaceTag + nameof(Light));

        2 usages Maxim Chihirov
        public static UIColor LightBlue =>
            UIColor.FromName(BackgroundNamespaceTag + nameof(LightBlue));

        Maxim Chihirov
        public static UIColor SheetsAndPopUps =>
            UIColor.FromName(BackgroundNamespaceTag + "Sheets & pop-ups");
    }
}
```

Рисунок 4.3.3 – Фрагмент коду з класу ColorPalette

ВИСНОВКИ

Досліджено предметну область, визначено та проаналізовано цільову аудиторію, потреби та проблеми користувачів, проаналізовано існуючі мобільні додатки на ринку, призначені для допомоги постраждалим людям через війну, проаналізовано типи користувачів, їх користувацькі болі та досвід, побудовано архітектуру мобільного додатку, спроектовано рішення та сценарії виконання певних дій користувачем на їх основі, створено прототипи, виконано прототипування, реалізовано сценарій додавання оголошення, розроблено логотип додатка та зібрано дизайн-систему в тому числі.

Обраними технологіями для розробки стали Xamarin, .NET. Для розробки програмного забезпечення було використано інтегроване середовище Rider.

Перспективи подальшої розробки: відображення на мапі не тільки пунктів допомоги, а й бомбосховищ, які знаходяться поряд. Також додатково можна оновити фільтрацію у розділі з мапою, щоб користувач міг обрати, що хоче побачити на мапі, бомбосховища або пункти бомбосховища. Якщо користувач не фільтруватиме таким чином, усі точки будуть відображені на карті, але матимуть різні кольори та іконки під час відображення, щоб користувач мав можливість інтуїтивно зрозуміти, де на мапі бомбосховище, а де пункт допомоги постраждалим.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Основні принципи розробки дизайну мобільних додатків [Електронний ресурс] - Режим доступу до ресурсу:
<https://www.management.com.ua/partners/2022/04/26/osnovni-printsipi-rozrobki-dizajnu-mobilnih-dodatkiv/>
2. Нижня панель навігації краща альтернатива для мобільних сайтів? [Електронний ресурс] - Режим доступу до ресурсу:
<https://prodesign.in.ua/2019/10/bottom-menu-best-navigation-pattern-for-mobile-web-sites/>
3. Jetpack Compose Navigation components [Електронний ресурс] - Режим доступу до ресурсу:
<https://www.composables.com/components/navigation>
4. The Thumb Zone: Designing For Mobile Users [Електронний ресурс] - Режим доступу до ресурсу:
<https://www.smashingmagazine.com/2016/09/the-thumb-zone-designing-for-mobile-users/>
5. Різниця в проектуванні нативних додатків iOS та Android [Електронний ресурс] - Режим доступу до ресурсу:
<https://ux.pub/editorial/razlichie-v-proektirovanii-nativnykh-prilozhenii-ios-i-android-9mn>
6. 8 етапів розробки дизайну мобільного додатка [Електронний ресурс] - Режим доступу до ресурсу:
<https://blog.ithillel.ua/ru/articles/8-etapov-razrabotky-ui-ux-dyzaina-mobylnoho-prylozheniya>
7. Проектування мобільного додатка [Електронний ресурс] - Режим доступу до ресурсу:
<https://wezom.com.ua/ua/blog/proektirovanie-mobilnogo-prilozheniya>

8. Розроблення UX/UI дизайну мобільних додатків [Електронний ресурс] - Режим доступу до ресурсу:
https://learn.ztu.edu.ua/pluginfile.php/242117/mod_resource/content/0/%D0%9B%D0%B5%D0%BA%D1%86%D1%96%D1%8F%205_6.pdf
9. Top Things to Know About UX Competitive Analysis [Електронний ресурс] - Режим доступу до ресурсу:
<https://uxplanet.org/top-things-to-know-about-ux-competitive-analysis-d91689fd8b36>
10. What are UX personas and what are they used for? [Електронний ресурс] - Режим доступу до ресурсу:
<https://www.uxdesigninstitute.com/blog/what-are-ux-personas/>
11. Інформаційна архітектура [Електронний ресурс] - Режим доступу до ресурсу: <https://ux.pub/zhmikhov/informatsiina-arkhitektura-4e6m>
12. Information Architecture for charity app [Електронний ресурс] - Режим доступу до ресурсу:
<https://dribbble.com/shots/3874744-Information-Architecture-for-charity-app>
13. НЕ СОРОМНО ЗАПИТАТИ: ЩО ТАКЕ ВАЙРФРЕЙМ, МОКАП І ПРОТОТИП [Електронний ресурс] - Режим доступу до ресурсу:
https://skvot.io/uk/blog/ne-soromno-zapitati-shcho-take-vayrfreyim-mokap-i-prototip?utm_term=&utm_campaign=blog&utm_source=google&utm_medium=cpc&utm_content=search&gclid=CjwKCAjw36GjBhAkEiwAKwIWYs1k-ytw-l_IM1CUFZF--GNUuio2dkpGHoW9syQfUVhjHpfBxX_yqBoCY6cQAvD_BwE
14. Never include visuals in wireframes; here's why [Електронний ресурс] - Режим доступу до ресурсу:
<https://nulab.com/learn/design-and-ux/never-include-visuals-wireframes-heres/>

15. What Are User Flows in UX Design? +Benefits, Types (2022)
[Електронний ресурс] - Режим доступу до ресурсу:
<https://whatfix.com/blog/user-flows/>
16. Wireframes for mobile UI design [Електронний ресурс] - Режим доступу до ресурсу:
<https://www.figma.com/community/file/848318135747364351/%F0%9F%93%B2Wireframes-for-mobile-UI-design>
17. Turn Design Concepts Into Prototypes With Wondershare Mockitt
[Електронний ресурс] - Режим доступу до ресурсу:
<https://www.theme-junkie.com/wondershare-mockitt-review/>
18. Кількість користувачів смартфонів в Україні збільшилася до 85% — дослідження [Електронний ресурс] - Режим доступу до ресурсу:
<https://ms.detector.media/mediadoslidzhennya/post/21573/2018-08-03-kilkist-korystuvachiv-smartfoniv-v-ukraini-zbilshylasya-do-85-doslidzhennya/>
19. IT-компанія Geniusee запустила мобільний додаток для взаємодопомоги «Бандеролька» [Електронний ресурс] - Режим доступу до ресурсу:
<https://novy.tv/ua/news/2022/05/30/it-kompaniya-geniusee-zapustyla-mobilnyj-dodatok-dlya-vzayemodopomogy-banderolka/>
20. Geniusee [Електронний ресурс] - Режим доступу до ресурсу:
<https://geniusee.com/>
21. ЕРАУНД [Електронний ресурс] - Режим доступу до ресурсу:
<https://twitter.com/eroundapp>
22. Human Interface Guidelines [Електронний ресурс] - Режим доступу до ресурсу:
<https://developer.apple.com/design/human-interface-guidelines>

23. Material Design [Електронний ресурс] - Режим доступу до ресурсу:

<https://m3.material.io/>

24. Інформаційна архітектура мобільного додатка “Helper”
<https://www.figma.com/file/dEoixg216sJfQ7bbCd8RnY/Information-Structure?type=whiteboard&node-id=0%3A1&t=TZa2slIW9VNW6OaQ-1>

25. Xamarin [Електронний ресурс] - Режим доступу до ресурсу:

<https://uk.wikipedia.org/wiki/Xamarin>

26. Model-View-ViewModel [Електронний ресурс] - Режим доступу до ресурсу: <https://uk.wikipedia.org/wiki/Model-View-ViewModel>

27. Посилання на проект [Електронний ресурс] - Режим доступу до ресурсу: <https://github.com/Mariaan84/dyploma>

28. User Personas. “Helper” [Електронний ресурс] - Режим доступу до ресурсу:

<https://www.figma.com/file/YVT3fJgS1JsMyP7RfrVeRh/User-persona?type=whiteboard&t=yVlaHTA2gIpBPs3p-0>

29. User Flow. “Helper” [Електронний ресурс] - Режим доступу до ресурсу:

<https://www.figma.com/file/paf5KAjBlrXi17QJxSLYlu/User-flow?type=design&t=yVlaHTA2gIpBPs3p-1>

30. Prototyping. “Helper” [Електронний ресурс] - Режим доступу до ресурсу:

<https://www.figma.com/file/ypzC56I723EAtZ27PZU1SX/Prototyping?type=design&t=yVlaHTA2gIpBPs3p-1>

31. Visual Design. “Helper” [Електронний ресурс] - Режим доступу до ресурсу:

<https://www.figma.com/file/51N2vKzSisWTREGLhHHhJi/Visual-design?type=design&t=yVlaHTA2gIpBPs3p-1>

32. Jesse J. G. The Elements of User Experience: User-Centered Design for the Web / James Garrett Jesse, 2002.

33. UX UI Design Wireframer. User Experience Mobile Design Notebook: Low-Fidelity Mobile Wireframe Screen Templates : For UI/UX Designers, Interaction Designers, Web/App Developers / UX UI Design Wireframer., 2023.

34. Fenniwald J. Mobile And Web Design: Basics Of Modern Web And Mobile Design Processes / Jay Fenniwald., 2023.

ДОДАТОК А

User Personas



Valeria Ivanenko

30 years · Housewife and mom

Polite and careful

Hard working

Productive

Technical skills

Internet: ★ ★ ★ ★ ★

Socials: ★ ★ ★ ★ ☆

Messengers: ★ ★ ★ ★ ★

Online payment: ★ ★ ★ ★ ☆

About

She left her hometown with her child because of the war. At the moment, she is unemployed, so she needs help for herself and her child. Lives in an unfamiliar city.

Goals

List of user goals

- Find aid stations with the necessary things for her and the child without wandering around the city. Doing this with a child is extremely uncomfortable.
- Help with things the same demanding people as her.
- Be able to get to the found aid station without problems along the route.
- Connect with people who help her or whom she herself helps.

Pain points & objections

List of points of frustration that the user has encountered

- Lack of marked points on the map with aid points. She has to look for information about the location in various social networks and Google.
- She has to search for ads with the help of applications not intended for this, groups in social networks and various forums.
- She has the desire and opportunity to help people in whatever way she can, but there is no application where she could post her ads.
- It is not always convenient to call the person whose ad was found, but in one case it is the only option for communication. prefers to chat more.

ДОДАТОК Б

Приклад сценарію взаємодії користувача (User flow)

