

Державний торговельно-економічний університет  
Кафедра інженерії програмного забезпечення та кібербезпеки

# ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА

на тему:

## «Програмний додаток для організації велодозвілля»

Студента 2м курсу, 2 групи,  
спеціальності 121 «Інженерія  
програмного забезпечення»  
освітньої програми «Інженерія  
програмного забезпечення»

\_\_\_\_\_

підпис студента

Чернюка Владислава  
Анатолійовича

Науковий керівник  
кандидат педагогічних наук,  
доцент кафедри інженерії  
програмного забезпечення та  
кібербезпеки

\_\_\_\_\_

підпис керівника

Котенко Наталія  
Олексіївна

Гарант освітньої програми  
кандидат педагогічних наук,  
доцент кафедри інженерії  
програмного забезпечення та  
кібербезпеки

\_\_\_\_\_

підпис керівника

Котенко Наталія  
Олексіївна

КИЇВ – 2023

Факультет інформаційних технологій

Кафедра інженерії програмного забезпечення та кібербезпеки

Освітній ступінь магістр

Освітня програма 121 «Інженерія програмного забезпечення»

**Затверджую**

Зав. кафедри інженерії програмного  
забезпечення та кібербезпеки

Криворучко О. В.

«13» грудня 2022 р.

### **Завдання на випускні кваліфікаційну роботу студентів**

Чернюку Владиславу Анатолійовичу

(прізвище, ім'я, по батькові)

1. Тема випускного кваліфікаційного проекту «Програмний додаток для організації велодозвілля».

Затверджена наказом ректора від "6" грудня 2022р. № 3285

2. Строк здачі студентом закінченого проекту (роботи) 1 грудня 2023р.

3. Цільова установка та вихідні дані до проекту (роботи)

Мета роботи – розроблення додатку, функціонал якого дає можливість спростити пошук людини для спільної велопогулянки а також ведення індивідуальної статистики поїздок на велосипеді.

Об'єкт дослідження – програмний додаток для організації велодозвілля.

Предмет дослідження – розробка архітектури та реалізація програмного додатку для організації велодозвілля.

4. Консультанти роботи із зазначенням розділів, які консультують:

Розділ	Консультант (прізвище, ініціали)	Підпис, дата	
		Завдання видав	Завдання прийняв

5. Зміст випускного кваліфікаційного проекту (перелік питань за кожним розділом)

ВСТУП

РОЗДІЛ 1. ВСЕБІЧНИЙ ОГЛЯД ВЕЛОПРОГУЛЯНКИ ЯК ВИДУ АКТИВНОГО ВІДПОЧИНКУ

1.1 Користь від велопогулянок

1.2 Популярність велотранспорту у світі

1.3 Велоіндустрія в Україні

1.4 Дослідження основних проблем та шляхів їх вирішення

1.5 Висновок до першого розділу

РОЗДІЛ 2. ПЛАНУВАННЯ ПРОГРАМНОГО ДОДАТКУ

2.1 Життєвий цикл розробки програмного забезпечення

2.2 Деталізація основних вимог до додатку

2.3 Поняття технологічного стека. Підбір технологій

2.4 Аналіз сутностей. Концептуальна, логічна та фізична моделі бази даних 29

2.5 Висновки до другого розділу

РОЗДІЛ 3. РОЗРОБКА ПРОГРАМНОГО ДОДАТКУ

3.1 Створення проекту у Firebase

3.2 Архітектура програмного забезпечення. State managent

3.3 Реалізація функціоналу авторизації

3.4 Модуль поїздок та статистики

3.5 Висновки до третього розділу

ВИСНОВКИ ТА ПРОПОЗИЦІЇ

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

ТЕХНІЧНЕ ЗАВДАННЯ

ПРОГРАМА ТА МЕТОДИКА ТЕСТУВАННЯ

ДОДАТКИ

## 6. Календарний план виконання проекту

№ пор.	Назва етапів випускного кваліфікаційного проекту	Строк виконання етапів проекту	
		за планом	фактично
1	2	3	4
1.	<i>Вибір теми випускного кваліфікаційного проекту</i>	07.11.2022	07.11.2022
2.	<i>Розробка та затвердження завдання на проєкт магістра (стац/заоч)</i>	13.12.2022	13.12.2022
3.	<i>Вступ та перелік літературних джерел</i>	24.02.2023	24.02.2023
4.	<i>Розробка технічного завдання</i>	15.03.2023	15.03.2023
5.	<i>Розділ 1. Всебічний огляд велопрогулянки як виду активного відпочинку</i>	10.04.2023	10.04.2023
6.	<i>Розділ 2. Планування програмного додатку</i>	24.05.2023	24.05.2023
7.	<i>Розділ 3. Розробка програмного продукту</i>	06.09.2023	06.09.2023
8.	<i>Розробка програми та методики тестування</i>	18.10.2023	18.10.2023
9.	<i>Написання наукової статті</i>	17.05.2023	17.05.2023
10.	<i>Керівництво користувача</i>	25.10.2023	25.10.2023
11.	<i>Висновки та пропозиції</i>	01.11.2023	01.11.2023
12.	<i>Здача випускного кваліфікаційного проекту на кафедру (перша перевірка)</i>	06.11.2023	06.11.2023
13.	<i>Підготовка автореферату та презентації доповіді</i>	06.11.2023	06.11.2023
14.	<i>Попередній захист випускного кваліфікаційного проекту</i>	20.11.2023 – 24.11.2023	20.11.2023 –24.11.2023
15.	<i>Здача зброшурованої випускного кваліфікаційного проекту</i>	01.11.2023	01.11.2023
16.	<i>Зовнішнє рецензування випускного кваліфікаційного проекту</i>	02.11.2023	02.11.2023
17.	<i>Підготовка до публічного захисту випускного кваліфікаційного проекту</i>	05.12.2023 – 06.12.2023	0 .12.2023

7. Дата видачі завдання «13» грудня 2022 р.

8. Науковий керівник випускного кваліфікаційного проекту Котенко Н.О.

(прізвище, ініціали, підпис)

9. Гарант освітньої програми Котенко Н.О.

(прізвище, ініціали, підпис)

10. Завдання прийняв до виконання студент Чернюк В.А.

Чернюк В.А.

(прізвище, ініціали, підпис)

# 11. Відгук керівника випускного кваліфікаційного проекту

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

Науковий керівник випускного кваліфікаційного проекту

\_\_\_\_\_ (підпис, дата)

Відмітка про попередній захист \_\_\_\_\_ (ПІБ, підпис, дата)

# 12. Висновок про випускний кваліфікаційний проект

Випускний кваліфікаційний проект студента \_\_\_\_\_ Чернюка В.А. (прізвище, ініціали) може бути допущена до захисту екзаменаційній комісії.

Гарант освітньої програми \_\_\_\_\_ Котенко Н.О. (прізвище, ініціали, підпис)

Завідувач кафедри \_\_\_\_\_ Криворучко О. В. (підпис, прізвище, ініціали)

« \_\_\_\_\_ » \_\_\_\_\_ 20 \_\_\_\_\_ р.

## АНОТАЦІЯ

Актуальність даної роботи полягає у відсутності зацікавленості населення у веденні активного способу життя, у даному випадку мова йде про велодозвілля. Можна виділити безліч факторів що так чи інакше впливають на це. Одним із таких факторів являється мобільне програмне забезпечення, а якщо бути точніше – їх кількість, яка близька нулю. Якщо ж поглянути на інші види спорту, наприклад фітнес, на відміну від велодозвілля, там ситуація значно краща – взяти хоча б мобільний ринок Google Play, що на Android. Проаналізувавши його, можна зробити висновок – додатків для фітнесу там дуже багато. Людина, що займається фітнесом, з легкістю вибере для себе потрібний застосунок, чого не скажеш про велодозвілля.

Мета цієї роботи заключається у створенні програмного додатку для велосипедистів, що дасть їм можливість аналізувати кожну поїздку та здійснювати пошук напарників. Таким чином цей додаток має на меті не тільки бути корисним для велосипедистів, а й збільшити зацікавленість людей у цьому виді активного відпочинку.

Випускна кваліфікаційна робота на тему “Програмний додаток для організації велодозвілля” містить 48 сторінок, 23 рисунка. Перелік використаних джерел налічує 9 найменувань.

**Ключові слова:** розробка мобільного додатку, велопрогулянка, база даних, Firebase, Flutter, віджет, авторизація, поїздки, статистика.

## ANNOTATION

The urgency of this work lies in the lack of interest of the population in leading an active lifestyle, in this case we are talking about leisure cycling. It is possible to single out many factors that in one way or another influence this. One of these factors is mobile software, or to be more precise – their number, which is close to zero. If we look at other sports, for example, fitness, in contrast to cycling, the situation is much better there – take at least the Google Play mobile market, which is on Android. After analyzing it, we can conclude that there are a lot of fitness apps out there. A person engaged in fitness will easily choose the right application for himself, which cannot be said about leisure cycling.

The purpose of this work is to create a application for cyclists that will allow them to analyze each trip and search for partners. Thus, this application aims not only to be useful for cyclists, but also to increase people's interest in this type of outdoor activity.

Graduation qualification work on the topic "Software application for organization leisure cycling" contains 48 pages, 23 drawings. The list of used sources includes 9 items.

**Keywords:** mobile app development, cycling, database, Firebase, Flutter, widget, authorization, trips, statistics.

## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

- ПЗ – Програмне Забезпечення;  
 SDLC – Software Development Life Cycle;  
 ПК – Персональний Комп’ютер;  
 UI – User Interface;  
 BaaS – Back-end as a Service;  
 API – Application Programming Interface;  
 SDK – Software Development Kit;  
 NoSQL – Not only SQL;  
 IDE – Integrated Development Environment;  
 MVC – Model View Controller;  
 MVVM – Model View ViewModel;  
 MVP – Model View Presenter;  
 GPS – Global Positioning System;

					<i>ДТЕУ 121– 02-23.МР</i>		
					<i>Програмний додаток для організації велодозвілля</i>		
					<i>Перелік умовних скорочень</i>		
<i>Зм</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>	<i>Стадія</i>	<i>Аркуш</i>	<i>Аркушів</i>
<i>Зав. каф.</i>		<i>Криворучко О.В.</i>		19.09.23	<i>ПС</i>	2	48
<i>Керівник</i>		<i>Котенко Н.О.</i>		19.09.23			
<i>Гарант</i>		<i>Котенко Н.О.</i>		19.09.23			
<i>Розроб</i>		<i>Чернюк В.А.</i>		19.09.23			
					Факультет інформаційних технологій, 2м курс, 2 група		



## ЗМІСТ

ВСТУП.....	4
РОЗДІЛ 1. ВСЕБІЧНИЙ ОГЛЯД ВЕЛОПРОГУЛЯНКИ ЯК ВИДУ АКТИВНОГО ВІДПОЧИНКУ .....	6
1.1 Користь від велопогулянок .....	6
1.2 Популярність велотранспорту у світі.....	12
1.3 Велоіндустрія в Україні.....	14
1.4 Дослідження основних проблем та шляхів їх вирішення .....	16
1.5 Висновок до першого розділу .....	18
РОЗДІЛ 2. ПЛАНУВАННЯ ПРОГРАМНОГО ДОДАТКУ .....	19
2.1 Життєвий цикл розробки програмного забезпечення .....	19
2.2 Деталізація основних вимог до додатку .....	21
2.3 Поняття технологічного стека. Підбір технологій.....	24
2.4 Аналіз сутностей. Концептуальна, логічна та фізична моделі бази даних.....	29
2.5 Висновки до другого розділу .....	33
РОЗДІЛ 3. РОЗРОБКА ПРОГРАМНОГО ДОДАТКУ.....	34
3.1 Створення проєкту у Firebase.....	34
3.2 Архітектура програмного забезпечення. State managment.....	36
3.3 Реалізація функціоналу авторизації .....	38
3.4 Модуль поїздок та статистики .....	42
3.5 Висновки до третього розділу.....	46
ВИСНОВКИ ТА ПРОПОЗИЦІЇ .....	47
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ .....	48
ТЕХНІЧНЕ ЗАВДАННЯ .....	49
ПРОГРАМА ТА МЕТОДИКА ТЕСТУВАННЯ .....	52
ДОДАТКИ.....	54

					<i>ДТЕУ 121– 02-23.МР</i>			
<i>Зм</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>	<i>Програмний додаток для організації велодозвілля</i>	<i>Стадія</i>	<i>Аркуш</i>	<i>Аркушів</i>
<i>Зав. каф.</i>		<i>Криворучко О.В.</i>		01.11.23		<i>Зміст</i>	3	48
<i>Керівник</i>		<i>Котенко Н.О.</i>		01.11.23				
<i>Гарант</i>		<i>Котенко Н.О.</i>		01.11.23				
<i>Розроб</i>		<i>Чернюк В.А.</i>		01.11.23		<i>Зміст</i>	Факультет інформаційних технологій, 2м курс, 2 група	

## ВСТУП

Питання актуальності спорту завжди мало важливе місце в культурному та фізичному житті суспільства, але кожен індивід по-різному оцінює значимість заняття фізичним вправами. Для однієї категорії людей, спорт – це марна трата часу, для іншої – це джерело позитивних емоцій та безкінечної енергії. Саме по собі виконання регулярних фізичних вправ щодня(або хоча б через день) робить людину не тільки більш здоровою, стійкою фізично до негативних факторів зовнішнього середовища, але і морально стійкою, оскільки людина повинна мати велику силу волі, щоб знаходити час та відволіктись від своїх буденних справ.

Метод пересування на двоколісному транспорті уже, безперечно, є швидшим, ніж ходити пішки. Іноді велосипеди стають більш вигідним ніж навіть персональні автомобілі або громадський транспорт, взяти хоча би в приклад ситуацію із заторами(оскільки вони сьогодні це уже не рідкість у великому місті). Завдяки велосипеду ви можете самостійно прокласти найкращий маршрут до точки не підлаштовуючись під карту пробок чи розклад, вартість проїзду у громадському транспорті. Вам не потрібно оплачувати місце для паркування – тримайте велосипед хоч вдома.

Таким чином, велопрогулянки – це не тільки різновид активного спорту, але і цілий суспільний рух, який, як і спорт в цілому, має на меті зміцнення стійкості організму шляхом здійснення кардіонавантаження, що має позитивний вплив на легені, м'язи та серцево-судинну систему. Цей рух можна частково віднести навіть до туризму, оскільки під час подолання значних дистанцій на велосипеді ви поступово відкриваєте для себе нові місця та знайомства.

**Об'єктом дослідження є програмний додаток для організації велодозвілля.**

					<i>ДТЕУ 121– 02-23.МР</i>			
					<i>Програмний додаток для організації велодозвілля</i>	<i>Стадія</i>	<i>Аркуш</i>	<i>Аркушів</i>
<i>Зм</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		<i>В</i>	<i>4</i>	<i>48</i>
<i>Зав. каф.</i>		<i>Криворучко О.В.</i>		<i>24.02.23</i>	<i>Вступ</i>	<i>Факультет інформаційних технологій, 2м курсе, 2 група</i>		
<i>Керівник</i>		<i>Котенко Н.О.</i>		<i>24.02.23</i>				
<i>Гарант</i>		<i>Котенко Н.О.</i>		<i>24.02.23</i>				
<i>Розроб</i>		<i>Чернюк В.А</i>		<i>24.02.23</i>				

**Предмет дослідження:** розробка архітектури та реалізація програмного додатку для організації велодозвілля.

**Мета роботи:** розробити додаток, функціонал якого дозволить спростити пошук людини для спільної велопогулянки а також ведення індивідуальної статистики поїздок.

**Задачі дослідження:**

- a) проаналізувати вплив ведення активного способу життя на здоров'я людини;
- b) визначити основні проблеми велодозвілля та шляхи їх вирішення;
- c) підготувати список вимог до ПЗ;
- d) спроектувати та створити back-end частину додатку;
- e) розробити клієнтську частину;

**Методи і технології розробки:** мобільна розробка під платформу Android, об'єктно-орієнтоване програмування мовою Dart з поєднанням фреймворку Flutter та Firebase в якості back-end частини.

					ДТЕУ 121–02-23.МР	Аркуш
Зм.	Аркуш	№ докум.	Підпис	Дата		5

## РОЗДІЛ 1

# ВСЕБІЧНИЙ ОГЛЯД ВЕЛОПРОГУЛЯНКИ ЯК ВИДУ АКТИВНОГО ВІДПОЧИНКУ

### 1.1 Користь від велопогулянок

Однією із причин існування та стимулом життя людини є щастя. Отримати ж його сьогодні можливо різними способами, проте задайте собі питання – чи є воно справжнім? У випадку з велопогулянкою можете сміло відповідати “так”.

Раніше ми розглянули корисні особливості велопогулянок в загальних аспектах, тепер же зробимо це на більш детальному рівні.

**Позитивний вплив на здоров'я.** Є одним із найважливіших пунктів, оскільки будь-яка людина, якій не байдуже за своє здоров'я, матиме на меті його не тільки зберегти, але і зміцнити:

а) *м'язи* – надзвичайно важливий елемент опорно-рухової системи, який, завдяки скороченню, здійснює свідому або несвідому діяльність (дихання, розмовляння, рухання і т.д.). Проявляючи фізичну активність ми нарощуємо мускулатуру і таким чином укріплюємо свій організм. Під час їзди на велосипеді ми циклічно крутимо педалі ногами, задіюючи (скорочуємо та розслабляємо) великі м'язи ніг – цей процес значно підвищує їх витривалість та надає їм більш рельєфного вигляду. Крім м'язів ніг, в процесі поїздки також беруть участь м'язи спини, груд та рук (вплив на них ідентичний).

б) *зміцнення нервової системи* є доволі цікавим аспектом велопогулянок. Виникнення питання на кшталт “яким чином все це пов'язано?” – це не рідкість у людей, які не часто користуються двоколісним “конем”. Проте

					<i>ДТЕУ 121–02-23.МР</i>			
					<i>Програмний додаток для організації велодозвілля</i>	<i>Стадія</i>	<i>Аркуш</i>	<i>Аркушів</i>
<i>Зм</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		<i>РІ</i>	<i>6</i>	<i>48</i>
<i>Зав. каф.</i>		<i>Криворучко О.В.</i>		10.04.23				
<i>Керівник</i>		<i>Котенко Н.О.</i>		10.04.23				
<i>Гарант</i>		<i>Котенко Н.О.</i>		10.04.23				
<i>Розроб</i>		<i>Чернюк В.А.</i>		10.04.23	<i>Всебічний огляд велопогулянки як виду активного відпочинку</i>	<i>Факультет інформаційних технологій, 2м курс, 2 група</i>		

від роботи та подарить гормони щастя. Опитування Cycleplan, яке вивчало переваги для здоров'я що відчули люди, які почали їздити на велосипеді, показало, що 75% велосипедистів помітили покращення свого психічного здоров'я після того, як сідали на сідло, а 8% навіть сказали, що це допомагає впоратися з депресією або тривогою. «Здорове тіло, здоровий дух» дійсно є правдою, і їзда на велосипеді може зіграти в цьому вирішальну роль. А ось нещодавній огляд 26-річних наукових досліджень Університету Торонто підтвердив те, що багато експертів уже давно і так припускають – фізичні вправи не тільки лікують депресію, але й можуть її запобігти. Насправді, дослідники підраховали, що неактивна доросла людина, яка займається спортом три рази на тиждень, може знизити ймовірність страждати від депресії на 19%.

- с) *серцево-судинну систему* не обійдемо стороною, оскільки регулярні велопогулянки сприяють зниженню розвитку серцевих захворювань. Кардіотренування, яке відбувається під час поїздки, пришвидшує кровообіг, що, у свою чергу, стимулює судинну систему та очищає її від відкладень на стінках кровоносних судин. Особливо це корисно для курців та для тих, хто працює на шкідливих виробництвах.
- д) *покращення роботи легенів* – невід'ємна частина поїздки, під час якої їх робота стає більш інтенсивнішою через збільшення частоти вдихів та видихів, що призводить до активного виведення токсинів та зменшує ризики виникнення хвороби легенів.
- е) завдяки проведенню дослідженню впливу велопоїздок на організм людини, було виявлено, що регулярні велопогулянки мають вплив на *зменшення ознак старіння*. Першою причиною цього стало посилення кровообігу, яке дозволяє клітинам шкіри ефективніше отримувати поживні речовини, насичуватися киснем та виводити токсини. Другою причиною визначили вироблення колагену – речовини, яка бореться зі зморшками.

- f) останнім часом велопоїздки стали популярним та швидким методом для *схуднення*. Це і не дивно, бо справа в тому, що прогулянка на велосипеді є легким, зручним, доступним, зручним та ефективним видом фізичної активності через низьку інтенсивність, аеробні та послідовні вправи. Рух зі швидкістю 20 км/год може спалити приблизно 700 калорій. Тому, щоб почати втрачати зайву вагу вам буде достатньо лише однієї години, проте важливо також зазначити факт, що організм буде спалювати калорії не тільки в процесі поїздки, але і протягом деякого часу після неї, тому варто підійти до цього з усією відповідальністю та не переїдати після поїздки.
- g) велопогулянки є корисним методом для *тренування зору*. Причиною цьому є те, що ми під час поїздки не тільки дивимося під колеса чи на дорогу, але і постійно фокусуємо всю увагу на околицях та пейзажах – цей процес змушує постійно міняти наш фокус. Саме такі вправи знижують ризики короткозорості та загострюють зір.
- h) *оздоровлення сну* через регулярні катання на велобайку сприяє виведенню з організму стресового гормону – кортизолу, який посилює дію адреналіну і норадреналіну, завдяки чому організм стає більш вразливим до стресових ситуацій. Цей же гормон буде заважати заснути після важкого робочого дня. Тому його виведення позбавить вас від безсоння.

**Питання збереження навколишнього середовища** поступово переходить на передній план. Екологічна ситуація стає все гіршою та гіршою, переходить у стан кризи. У великих містах все частіше можна почути скарги щодо якості повітря, води, продуктів, і як результат цього всього – збільшення хвороб, зменшення якості життя і його тривалості. Особливе місце у цій ситуації займає процес зміни клімату, який частково є результатом діяльності людини, а саме: спалення вугілля, нафти та газу, використання неекологічних заводів і тд. Пасивно спостерігаючи це все виправити просто неможливо, тому частина людства вже почала вживати активні заходи(перехід на використання альтернативної електроенергії, тощо). Тоді ж яку

										Аркуш
Зм.	Аркуш	№ докум.	Підпис	Дата	ДТЕУ 121-02-23.МР					8

роль відіграє використання велотранспорту? Відповідь на це питання буде розкрито в наступних тезах:

- а) зменшення частки парникових газів в атмосфері та частоти появ парникового ефекту. Парниковий ефект – явище, яке виникає внаслідок затримки та відбиття енергії сонячних променів від поверхні різними газами(зокрема вуглекислого). Основною причиною цього процесу на сьогодні є транспорт, а якщо бути точніше – їхні двигуни, оскільки саме вони спалюють паливо та виділяють вуглекислий газ( $CO_2$ ). Проте різні види транспорту мають і різний ступінь виділення. Таким чином, на графіку (рис. 1.1) нижче можна побачити, що найбільш безпечним є залізничний, а найбільший вплив має звичайний автомобільний транспорт. Найнеприємнішим є два наслідки парникового ефекту, які можуть призвести навіть до глобальної катастрофи – потепління клімату. У першому випадку відбудеться значне збільшення в основних зернових районах, зокрема в Україні. Другий же випадок є доволі актуальним, оскільки його можна вже спостерігати – це підйом рівня Світового океану на 2-3 метри за рахунок танення полярних льодових шапок, результатом чого виявиться затоплення найбільш заселених частин суходолу.

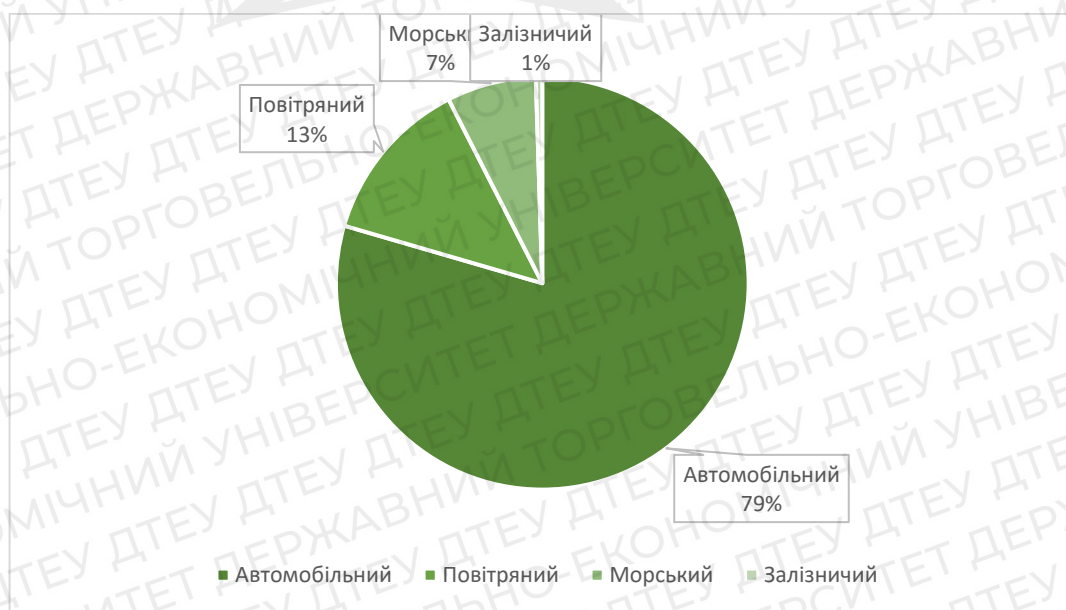


Рис. 1.1. Частка викидів парникових газів різними видами транспорту

Зм.	Аркуш	№ докум.	Підпис	Дата

Основним же “двигуном” велобайків є наша енергія, тому, вибравши його як альтернативу для поїздки на навчання чи роботу, ви зробите свій внесок у збереження клімату.

б) виключаємо шумове забруднення – однієї з форм хвильового забруднення навколишнього середовища. Найчастіше ця форма забруднення зустрічається у великих містах, а сільські райони не страждають цією хворобою. Джерел виникнення – переважно транспортні засоби, фабрики, деякі інфраструктурні споруди(наприклад аеропорти). На перший погляд, людині, яка проживає поряд з вище перерахованими джерелами шуму, може здатися, що ніякого впливу немає, проте це не так, а причиною цього є звикання(не путати з адаптацією). Постійний вплив шуму має наслідки як на людей, так і на тварин. Для людей – це безсоння(що скажеться на продуктивності), при умові перевищення 65-85 децибел можуть виникати проблеми із серцем, а також із органами слуху. Брати ж наші менші зазнають більше страждань. Тварини мають шанс втратити слух чи зазнати дезорієнтацію. Таким чином, шумове забруднення спричиняє дисбаланс у природному перебігу екосистем. Велосипед є “тихим” видом транспорту, тому його використання значно зменшує рівень шуму та його вплив.

с) забруднення води автомобільними засобами. Попадання таких речовин(що використовуються для обслуговування транспортних засобів) як гальмівної рідини, антифризу чи моторного мастило є одним із основних джерел забруднення води. Чого не скажеш про велосипеди.

**Заощадження на обслуговуванні та покупці транспортного засобу.** Коли мова починає йти про витрати, то значна частина суспільства має на меті заощадити лишню копійчину так, що товар буде відповідати їхнім очікуванням. Купуючи велобайк ви не тільки заощаджуєте, а і навіть робите інвестиції (під інвестицією необхідно розуміти поступове зміцнення стану здоров’я, яке було описано вище) у майбутнє, успішність яких буде залежати уже тільки від вас самих, як це банально

									Аркуш
									10
Зм.	Аркуш	№ докум.	Підпис	Дата	ДТЕУ 121-02-23.МР				



би не звучало. Розглянемо детальний список на чому ж таки можна зекономити використовуючи велосипед замість автомобілю:

- а) відсутність витрат на технічне обслуговування. Утримання автомобіля часто займає ліву частину витрат після купівлі авто (рис. 1.2). Чого тільки варте постійне поповнення баку. Бувають випадки, коли на ремонт уходить коштів навіть більше, ніж на купівлю самого авто. Тому необхідно трезезо оцінювати своє фінансове положення. Велосипед же, у свою чергу, не вимагає подібних витрат.

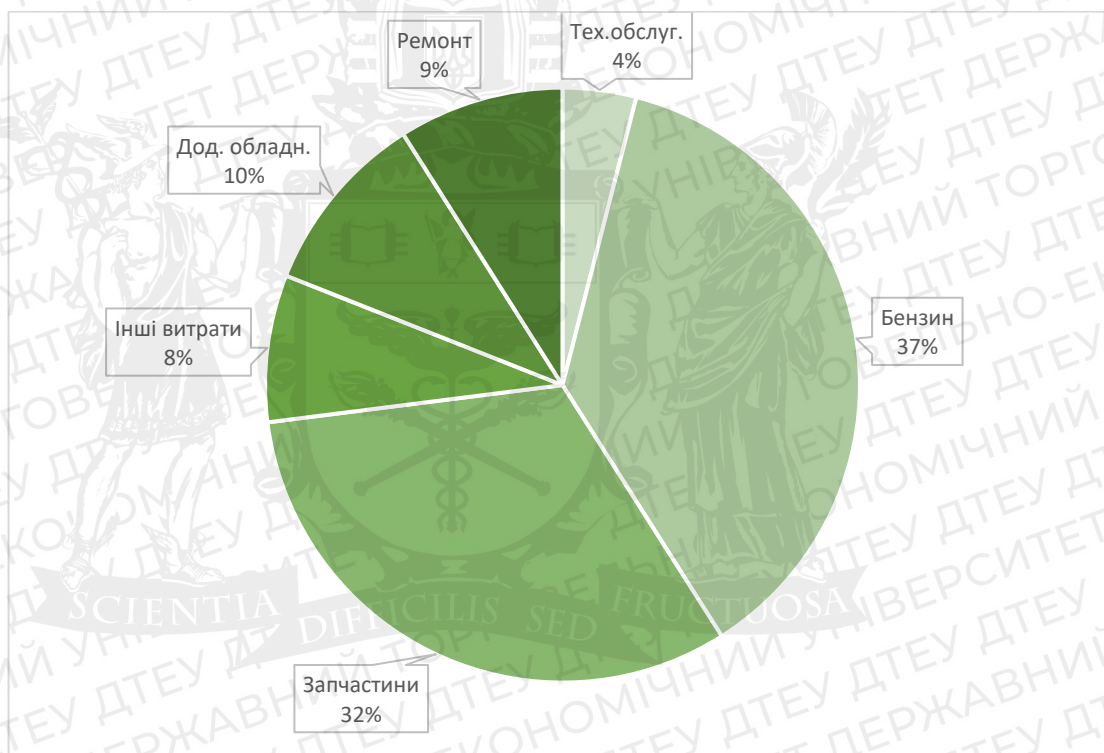


Рис. 1.2. Сумарні витрати на автомобіль

- б) економія часу. Через свої габарити автомобілю не під силу проїхати там, де це може зробити велобайк. Ситуацію із заторами також згадаємо, оскільки сівши за кермо двоколісного автоматично відпаде необхідність витратити власний дорогоцінний час на звичайне безкорисне простоювання.

**Укріплення соціальних зв'язків.** Активний відпочинок – найкращий вибір для проведення вільного часу з родиною чи у колі друзів, що чудово зміцнить

Зм.	Аркуш	№ докум.	Підпис	Дата

відносини. Велопогулянку з легкістю можна зробити сімейною традицією, прищеплюючи любов до спорту у дітей.

## 1.2 Популярність велотранспорту у світі

Якщо ми заглянемо в актуальний на сьогодні рейтинг популярності видів активного відпочинку, то побачимо, що поїздки на велосипеді займають п'яте місце, після спортивної ходьби, плавання, занять на тренажері та туризму. Проте, крім активного відпочинку, велобайк широко використовується в якості виду транспортного засобу (рис. 1.3).

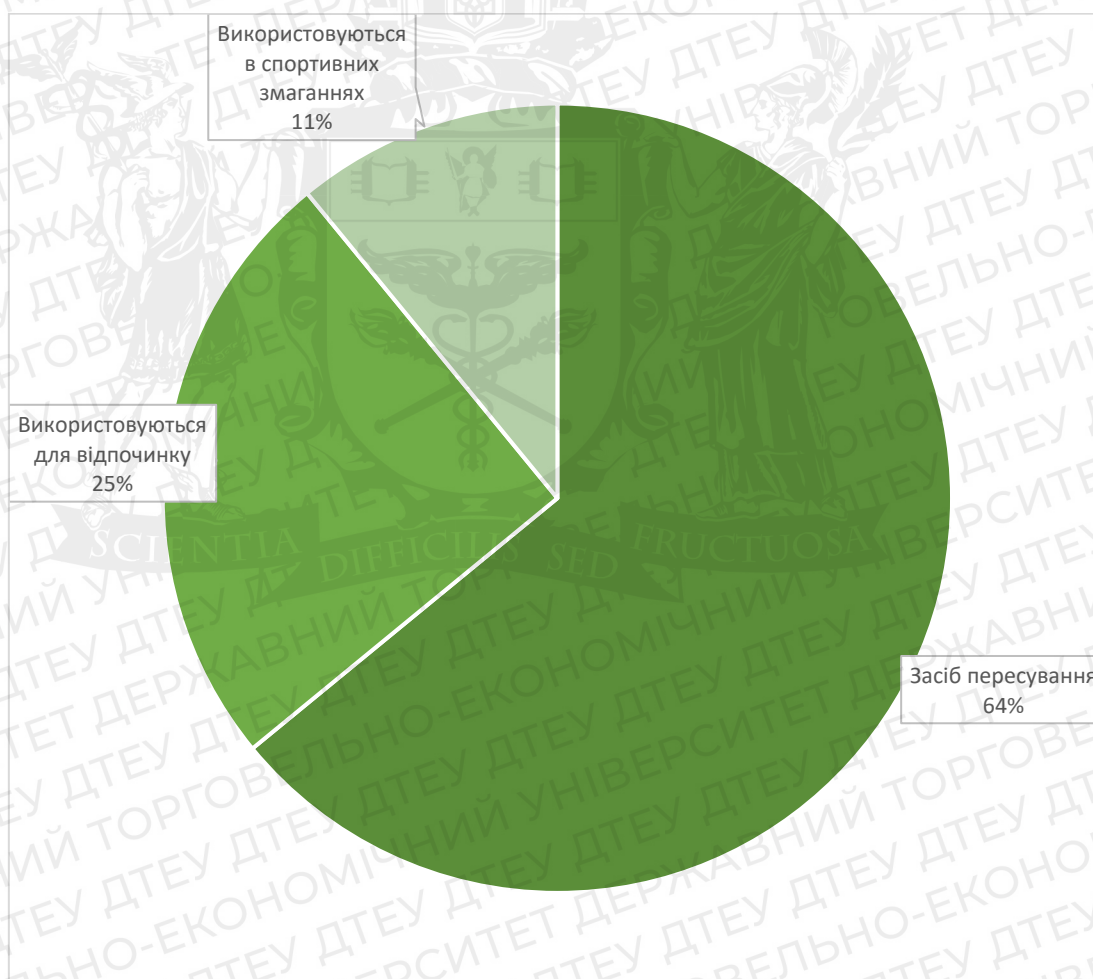


Рис. 1.3. Способи використання велотранспорту

У країнах з більш розвинутою економікою велосипед займає місце другорядного засобу пересування чи одного із видів активного відпочинку. Спосіб використання часто залежить від самої країни, зокрема її розвиненості (рис. 1.4). Так, у Китаї, США та низці європейських країн, де двухколісного може

Зм.	Аркуш	№ докум.	Підпис	Дата

дозволити майже кожний мешканець, велосипед не є основним засобом транспорту(можна зробити припущення, що основним транспортним засобом у цій ситуації є автомобіль), а виступає в ролі засобу для активного відпочинку.

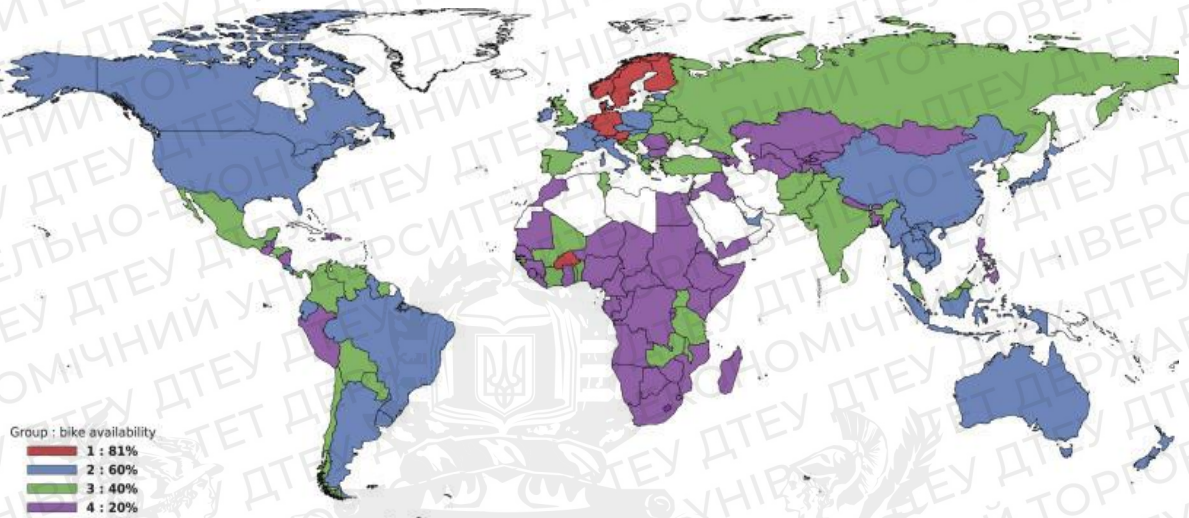


Рис. 1.4. Доступність велосипедів по країнам

Цікавим фактом є те, що велосипед може позначати соціальне та економічне положення людини у суспільстві(її статус). Наприклад, суспільство розвинених країна сприймає велосипедистів як звичайних учасників дорожнього руху. Чого не скажеш про країни, що розвиваються, – поїздка на велобайку дуже часто сприймається як бідність, або навіть зустрічаються погляди з презирством, оскільки “людина навіть не може оплатити квиток на автобус”. Тому деякі люди навмисне надають перевагу витратити частину власної заробітної плати на поїздку автобусом, в іншій же ситуації – ходити пішки.

Підсумувати популярність використання велосипедом можна спеціальним рейтингом (рис. 1.5), в якому побачимо, що країною, в якій велосипед набув найбільшого призначення є Нідерланди.

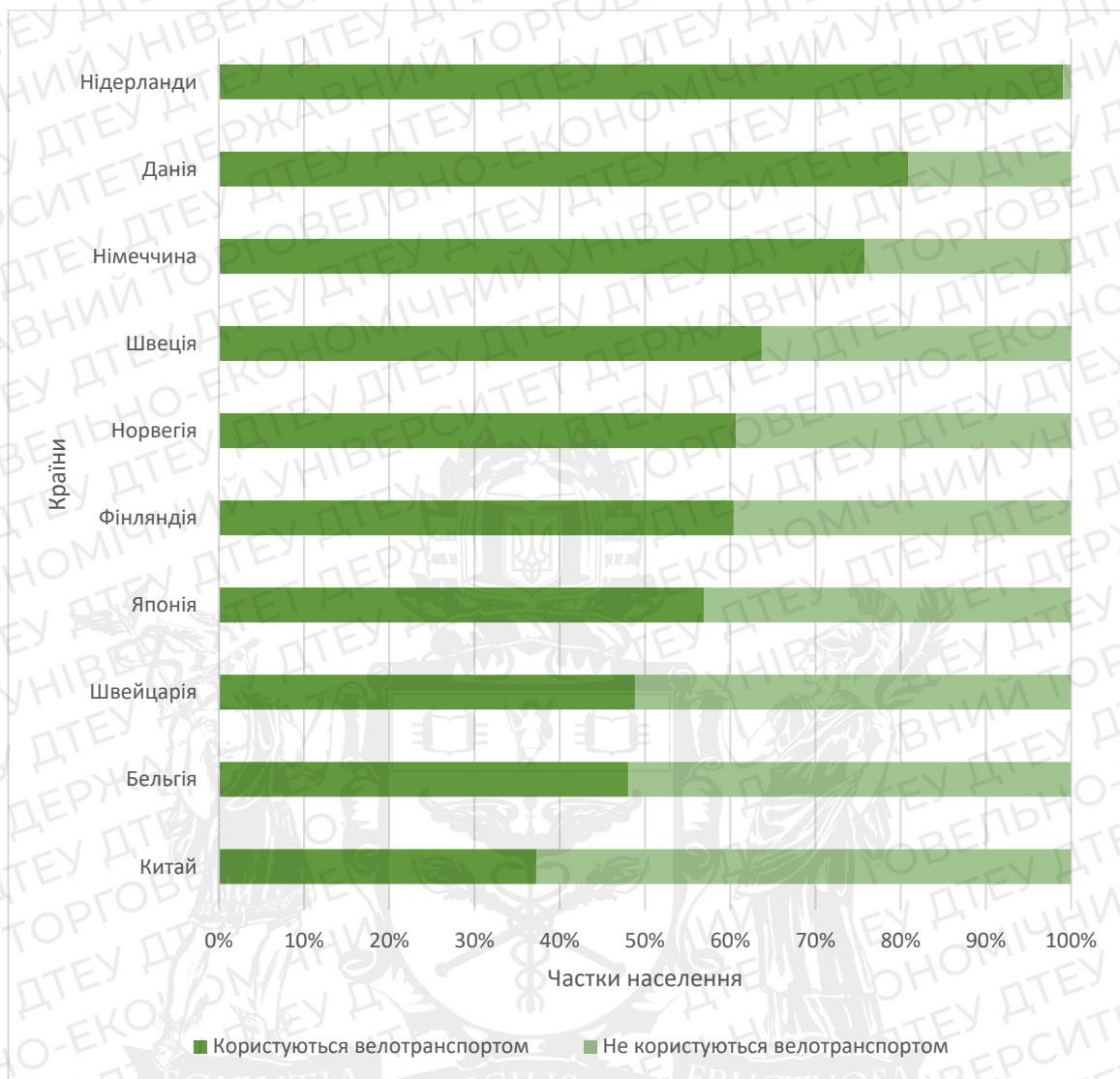


Рис. 1.5. Графік популярності користування велобайком

### 1.3 Велоіндустрія в Україні

Бути велосипедистом в Україні, скажемо відверто – справа не з простих. Причиною цього, в першу чергу, є *відсутність велосипедної інфраструктури*. Для пояснення такого терміну як “велосипедна інфраструктура” необхідно розповісти спочатку про її складові:

1. **Шляхи.** Дуже часто цей фактор вважають відповіддю, проте це лише невід’ємна її частина. Виділяють два основних типи велосипедних шляхів: велодоріжки, що можуть бути суміщені з пішохідними доріжками або розташовані окремо, і велосмути – елементи проїзної частини, призначені для руху велосипедистів.

2. **Засоби організації дорожнього руху** – основні чинники регулювання пішохідних та транспортних потоків. До них відносять: засоби інформування(дорожні знаки, світлофори, дорожня розмітка), пристрої, які забезпечують функціонування засобів інформації(контролери, детектори, комп'ютери, засоби диспетчерського зв'язку та інші) і засоби подолання перепадів висот(рампи, ліфти, підйомники).

3. **Об'єкти сервісу** – споруди, призначення яких полягає в обслуговуванні велосипедистів(заклади харчування, технічного обслуговування, пункти зберігання велотранспорту).

4. **Елементи благоустрою** – засоби, що надають позитивних емоцій, естетичної насолоди учасникам руху(скульптури, фонтани, лавки, озеленення).

5. **Штучні споруди** – інженерні споруди, ціль яких – спрощення подолання природних перешкод(мости, тунелі і т.д.).

6. **Соціальний захист користувачів велотранспорту.**

Незважаючи на відсутність елементів велосипедної інфраструктури – люди все ж таки продовжують купувати та використовувати велотранспорт. Однією з причин стало й поширення коронавірусу.

Пандемія зробила свій внесок, через страх зараження люди почали надавати перевагу альтернативним, більш безпечним(з точки зору відсутності контактів з іншими особами) засобам пересування. “Спостерігаючи ситуацію щодо індивідуального транспорту в Європі під час пандемії, я розумів, що щось подібне станеться і в нас” – такими є твердження Олексія Кушки(власника сайту [veliki.ua](http://veliki.ua) та магазину с продажу товарів для активного відпочинку [shop.veliki.ua](http://shop.veliki.ua), співзасновник сервісу з прокату велосипедів [Vike Now](http://Vike Now)). За його словами попит на велосипеди у пандемію виріс у 2-4 рази.

Інша ситуація - мешканець Києва Лев вирішив купити велосипед під час карантину, щоб добиратися до місця робочого розвезення. «На початку карантину дивився велосипеди на OLX вартістю до 4000 грн. Пропозицій було багато як по Києву, так і загалом по Україні. Коли зайшов на платформу за місяць – у цьому

										Аркуш
Зм.	Аркуш	№ докум.	Підпис	Дата						15

ДТЕУ 121-02-23.МР

ціновому сегменті залишилися лише дитячі велосипеди», — каже Лев. У OLX(один із самих популярних інтернет-магазинів в Україні) зазначають, що на платформі купували як нові, так і вживані велосипеди. Попит на останній був вищим. У період з березня по вересень у Дніпропетровській області замовили вживаних велосипедів на 27% більше, ніж нових, у Харківській — на 15%, у Київській — на 14%.

На Київщині піковий попит на вживані велосипеди був у травні. За даними OLX Доставки - середній чек складав 3 280 грн. У наступні місяці він виріс, і в серпні дорівнював 4 166 грн.

На Львівщині бум попиту на велосипеди припав на квітень. Середній чек - 4490 грн. У наступні місяці кількість замовлень трохи знизилася. Також у Івано-Франківській області у квітні порівняно з травнем зросла кількість покупок велосипедів на 61%.

У цих умовах стрімко розвинувся дефіцит на малобюджетні моделі, що привело до збільшення популярності велопрокатів. Одним із лідерів є bikepow – українська система громадського велопрокату. Як зазначено на їх сайті, кількість оренд в день сягає 5000, а ціна за хвилину користування 0,5 гривень, тобто в час – 30 грн.

З однієї сторони ситуація не може не радувати, все більша частина людей починає входити у велоіндустрію, з іншою ж – ціна цих змін занадто велика.

#### **1.4 Дослідження основних проблем та шляхів їх вирішення**

Неможливо не помітити стрімкий технологічний прогрес, що має все більший і більший вплив на життя кожного. Уже давно не рідкість те, що людина задля спрощення життя, створює технологічні пристрої, які часто зосереджені на тому, щоб зменшити(або зовсім прибрати) фізичну частку в тій чи іншій сфері діяльності. Таким чином, розвиток технологій, в більшій мірі, спричиняє деградацію фізичної якості життя суспільства, це і є перша проблема – для її рішення людина починає замислюватися та переглядати власний стиль життя, додаючи в нього часточку спорту. Звісно з різних причин можливість вибрати

									Аркуш
Зм.	Аркуш	№ докум.	Підпис	Дата	ДТЕУ 121-02-23.МР				16

велоспорт мають не усі(наявність травм, які не припустимі для поїздки на велосипеді, чи бажання людини), у цьому випадку варто дивитися у бік більш безпечних для вас видів активного відпочинку. Так або інакше, вирішивши що прогулянки на велосипеді – це те, що вам потрібно, ви перейдете у наступний етап, де зустрінетеся з іншими причинами, які можуть негативно вплинути на перший досвід велосипедиста. З цих причин і створюються сервіси, різні мобільні додатки, ведуться передачі/блоги в інтернеті чи на телебаченні, пишуться статті, їх спільна ціль – мінімізувати негативні фактори(тобто проблеми) з якими ви можете зустрітися(рекомендації по купівлі велосипеда для новачків, найкращі місця для поїздки або звичайний пошук компанії) через надання рекомендацій чи готових рішень щоб їх обминути.

У даній роботі мова буде йтися про розробку *програмного додатку* для велодозвілля. Всесвітньо відомі онлайн-магазини додатків, такі як Google Play чи Apple Store, на сьогодні мають досить бідний каталог додатків для велопоїздки, функціонал яких не є широким та лиш частково заключається у вирішенні чи спрощенні буденних проблем велосипедистів.

Мабуть будь-який розробник, який створює програмне забезпечення, прагне, щоб його праця не була марною. Тому враховування величезної кількості факторів є дуже важливим щоб наш додаток міг конкурувати з іншими та став корисним у загальному плані для потенційних користувачів. Як висновок, визначимо основні тези про функціональні особливості додатку, що зроблять його корисним:

- a) *пошуком компаньйона або цілої компанії* для велопоїздок – не кожен має друга, який також не проти виділити час для прогулянки на велосипеді. Тому для рішення цієї проблеми було поставлено завдання по розробці функціоналу в додатку, що дозволить кожному користувачеві знайти у своєму регіоні людину для спільної поїздки.
- b) *відслідковування власної статистики*. З кожної поїздки велосипедист має вижимати максимум як користі для власного здоров'я, так і

інформації. Наявність статистики дасть можливість аналізувати маршрут та те, як Ви пройшли його.

с) *інтерактивна карта з відслідковуванням місцезнаходження ваших друзів у режимі реального часу*. Загубитися через відставання або випередження від ваших друзів стане в рази складніше.

### 1.5 Висновок до першого розділу

Таким чином, у першому розділі було проаналізовано та визначено головні фактори впливу ведення активного способу життя, зокрема велопогулянок, на здоров'я людини. Згадаємо основні: зміцнення м'язів, зменшення вірогідності появи депресії, зниження розвитку серцево-судинних захворювань, похудання. Окрім цього, середовище теж позитивно реагує на збільшення частки велосипедистів у суспільстві, а саме:

- а) зменшення викидів парникових газів в атмосфері разом;
- б) мінімізація рівня шумового забруднення;

З'ясовано, що ступінь популярності та способи використання велосипедів у багатьох країнах відрізняються, а причиною цього являється соціальне та економічне положення людей.

Ситуацію зі станом розвитку велосипедної інфраструктури і популярності велодозвілля в Україні не обійшли стороною. Була зазначено про низький стан розвитку велоіндустрії, проте тенденція має позитивний курс.

Досліджено основні проблеми, з якими може зустрітися новачок або людина з досвідом у сфері велодозвілля. Поставлено завдання на розробку програмного продукту, ціллю якого є допомога з пошуком компаньйона для спільної поїздки та ведення індивідуальної статистики.

					ДТЕУ 121–02-23.МР	Аркуш
Зм.	Аркуш	№ докум.	Підпис	Дата		18



## РОЗДІЛ 2

### ПЛАНУВАННЯ ПРОГРАМНОГО ДОДАТКУ

#### 2.1 Життєвий цикл розробки програмного забезпечення

Процес розробки будь-якого програмного продукту представляє собою певний набір послідовних етапів, які, поєднуючись, утворюють так званий *життєвий цикл розробки програмного (SLDC)*. Цей цикл охоплює проміжок часу від моменту зародження самої ідеї і до моменту реалізації на ринку повноцінного продукту з подальшою підтримкою.

Поділ життєвого циклу на етапи відбувається через вплив певних факторів. Першим фактором являється *комплексність* сьогоденних додатків та вимог до них. Функціонал, архітектура сучасного програмного забезпечення значно складніша, ніж була раніше. Причиною цьому є розвиток бізнесу.

Комплексні програми вимагають від себе певну послідовність чітких дій під час розробки, а не їх хаотичність. Досвід багатьох компаній і розробників показує, що у разі разі наявного безладу у проекті, несистематизованості вимог, цілей чи термінів – проект має катастрофічні шанси зазнати провалу.

Таким чином, саме задля мінімізації шансів втрат фінансових ресурсів і провалу проекту загалом, були визначені етапи розробки, які дають ряд таких *переваг*:

- а) *чіткість описаних цілей*, що стоять в основі кожного етапу. Їх знання дозволяють розробникам та клієнтам відслідковувати прогрес у створенні ПЗ, правильно виставляти терміни.

					<i>ДТЕУ 121–02-23.МР</i>			
						<i>Стадія</i>	<i>Аркуш</i>	<i>Аркушів</i>
<i>Зм</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>	<i>Програмний додаток для організації велодозвілля</i>	<i>P2</i>	<i>19</i>	<i>48</i>
<i>Зав. каф.</i>		<i>Криворучко О.В.</i>		<i>24.05.23</i>				
<i>Керівник</i>		<i>Котенко Н.О.</i>		<i>24.05.23</i>				
<i>Гарант</i>		<i>Котенко Н.О.</i>		<i>24.05.23</i>				
<i>Розроб</i>		<i>Чернюк В.А</i>		<i>24.05.23</i>	<i>Планування програмного додатку</i>	<i>Факультет інформаційних технологій, 2м курс, 2 група</i>		

б) *послідовність*. Як було раніше сказано – етапи життєвого циклу розробки є послідовними. Це означає, що не можна перейти до іншого етапу доти, доки не буде готовий та затверджений з менеджером і клієнтом повністю виконаний поточний етап. Завдяки чому розробники фокусуються не на всьому об’ємі задач у проекті, а лише над певною її частиною.

Поділ етапів життєвого циклу одного програмного забезпечення у більшості випадків не залежить від платформи, поставленого завдання чи компанії, що створювала це ПЗ. Тому список етапів можна привести до наступного вигляду:

- а) етап *дослідження та аналізу*;
- б) етап *планування*;
- с) етап *розробки*;
- д) етап *випуску продукту та здійснення підтримки*;

Життєвий цикл починається з етапу *дослідження та аналізу*. Під час нього розробники формулюють список вимог до кінцевого продукту та функціоналу. Відбувається цей процес разом у парі з аналізом сфери бізнесу та ринку, куди має вийти продукт.

Як тільки список функціоналу був сформований, а вимоги визначені – розпочинається *етап планування*. На цій стадії розробники та системні архітектори проектують високорівневий дизайн системи, визначають список технологій, що має бути задіяний під час розробки.

Грунтуючись на спроектованій архітектурі, дизайні та вимогам до функціоналу, розробники починають писати код програми. Це і є *етап розробки*. У більшості випадків, задля кращого відслідковування прогресу, сам процес розробки ділять на певні частини або як це називається по-іншому – модулі. Наприклад модуль авторизації у програмі для чату і тд. Крім самої розробки продукту, цей етап також включає процес його тестування.

Як тільки код програми був написаний, а усі її частини протестовані – настає *момент випуску програмного продукту* на ринок. Прикладом найбільш популярних на сьогодні ринків мобільних додатків є *Play Store (Android)* чи *App*

									Аркуш
									20
Зм.	Аркуш	№ докум.	Підпис	Дата	ДТЕУ 121-02-23.MP				

*Store* (iOS). Після випуску підключається відділ технічної підтримки, який забезпечує зворотній зв'язок з користувачами, консультування та підтримку.

Весь перший розділ випускної кваліфікаційної роботи був присвячений як раз таки лише першому етапу життєвого циклу програмного забезпечення, додатково окресливши рамки функціоналу мобільного додатку. В цьому ж розділі ми перейдемо до етапу планування.

## 2.2 Деталізація основних вимог до додатку

Список вимог до додатку прийнято об'єднувати у спеціальний документ, що називається специфікацією вимог. Крім вимог, у ньому зазначені й інші характеристики продукту: його мета, абрєвіатури, обмеження і так далі.

Вимоги прийнято поділяти на:

- a) *функціональні*, що стосуються загальної логіки роботи додатку;
- b) *нефункціональні*, що стосуються зовнішнього інтерфейсу користувача (UI) та обмеження в розробці;

Чіткість, якість сформульованих вимог оцінюється завдяки спеціальним критеріям. Цих критеріїв доволі багато, тому розглянемо лише основні.

*Атомарність* визначає, що вимога не повинна мати властивість додатково розділятися на декілька інших самостійних вимог. Якщо таке можливо, то це викличе більше питань під час розробки.

Критерієм, по якому визначається чи суперечить вимога іншим вимогам, архітектурі системи, називається *послідовність*.

*Здійсненність* вказує на можливість реалізації тієї чи іншої вимоги на певній системі.

Іншими критеріями є *завершеність*, *відстеження*, *актуальність*, *здійсненість*, *зрозумілість*, *верифікованість*, *обов'язковість*, *ідентифікація*, *модифікація* та *повнота*.

У попередньому розділі було визначено такий набір функціоналу додатку:

- a) надання можливості спільних поїздок з іншими користувачами;

Зм.	Аркуш	№ докум.	Підпис	Дата

- b) ведення статистики поїздок;
- c) використання карти як навігатора під час поїздки;

У даному вигляді вони не відповідають критеріям якості вимог, зокрема атомарності, тому виникає необхідність у додатковій їх деталізації.

Вищеописані пункти мають важливу особливість. Цією особливістю є прив'язаність до конкретного користувача. Неможливо зберегти дані про поїздку людини не ідентифікуючи її саму. Те ж саме можна сказати про процес поїздки. Виходячи з цього, стає більш зрозуміло, що нашою першою вимогою стане *використання системи аккаунтів*. Це допоможе нам ідентифікувати користувачів і зберегти їх дані.

При додаванні системи аккаунтів варто відразу враховувати і те, як користувач зможе увійти чи зареєструватися, які дані для цього знадобляться. Розрізняють велику кількість способів входу в аккаунт – вхід через Google, за номером телефону, за поштою і тд. У нашому випадку вистачить і класичного – *вхід з використанням електронної пошти користувача та паролю*. З цього випливає, що користувач, створюючи новий аккаунт, повинен обов'язково *вводити пароль і свою електронну пошту*, адже вона потім знадобиться для входу.

Підіб'ємо підсумки по вимогам щодо системи аккаунтів:

- a) користувач повинен мати *можливість створення аккаунту*. Обов'язкове використання пошти та паролю.
- b) користувач повинен мати *можливість увійти у власний аккаунт* використовуючи пошту та пароль.

Настав час скласти список вимог для *процесу запуску поїздки*. Перш за все, визначимо ролі учасників. Користувачем, що запросив іншого надалі буде називатися *ініціатором поїздки*. Той, хто отримав запрошення і прийняв його – *запрошеним*. Сформулюємо список етапів поїздки:

- a) *початковий етап*. На екрані має відображатися карта з локацією самого користувача, а навколо – інші користувачі, яких можна запросити на спільну поїздку. Важливо, що відображатися мають лише люди, що

					ДТЕУ 121–02-23.МР	Аркуш
Зм.	Аркуш	№ докум.	Підпис	Дата		22

знаходяться поряд з Вами. Якщо країни чи міста користувачів різні – реалізувати спільну поїздку буде надважким завданням.

- b) *старт*. Цей етап має розпочинатися як тільки хтось прийняв запрошення. Його особливість – вибір спільної точки фінішу. Обиратиме спільну точку ініціатор, в той час запрошений має її підтвердити. Як тільки точка буде вибрана і підтверджена – розпочнеться відлік до початку наступного етапу, який називається “поїздка”.
- c) під час *етапу поїздки* учасники мають бачити лише один одного на карті і нікого іншого. Для кожного учасника будується індивідуальний маршрут до точки фінішу.
- d) як тільки обидва користувачі прибудуть до точки – починається етап *завершення*, який характерний відображенням статистика поїздки та її збереження.

Вимоги до самих етапів визначено, проте варто також додати ще одну – *можливість відмінити поїздку у будь-якій її етап*. Ніхто не застрахований від попадання у випадкові ситуації і бажання відмінити поїздку цілком матиме сенс. Важливо зазначити, що у цьому разі дані статистики *не мають бути збережені*.

Таким чином, було розглянуто основні вимоги до поїздок. Залишилося розібратися зі статистикою. Перш за все, статистика потрібна для відслідковування індивідуальних показників, тому краще зробити акцент на самих даних. Кожна успішно завершена поїздка на двухколісному повинна має характеризуватися наступними показниками:

- a) зображення маршруту;
- b) час старту;
- c) час фінішу;
- d) загальна тривалість;
- e) середня швидкість;
- f) учасники;
- g) максимальна швидкість;

Зм.	Аркуш	№ докум.	Підпис	Дата

ДТЕУ 121–02-23.МР

Аркуш

23

А завершимо тему з вимогами на визначенні однієї із основних нефункціональних вимог – *вибору платформи*.

Додаток має бути завжди “поруч” з користувачем в момент поїздки, адже потрібно слідкувати за маршрутом. Тому варіант зі створенням програми на персональні комп’ютери відразу відпадає. З інших напрямків залишається створення мобільного додатку або веб-сайту.

Розглядаючи питання вибору платформи для реалізації нашого продукту, ми повинні взяти до уваги декілька факторів, які обов’язково мають бути втілені у вибраному напрямку:

- а) можливість побудови зручного інтерфейсу користувача;
- б) наявність висока продуктивність (особливо при використанні карт);

Коли мова заходить про зручний інтерфейс під мобільні пристрої і одночасну швидкодію, то фаворитом відразу стають *мобільні додатки*, а не сайти. Якщо ж говорити про загальну зручність (не про інтерфейс користувача), то і тут мобільні додатки обходять сайти. Рядовому користувача значно простіше розблокувати телефон і відкрити додаток, ніж, знову ж таки, розблокувати телефон, запустити браузер, пошукати посилання на сайт, перейти на сам сайт.

Ще однією особливістю мобільних додатків є можливість надання офлайн доступу до даних. Так, навіть при відсутності доступу в інтернет користувач зможе переглядати власну статистику.

### 2.3 Поняття технологічного стека. Підбір технологій

Успішність будь-якої розробки програмного забезпечення напряму залежить від правильного підбору технологічного стека. Технологічним стеком називають певний набір технологій. Але що таке сама технологія? Технологія це збірне поняття, яке присвоюється будь-якому інструменту, яке було використане під час розробки.

Тому вживаючи термін “технології”, мають на увазі одне із наступних:

- а) мови програмування (Dart, Java, C# і тд.);

- b) фреймворки (Flutter, WPF і тд.);
- c) різноманітні бібліотеки (Bloc, Provider, Telerik і тд.);
- d) середовища розробки (Microsoft Visual Studio Code, JetBrains і тд.);
- e) інші програмні засоби (програми для проектування архітектури, побудови блок-схем і тд.);

Кожне програмне забезпечення, що працює з локальною чи інтернет мережами складається з двох частин (рис. 2.1) – клієнтської(front-end) та back-end. Для кожної з цих частин технологічний стек підбирається окремо.



Рис. 2.1. Частина програмного забезпечення та їхній зв'язок

Back-end називають внутрішню стороною сервісу, що працює окремо і незалежно від самого додатку. Її ціллю є своєчасне реагування на запити отримання чи відправки даних від сторони клієнта. Крім цього, у список задач серверної сторони також входить слідкування за тим, щоб ці дані завжди залишалися у безпеці та були доступними лише тим, хто має на це право.

Користувач самостійно не має можливості надсилати і отримувати дані – в цьому йому допомагає front-end завдяки користувацькому інтерфейсу. UI додатку називають набір графічних елементів керування і відображення інформації. До набору можуть входити кнопки, поля вводу, зображення та інше. Мобільні додатки, веб-сайти – це, в більшості випадків, і є клієнтською стороною.

Оскільки наш мобільний додаток повинен зберігати дані користувачів віддалено, виникає необхідність в першу чергу підібрати набір технологій для back-end.

Як було сказано вже раніше – back-end є окремою частиною додатку. Це означає необхідність у розробці з нуля окремого програмного забезпечення, а отже – додаткові затрати часу і фінансових ресурсів. Ці ресурси мають йти не тільки у найм розробника, а й на оренду серверної частини. У програмному забезпеченні малого масштабу, що має невеличкий бюджет, тут починаються великі проблеми. Але так було до появи BaaS (Back-end as a Service). Звісно BaaS не є повноцінною заміною вручну написаної back-end частини, проте є одним із найкращих рішень для невеличких додатків.

Розглянемо цю тему через те, що мобільний застосунок для спільних поїздки на велосипеді не має широкого набору функцій (пригадаємо що загалом їх 3), фінансових чи часових можливостей, і вибір BaaS є цілком і повністю логічним кроком у формуванні технологічного стеку.

Так ось, замість того, щоб писати back-end з нуля, придумали BaaS. BaaS мають вбудований готовий набір рішень по back-end частині, доступ до яких надається через API (інтерфейс для обміну даними) та SDK (набір готових інструментів для розробника). Крім готових рішень, постачальники цих послуг можуть надавати й інші функції, наприклад:

- a) вбудовані бази даних;
- b) спрощення масштабованості;
- c) аналітика даних;
- d) інструменти для аутентифікації;

BaaS має і мінуси, до яких відносять платність (але вона ніщо, в порівнянні з тим, скільки фінансових ресурсів може уйти на розробку власної back-end частини), постійна прив'язаність до постачальника послуг, менший контроль за кодом та проблеми з реалізацією унікальної моделі додатку.



Сьогодні на ринку є багато компаній, які надають послуги BaaS, найбільш популярними та надійними платформами є Firebase та AWS. Вибір між цими постачальниками буде *на користь першого*. Причиною є краща документація, широкий набір функцій, доступність по ціновому питанню (безкоштовність, хоча й з обмеженим функціоналом) та зручністю у подальшому використанні в кодї (наявність бібліотек і тд.). З усього наявного функціоналу, нам стане у нагоді лише наступне:

1. *Firebase Authentication* – для авторизації користувачів.
2. *Firebase Storage* – зберігання файлових даних авторизованих користувачів.
3. *Firebase Realtime Database* – NoSQL база даних, має низьку затримку в отриманні даних та дозволяє робити часті запити оновлення даних.
4. *Firestore* – NoSQL база даних, більш структурована, ніж *Firebase Realtime Database*, що є однією із причин можливості до автоматичної масштабованості та гнучкості запитів.

Отже, зі стеком технологій для back-end частинини ми визначились. Час перейти до підбору технологій для front-end.

Спершу виберемо на якій саме мові програмування та фреймворку буде написаний додаток. Ця галузь доволі різноманітна, але найбільш використовуваними є Flutter (мова програмування – Dart), React Native (мова програмування – JavaScript) та Xamarin(мова програмування – C#), Kotlin. Беручи до уваги знання та досвід роботи з Flutter у виконавця – *зупинимось на ньому*.

Flutter є фреймворком мови програмування Dart, що дозволяє створювати крос-платформні додатки (під комп'ютер, під мобільні пристрої і навіть під годинники). Відносно молодий фреймворк, як і мова програмування, були створені зусиллями компанії Google і був націлений замінити JavaScript при розробці сайтів. Як можна вже зрозуміти – широкого використання у вебї Flutter не набув, проте у сфері розробки мобільних додатків його очікував успіх. Крок за кроком, цей фреймворк, а якщо бути точніше – мова програмування Dart, за підсумками 2022-

									Аркуш
									27
Зм.	Аркуш	№ докум.	Підпис	Дата					

го зайняла 3 місце (рис. 2.2) в рейтингу використання мов програмування у мобільній розробці.

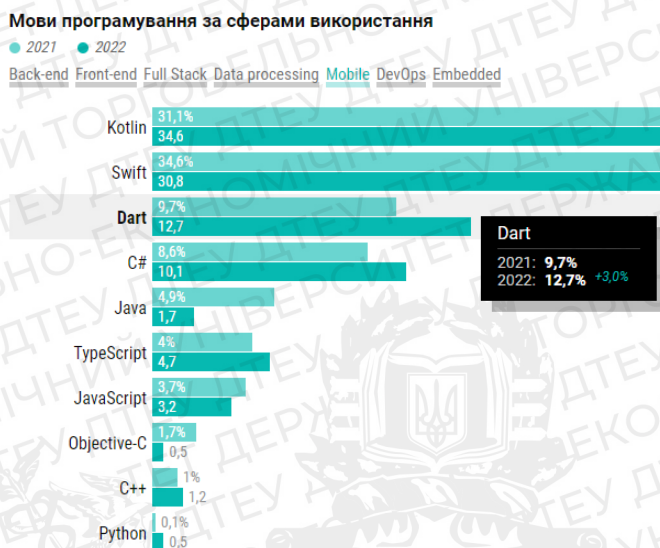


Рис. 2.2. Рейтинг використання мов програмування у мобільній розробці за 2021 та 2022-гі роки

Причини такого результату криються у багатьох факторах. Першим є продуктивність, адже маючи власний графічний движок, що зветься Skia, Flutter здатний відмальовувати елементи інтерфейсу використовуючи графічний процесор у пристрої. За рахунок цього компіляція відбувається значно швидше.

У стек технологій мобільного додатку добавим також декілька головних бібліотек:

1. *Provider*, який, забігаючи наперед, є менеджером станів. Мова про нього буде йтися у розділі з розробкою.
2. *Firestore Core* – можливість використання Firebase API.
3. *Firestore Auth* – набір функцій для авторизації.

Ще однією технологією буде *Figma*. У порівнянні з іншими, вона не використовується у процесі розробки, але відіграє головну роль у процесі розроблення графічного представлення інтерфейсу користувача.

Завершимо процес визначення технологічного стеку вибором IDE (інтегроване середовище розробки). У ролі IDE буде виступати *Microsoft Visual Studio Code*, який завдяки своїй простоті та зручності став гарним рішенням для розробки багатьох програм.

## 2.4 Аналіз сутностей. Концептуальна, логічна та фізична моделі бази даних

Аналіз сутностей та об'єктів є важливим під час процесу планування, оскільки допоможе правильно підібрати тип та саму базу даних, зробить процес розробки мобільного додатку та роботи з базою даних значно простішим.

Поняття “сутність” можна визначити як чітко структурований набір даних. Зв'язки між сутностями на високому рівні абстракції графічно утворюють концептуально модель. Конкретно у нашому випадку, під час проектування концептуальної та інших моделей, важливо врахувати один нюанс – використання різних сервісів (Firestore, Firebase Authentication, Firebase Realtime Database) для зберігання інформації. Тобто, враховуючи особливості цих баз даних, одна сутність може бути наприклад занесена у Firestore, інша – у Realtime Database і тд.

Відповідно до вимог, наш додаток потенціально працюватиме з наступними сутностями (рис. 2.3):

- а) користувач – базова інформація користувача. Зберігатиметься в Firebase Authentication. Цікавим є те, що використання Firebase Authentication надає не тільки API та SDK для авторизації, а й можливість зберігання суворо визначеною сервісом основної інформації про користувачів (такої як електронна пошта чи пароль).
- б) дані користувача – детальні дані користувачів. Поля розберемо під час створення логічної моделі. Зберігатиметься у Firestore, оскільки оновлення даних не буде проводитися часто (пригадаємо, що Firebase Realtime Database використовується для збереження даних, що часто оновлюються).

- с) доступні для спільної поїздки користувачі. Основними даними, забігаючи наперед, тут будуть координати локації місцезнаходження. На перший погляд, логічно було б занести цей показник у попередню визначену в цьому списку сутність. Проте, дані про локацію мають постійно оновлюватися щоб показувати актуальне розміщення користувача. Тому тут знадобиться Firebase Realtime Database.
- д) завершені поїздки (статистичні показники). Використовуватиметься Firestore.
- е) етап поїздки. Також зберігатиметься у Firestore.
- ф) поїздка. Особливістю цієї сутності є її тимчасовість та постійність у зміні стану (етапів), що створює умови до втілення цієї сутності у Firebase Realtime Database.

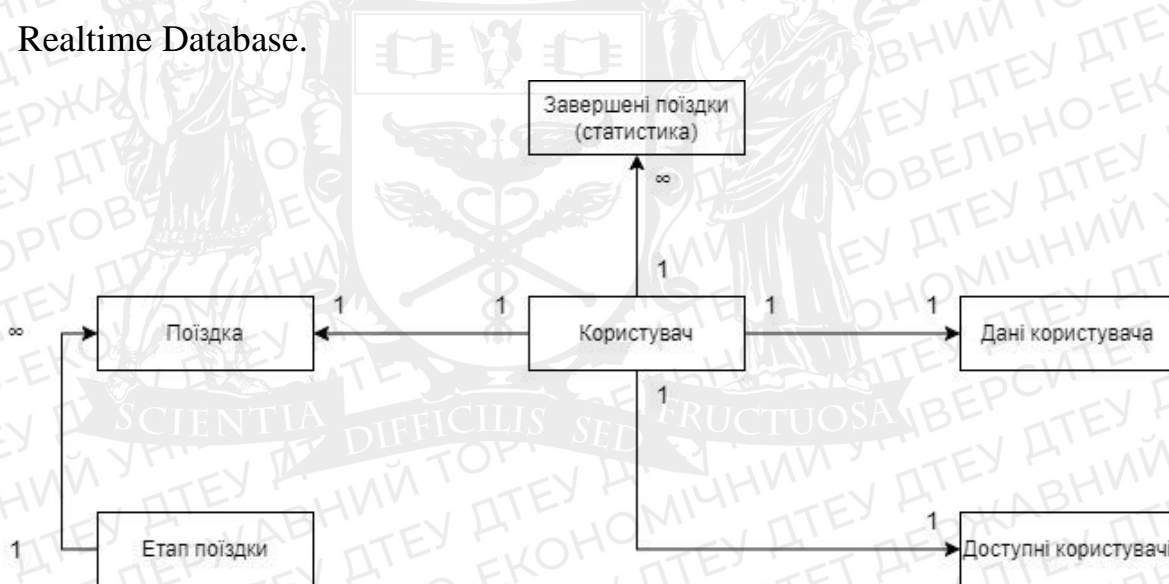


Рис. 2.3. Концептуальна модель зв'язків в базі даних між основними сутностями

Розглянемо тепер побудову логічної моделі з визначенням полів для головних сутностей (рис. 2.4):

а) користувачі (Users):

- 1) **UID** - унікальний ідентифікатор користувача, що допоможе у прив'язуванні таких даних як поїздки, особиста статистика.
- 2) **Nickname** – нікнейм користувача, використовуватиметься для ідентифікації самого користувача в додатку.

Зм.	Аркуш	№ докум.	Підпис	Дата
-----	-------	----------	--------	------

- 3) **Email** – буде необхідний для авторизації в аккаунт.
- 4) **Password** – пароль користувача.
- 5) **CreatedDate** – дата створення аккаунту.

**b) поїздки(Trips):**

- 1) **TID** – ключ-ідентифікатор поїздки.
- 2) **Participants** – учасники.
- 3) **SID** – зовнішній ключ, який посилається на конкретний етап таблиці TripStages.
- 4) **EndPoint** – кінцева точка, фініш.

**c) завершені поїздки:**

- 1) **CTID** - ключ-ідентифікатор завершеної поїздки.
- 2) **Title** – назва поїздки, що вводиться після її завершення.
- 3) **CreatedById** – UID користувача, що зберіг поїздку.
- 4) **OtherParticipants** – інші учасники
- 5) **StartedAt** – час старту.
- 6) **FinishedAt** – час фінішу.
- 7) **Duration** – тривалість поїздки.
- 8) **MaxSpeed** – максимальна швидкість користувача, що була досягнута під час поїздки.
- 9) **AverageSpeed** – середня швидкість користувача.
- 10) **Path** – пройдений користувачем шлях.

**d) детальна інформація користувачів:**

- 1) **UID** – зовнішній ключ, який посилається на користувача.
- 2) **AvatarURL** – посилання на зображення аватару, що зберігається у Firebase Storage.
- 3) **TotalAverageSpeed** – загальна середня швидкість користувача.

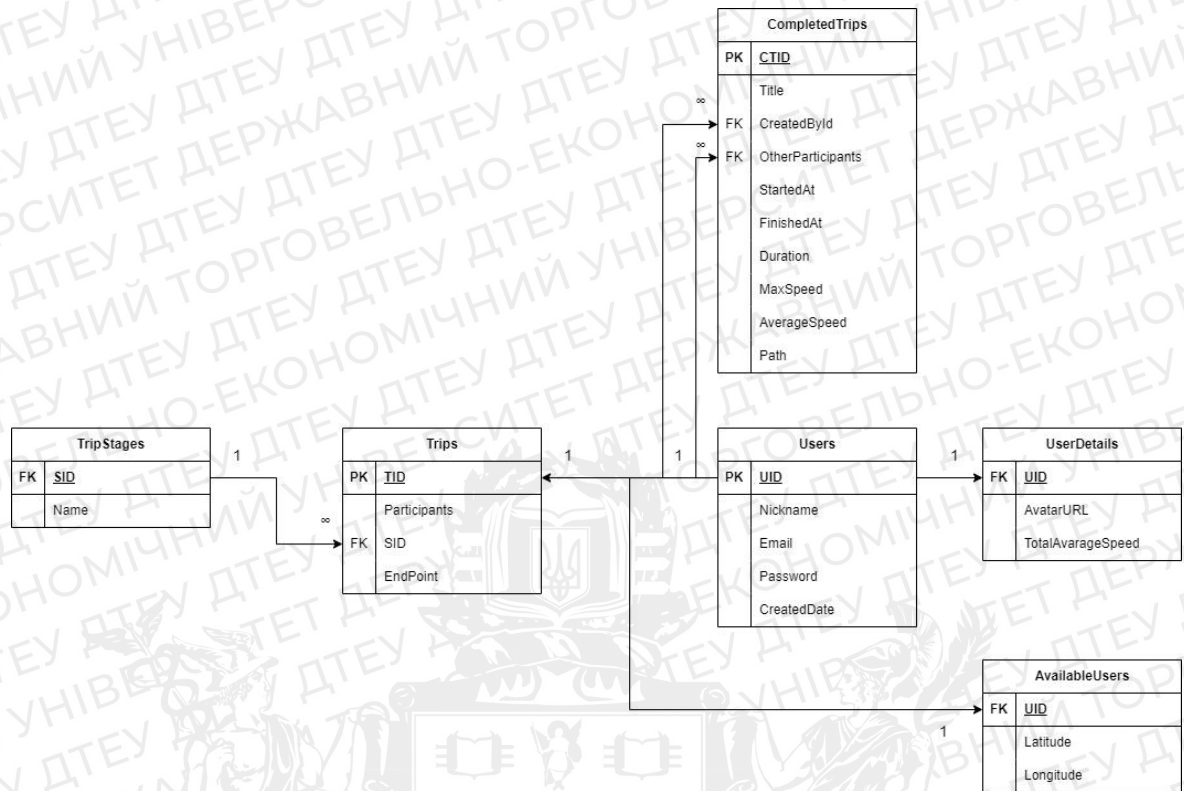


Рис. 2.4. Логічна модель бази даних

Останнім етапом проектування бази даних є розробка фізичної моделі (рис. 2.5). І хоча в нас база даних не є реляційною, але все ж ми можемо припустити який тип даних матиме те чи інше поле.

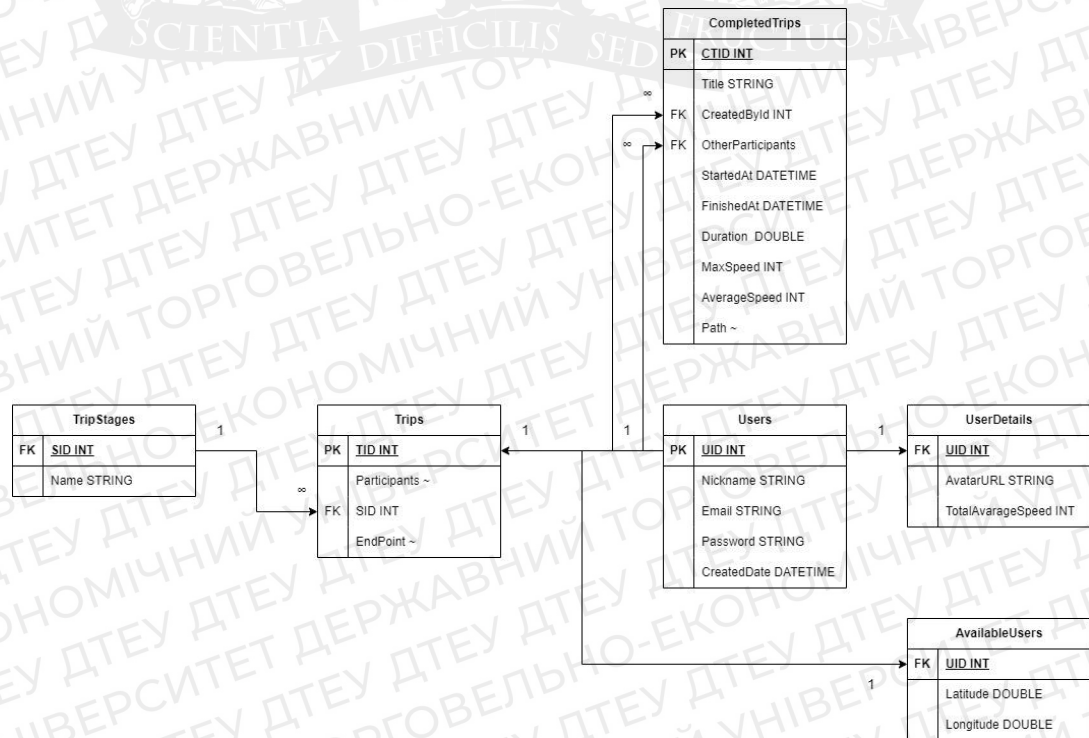


Рис. 2.5. Фізична модель бази даних

Такі ж поля, як Participants(Trips), EndPoint(Trips), Path(CompletedTrips) та OtherParticipants – не мають визначеного типу. Причина криється у комплексності структури цих даних.

У Participants суворо не обмежується кількість учасників поїздки, тому це поле може мати як два, так і більше вкладених структур, що складаються з UID, CurrentSpeed(швидкість користувача на даний момент), Location (яка у свою чергу поділяється на Longitude та Latitude).

EndPoint та Path – набір точок на карті, які, подібно до локації користувача, визначаються завдяки Longitude та Latitude. Причому EndPoint у таблиці Trips складатиметься лише тільки з однієї точки, в той час як Path з таблиці CompletedTrips вмщуватиме цілий набір точок. Завдяки цьому набору потім можна буде відтворити пройдений учасником шлях.

## 2.5 Висновки до другого розділу

У результаті роботи над другим розділом були оглянуті етапи поділу життєвий цикл розробки програмного забезпечення: дослідження, планування, розробка й випуск. Чинником, що став в основі поділу, є поступова зміна комплексності програм, їх архітектур та функціонального набору в залежності від вимог бізнесу.

Складено список функціональних та нефункціональних вимог до додатку. На основі факторів продуктивності та зручності прийнято рішення зупинитися на розробці програмного додатку саме для мобільної платформи.

В якості головних технологій для розробки було обрано:

- a) Dart та його фреймворк Flutter для створення мобільного додатку;
- b) Firebase у ролі back-end з такими сервісами, як Firebase Authentication, Firebase Storage, Firebase Realtime Database, Firestore;
- c) бібліотеки Provider, Firebase Core, Firebase Auth, Cloud Firestore;
- d) інструмент створення графічних інтерфейсів Figma;

## РОЗДІЛ 3

### РОЗРОБКА ПРОГРАМНОГО ДОДАТКУ

#### 3.1 Створення проекту у Firebase

Наявний набір необхідних даних у вигляді вимог та технологічного стеку, визначених раніше, дає можливість перейти до етапу розробки програмного забезпечення. Відправною точкою стане створення та налаштування back-end частини, а саме – Firebase.

Перш за все, нам стане у нагоді Google акаунт. Без нього створити проект у Firebase не вийде. Як тільки розробник зареєструє такий акаунт – він зможе перейти до консолі (рис. 3.1).



Рис. 3.1. Консоль Firebase

Далі залишається лише натиснути кнопку “Add project”, після чого потрібно ввести ім’я проекту. В якості ім’я обирають зазвичай назву самого додатку. В разі чого – назва може бути змінена пізніше в налаштуваннях після створення проекту.

Увівши назву, нам запропонують підключити Google Analytics. На ранній стадії розробки вона не матиме важливої ролі, тому відключаємо.

					<i>ДТЕУ 121– 02-23.МР</i>			
						<i>Стадія</i>	<i>Аркуш</i>	<i>Аркушів</i>
<i>Зм</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>	<i>Програмний додаток для організації велодозвілля</i>	<i>РЗ</i>	<i>34</i>	<i>48</i>
<i>Зав. каф.</i>		<i>Криворучко О.В.</i>		<i>06.09.23</i>				
<i>Керівник</i>		<i>Котенко Н.О.</i>		<i>06.09.23</i>				
<i>Гарант</i>		<i>Котенко Н.О.</i>		<i>06.09.23</i>				
<i>Розроб</i>		<i>Чернюк В.А</i>		<i>06.09.23</i>				
<i>Розробка програмного додатку</i>						<i>Факультет інформаційних технологій, 2м курс, 2 група</i>		



Як тільки проект створиться – ми потрапимо на його адмін-панель (рис. 3.2).

Завдяки панелі ми матимемо здатність працювати з цілою back-end частиною.

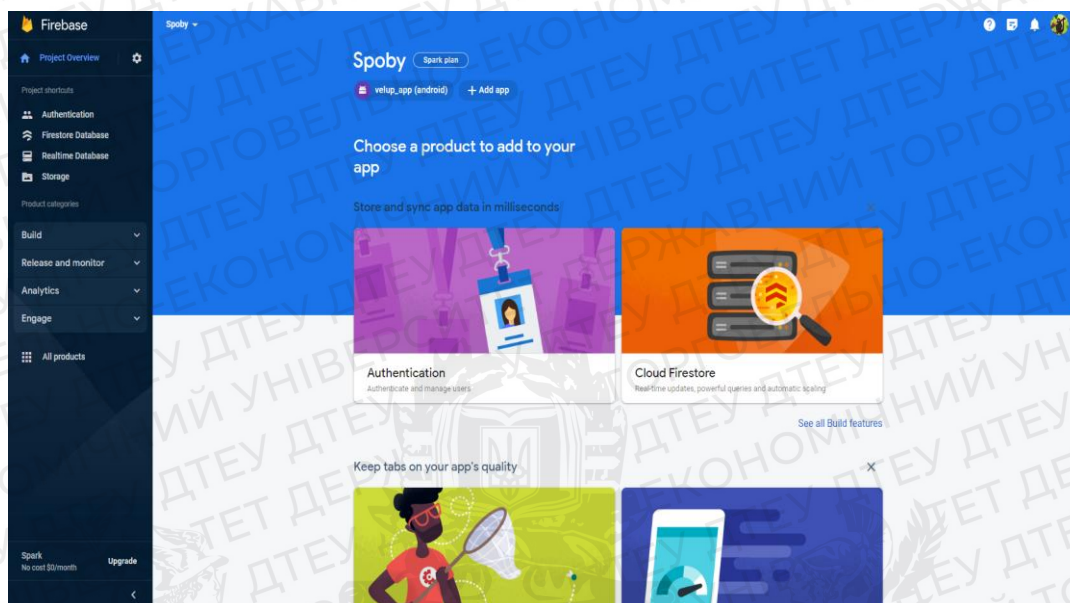


Рис. 3.2. Адмін-панель мобільного додатку у Firebase

Сервіси у Firebase підключаються окремо. Тобто проекти не матимуть зайвий функціонал. Набір функцій додатку для велопрогулянок, як це вже було визначено в попередньому розділі, складатиметься з Authentication, Storage, Realtime Database та Firestore. Під час та після додавання кожен сервіс можна налаштувати окремо. Наприклад для автентифікації відразу вказується метод входу, у нашому випадку – це введення користувачем власної електронної пошти та паролю.

Після додавання усіх сервісів матимемо за допомогою сайту та спеціального файлу налаштування підключити наш Android додаток у проект Firebase. Це робиться за допомогою додавання конфігураційних файлів Firebase у корінь Flutter проекту. Поєднавши додаток з проектом, залишається лише додати в залежності мобільного додатку Firebase SDK, який і дасть можливість використовувати функціонал Firebase у мобільному додатку.

На цьому з базовим налаштуванням back-end покінчено. Перейдемо до основної частини цього розділу – розробки клієнтської частини.

Зм.	Аркуш	№ докум.	Підпис	Дата

### 3.2 Архітектура програмного забезпечення. State managent

Архітектура розробки програмного забезпечення – це розроблена структура додатка, яка є визначенням взаємодії компонентів інтерфейсу із внутрішніми процесами програми. Простіше кажучи, це своєрідний підхід, який визначає які функції за що відповідають та як вони взаємодіють між собою.

Раніше архітектура не мала вагомого значення у процесі розробки програмного продукту. Причиною цьому була простота коду і функціоналу програм. Адже, маючи примітивне ПЗ, буквально не було необхідності у зайвому його ускладненні різноманітними архітектурними рішеннями. Тому на архітектурі просто не зосереджувалися і таким чином економили час.

Поступово функціонал програм розростався і еволюціонував у більш складні алгоритми. Самі програми почали ставати негнучкими і некерованими, а перевага у часі стала менш помітною. Кожне внесення змін робилося з великими зусиллями і затратами часу. Стало очевидним, що ситуацію потрібно змінювати, і як наслідок – появився напрямок по проектуванню архітектури. Його ідеєю став поділ програми на незалежні між собою частини коду, які будуть виконувати лише суворо визначену їм роль. З часом підходи отримали більш чітку структуру і стали називатися паттернами архітектури ПЗ.

Традиційно базовим паттерном називають MVC. З нього, розвиваючись, виникли й інші підходи до архітектури – MVVM, MVP та інші. Назва “MVC” розшифровується як Model View Controller. Ці слова описують фундаментальні компоненти поділу програмного забезпечення. Кожен компонент, відповідно до загальної ідеї архітектури, виконує власну роль.

У мобільній розробці частіше за все використовуються MVVM підхід. Для створення мобільного додатку на flutter класичним також вважають MVVM. Він і стане в основі архітектури нашого додатку для велопрогулянок. Розглянемо цей паттерн більш детально.

Паттерн MVVM поділяється на 3 основні компоненти – Model, View, ViewModel. Зі складових можна побачити подібність до MVC, проте порядок взаємодії між компонентами зовсім інший (рис. 3.3).

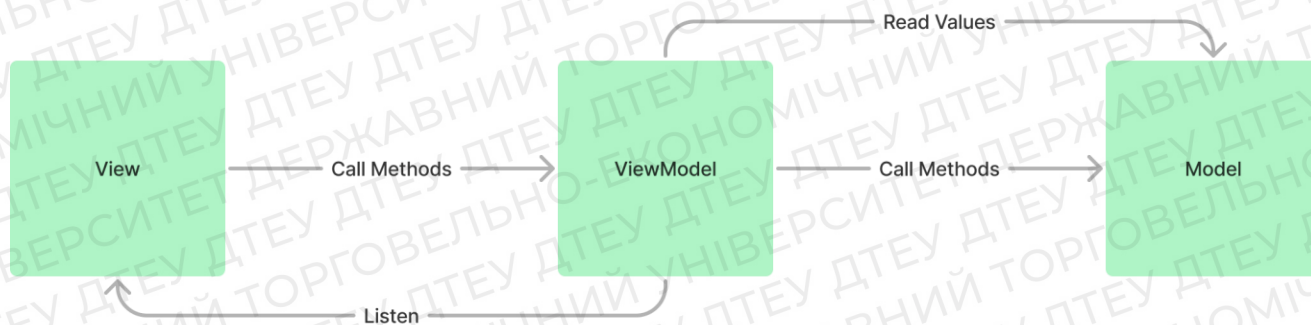


Рис. 3.3. Компоненти MVVM

У Flutter інтерфейс користувача представлений набором віджетів, які разом утворюють дерево вкладених віджетів. Віджети можуть мати як власне візуальне представлення (віджет-кнопка, віджет-текст), так і передбачають відсутність графічного елемента на екрані. Віджети другого типу зазвичай або надають функціонал, або надають доступ до даних іншим вкладеним віджетам далі по дереву.

Представлення (View) пов'язане з ViewModel тим, що викликає у ньому методи. Наприклад користувач натиснув кнопку, і, замість того щоб метод викликався у View, він викликається у ViewModel. У свою чергу, ViewModel здатний використовувати методи від моделі.

Задачею моделі (Model) є робота з даними, їх отримання та підготовка. Інформацію про оновлення даних отримує ViewModel, який потім передає їх у компонент View.

Вище було подана базова інформація щодо архітектурного паттерну MVVM. Розподіл на такі функціональні обов'язки дає розробнику значні переваги, серед яких:

- a) легка супроводжуваність;
- b) зручність тестування;
- c) масштабованість;

Крім самої архітектури, у мобільних додатках, зокрема що були написані на Flutter, існує така особливість як "state management". Що взагалі називають "state" і навіщо ним керувати?

Уся причина криється у віджетах, а точніше – їх поділі на Stateless та Stateful. Станом (“state”) називають набір даних для представлення у певний момент існування віджета. Stateless тип вказує на відсутність “пам’яті”, стану у віджета, що приводить до його незмінності у процесі роботи програми (але існують винятки). Зовсім протилежним до цього є Stateful, який може змінюватися у процесі існування в залежності від стану що має.

Наприклад наш головний віджет має стан, нехай це буде якесь число. Це число має відображатися і у вкладеному віджеті. Якщо рівень вкладеності незначний, то проблем з передачею значення не буде. В іншому випадку доведеться поступово “спускати” значення, передаючи в якості аргумента аж до потрібного віджета. Тут і починаються проблеми, адже код стає менш читабельним, а при зміні значення відбудеться послідовний виклик ланцюжку методів і кожний віджет, що передає значення – буде перебудований. Цей процес створить помітне падіння у продуктивності мобільного додатку. Для вирішення цієї проблеми і були створенні такі бібліотеки як BloC, Provider чи Redux. У нашому додатку, через простоту та гнучкість (яка стане у нагоді під час роботи з MVVM), ми будемо використовувати саме Provider.

### 3.3 Реалізація функціоналу авторизації

Авторизація є однією із визначених вимог цього додатку. Схема по авторизації має наступний вигляд (рис. 3.4).

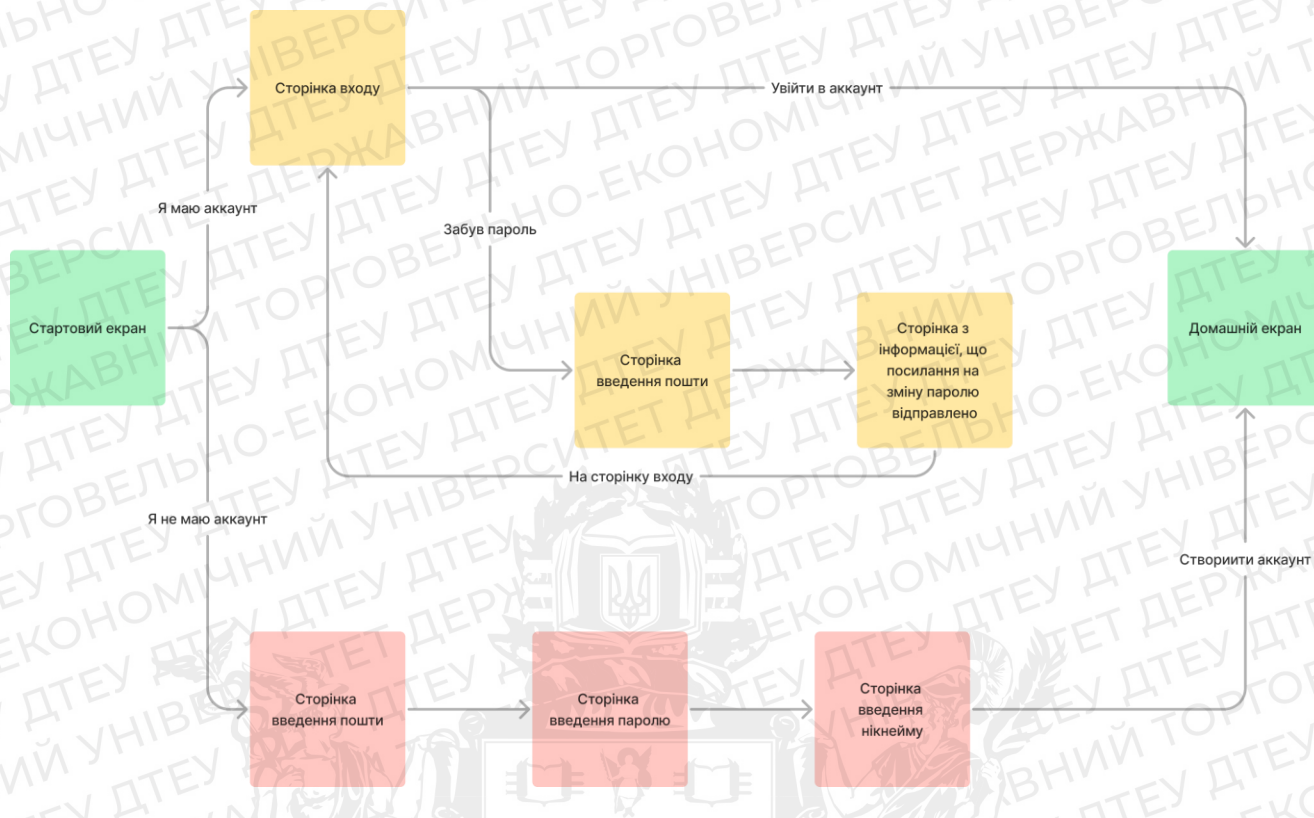


Рис. 3.4. Схема авторизації

Щоб надати незареєстрованим користувачам можливість створити аккаунт, а зареєстрованим дозволити увійти у нього – реалізуємо “початковий” екран (рис. 3.5). Він стане відправною точкою на нашому шляху авторизації.

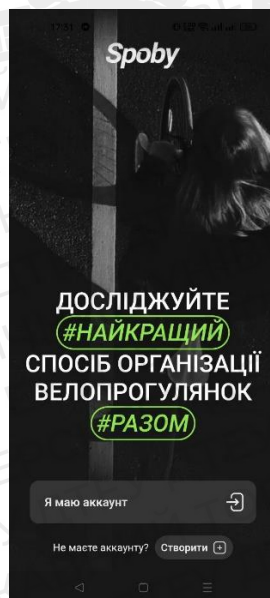


Рис. 3.5. Стартовий екран авторизації

Зм.	Аркуш	№ докум.	Підпис	Дата

Якщо користувач натисне кнопку “Я маю аккаунт” на стартовій сторінці, то сторінка зміниться на вхід, а для входу потрібно буде ввести електронну пошту та пароль (рис. 3.6). У разі правильності введених даних Вас перекине на домашню сторінку.

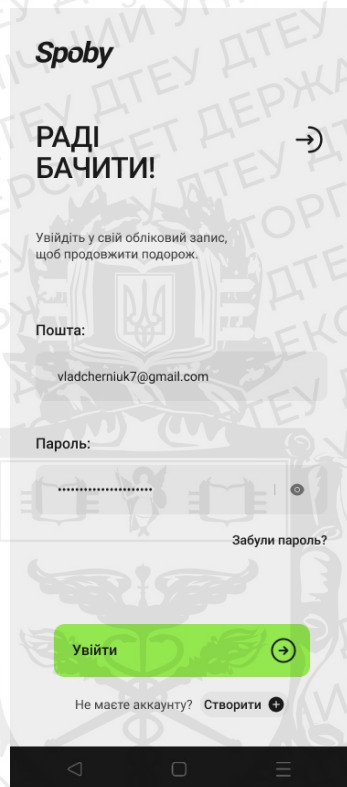


Рис. 3.6. Сторінка входу в аккаунт

Розглянемо цей функціонал з точки зору коду. Відповідно до паттерну MVVM ми маємо View (Додаток А), що представлена деревом вкладених віджетів, ViewModel (рис. 3.7) та самої моделі. Важливо, що ViewModel впроваджується у представлення за допомогою Provider (Додаток В) та використовує методи моделі (наприклад signIn). ViewModel має також власний стан (Додаток Г). Цей стан, в першу чергу, покликаний для збереження введених користувачем даних (пароль та пошту).

Модель авторизації (Додаток Б) представлена набором полів та методів з вибраної раніше бібліотеки Firebase Auth. Застосування ця модель знайшла не тільки у ViewModel сторінки входу, а й на усіх сторінках, що відносяться до авторизації.

```

class LoginViewModel extends ChangeNotifier {
  final _authenticationService = AuthenticationService();
  final _state = _LoginViewModelState();
  final BuildContext _buildContext;
  _LoginViewModelState get state => _state;

  LoginViewModel(this._buildContext);

  void changeEmail(String value) {
    if (_state.email == value) return;
    _state.email = value;
    notifyListeners();
  }

  void changePassword(String value) {
    if (_state.password == value) return;
    _state.password = value;
    notifyListeners();
  }

  Future<void> onAuthButtonPressed() async {
    final email = _state.email;
    final password = _state.password;

    if (email.isEmpty || password.isEmpty) return;

    _state.isAuthInProgress = true;
    notifyListeners();

    try {
      await _authenticationService.signIn(email: email, password: password);
      _state.isAuthInProgress = false;
      Navigator.of(_buildContext).pushNamedAndRemoveUntil(
        RouteKeys.authScreenLoader, (route) => false);
      notifyListeners();
    } on FirebaseAuthException catch (e) {
      CustomSnackBar.setError(_buildContext, e.code);
      _state.isAuthInProgress = false;
      notifyListeners();
    } catch (exception) {
      CustomSnackBar.setError(_buildContext, exception.toString());
      _state.isAuthInProgress = false;
      notifyListeners();
    }
  }
}

```

Рис. 3.7. ViewModel сторінки входу

Процес реєстрації відбувається поетапно (рис. 3.8). Кожний етап характеризується зміною екранів. Спочатку пропонується увести пошту, пароль, а вкінці – нікнейм. Завдяки цьому процес реєстрації є більш “легким” та інтуїтивно зрозумілим для користувача. ViewModel є загальною для цих сторінок (Додаток Д), а її станом являється сам етап (Додаток Е).

Зм.	Аркуш	№ докум.	Підпис	Дата

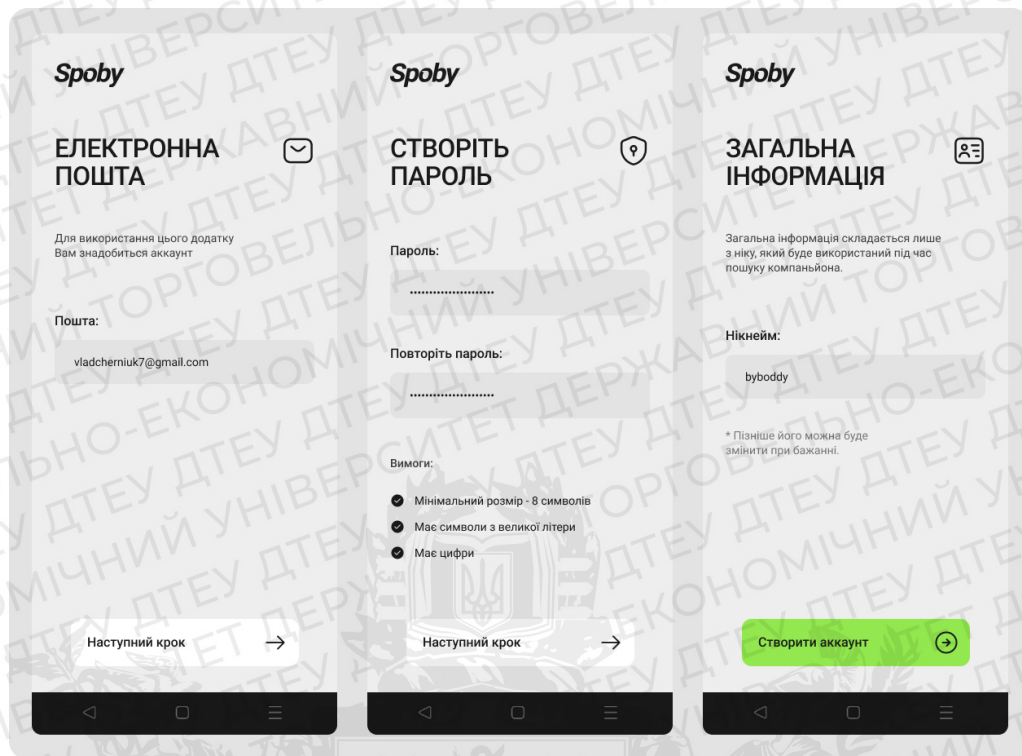


Рис. 3.8. Екрани реєстрації

Окрім створення аккаунту та входу була додана також функція по скиданню паролю (Додаток Ж), адже нерідко зустрічаються випадки, коли власники аккаунтів забувають власний пароль. Відсутність цієї можливості створить лише негативний досвід користуванням нашим мобільним додатком. Для скидання паролю потрібно натиснути кнопку “Забули пароль?”. Екран зміниться і додаток попросить увести пошту, до якої прив’язаний аккаунт. Після введення прийде електронний лист з посиланням на зміну паролю. Змінивши пароль, користувач знову зможе увійти у власний аккаунт.

### 3.4 Модуль поїздок та статистики

Після авторизації користувач потрапляє на власну домашню сторінку (рис 3.9). На ній, відповідно до вимог, знаходиться наступна інформація:

- аватар та нікнейм;
- статистична інформація про середню швидкість користувача за увесь час;
- останні завершені поїздки;



Тут же можна перейти до налаштування самого додатку(Додаток И) та редагування інформації аккаунту (Додаток К) з кнопкою для виходу з аккаунту.

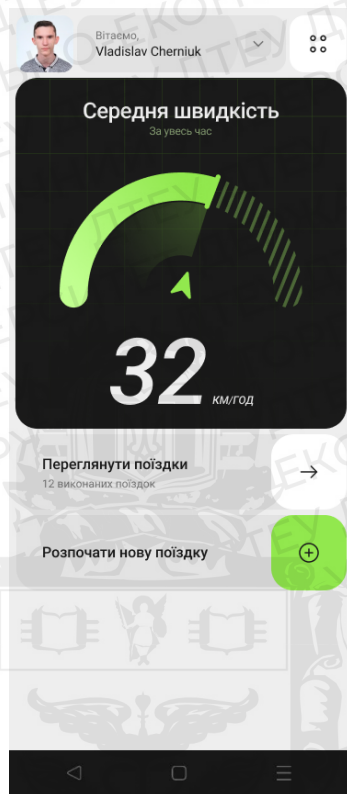


Рис. 3.9. Домашня сторінка

Щоб розпочати поїздку – потрібно натиснути відповідну кнопку на домашній сторінці, після чого включити GPS на телефоні. Це дасть можливість показати Ваше місцезнаходження на карті і як наслідок – отримати запрошення від інших користувачів на спільну поїздку. Карта (Додаток Л) відображає у режимі реального часу Ваше місцезнаходження лише тоді, коли ви знаходитесь на цій сторінці. Навколо користувача на карті є спеціальна зона, потрапивши в яку, інші користувачі стануть для Вас помітними. Таким чином не дається можливість брати участь у спільній поїздки людям, що знаходяться далеко від Вас.

Метод (рис. 3.10), який відповідає за отримання від Firebase інформації про розміщення інших користувачів, знаходиться у ViewModel сторінки карти.

```

Future<void> showAllUsersNear() async {
  _getCurrentLocation();

  try {
    _state.otherUsers = await _tripService.getAllUsersNear(currentLocation);
    notifyListeners();
  } on TripException catch (e) {
    CustomSnackBar.setError(_buildContext, e.code);
    notifyListeners();
  } catch (expection) {
    CustomSnackBar.setError(_buildContext, expection.toString());
    notifyListeners();
  }
}

```

Рис. 3.10. Метод отримання локацій користувачів поряд

А для того, щоб могли знайти нашу локацію, викликається інший метод – `getCurrentLocation` (Рис.3.11).

```

void _getCurrentLocation() async {
  LocationPermission permission;

  permission = await Geolocator.checkPermission();
  if (permission == LocationPermission.denied) {
    CustomSnackBar.setError(_buildContext,
      "Доступ до Вашої локації закритий. Включіть будь ласка GPS в налаштуваннях");
  } else {
    var position = await Geolocator.getCurrentPosition(
      desiredAccuracy: LocationAccuracy.high);
    currentLocation = Location(position.latitude, position.longitude);
    _tripService.sendCurrentPosition(currentLocation);
    notifyListeners();
  }
}

```

Рис. 3.11. Метод отримання локації

Перш за все, для старту поїздки потрібно відправити запрошення (Додаток У) іншому користувачеві, який розмішений поруч. Приймати це запрошення чи відхиляти – рішення індивідуальне і залежить лише від бажання людини.

У разі прийняття запрошення, той, хто запросив – має обрати точку фінішу (Додаток Ф), тобто місце, де поїздка завершиться, а людина яку запросили – підтвердити її. Якщо все пройшло вдало, то поїздка починається, а карта автоматично у режимі реального часу буде навігацію до фінішу (Рис. 3.12).

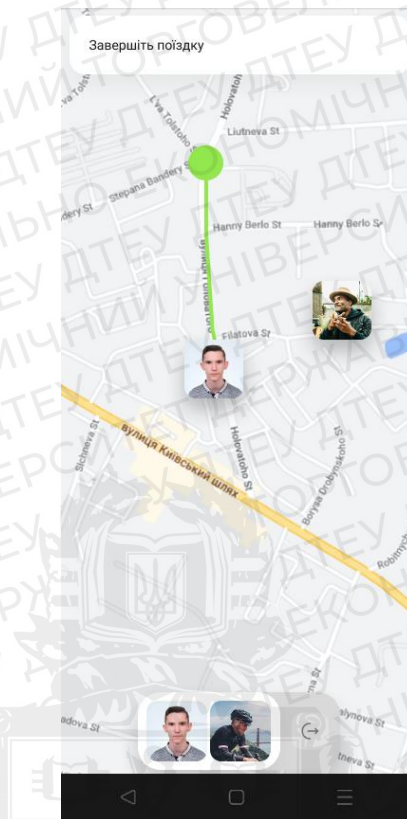


Рис. 3.12. Процес поїздки

Після завершення поїздки, прибувши до фінішу, показується екран зі статистикою пройденого маршруту із можливістю збереження даних (рис. 3.13).

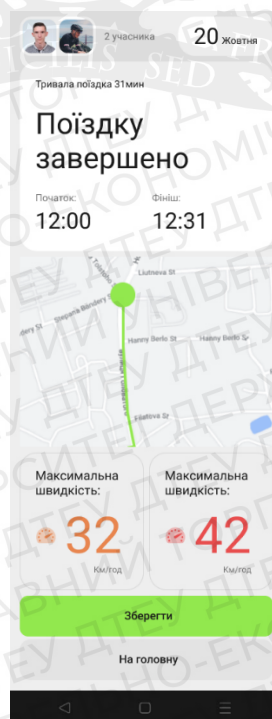


Рис. 3.13. Статистика користувача за пройдений маршрут

Зм.	Аркуш	№ докум.	Підпис	Дата



## ВИСНОВКИ ТА ПРОПОЗИЦІЇ

Проблематика ролі спорту та ведення активного способу життя ніколи не втратить актуальності. Людина постійно потребує як фізичний розвиток, так і психологічну стійкість. Одним із джерел цього є саме ведення активний способу життя, зокрема у цій роботі мова йде про регулярні велопогулянки.

Поїздки на двухколісному не тільки позитивно впливають на людину, а й приносять користь навколишньому середовищу. Проте, з часом велосипедисти можуть зустрітися з певними нюансами, які приводять до отримання негативного досвіду під час поїздок. До них можна віднести:

- a) проблема з пошуком компаньйона для спільної поїздки;
- b) необхідність в аналізі пройденого маршруту;

У процесі виконання випускної кваліфікаційної роботи було розроблено мобільний додаток, призначення якого і заключається у вирішенні вищезгаданих проблем. В якості ролі мови програмування виступав Dart, а в ролі фреймворка – Flutter. Обидві технології чудово справилися з поставленим завданням.

Розглянуто на практиці основні етапи життєвого циклу розробки програмного забезпечення. Під час етапу проектування визначили базові вимоги до мобільного додатку, на основі вимог підібрали технологічний стек та спроектували базу даних. Етап розробки складався зі створення та використання проекту у Firebase, Flutter, а також огляду та втілення архітектури MVVM під час реалізації функціоналу.

					<i>ДТЕУ 121– 02-23.МР</i>			
						<i>Стадія</i>	<i>Аркуш</i>	<i>Аркушів</i>
<i>Зм</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>	<i>Програмний додаток для організації велодозвілля</i>	<i>ВП</i>	<i>47</i>	<i>48</i>
<i>Зав. каф.</i>		<i>Криворучко О.В.</i>		<i>01.11.23</i>				
<i>Керівник</i>		<i>Котенко Н.О.</i>		<i>01.11.23</i>				
<i>Гарант</i>		<i>Котенко Н.О.</i>		<i>01.11.23</i>				
<i>Розроб</i>		<i>Чернюк В.А</i>		<i>01.11.23</i>	<i>Висновки та пропозиції</i>	<i>Факультет інформаційних технологій, 2м курс, 2 група</i>		

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. 75% Notice A Boost To Their Mental Health After Cycling – Survey [Електронний ресурс]. – Режим доступу: <https://www.cycleplan.co.uk/cycle-savvy/cycling-mental-health-survey/> (дата звернення: 16.08.2023).
2. Про український проект BikeNow [Електронний ресурс]. – Режим доступу: <https://bikenow.com.ua/pro-nas/> (дата звернення: 16.08.2023).
3. Firebase Products [Електронний ресурс]. – Режим доступу: <https://firebase.google.com/products-build> (дата звернення: 23.10.2023).
4. Choose a Database: Cloud Firestore or Realtime Database [Електронний ресурс]. – Режим доступу: <https://firebase.google.com/docs/database/rtdb-vs-firestore> (дата звернення: 23.10.2023).
5. Документація Flutter [Електронний ресурс]. – Режим доступу: <https://flutter.dev/docs> (дата звернення: 24.10.2023).
6. Документація по Dart [Електронний ресурс]. – Режим доступу: <https://dart.dev/guides> (дата звернення: 24.10.2023).
7. Документація по Firebase [Електронний ресурс]. – Режим доступу: <https://firebase.google.com/docs/build> (дата звернення: 24.10.2023).
8. Документація бібліотеки Provider [Електронний ресурс]. – Режим доступу: <https://pub.dev/packages/provider> (дата звернення: 24.10.2023).
9. Get Started with Firebase Authentication on Flutter [Електронний ресурс]. – Режим доступу: <https://firebase.google.com/docs/auth/flutter/start> (дата звернення: 24.10.2023).

					<i>ДТЕУ 121–02-23.МР</i>		
					<i>Програмний додаток для організації велодозвілля</i>		
<i>Зм</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>	<i>Стадія</i>	<i>Аркуш</i>	<i>Аркушів</i>
<i>Зав. каф.</i>		<i>Криворучко О.В.</i>		06.09.23	<i>СВД</i>	48	48
<i>Керівник</i>		<i>Котенко Н.О.</i>		06.09.23			
<i>Гарант</i>		<i>Котенко Н.О.</i>		06.09.23			
<i>Розроб</i>		<i>Чернюк В.А.</i>		06.09.23			
					<i>Список використаних джерел</i>		
					Факультет інформаційних технологій, 2м курс, 2 група		

## ТЕХНІЧНЕ ЗАВДАННЯ

### 1. Загальні відомості:

- 1.1. Дане технічне завдання є основним документом, що визначене список вимог та порядок розробки мобільного додатку для велодозвілля.
- 1.2. Крайній термін готовності додатку – 24.11.2023.

### 2. Мета додатку:

- 2.1. Метою мобільного додатку для велодозвілля є допомога велосипедистам з пошуком компаньйона для спільної поїздки, а також можливість слідкувати за індивідуальними показниками після завершення маршруту.

### 3. Вимоги до додатку:

#### 3.1. Загальні:

- 3.1.1. Крім клієнтської частини (самого мобільного додатку), також необхідно розробити back-end частину з базою даних для збереження інформації користувачів.
- 3.1.2. Наявність локалізації (переводу інтерфейсу) – англійської та української.
- 3.1.3. Інформація, що відображається на екрані користувача, має бути завжди актуальною і синхронізованою з базою даних.

#### 3.2. Функціональні:

##### 3.2.1. Система аккаунтів:

- 3.2.1.1. Наявність можливості реєстрації та входу в облікові записи.
- 3.2.1.2. Редагування особистих даних (пошти, паролю і тд.) за умови, що користувач авторизований. Винятком є скидання паролю.

##### 3.2.2. Процес поїздок:

					<i>ДТЕУ 121–02-23.МР</i>			
					<i>Програмний додаток для організації велодозвілля</i>	<i>Стадія</i>	<i>Аркуш</i>	<i>Аркушів</i>
<i>Зм</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		<i>ТЗ</i>	49	48
<i>Зав. каф.</i>		<i>Криворучко О.В.</i>		06.09.23				
<i>Керівник</i>		<i>Котенко Н.О.</i>		06.09.23				
<i>Гарант</i>		<i>Котенко Н.О.</i>		06.09.23				
<i>Розроб</i>		<i>Чернюк В.А</i>		06.09.23	<i>Технічне завдання</i>	Факультет інформаційних технологій, 2м курс, 2 група		





#### **4. Вимоги до складу і параметрів технічних засобів:**

4.1. Цільова платформа для розробки – Android.

4.2. Список вимог до платформи:

4.2.1. Мінімальна сумісна версія – Android 6.0 (SDK 23+).

4.2.2. Вертикальна орієнтація:

4.2.2.1. Телефони – наявна.

4.2.2.2. Планшети – наявна.

4.2.3. Горизонтальна орієнтація:

4.2.3.1. Телефони – відсутня.

4.2.3.2. Планшети – відсутня.

#### **5. Стадії та етапи розробки:**

5.1. Етап дослідження:

5.1.1. Термін готовності – до 24.05.2023.

5.2. Етап планування:

5.2.1. Термін готовності – до 06.09.2023.

5.3. Етап розробки

5.3.1. Термін готовності – до 24.11.2023.

#### **6. Порядок контролю і приймання:**

6.1. У процесі та після завершення розробки мобільного додатку, функціонал має бути протестований на правильність роботи.

6.2. Види тестування для проведення:

6.2.1. Функціональне.

6.2.2. Usability.

Зм.	Аркуш	№ докум.	Підпис	Дата

ДТЕУ 121-02-23.MP

Аркуш

51

## ПРОГРАМА ТА МЕТОДИКА ТЕСТУВАННЯ

Тестування є невід'ємною частиною розробки програмного забезпечення, адже без нього просто неможливо визначити якість мобільних додатків. У разі постійного виникнення помилок чи інших незручностей, користувач не буде користуватися цим ПЗ, а просто знайде альтернативу. Уникнути цей сценарій або довести можливість його появи до мінімуму допоможе лише постійне тестування.

У мобільному додатку для велопогулянок було проведено функціональне та usability тестування. Під час функціонального вдалося перевірити наступне:

- a) коректність запуску програми (відображення Splash Screen);
- b) роботу додатку у фоновому режимі;
- c) вхід в аккаунт;
- d) реєстрацію;
- e) скидання паролю;
- f) відображення помилок, зокрема за допомогою снєкбарів;
- g) скролл чи свайп елементів;
- h) використання різних типів підключень;
- i) орієнтацію екрану;
- j) запуск та процес поїздки;

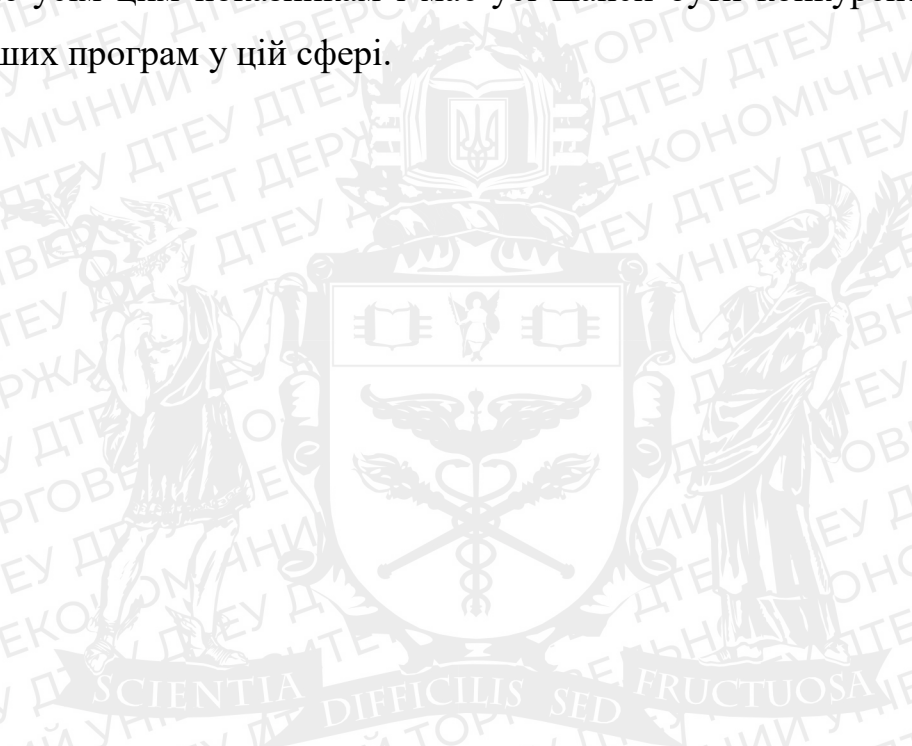
Usability тестування, що з англійської означає зручність використання, допомагає оцінити додаток з точки зору рівня "дружелюбності" до користувачів, тобто зручність та ефективність графічного інтерфейсу. У список перевіреного увійшло:

- a) зручність натиснення кнопок;
- b) читабельність тексту (розмір і тд) та його вирівнювання;
- c) наявність плейсхолдерів у полях вводу інформації;

					<i>ДТЕУ 121–02-23.МР</i>			
						<i>Стадія</i>	<i>Аркуш</i>	<i>Аркушів</i>
<i>Зм</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>	<i>Програмний додаток для організації велодозвілля</i>	<i>ПМТ</i>	52	48
<i>Зав. каф.</i>		<i>Криворучко О.В.</i>		06.09.23				
<i>Керівник</i>		<i>Котенко Н.О.</i>		06.09.23				
<i>Гарант</i>		<i>Котенко Н.О.</i>		06.09.23				
<i>Розроб</i>		<i>Чернюк В.А</i>		06.09.23	<i>Програма та методика тестування</i>	Факультет інформаційних технологій, 2м курс, 2 група		

- d) коректне відображення неактивних елементів (наприклад кнопок);
- e) роботу маршрутів між екранами (наприклад повернення назад через нажаття кнопки в інтерфейсі);
- f) відповідність до дизайну макетів у Figma;
- g) правильність переводу інтерфейсу на інші мови (англійська);

Результати тестування показали, що мобільний додаток у повній мірі відповідає усім цим показникам і має усі шанси бути конкурентоспроможним з поміж інших програм у цій сфері.



## ДОДАТКИ

Додаток А

Лістинг коду View частини сторінки входу в аккаунт

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

  @override
  Widget build(BuildContext context) {
    final model = context.read<LoginViewModel>();

    double screenHeight = context.mediaQuery.size.height;
    double distanceBetweenFields = screenHeight * 0.01;

    final authButtonState =
      context.select((LoginViewModel value) => value.state.authButtonState);

    final onPressed = authButtonState == ViewModelAuthButtonState.canBePressed
      ? model.onAuthButtonPressed
      : null;

    return AuthContainer(
      authStepTitle: "Welcome \nBack",
      authStepDescription: "Log in to your account \nto continue cycling.",
      iconData: IconOnlyBold.login,
      middleWidgetsColumn: Column(
        crossAxisAlignment: CrossAxisAlignment.end,
        children: [
          DefaultInput(
            LocaleKeys.fields_email_title.tr(),
            TextInputType.emailAddress,
            isNumber: false,
            hintText: LocaleKeys.fields_email_hint.tr(),
            onChanged: model.changeEmail,
          ),
          SizedBox(
            height: distanceBetweenFields,
          ),
          DefaultInput(
            LocaleKeys.fields_password_title.tr(),
            isPassword: true,
            TextInputType.text,
            isNumber: false,
            hintText: "●●●●●●●●●●●●●●●●",
            onChanged: model.changePassword,
          ),
        ],
      ),
    );
  }
}
```

```

        SizedBox(
          height: distanceBetweenFields,
        ),
        TextBtn(
          text: LocaleKeys.actions_reset_password.tr(),
          onPressedCallback: () => {
            Navigator.pushNamed(context, RouteKeys.resetPassword),
          },
        )
      ],
    ),
    bottomControls: BottomControlsContainer(
      bottomContent: Column(
        children: [
          PrimaryBtn(
            isEnabled:
              authButtonState == ViewModelAuthButtonState.disabled ||
              authButtonState == ViewModelAuthButtonState.authInProgress,
            text: LocaleKeys.actions_login.tr(),
            iconData: IconOnlyLight.login,
            style: PrimaryBtnStyle.accent,
            onPressedCallback: () => onPressed?.call(),
          ),
          const SizedBox(height: 8),
          Row(
            mainAxisAlignment: MainAxisAlignment.min,
            children: [
              Text(
                "Don't have an account?",
                style: Theme.of(context).textTheme.bodySmall,
              ),
              const SizedBox(width: 8),
              PrimaryBtn(
                text: "Create",
                iconData: IconOnlyLight.plus,
                style: PrimaryBtnStyle.grey,
                size: PrimaryBtnSize.small,
                onPressedCallback: () => {
                  Navigator.pushNamed(context, RouteKeys.signUp),
                },
              ),
            ],
          ),
        ],
      ),
    ),
  );
}

```

## Лістинг коду моделі авторизації

```
import 'package:firebase_auth/firebase_auth.dart';

class AuthenticationService {
  final FirebaseAuth _firebaseAuth = FirebaseAuth.instance;

  bool isAutheficated() {
    return _firebaseAuth.currentUser != null;
  }

  Future<void> signIn({required String email, required String password}) async {
    await _firebaseAuth.signInWithEmailAndPassword(
      email: email, password: password);
  }

  Future<void> requestPasswordReset({required String email}) async {
    await _firebaseAuth.sendPasswordResetEmail(email: email);
  }

  String requestInitialData() =>
    _firebaseAuth.currentUser!.displayName ?? "Незнайомець";

  Future<void> signUp(
    {required String email,
    required String password,
    required String name,
    required String surname}) async {
    await _firebaseAuth.createUserWithEmailAndPassword(
      email: email, password: password);

    await signIn(email: email, password: password);
    await _firebaseAuth.currentUser!.updateDisplayName(name + surname);
  }

  Future<void> signOut() async {
    _firebaseAuth.signOut();
  }
}
```

## Лістинг коду впровадження ViewModel у View сторінки входу

```
Widget createLoginPage() {  
  return ChangeNotifierProvider(  
    create: (context) => LoginViewModel(context),  
    child: const LoginPage(),  
  );  
}
```



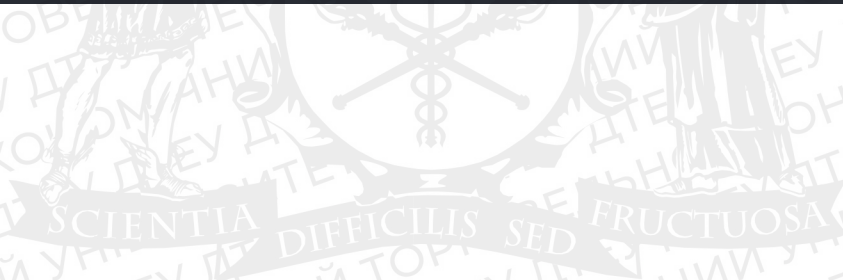
## Лістинг коду стану ViewModel сторінки входу

```
enum ViewModelAuthButtonState { canBePressed, authInProgress, disabled }

class _LoginViewModelState {
  String email = '';
  String password = '';
  bool isAuthInProgress = false;

  ViewModelAuthButtonState get authButtonState {
    if (isAuthInProgress) {
      return ViewModelAuthButtonState.authInProgress;
    } else if (email.isNotEmpty && password.isNotEmpty) {
      return ViewModelAuthButtonState.canBePressed;
    } else {
      return ViewModelAuthButtonState.disabled;
    }
  }
}

_LoginViewModelState();
```





## Лістинг коду ViewModel сторінки реєстрації

```
class SignUpViewModelPage extends ChangeNotifier {  
  final SignUpVMState _state = SignUpVMState();  
  final _authenticationService = AuthenticationService();  
  final BuildContext _buildContext;  
  
  SignUpVMState get state => _state;  
  
  SignUpViewModelPage(this._buildContext);  
  
  void changeEmail(String value) {  
    if (_state.email == value) return;  
    _state.email = value;  
    notifyListeners();  
  }  
  
  void changePassword(String value) {  
    if (_state.password == value) return;  
    _state.password = value;  
    notifyListeners();  
  }  
  
  void changeRepeatPassword(String value) {  
    if (_state.repeatedPassword == value) return;  
    _state.repeatedPassword = value;  
    notifyListeners();  
  }  
  
  void changeName(String value) {  
    if (_state.name == value) return;  
    _state.name = value;  
    notifyListeners();  
  }  
  
  void changeSurname(String value) {  
    if (_state.surname == value) return;  
    _state.surname = value;  
    notifyListeners();  
  }  
  
  // Email validation  
  void toPasswordBtnPressed() {  
    state.currentStep = SignUpSteps.passwords;  
    notifyListeners();  
  }  
}
```

```

// Email validation
void toDetailsBtnPressed() {
  state.currentStep = SignUpSteps.details;
  notifyListeners();
}

Future<void> onSignUpButtonPressed() async {
  String email = state.email;
  String password = state.password;
  String name = state.name;
  String surname = state.surname;

  state.currentStep = SignUpSteps.registrationLoading;
  notifyListeners();

  try {
    await _authenticationService.signUp(
      email: email, password: password, name: name, surname: surname);
    Navigator.of(_buildContext).pushNamedAndRemoveUntil(
      RouteKeys.authScreenLoader, (route) => false);
    notifyListeners();
  } on FirebaseAuthException catch (e) {
    CustomSnackBar.setError(_buildContext, e.code);
    notifyListeners();
  } catch (expection) {
    CustomSnackBar.setError(_buildContext, expection.toString());
    notifyListeners();
  }
}
}

```

## Лістинг коду стану ViewModel сторінки реєстрації

```
enum SignUpSteps { email, passwords, registrationLoading, details }

class SignUpVMState {
    String email = "";
    bool get isEmailValidated => RegExp(
        r"^[a-zA-Z0-9.a-zA-Z0-9.!#$%&'*/+=/?^_`{|}~]+@[a-zA-Z0-9]+\.[a-zA-Z]+"
    ).hasMatch(email);

    String password = "";
    String repeatedPassword = "";

    String name = "";
    String surname = "";

    SignUpSteps currentStep = SignUpSteps.email;

    bool get isPasswordValidated =>
        PasswordValidator.isPasswordValidated(password) &&
        password == repeatedPassword;

    SignUpVMState();
}
```

Сторінка введення пошти для скидання паролю

**Spoby**

**СКИДАННЯ  
ПАРОЛЮ**



Для скидання паролю - введіть  
електронну пошту від Вашого акаунту.

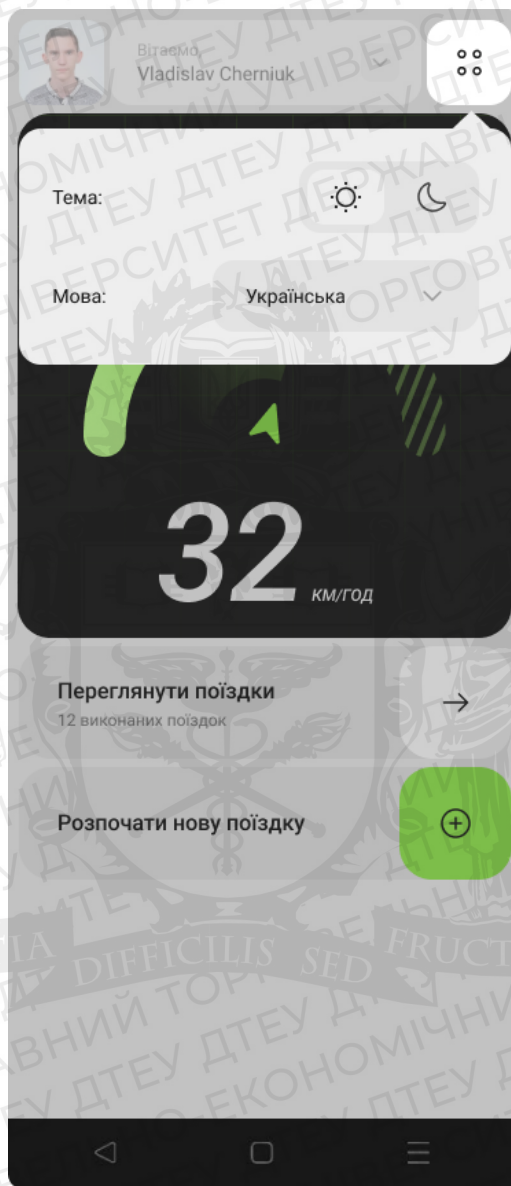
Пошта:

vladcherniuk7@gmail.com

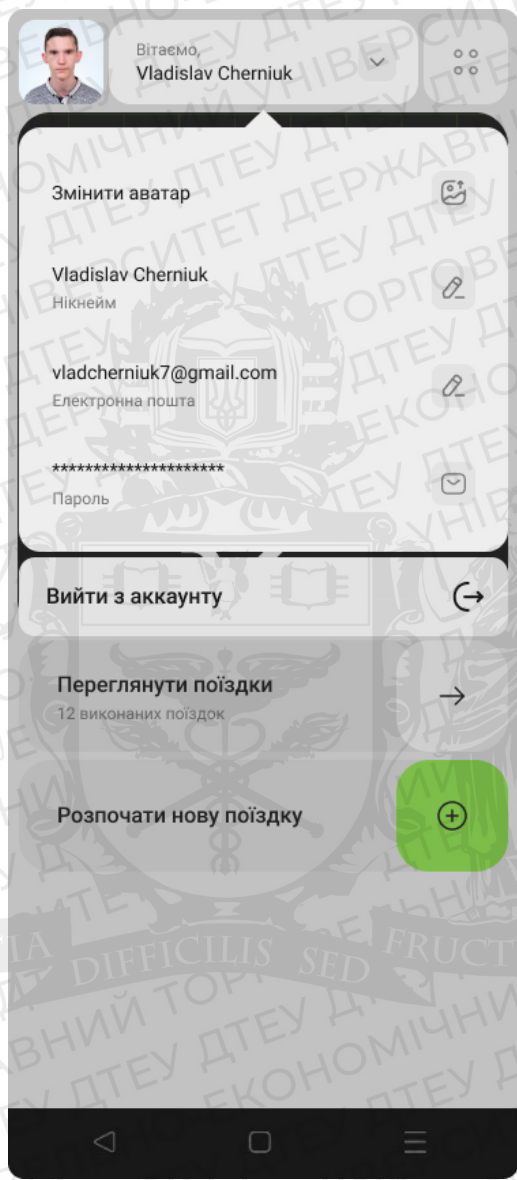
Відправити лист



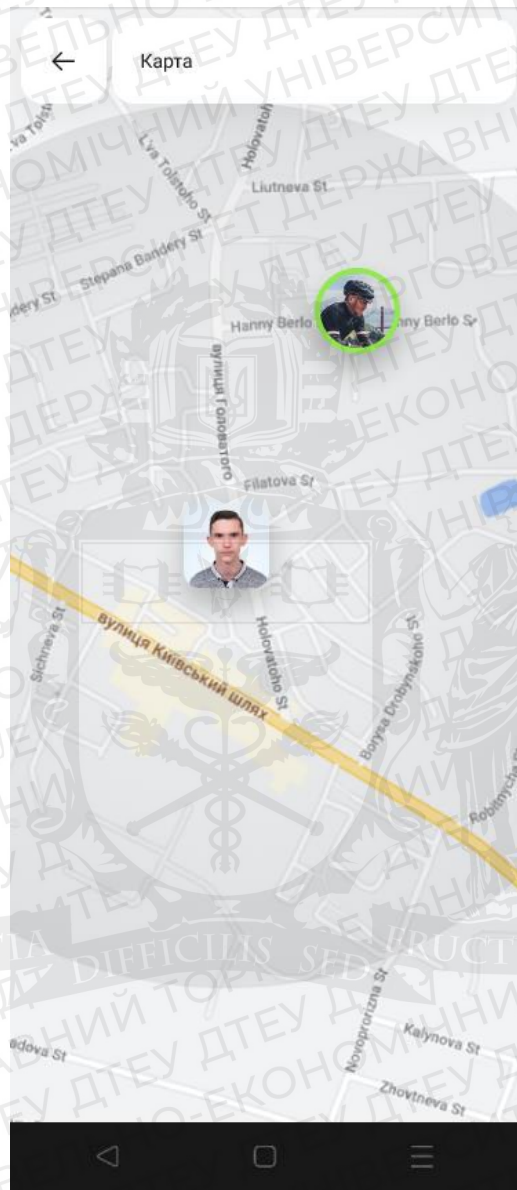
Екран налаштування додатка



Екран редагування профілю




Карта з активними поряд користувачами



Перегляд збереженої статистики

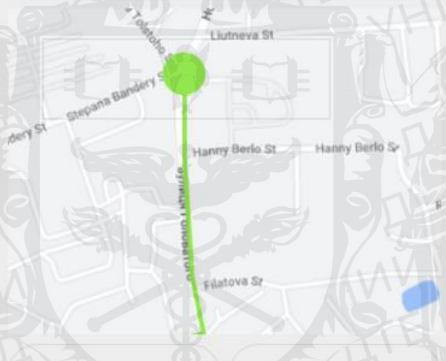
← Перегляд поїздки

 2 учасника **20 Жовтня**

Тривала поїздка 31мин

### Моя найкраща поїздка

Початок: **12:00**      Фініш: **12:31**

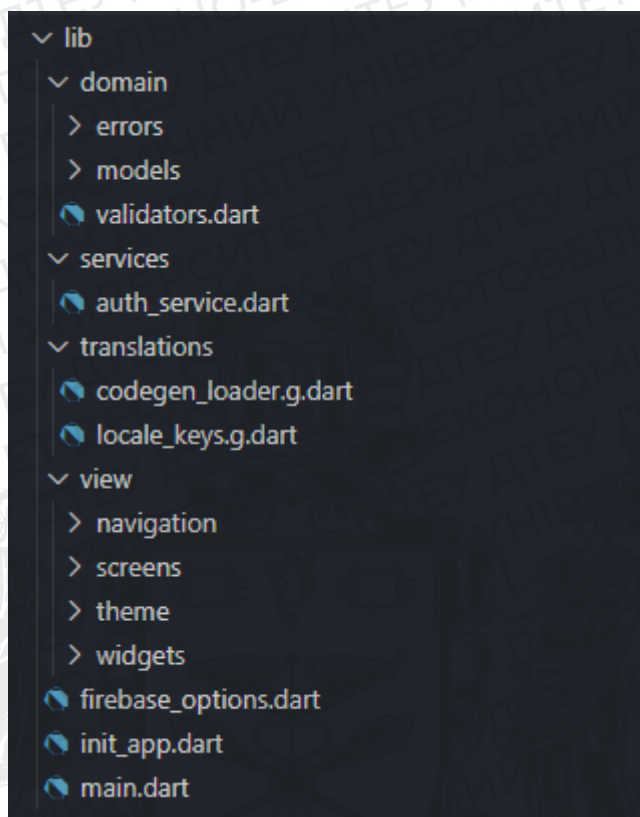


Середня Швидкість: **32** Км/год

Максимальна швидкість: **42** Км/год



### Файлова архітектура проекту



SCIENTIA DIFFICILIS SED FRUCTUOSA

## Лістинг коду для підключення бібліотек

```
dependencies:  
  flutter:  
    sdk: flutter  
  
  #Lints  
  flutter_lints: ^1.0.4  
  
  # Used to create custom color theme and get availability to change theme  
  # modes.  
  adaptive_theme: ^2.3.0  
  
  # Language localization for app.  
  easy_localization: ^3.0.0  
  
  # Icons set for UI.  
  iconsax: ^0.0.8  
  
  # Icons set for UI.  
  iconly: ^1.0.1  
  
  video_player: ^2.6.0  
  
  #Vector animations.  
  lottie: ^2.6.0  
  
  get: ^4.3.8  
  
  # Animations for UI.  
  animations: ^2.0.2  
  
  # Customizable switch  
  flutter_switch: ^0.3.2  
  
  firebase_core: ^2.15.1  
  
  firebase_auth: ^4.9.0  
  
  # Set app icon  
  icons_launcher: ^2.1.3  
  
  # Splash screen for app  
  flutter_native_splash: ^2.3.2  
  provider: ^6.0.5  
  
  top_snackbar_flutter: ^3.1.0
```

## Лістинг коду точки запуску додатка

```
void main() async {  
  WidgetsFlutterBinding.ensureInitialized();  
  await Firebase.initializeApp();  
  await EasyLocalization.ensureInitialized();  
  
  final savedThemeMode = await AdaptiveTheme.getThemeMode();  
  
  SystemChrome.setPreferredOrientations([  
    DeviceOrientation.portraitUp,  
    DeviceOrientation.portraitDown,  
  ]);  
  
  runApp(  
    //Localization Setup  
    EasyLocalization(  
      supportedLocales: const [Locale('en'), Locale('uk')],  
      path: 'assets/translations',  
      fallbackLocale: const Locale('en'),  
      assetLoader: const CodegenLoader(),  
      child: InitApp(savedTheme: savedThemeMode),  
    ),  
  );  
}
```

SCIENTIA DIFFICILIS SED FRUCTUOSA

## Лістинг коду ініціалізація

```
class InitApp extends StatelessWidget {
  const InitApp({Key? key, required this.savedTheme}) : super(key: key);
  final AdaptiveThemeMode? savedTheme;

  static final navigation = RouteNavigation();

  @override
  Widget build(BuildContext context) {
    SystemChrome.setSystemUIOverlayStyle(const SystemUiOverlayStyle(
      statusBarColor: Colors.transparent,
      statusBarIconBrightness: Brightness.dark));
    return AdaptiveTheme(
      light: kLightTheme,
      dark: kDarkTheme,
      initial: AdaptiveThemeMode.light,
      builder: (light, dark) => GetMaterialApp(
        routes: navigation.routes,
        initialRoute: RouteKeys.authScreenLoader,
        debugShowCheckedModeBanner: false,
        theme: light,
        darkTheme: dark,
        localizationsDelegates: context.localizationDelegates,
        supportedLocales: context.supportedLocales,
        locale: context.locale,
        title: 'Spoby',
      ),
    );
  }
}
```



Екран відправлення запрошення та очікування відповіді

