

Міністерство освіти і науки України  
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії  
(повна назва факультету)

Кафедра комп'ютерних наук  
(повна назва кафедри)

# КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

магістр

(назва освітнього ступеня)

на тему: Розробка клієнт-серверного застосунку для систематизації та синхронізації даних геопозиціонування велосипедистів

Виконав: студент  
спеціальності

VI курсу, групи САМ-61  
124 Системний аналіз  
(шифр і назва спеціальності)

(підпис)

Берега Ю.А.

(прізвище та ініціали)

Керівник

(підпис)

Никитюк В.В.

(прізвище та ініціали)

Нормоконтроль

(підпис)

Мацюк О.В.

(прізвище та ініціали)

Завідувач кафедри

(підпис)

Боднарчук І.О.

(прізвище та ініціали)

Рецензент

(підпис)

Муж В.В.

(прізвище та ініціали)

Тернопіль  
2021

Міністерство освіти і науки України  
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії  
(повна назва факультету)

Кафедра комп'ютерних наук  
(повна назва кафедри)

ЗАТВЕРДЖУЮ

Завідувач кафедри

Боднарчук І.О.  
(підпис) (прізвище та ініціали)

«   » грудня 2021 р.

**ЗАВДАННЯ  
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

на здобуття освітнього ступеня Магістр  
(назва освітнього ступеня)

за спеціальністю 124 Системний аналіз  
(шифр і назва спеціальності)

Студенту Березі Юрію Анатолійовичу  
(прізвище, ім'я, по батькові)

1. Тема роботи Розробка клієнт-серверного застосунку для систематизації та синхронізації даних геопозиціонування велосипедистів

Керівник роботи Никитюк В.В., к.т.н., доцент кафедри КН  
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від «28» жовтня 2021 року № 4/7-910

2. Термін подання студентом завершеної роботи 22 грудня 2021р.

3. Вихідні дані до роботи Наукові публікації про методи синхронізації даних та реалізація методів комунікації між мікросервісами та клієнтами.

4. Зміст роботи (перелік питань, які потрібно розробити)

Вступ. 1 Аналітичний огляд літературних та інших джерел. Системний аналіз та обґрунтування проблеми. 2 Проектна частина. 3 Застосування застосунку та опис реалізованих функцій. 4 Охорона праці та безпека в надзвичайних ситуаціях. Висновки.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)

## 6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Охорона праці	Дмитроца Л.П., к.т.н., доцент кафедри КН		
Безпека в надзвичайних ситуаціях	Клепчик В.М., проректор з адміністративно-господарської роботи та будівництва		

7. Дата видачі завдання \_\_\_\_\_ 20 вересня 2021 р.

## КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1.	Ознайомлення з завданням до кваліфікаційної роботи	20.09.21-21.09.21	Виконано
2.	Підбір наукових джерел для проблематики доцільності розробки велосодатку в період пандемії Covid-19	22.09.21-04.10.21	Виконано
3.	Визначення основних функцій додатку, формування поставленого завдання	04.10.21-8.10.21	Виконано
4.	Вибір архітектури проєкту та реалізація серверної частини	11.10.21-17.10.21	Виконано
5.	Реалізація клієнтської частини	18.10.21-20.10.21	Виконано
6.	Тестування	20.10.21-21.10.21	Виконано
7.	Оформлення розділу «Аналітичний огляд літературних та інших джерел. системний аналіз та обґрунтування проблеми»	23.10.21-29.10.21	Виконано
8.	Оформлення розділу «Проектна частина»	01.11.21-05.11.21	Виконано
9.	Оформлення розділу «Застосування застосунку та опис реалізованих функцій»	08.11.21-12.11.21	Виконано
10.	Виконання завдання до підрозділу «Охорона праці»	09.11.20-15.11.20	Виконано
11.	Виконання завдання до підрозділу «Безпека в надзвичайних ситуаціях»	16.11.20-22.11.20	Виконано
12.	Оформлення кваліфікаційної роботи	22.11.21-26.11.21	Виконано
13.	Нормоконтроль	29.11.21-3.11.21	Виконано
14.	Перевірка на плагіат	01.12.21	Виконано
15.	Попередній захист кваліфікаційної роботи	08.12.21	Виконано
16.	Захист кваліфікаційної роботи	22.12.21	

Студент

\_\_\_\_\_ (підпис)

Берега Ю.А.

\_\_\_\_\_ (прізвище та ініціали)

Керівник роботи

\_\_\_\_\_ (підпис)

Никитюк В.В.

\_\_\_\_\_ (прізвище та ініціали)



## АНОТАЦІЯ

Розробка клієнт-серверного застосунку для систематизації та синхронізації даних геопозиціонування велосипедистів // Кваліфікаційна робота освітнього рівня «Магістр» // Береза Юрій Анатолійович // Тернопільський національний технічний університет імені Івана Пулюя, факультет комп'ютерно-інформаційних систем і програмної інженерії, кафедра комп'ютерних наук, група САМ-61 // Тернопіль, 2021 // с. 63, рис. – 20, табл. – 4, бібліогр. – 50, додат. – 2.

Ключові слова: геолокація, вело-додаток, клієнт-сервісна архітектура, база даних, Signal-R, Identity4, rest API, консолідація даних.

У кваліфікаційній роботі проаналізовано доцільності розробки додатку. Досліджено вплив пандемії Covid-19 на популярність велотранспорту. Досліджено розвиток урбанізації міст для поліпшення інфраструктури велотранспорту.

Визначено основних потреб велосипедистів.

На основі аналізу та досліджень було розроблено клієнт-серверний застосунок для систематизації та синхронізації даних геопозиціонування велосипедистів, який б задовольняв більшість потреб користувачів автотранспорту.

## ANNOTATION

Development of the client-server application for geopositioning data systematization and synchronization in cycling // Qualification thesis Master Degree // Yurii Bereza // Ternopil' Ivan Pul'uj National Technical University, Faculty of Computer Information System and Software Engineering, Department of Computer Science, group SAm-61 // Ternopil, 2021 // Pages 63, Fig. –20, Tables –4, References – 50, Annexes – 2.

Key words: geolocation, bicycle application, client-service architecture, dinh database, Signal-R, Identity4, rest API, data consolidation.

In the qualification work, the expediency of application development is analyzed. The impact of the Covid-19 pandemic on the popularity of cycling has been researched. The development of urbanization of cities to improve the infrastructure of cycling is researched.

The main needs of cyclists have been identified.

Based on the analysis and research, a client-server application was developed to systematize and synchronize cycling geopositioning data, which will meet most of the needs of cycling users.

## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

BITS – Birla Institute of Technology & Science;

GPS (англ. Global Positioning System) – система глобального позиціонування;

UI (англ. user interface) – користувацький інтерфейс;

UX (англ. user experience) – досвід користування;

БД – база даних;

API (англ. Application programming interface) – прикладний програмний інтерфейс;

ОС – операційна система;

ПК – персональний комп'ютер;

НПАОП – нормативно-правові акти з охорони праці;

ПУЕ – правила улаштування електроустановок;

ПТЕЕС – правила технічної експлуатації електроустановок споживачів;

## ЗМІСТ

ВСТУП.....	8
1 АНАЛІТИЧНИЙ ОГЛЯД ЛІТЕРАТУРНИХ ТА ІНШИХ ДЖЕРЕЛ. СИСТЕМНИЙ АНАЛІЗ ТА ОБҐРУНТУВАННЯ ПРОБЛЕМИ.....	9
1.1 Популяризація велотранспорту.....	9
1.2 Потреби велосипедистів .....	12
1.2.1 Навігація.....	13
1.2.2 Дані руху .....	13
1.2.3 Статистичні дані .....	14
1.2.4 Пошук існуючих веломаршрутів .....	15
1.2.5 Пошук найближчих послуг велопрокату .....	16
1.3 Вимоги до додатку .....	16
1.4 Висновки до першого розділу .....	17
2 ПРОЄКТНА ЧАСТИНА .....	18
2.1 Постановка завдання.....	18
2.2 Архітектура проєкту .....	19
2.2.1 Identity server.....	20
2.2.2 Data storage service .....	22
2.2.3 Notification service.....	24
2.3 Бази даних .....	25
2.3.1 Серверні бази даних .....	25
2.3.2 Локальна база даних .....	26
2.4 Консолідація баз даних.....	27
2.5 Комунікація між сервісами та клієнтами.....	29
2.6 Мобільні клієнти.....	31



2.6.1 Інструменти розробки .....	31
2.6.2 Опис екранів додатку .....	32
2.7 Висновки до другого розділу .....	42
3 ЗАСТОСУВАННЯ ЗАСТОСУНКУ ТА ОПИС РЕАЛІЗОВАНИХ ФУНКЦІЙ .....	43
3.1 Опис реалізованих функцій.....	43
3.2 Тестування.....	43
3.3 Модифікація оновлень та розповсюдження додатку .....	48
3.4 Висновки до третього розділу .....	49
4 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ .....	50
4.1 Організація робочого місця для користувача ПК .....	50
4.2 Електробезпека для користувачів ПК.....	53
4.3 Висновки до четвертого розділу .....	56
ВИСНОВКИ .....	57
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	58

## ВСТУП

**Актуальність теми роботи.** Впродовж останніх пару років, пандемія Covid-19 спричинила різке зростання популярності велотранспорту, окрім пандемії на популярність велотранспорту вплинули: урбанізація міст, розробка зручної велотранспортної інфраструктури, політика уряду. Все це призвело в необхідність створення зручного застосунку, який б задовільнив потреби більшості користувачів велотранспорту.

**Метою кваліфікаційної роботи** є розробка клієнт-серверного застосунку для систематизації та синхронізації даних геопозиціювання велосипедистів.

Досягнення поставленої мети викликало необхідність виконання таких завдань:

- дослідити доцільність розробки застосунку;
- визначення основних вимог для розробки застосунку;
- вибір архітектури застосунку;
- реалізація сервера;
- реалізація клієнта;
- тестування застосунку.

**Об'єктом дослідження** – потреби користувачів велотранспорту.

**Предмет дослідження** – урбанізація міст, та покращення велоінфраструктури, вплив політики та пандемії Covid-19 на популяризацію велотранспорту.

**Науковою новизною роботи** є створення нового конкурентоспроможного вело-додатку.

**Практичне значення** одержаних результатів. Визначення основних вимог для створення застосунку, вибір архітектури розробки, вибір інструментів розробки, реалізація серверу, створення баз даних, реалізація клієнтів, реалізація комунікації клієнтів та сервера, тестування застосунку.

# 1 АНАЛІТИЧНИЙ ОГЛЯД ЛІТЕРАТУРНИХ ТА ІНШИХ ДЖЕРЕЛ. СИСТЕМНИЙ АНАЛІЗ ТА ОБҐРУНТУВАННЯ ПРОБЛЕМИ

## 1.1 Популяризація велотранспорту

В сучасному світі більшість людей працюють в офісі, ведуть малорухливий спосіб життя, страждають на ожиріння. Останні роки популяризація велотранспорту значно зросла. Велосипед завжди був одним з найшвидших, мобільним та найбільш надійним видом транспорту. Основною ідеєю цього виду транспорту, проведення руху за допомогою людських м'язів. Існує багато типів велосипедів: двоколісні (найпоширеніший), триколісні та екзотичні (одноколісні, чотирьох колісні та інші). Понад один мільярд велосипедів використовують у наш час, таким чином даний транспорт вважається найрозповсюдженішим видом транспортом. Він здобув цей статус, завдяки не великій собівартості та вартості самообслуговування (вартість деталей), не великій вазі та розмірам, відсутності в потребі пального (не включаючи електровелосипеди) [1].

Протягом останніх 5 років популярність велосипедів в розвинених країнах значно зросла, завдяки екологічності та бажанню людей до фізичного навантаження. Пандемія Covid-19 сприяла різкому зростанню продажів велосипедів, як так званий «Bicycle boom». Збільшилася тривожність людей у зв'язку з проблемою доступності громадського транспорту в період пандемії та потребою у фізичному навантаженні, оскільки більшість людей працюють віддалено вдома, – це все наштовхує людей використовувати одну з найосновніших форм мобільного транспорту – велосипеди та інші велотранспорт [2].

За даними роздрібної служби відстеження роздрібної торгівлі NPD [3], у Сполучених Штатах зросли продажі велосипедів в березні 2020 року, для деяких категорій велосипедів спостерігалися темпи зростання, понад 100 відсотків більше продали більше велосипедів порівняно з попереднім роком. «Споживачі шукають заходи для відпочинку на відкритому повітрі та для дітей, щоб краще переносити період пандемії вдома», – сказав Дірк Соренсон, аналітик спортивної індустрії NPD,

додавши, що дитячі велосипедів є доступні за ціною, особливо добре продавались велосипеди для відпочинку дорослих. Результати зростання продажів наведено на рисунку 1.1 [4].

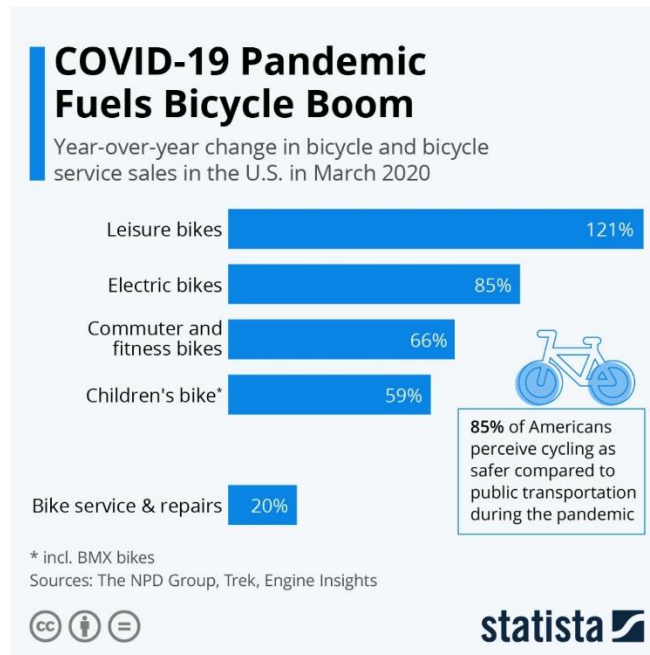


Рисунок 1.1 – Ріст продажів велосипедів в період пандемії Covid-19 «Bicycle Boom» в США

Переглянувши рисунок 1.1 можемо зробити висновок, що пандемія Covid-19 вплинула на зростання популярності велотранспорту.

Також про зростання популярності велотранспорту підтверджує опитування BITS. Опитування розпочалося минулого року у вересні, за цей час анкети опитування заповнили понад сім тисяч мешканців міст та регіонів північних країн Європи [5].

В цілому, результати опитування BITS продемонстрували, що їзда на велосипеді є популярною діяльністю, більше 75% респондентів назвали себе «цілковитими велосипедистами», які їздять на велосипеді принаймні раз на тиждень. Цікаво, що відповіді дозволили сприймати різні групи. Перша група – люди з високою освітою, друга – люди, які живуть разом з друзями, студентами (працівниками, які працюють повний або неповний робочий день), а також третя –

пенсіонери, як правило, частіше їздять на велосипедах, також до третьої групи віднесли безробітних та осіб, які ведуть домашнє господарство повний робочий день.

Деякі опитані особи вважають велосипед транспортним засобом, а інші – чистим задоволенням. У будь-якому випадку, вони використовують велосипед для різних цілей. Більше половини опитаних їздять кілька разів на тиждень на шопінг, для відвідування сім'ї та друзів або для подорожей, до половини респондентів використовують велосипед як транспорт пересування до роботи та до школи. Тим не менш, їзда на велосипеді – це набагато більше, ніж просто пересування з одного місця на інше. Кожен третій респондент їздить на велосипеді для дозвілля кілька разів на тиждень.

Опитування показало, що люди хочуть більше їздити на велосипеді. Наприклад, 30% респондентів хотіли б більше їздити на велосипеді, щоб їздити на роботу, 43% хотіли б більше їздити на велосипеді для шопінгу та подорожей для дозвілля, 53% хотіли б більше їздити на велосипеді, як для відпочинку.

Половина респондентів зазначили, що, окрім погодних факторів та факторів відстані, існує чи мало причин, які заважають їм сьогодні більше їздити на велосипеді. Незважаючи на те, що безпечні велосипедні маршрути та велопарковки покращуються, ці проблеми залишаються важливими для людей, які користуються велосипедами. Дійсно, 40% респондентів незадоволені ініціативами урядів щодо велосипедної політики у своїх регіонах. Особливо щодо стану та обслуговування (45%) велодоріжок, а також безпеки велосипедних доріжок (39%) та переходів (39%).

Відповідно до опитування BITS, немає жодних сумнівів, що всі люди повинні їздити на велосипед – чи то просто для розваги, для підтримки фізичної форми, для зміцнення свого стану здоров'я чи для збереження довкілля. Хоча мотивація є важливим чинником їзди на велосипеді, але не всім вдається втілити її на практиці.

Іншим каталізатором популяризації велотранспорту є урбанізація міст, створення зручної інфраструктури, велодоріжок для велосипедистів, міські адміністрації по всій Європі та за її межами досягли реального прогресу в плануванні та забезпеченні велосипедного руху у своїх містах.

Щодня європейські міста демонструють, що скорочення використання приватних автомобілів є не просто бажаним, а й реалізованими багатьма містами, приймаються відповідні закони, розбудовується сприятлива інфраструктура, які стимулюють громадян перейти на користування громадським транспортом та велосипедами. Міста не пригнічують розвитком вело-інфраструктури своєму економічному зростанню та доступу до торгових центрів. Насправді, міста отримують покращення розвитку та розбудові міст, оскільки нестримне використання автомобілів для індивідуальних поїздок для більшості громадян більш не сумісне з простотою мобільності.

Сьогодні міста визнають особисті та суспільні переваги, які потенційно можуть покращити умови для їзди на велосипеді, тому що – це рентабельно, швидко, екологічно, гнучко та ефективно, таким чином спрямовується політика, яка сприяє просування цього виду транспорту, реалізується комфортна інфраструктура, змінюється соціальна згуртованість економічного розвитку.

Деякі країни винагороджують громадян покупкою нових велосипедів або покривають витрати на устаткування технічного обслуговування велосипедів, намагаючись заохотити пасажирів під час пандемії. Наприклад, Італія виділила 210 млн. євро на програму повернення готівки, така схожа програма була задіяна і у Франції [6].

З вище наведених причин популярності можна зробити висновок, що велосипедний ринок, значно виріс, робиться вся сприятлива інфраструктура для його зростання, велосипед є корисним для здоров'я, та екологічним видом транспорту.

## **1.2 Потреби велосипедистів**

Існує чи мало додатків та девайсів, якими користуються власники велосипедів, які забезпечують певні потреби такі як: навігація, пошук найближчих веломагазинів, пошук веломаршрутів, перегляд статистики та хронології поїздки, перегляд даних руху в реальному часі.

### 1.2.1 Навігація

Навігація є однією з основних функцій додатку, вона спрямовує (підказує) напрям руху транспортного засобу до цілі.

Існує багато додатків які працюють з мапами та навігацією. Найбільш поширені: GoogleMaps, AppleMaps, BingMaps, OpenStreetMaps. Особливістю роботи з вищенаведеними додатками є взаємодія з ними, достатньо лише викликати додаток з спеціальним текстом спрямування, тоді як він відкриється з навігацією з покроковими вказівками. Серед відомих, існує також вузькоспеціалізований додаток BikeMaps призначений для пошук веломаршрутів, та прокладання їх, особливістю додатка є те, що можна взаємодіяти з іншими користувачами, ділитися маршрутами, вибирати з них найкращі, вносити свої корективи.

### 1.2.2 Дані руху

Не менш важливою основною функцією додатку є перегляд даних руху, це може бути такі дані, як: швидкість руху, пульс велосипедиста, кількість спалених калорій, максимальна швидкість руху, максимальна висота та мінімальна висота спуску / підйому.

*Швидкість руху* – це переміщення тіла не за весь час руху тіла, а за одиницю часу. Іноді кажуть, що швидкість чисельно дорівнює шляху, що тіло проходить за одиницю часу. Вірним можна вважати обидва твердження.

Дані швидкості руху велосипедиста можна отримати багатьма способами, найбільш розповсюджені способи виміру: електронний та за допомогою системи супутникового позиціонування (GPS).

Електронним способом швидкість вимірюється за допомогою оптичного, магнітного або механічного датчика, який генерує імпульси струму при обертанні шпинделя. Імпульси виробляються електронною схемою та виводяться на екран.

За допомогою супутникового позиціонування (GPS) вимірюється швидкість за формулою: відстань поділити на одиницю часу. Цей метод буде використовуватися для виміру в додатку, оскільки кожен смартфон оснащений GPS трекером. За допомогою GPS трекера будемо визначати максимальну швидкість на маршруті.

*Пульс велосипедиста.* У вело-додатках дані пульсу велосипедиста беруться з вже існуючі функції, які розташовані у смарт-годинниках, які містять спеціальний сканер.

*Кількість спалених калорій.* В середньому людина катаючись одну годину на велосипеді спалює від 450 до 740 калорій. Кількість калорій залежить від ваги велосипедиста, швидкості та часу. Кількість спалених калорій (K). Для визначення кількості калорій спалених у русі для чоловіків визначається за формулою [7]:

$$K = \frac{(Y*0.2017 - W*0.09036 + C*0.6309 - 55.09.69)*t}{4.184}$$

Для жінок кількість спалених калорій у русі визначається за формулою [7]:

$$K = \frac{(Y*0.074 - W*0.05741 + C*0.4472 - 20.4022)*t}{4.184}$$

де Y – вік велосипедиста, W – вага велосипедиста, C – частота серцевих скорочень, t – час.

*Максимальна та мінімальна висота спуску та підйому* та спуску велосипедист отримує через проходження певної відстані методом визначення певної геолокації місця розташування, через спеціальний сервіси (Google Maps, Open Street Maps, Bing Maps, GeoInfo) відправивши дані маршруту можна отримати інформацію про рельєф місцевості (висоту місця знаходження над рівнем моря).

### **1.2.3 Статистичні дані**

Для відображення необхідної інформації в статистиці даних слід розуміти, що не потрібно перенасичувати звіти інформацією, а слід дати можливість користувачу вибрати те, що його зацікавить. У додатку буде реалізована кастомна статистика, яку користувач може сам налаштувати під власні потреби.

Статистика даних буде запропонована для відображення по: день, тиждень, місяць, рік. Користувач матиме можливість додавати необхідну інформацію, для



відображення: протяжність (скільки проїхав), серцевий ритм, спалені калорії, рівень підняття та спуску (максимальна набрана висота на підйомі, мінімальна висота при спуску), швидкість руху на дистанції та інші.

Це дасть перевагу над існуючими програмами, право вибору статистики яку б користувач хотів бачити, одночасно це позбавить нагромадженню даних.

#### 1.2.4 Пошук існуючих веломаршрутів

Функція пошук існуючих веломаршрутів, дозволить користувачам створювати свої вело маршрути, публікувати їх для інших, оцінювати вже існуючі вело маршрути тощо.

Веломаршрут складається з двох або декількох точок геолокації, з'єднаних між собою. Кожна геолокаційна точка має певне призначення в маршруті (стоянка, туристичне місце, історична споруда та інші мітки). Приклад веломаршруту наведено на рисунку 1.2 [8].

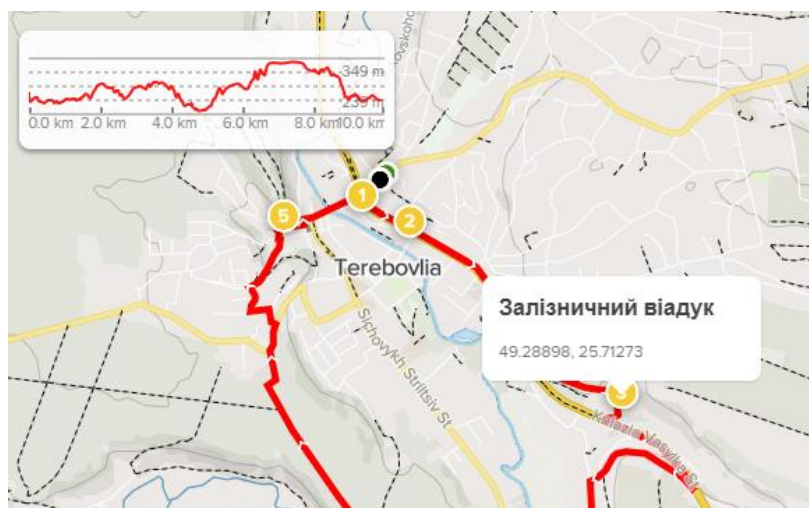


Рисунок 1.2– Веломаршрут міста Тереховля

У рисунку 1.2 зображено веломаршрут міста Тереховля, який охоплює всі історичні визначні місця, цей маршрут можна відкрити в режимі руху, який буде показувати місце розташування велосипедиста та буде спрямовувати його в правильному напрямку.

Цей функціонал є додатковим, він залучить ще більшу аудиторію користувачів, які цікавляться велотуризмом та подорожами, оскільки більшість історичних та туристичних міст пропонують прокат велотранспорту.

### **1.2.5 Пошук найближчих послуг велопрокату**

Не менш корисною буде функціональність пошуку найближчих точок велопрокату, вона буде корисною для туристів, які подорожують, відвідують нові місця, які залюбки б орендували велосипед на період подорожі. Дані для пошуку точок прокату велотранспорту можна отримувати з відкритих баз даних Google companies або зі власної бази даних з магазинами-партнерами. Пошук буде здійснюватися в найближчому радіусі десяти кілометрів від місця розташування.

## **1.3 Вимоги до додатку**

В результаті дослідження популярності автотранспорту, та основних потреб велосипедиста, можемо сформувані основні вимоги для створення додатку, додаток повинен містити функціонал:

- показувати поточну швидкість велотранспорту;
- визначати, вираховувати пройдений шлях, відображати та час у русі;
- визначати кількість спалених калорій;
- прокладати маршрути до місця призначення.

Не менш корисними будуть додаткові функції, які б залучили більшу аудиторію користувачів:

- аналіз та статистика поїздок;
- пошук найближчих веломагазинів або компаній, якій займаються велопрокатом;
- створення веломаршрутів та велокарт;
- функції комунікації між користувачами.

Всі ці вище перелічені вимоги будуть взяті до уваги під час розробки мобільного додатку.

## **1.4 Висновки до першого розділу**

В даному розділі досліджено популяризацію велотранспорту, серед них слід підкреслити наступні чинники:

- пандемія Covid-19;
- розвиток велоінфраструктури;
- позитивний вплив здоров'я;
- екологічність та інші чинники.

Також в даному розділі було визначено основні потреби користувачів велотранспорту, перелічено основні функції, які слід розробити в додатку, а саме:

- показ даних руху (швидкість, пульс, відстань, спалені калорії);
- створення веломаршрутів, взаємодія з мапою;
- аналіз статистики руху (побудова діаграм, визначення прогресу активності);
- пошук найближчих точок велопрокату.

## 2 ПРОЄКТНА ЧАСТИНА

### 2.1 Постановка завдання

Реалізувати клієнт-серверний додаток, який задовольнятиме потреби користувачів велосипедів. Додаток повинен містити в собі такі функції та особливості:

- Функції авторизації та реєстрації;
- Механізм шифрування даних користувачів
- Функції вирахування місцезнаходження за допомогою GPS;
- Функції обчислення швидкості руху;
- Функція підрахунку спалених калорій;
- Механізм синхронізації даних з смарт-годинником;
- Функції візуалізації руху, збереження даних траєкторій руху;
- Функції візуалізація статистики даних;
- Функції консолідації даних, збору інформації.

Клієнти додатку (мобільні програми) повинні містити зручний сучасний графічний інтерфейс (UI / UX), повинно бути реалізовано:

- вікно заставки (splash screen);
- вікно входу (Login screen);
- вікно реєстрації;
- вікно перегляду даних в реальному часі;
- вікно перегляду історії поїздок;
- вікно перегляду статистики даних;
- вікно маршрутизації;
- вікно пошуку найближчих точок велопрокату;
- вікна вибору та створення маршруту.

Окрім проектування та розробки, додаток також повинен бути протестований та оптимізований, не повинен містити в собі критичних помилок (bugs) та відповідати всім поставленим вимогам.

## 2.2 Архітектура проєкту

Для реалізації проєкту було обрано мікросервісну архітектуру.

Мікросервісна архітектура – стиль розробки проєктних рішень, за яким єдиний застосунок будується як сукупність невеликих сервісів, кожен з яких працює у своєму власному середовищі (процесі) та спілкується з рештою, використовуючи прості та швидкодіючі протоколи передачі даних. Ці сервіси будуються навколо бізнес-потреб і розгортаються незалежно один від одного з використанням зазвичай повністю автоматизованого середовища. Існує абсолютний мінімум централізованого керування цими сервісами[17].

Мікросервіси є невеликими та незалежними або слабо з'єднаними, для обслуговування яких не потрібно багато людських ресурсів. Кожен сервіс є окремою кодовою базою. Основною перевагою мікро сервісів є те, що оновлюючи один мікросервіс не потрібно перебудовувати чи оновлювати інший сервіс. Також перевагою є те, що кожен сервіс має доступ лише до поверхневого ресурсу іншого сервісу, не має доступ до прихованих служб. Кожен сервіс може бути написаний з використанням різних мов програмування і технологій зберігання даних [18]. Схема архітектури проєкту зображена на рисунку 2.1.

На рисунку 2.1 видно, що дані зберігаються на окремих, розподілених базах даних, для кожної бази даних існує свій мікросервіс, який може виконувати певний стек транзакцій. Всі сервіси можуть комунікувати між собою. Клієнти можуть бути різноманітними від мобільних телефонів, до веб-сайтів та стільникових програм.

Комунікація клієнтів виконується через HTTPS протокол передачі даних, з використанням токенів, які забезпечують надійний та захищений спосіб даних. Токен можна отримати звернувшись до сервера авторизації (Identity server). Data storage service може також звертатися до сервісу сповіщень (Notification service), а сервіс сповіщень в свою чергу надсилати клієнтам сповіщення.

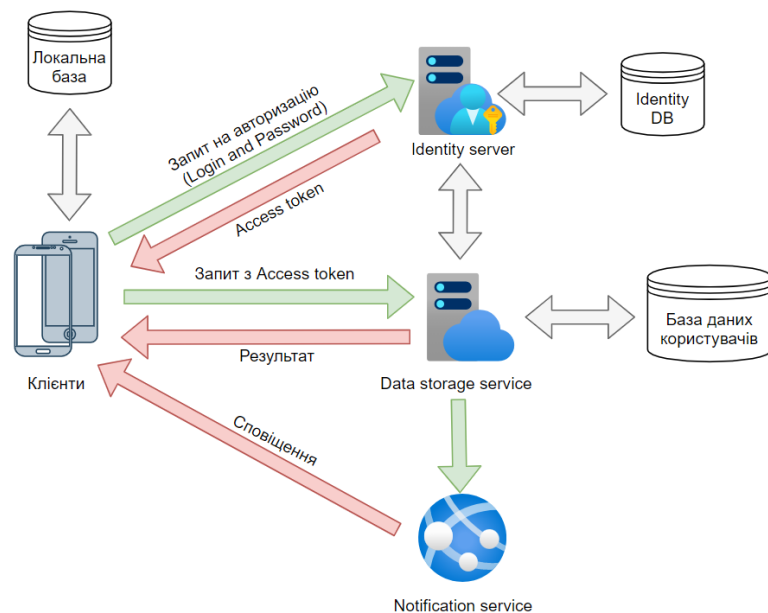


Рисунок 2.1 – Мікросервісна архітектура проєкту

### 2.2.1 Identity server

Identity server – сервер авторизації, який перетворює звичайні методи авторизації (відбиток пальця, пароль та логін, цифровий ключ, вхід за допомогою сторонніх сервісів та соцмереж та інші) в цифровий код – access token. За допомогою токенів можна захищено отримати дані з дружніх ресурсів та мікро сервісів. Схема авторизації наведена на рисунку 2.2 [19].

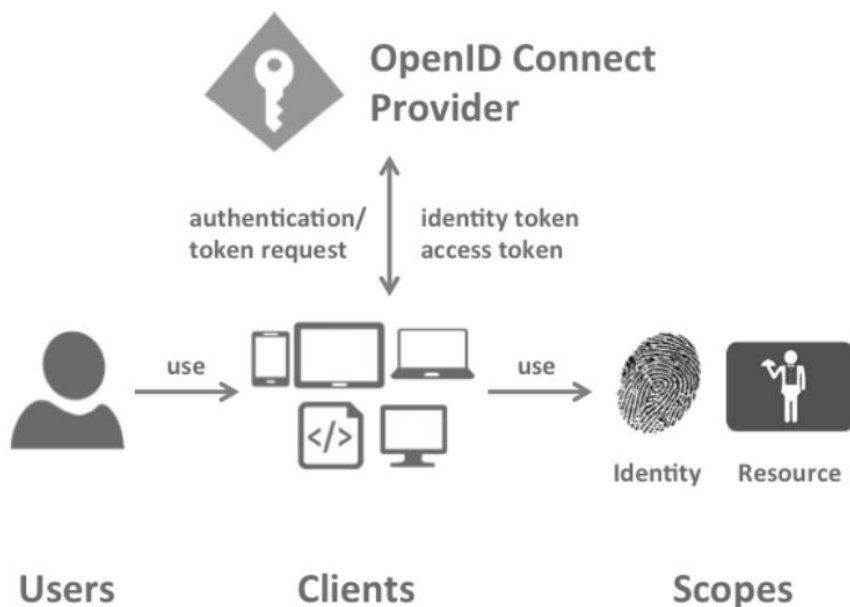


Рисунок 2.2 – Схема авторизації Identity server

Користувач використовуючи різні пристрої, звертається до серверу авторизації, в разі успішної авторизації отримує access token. Токен містить в собі зашифровані дані ключем, про який знає лише сервер. Користувач використовуючи токен матиме можливість отримати доступ до ресурсів Identity сервера. Також токен має термін дії, в випадку, якщо термін дії токена закінчиться, користувач не зможе отримати доступ до ресурсів Identity сервера.

Розгортання сервера авторизації в проєкті не займає багато часу, в проєкті використовується identity4 [20].

Спочатку налаштовується конфігурація сервера (див. лістинг 2.1).

#### Лістинг 2.1 – Конфігурація Identity server

```
public void ConfigureServices(IServiceCollection services)
{
    var builder = services.AddIdentityServer()
        .AddDeveloperSigningCredential()
        .AddInMemoryApiScopes(Config.ApiScopes)
        .AddInMemoryClients(Config.Clients);
}
```

Далі додаються ресурси сервера, для яких сервер буде надавати доступ (див. лістинг 2.2).

#### Лістинг 2.2 – Ресурси

```
public static class Config
{
    public static IEnumerable<ApiScope> ApiScopes =>
        new List<ApiScope>
        {
            new ApiScope("api1", "BikeTravel API")
        };
}
```

Також слід додати клієнтів сервера, для яких сервер буде надавати доступ (див. лістинг 2.3). Список клієнтів з часом можна розширювати.

### Лістинг 2.2.3. – Клієнти

```
public static IEnumerable<Client> Clients =>
    new List<Client>
    {
        new Client
        {
            ClientId = "MobileClient",

            AllowedGrantTypes = GrantTypes.ClientCredentials,
            ClientSecrets =
            {
                new Secret("secret".Sha256())
            },
            AllowedScopes = { "api1" }},
        new Client
        {
            ClientId = "WEBClient",

            AllowedGrantTypes = GrantTypes.ClientCredentials,
            ClientSecrets =
            {
                new Secret("secret".Sha256())
            },
            AllowedScopes = { "api1" } }
    };
```

На цьому налаштування сервера авторизації завершується, залишається лише реалізувати контролер та способи авторизації.

В проєкті будуть реалізовуватися два способи авторизації через Login та пароль, також через соціальні мережі такі як Facebook та Google.

### 2.2.2 Data storage service

Сервіс який буде обробляти та зберігати дані додатку – Data storage service. В проєкті буде реалізована REST API архітектура, в якому клієнти можуть оновлювати дані по HTTP протоколу.

API – це набір визначень та протоколів для створення та інтеграції прикладного програмного забезпечення. Іноді його називають договором між постачальником інформації та користувачем інформації встановлюючи вміст, необхідний серверу, та вміст, необхідний клієнту (відповідь) [21].



REST – це набір архітектурних обмежень, а не протокол або стандарт. Розробники API можуть реалізовувати REST різними способами.

Запит POST – вставляє дані.

Запит GET – видає дані.

Запит PUT – оновлює дані.

Запит DELETE – видаляє дані.

Запит PATCH – оновлює певне поле моделей даних.

Для захищеного отримання та передачі даних буде використовувати технологія авторизації OAuth2.

OAuth є відкритим стандартом для доступу делегації, зазвичай використовується як спосіб для користувачів Інтернету для надання веб-сайтів або додатків доступу до їх інформації на інших сайтах, але не даючи їм паролі [22].

Для кращого розуміння та простоти розробки комунікації клієнтів з сервісами було використано технологію Open API Swagger. На рисунку 2.3 зображено Open API у браузері.

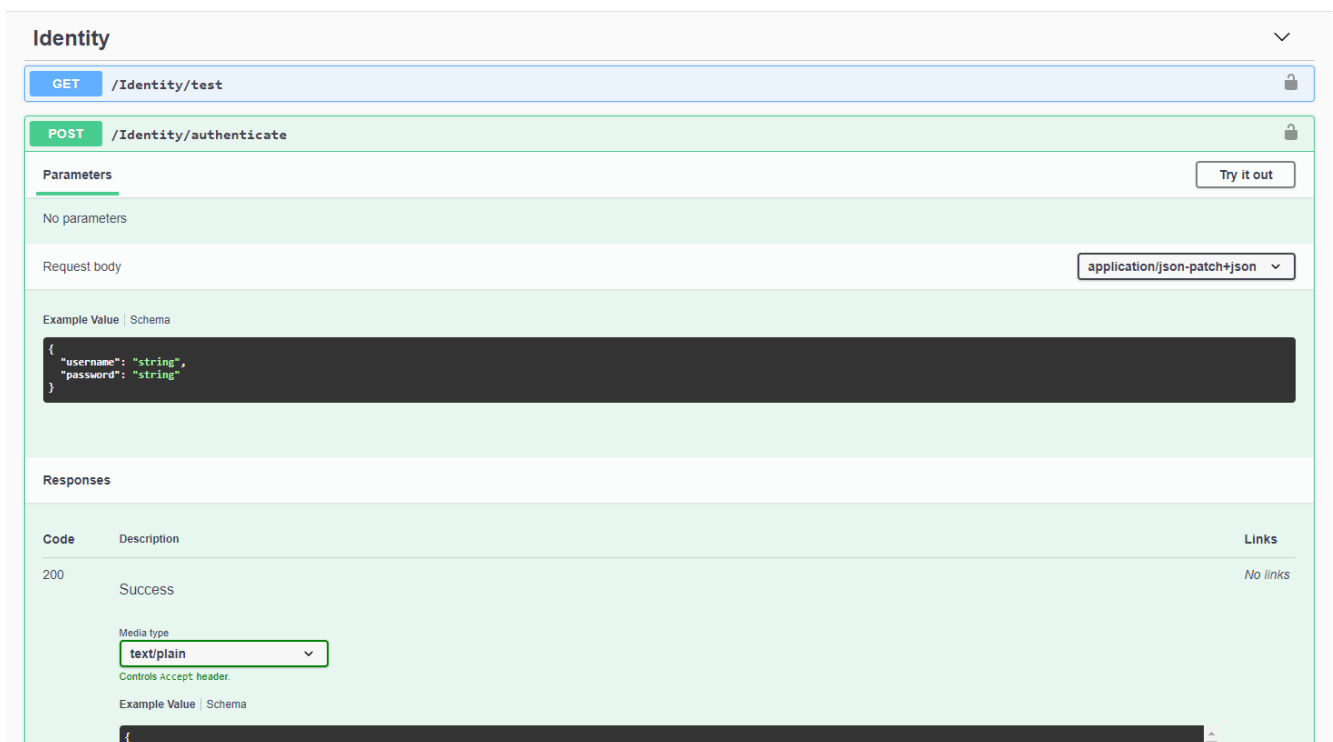


Рисунок 2.3 – Open API Swagger

Swagger (Open API) – це не залежна від мови специфікація для опису REST API. Вона дозволяє комп'ютерам і користувачам краще зрозуміти можливості REST API без прямого доступу до вихідного коду. Її основні цілі:

- звести до мінімуму обсяг робіт, необхідних для з'єднання окремих служб;
- скоротити час, необхідний для точного документування служби [24].

У лістингу 2.4 наведено код конфігурації Swagger.

#### Лістинг 2.4 – Конфігурація Swagger

```
public void Configure(IApplicationBuilder app)
{
    app.UseSwagger();
    app.UseSwaggerUI(c => { c.SwaggerEndpoint("/swagger/v1/swagger.json",
"BikeTravel API"); });

    app.UseRouting();
    app.UseEndpoints(endpoints =>
    {
        endpoints.MapControllers();
    });
}
```

### 2.2.3 Notification service

Notification service – це сервіс сповіщень між сервером та клієнтами. Сервер сповіщень буде реалізовуватися в проєктів використовуючи вже існуючий інструмент Microsoft Azure notification hub Основні переваги роботи сервісу:

- швидка трансляція повідомлень на клієнти;
- динамічність повідомлень;
- підтримка мультимовних пакетів локалізації [25].

Сервіс буде транслювати повідомлення з сервера на мобільні клієнти iOS та Android. Для того щоб сервіс міг транслювати повідомлення необхідно додаток зареєструвати в Firebase – лише для Android, для iOS реєстрація не потрібна [26].

## 2.3 Бази даних

Для реалізації баз для мікросервісів, бази даних повинні відповідати наступним вимогам та принципам:

- Служби повинні бути слабо пов'язані, щоб їх можна було розробляти, розгортати та масштабувати незалежно.
- Деякі бізнес-операції повинні застосовувати інваріанти, які охоплюють декілька служб.
- Деякі бізнес-операції мають запитувати дані, які належать кільком службам.
- Деякі запити мають об'єднувати дані, які належать кільком службам.
- Для масштабування бази даних іноді потрібно реплікувати та розділяти. Див. Масштабний куб.
- Різні служби мають різні вимоги до зберігання даних. Для деяких служб найкращим вибором є реляційна база даних. Іншим службам може знадобитися база даних NoSQL, наприклад MongoDB, яка добре зберігає складні, неструктуровані дані, або Neo4J, яка розроблена для ефективного зберігання та запитів даних графіка [27].

У проєкті передбачено три бази даних:

- база серверу авторизації;
- база даних користувачів додатку;
- локальна база користувачів додатку.

### 2.3.1 Серверні бази даних

Бази даних серверів було вирішено розробити в середовищі MS SQL Server за допомогою технології CodeFirst. Розробники MS Entity framework розробили методика побудови баз даних за допомогою програмного коду, структура баз даних реалізується за допомогою змін в кодї з використанням процедури міграції даних. Схема роботи CoreFirst зображена на рисунку 2.4 [28].

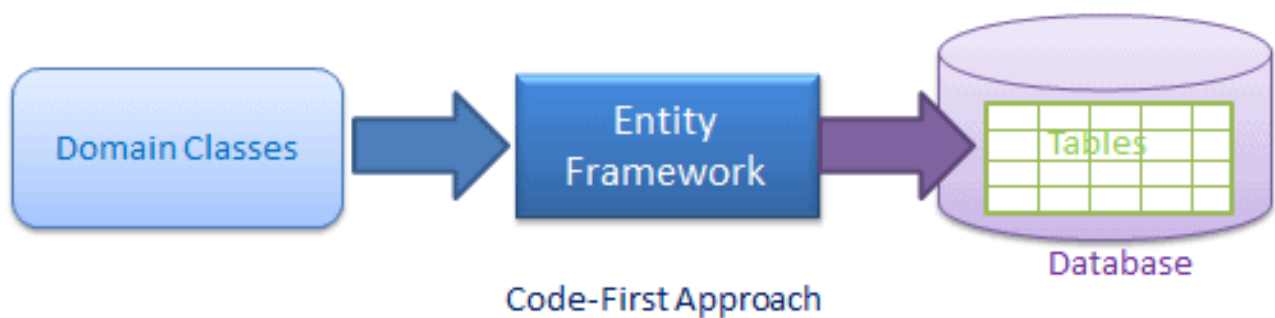


Рисунок 2.4 – Структура роботи CodeFirst

У базі даних авторизації зберігається інформація, яка необхідна для авторизації користувачів, їхні права та ролі, `customerId`.

У базі даних користувачів додатку зберігаються персональні дані користувачів, дані їхніх поїздок (на основі цих даних формулюється статистика), дані маршрутів та інші дані представлені у додатку А.

З додатка А видно, що таблиця «CustomerAccount» має зв'язок один до багатьох з таблицями: CustomerActivity; CustomerRecords; CustomerAppSettings; CustomerDevices; Trips. Таблиця «CustomerRecords» має зв'язок один до одного з таблицею Activity також зв'язок один до багатьох з таблицею Pulse. Таблиця «CustomerActivity» має зв'язок один до одного з таблицею «Activity». Таблиця RentPoints не має зв'язків з іншими таблицями.

### 2.3.2 Локальна база даних

У проєкті передбачена локальна база даних. База даних буде розроблена з використанням не реляційної бази даних MongoDB.

MongoDB – це крос-платформна програма для роботи з документами, орієнтована на джерела. Класифікований як програма бази даних NoSQL, MongoDB використовує документи, подібні до JSON, з обов'язковими схемами. MongoDB розроблено компанією MongoDB Inc. та ліцензовано під загальнодоступною ліцензією на стороні сервера (SSPL) [29].

У локальній базі даних будуть зберігатися проміжні дані поїздок користувачів, для подальшої синхронізації з сервером при наявності мережі інтернет, це зроблено

для того, щоб заподіяти нагромадженню даних на телефоні, та для того щоб програма не розросталася у дуже великі об'єми пам'яті [30].

## 2.4 Консолідація баз даних

В проєкті будуть використовувати сторонні open source бази даних, для пошуку інформації та об'єктів. Проєкт буде містити на стороні сервера консолідацію даних.

Консолідація даних – це процес пошуку та об'єднання даних з різних джерел в одну єдину базу даних [31].

У проєкті консолідація буде відбуватися для пошуку найближчих точок велопрокату, пошук буде відбуватися у базах даних Google Maps Places та у OpenStreetMaps Places, також дані будуть братися з бази даних додатку, в якій зберігаються магазини-партнери, які надають послуги велопрокату.

Консолідація даних буде відбуватися програмним шляхом, за допомогою ключів API сторонніх джерел даних. Для того щоб дані відображалися без дублікатів будемо фільтрувати дані за назвою компанії, видаляючи дублікати даних. Результат пошуку велопрокату буде сортувати від найменшої до найбільшої відстані розташування. Схему консолідації даних проєкту зображено на рисунку 2.5.

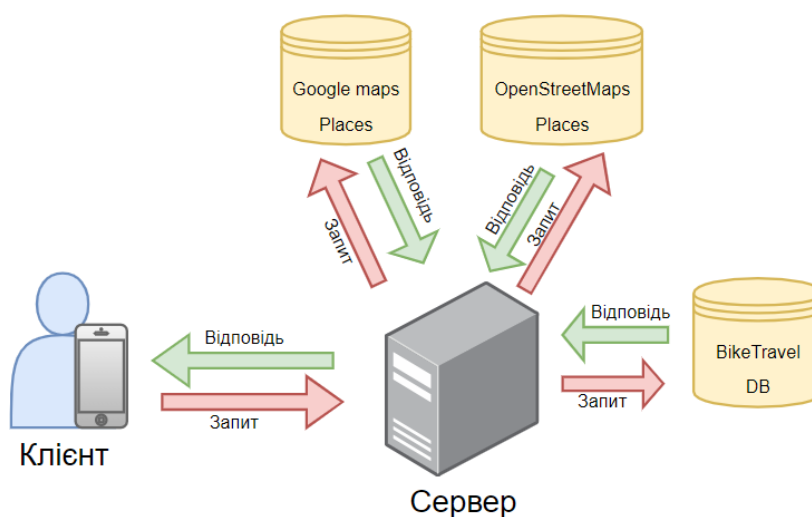


Рисунок 2.5 – Консолідація даних

Клієнт надсилає запит найближчого вело прокату в радіусі 10 км на сервер. Сервер опрацьовує запит, надсилає запити до джерел даних (Google Maps Places, OpenStreetMaps та BikeTravel базу даних). Після отримання відповіді з усіх джерел даних, сервер опрацьовує дані та формулює результат видаляючи дублікати даних, перевіряє інформацію на валідність. Після формулювання результату, відправляє його клієнту, який робив запит.

## **2.5 Комунікація між сервісами та клієнтами**

В проєкті комунікація між клієнтами та сервісами буде реалізовуватися за допомогою потужного інструменту SignalR.

SignalR – це безкоштовна бібліотека програмного забезпечення з відкритим кодом для Microsoft ASP.NET, яка дозволяє коду сервера надсилати асинхронні сповіщення веб-програмам на стороні клієнта. Бібліотека містить компоненти JavaScript на стороні сервера та клієнта [32].

SignalR можна умовно розділити на 2 частини:

- перша частина – поверхнева, реалізує в собі зручний інтерфейс для написання коду, який використовується в контролерах на стороні сервера, та в класах оброблювачів повідомлень на стороні клієнта.

- друга частина – внутрішня, яка містить в собі реалізацію процесів, які працюють безпосередньо з протоколами передачі та захисту даних.

Можливості SignalR дуже великі, його можна швидко конфігурувати з `identityserver` та отримати необхідну аутентифікацію, або з легкістю розбивати користувачів на певні класи та підкласи, що значно спростить реалізацію. На рисунку 2.6 наведено основний принци роботи SignalR [33].

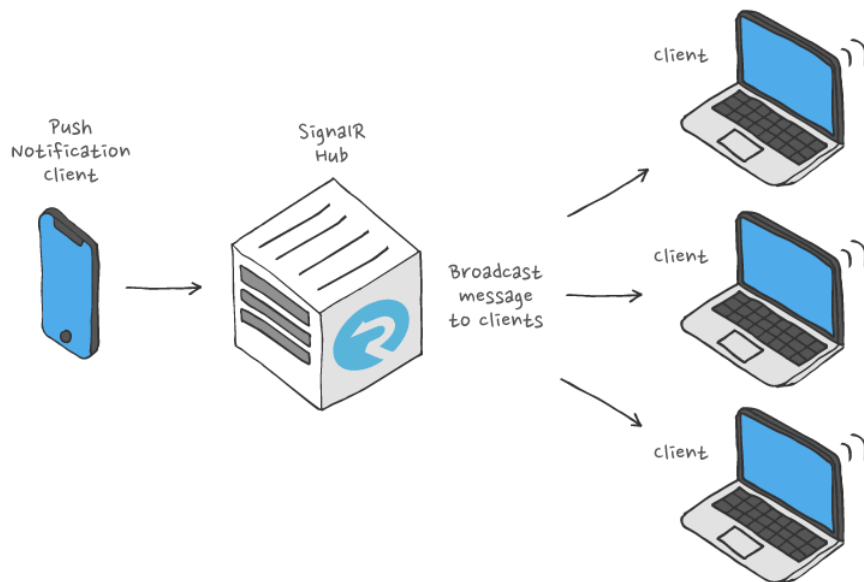


Рисунок 2.6 – Принцип роботи SignalR

На рисунку 2.6 зображено надсилання даних з одного клієнта на SignalR Hub, який знаходиться на стороні сервера. SignalR Hub опрацьовує запит та надсилає оновлені дані всім клієнтам, які підписані на оновлення даних.

Для реалізації SignalR у проекті потрібно налаштувати в стартовий клас сервера. Налаштування в стартовому класі наведено у лістингу 2.5.

Лістинг 2.5 – Конфігурація SignalR в стартовому класі на стороні сервера

```
public void ConfigureServices(IServiceCollection services)
{
    ...
    services.AddSignalR();
    ...
}
...
public void Configure(IApplicationBuilder app, IWebHostEnvironment env,
    ServerDbContext dbContext)
{
    ...
    app.UseEndpoints(endpoints =>
    {
        endpoints.MapDefaultControllerRoute();
        endpoints.MapHub<UpdateHub>("/update-hub");
    });
    ...
}
```

Також необхідно створити UpdateHub, завдяки якому будуть надсилатися оновлення даних клієнтам. У лістингу 2.6 наведено реалізацію UpdateHub.

### Лістинг 2.6 – Реалізація UpdateHub

```
[Authorize]
public class UpdatesHub : Hub
{
    private readonly IRepository _repository;
    public UpdatesHub(IRepository repository)
    {
        _repository = repository;
    }
    public async override Task OnConnectedAsync()
    {
        var user = await
_repository.GetUserById(Guid.Parse(Context.User.Claims.First(p => p.Type ==
"Id").Value));

        await Groups.AddToGroupAsync(Context.ConnectionId,
user.CustomerCompanyId.ToString());
    }
}
```

Оновлення даних будуть надсилатися через функцію Client.SendAsync(), приклад використання наведено у лістингу 2.7.

### Лістинг 2.7 – Надсилання оновлення статистичних даних клієнту

```
...
await _hubContext.Clients.Client(id).SendAsync("UpdateStatistics", data);
...
```

де перший параметр функції – назва методу, другий параметр – дані які надсилаємо.

Для отримання оновлення даних на стороні клієнта повинно бути реалізовано підписку на оновлення даних, та обробку відповідних повідомлень. Реалізацію підписки та хендлінгу повідомлень наведено у лістингу 2.8.



Лістинг 2.8 – Реалізація підписки та хендлінгу повідомлень на стороні клієнта

```
private async Task ConnectionHub()
{
    Connection = new HubConnectionBuilder()
        .WithUrl(_BaseUrl + "/update-hub", option => {
            option.AccessTokenProvider = () =>
Task.FromResult(User.jwtToken);
        })
        .WithAutomaticReconnect()
        .Build();

    Connection.Closed += async (error) =>
    {
        await Task.Delay(new Random().Next(0, 5) * 1000);
        await Connection.StartAsync();
    };

    Connection.On<StatisticsData>("UpdateStatistics", (data) =>
    {
        StatisticsUpdated.Invoke(data);
    });
}
...
```

На цьому етапі можна одержати повідомлення від сервера на стороні клієнта. Можна вважати, що реалізація комунікації клієнта та сервера завершено. Для подальшого можна її розширити, добавивши необхідні real-time функції.

## 2.6 Мобільні клієнти

### 2.6.1 Інструменти розробки

Мобільні додатки будуть розроблятися для популярних ОС, такі як Android та iOS [34].

Для ОС Android додаток буде розроблятися в інтегрованому середовищі Android studio, на мові програмування Kotlin [35].

Для ОС iOS додаток буде розроблятися в середовищі розробки XCode, на мові програмування Swift [36].

### 2.6.2 Опис екранів додатку

*Логотип додатку.* На логотипі додатку зображено велосипед білим кольором на зеленому тлі квадратної форми з заокругленими кутами. Логотип додатку зображено на рисунку 2.7.



Рисунок 2.7 – Логотип додатку

Велосипед на логотипі, вказує на призначення додатку – для велосипедистів, зелений колір тла символізує приязність додатку для заохочення до здорового способу життя.

На рисунку 2.8 зображено екран заставки (splash screen).

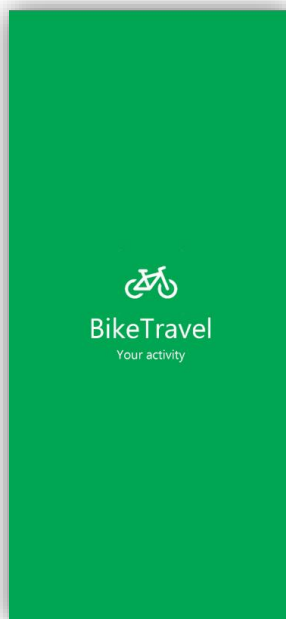


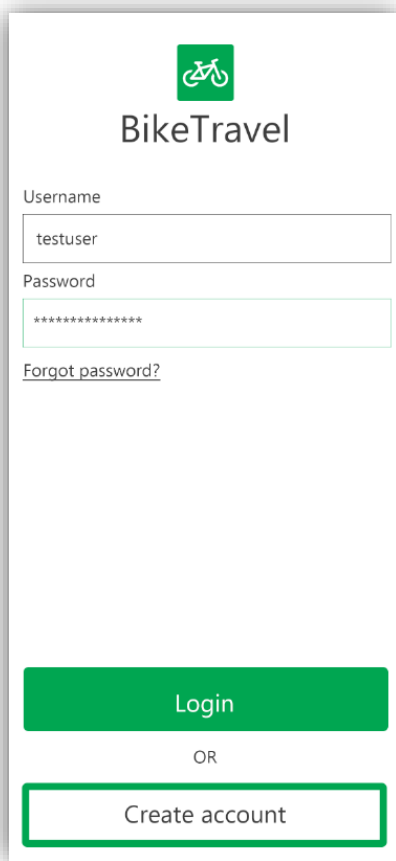
Рисунок 2.8 – Splash screen

Splash screen відображається щоразу, коли користувач відкриває програму, цей скрін слугує для підтягування конфігурацій, оновлень програми перед тим, як користувач побачить контент застосунку.

*Екрани входу, реєстрації та відновлення паролю.* Користувач матиме можливість самостійно створити свій аккаунт, якщо його не існує.

У випадку якщо користувач забув пароль для входу у додаток, він зможе його відновити натиснувши на посилання «Forgot password?», де далі необхідно буде ввести свою електронну пошту та натиснути кнопку «Next», після чого на пошту користувача прийде електронний лист з кодом підтвердження. Натиснувши на посилання у листі, користувачу відкриється вікно у якому необхідно ввести код підтвердження та новий пароль.

На рисунку 2.9 зображено екран входу (Login screen)



The image shows a mobile application login screen for 'BikeTravel'. At the top center is a green square icon with a white bicycle symbol. Below the icon is the text 'BikeTravel'. Underneath, there are two input fields: 'Username' with the text 'testuser' and 'Password' with '\*\*\*\*\*'. Below the password field is a link that says 'Forgot password?'. At the bottom of the screen, there is a solid green button labeled 'Login', the text 'OR' in the center, and a button with a green border labeled 'Create account'.

Рисунок 2.9 – Login screen

Login screen містить у собі перевірку на валідність паролю, у випадку коли користувач введе невірний логін або пароль відобразиться повідомлення «Wrong

Username or password». Коли користувач не ввів одне з полів відобразиться повідомлення «Username is empty» або «Password is empty». У випадку коли користувач ввів не валідні дані відобразиться повідомлення «Invalid username» або «Invalid password».

*Вкладки програми.* Після введення вірних даних для входу (login and password) користувач успішно увійде до програми.

В програмі реалізовано чотири вкладки:

- My Activity;
- Trips;
- History;
- More.

У вкладці «My Activity» відображаються дані в реальному часі. Користувач на цій вкладці має можливість записувати дані руху, як окремий фрагмент їзди, так і переглядати дані в реальному часі в момент руху.

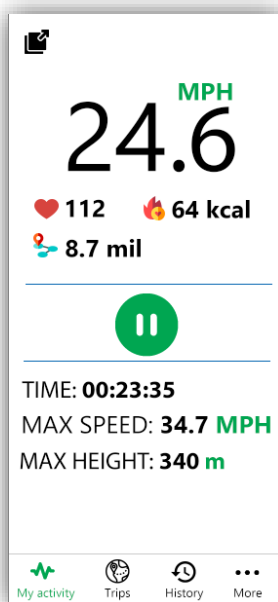


Рисунок 2.10 – My activity tab – run record is started

Вкладна «My Activity» зображена на рисунках 2.10 та 2.11.

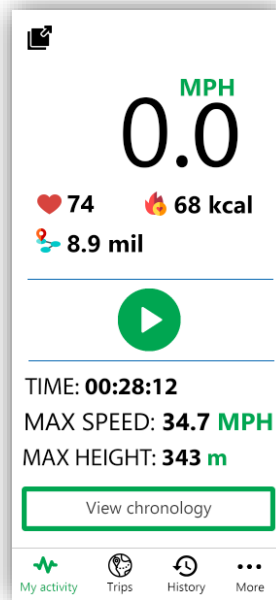


Рисунок 2.11 – My activity tab – run record is stopped

My activity tab реалізує в собі відображення даних в режимі реального часу тобто в русі. Відображається швидкість користувача, кількість серцевих скорочень за хв. – ця інформація береться з синхронізованих смарт годинників, відображаються спалені калорії шляхом калькуляції даних, а також відображається відстань, яку проїхав користувач.

Натиснувши на кнопку «Play», розпочнеться запис фрагменту їзди в реальному часі, буде обчислюватися окремо відстань, записуватиметься хронологія маршруту, максимальна швидкість, висота підйому, зміни серцевих скорочень, час.

При натисканні на кнопку «Stop», зупиниться запис фрагменту їзди, запис буде збережено у локальній базі даних та буде надіслано на сервер після встановлення з'єднання з інтернетом. Також появиться кнопка «View chronology» яка перенаправить користувача на детальний огляд фрагменту запису.

При натисканні на кнопку «Full screen» відкривається додаток в повноекранному режимі. Повноекранний режим наведено на рисунку 2.12.

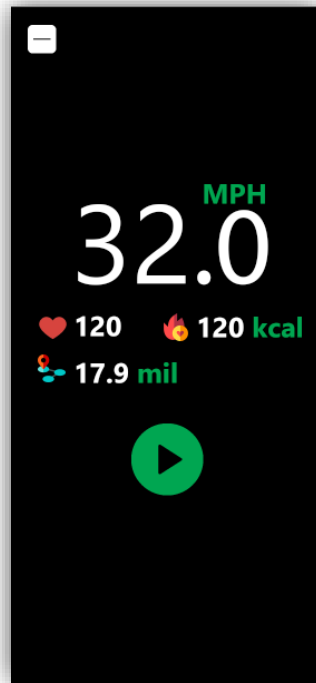


Рисунок 2.12 – Full screen mode

Повноекранний режим, містить всю необхідну інформацію у русі:

- швидкість руху;
- пульс;
- спалені калорії;
- відстань. Також повноекранний режим містить функції запису фрагменту їзди та кнопку «Exit full screen mode».

Чорний колір екрану обрано не випадково, дослідження довели, що акумулятор смартфона при темних кольорах розряджається повільніше. Також темний колір забезпечує зручність перегляду інформації під час руху.

При натисканні на кнопку «Exit full screen mode», користувач повернеться назад до звичайного режиму роботи додатку.

На вкладці «Trips» користувач може планувати поїздки, створювати власні, ділитися ними з друзями через соціальні мережі. Та вибирати вже існуючі веломаршрути. Знімок вкладки Trips наведено на рисунку 2.13.

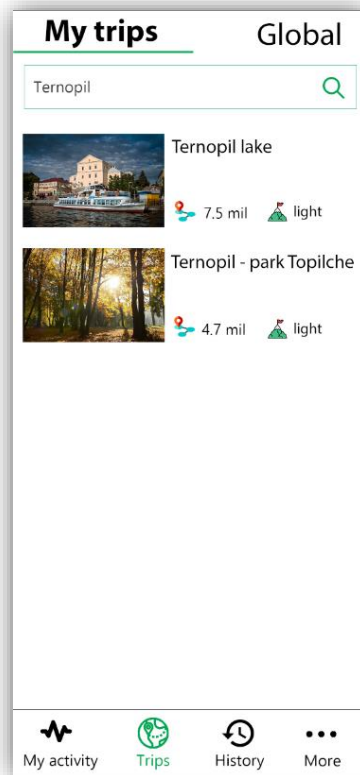


Рисунок 2.13 – Trips tab – My trips

На вкладці Trips у верхній частині екрану розташовані навігація для двох внутрішніх вкладок «My trips» та «Global».

На вкладці My trips користувачу відображаються всі обрані маршрути користувача, які він додав з вкладки Global.

Обравши маршрут на вкладці My trips користувач має змогу свайпнути (провести пальцем) ліворуч та праворуч, при свайпі ліворуч користувач може редагувати маршрут або поділитися з друзями, при свайпі праворуч користувач може видалити маршрут.

На вкладці My trips користувачу відображаються всі маршрути, які він додає в обрані з вкладки Global.

На вкладці Global користувачу відображаються загальнодоступні маршрути, також реалізована можливість створення власних маршрутів.

Для створення власних маршрутів необхідно натиснути на кнопку «Create new trips», після чого відобразиться екран Creating trips, у якому необхідно буде заповнити інформацію про маршрут та натиснути на кнопку «Verify my trips». Після натиснення

кнопки «Verify my trips», маршрут буде відправлено на перевірку, у разі успішної перевірки, користувач отримує системне сповіщення у вигляді push notification.

На вкладках «My trips» та «Global» реалізовано фільтри, де користувач має можливість прискорити пошук маршруту ввівши додаткові критерії пошуку. Більш розширений фільтр реалізовано на вкладці Global він містить:

- країну;
- регіон;
- категорію;
- пошук по назві.

Обравши маршрут на вкладці Global, користувач має можливість свайпнути ліворуч та праворуч, при свайпі ліворуч користувач може запропонувати зміни в маршруті, або поділитися маршрутом з друзями через соціальні мережі, при свайпі праворуч користувач зможе додати до своїх маршрут.

Натиснувши на маршрут, відкриється екран з детальною інформацією про маршрут. Екран маршруту зображено на рисунку 2.14.

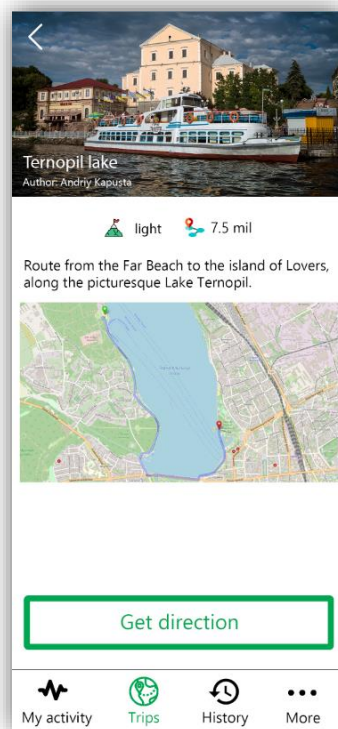


Рисунок 2.14 – Екран маршруту «Ternopil lake»



При натисканні на кнопку «Get direction» відкриється маршрут в запропонованому середовищі Maps.

На вкладці «History» користувач матиме можливість переглянути історію своєї активності, зробити аналітику активності та відображати її в різних діаграмах, також має можливість створювати власні кастомні діаграми. Скрін вкладки History наведено на рисунку 2.15.

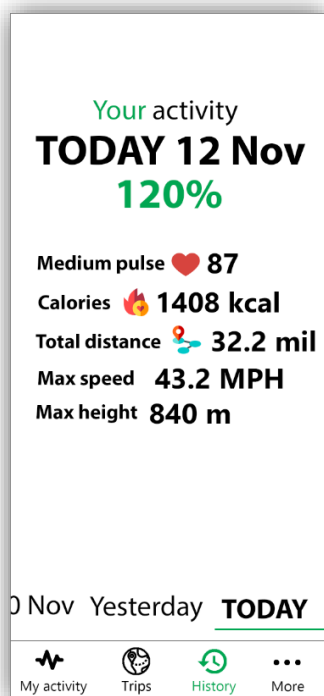


Рисунок 2.15 – Вкладка History

Користувач на вкладці History має можливість переглядати детальну інформацію по датах. Також має можливість свайпнути до низу. При свайпі до низу відобразиться увесь список всіх збережених записів поїздок за обраною датою та конфігурації створення кастомних діаграм.

На вкладці «More» розташовані наступні пункти:

- My account;
- Search bike rent point;
- Settings;
- Terms of use. Знімок вкладки «More» наведено на рисунку 2.16.

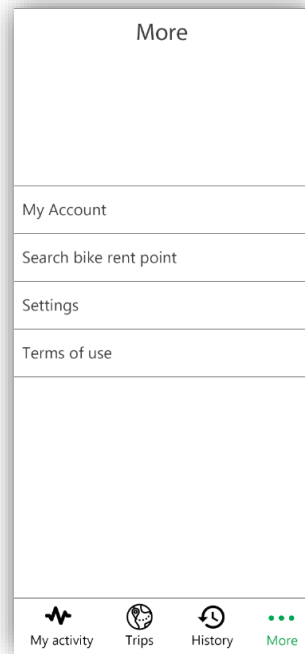


Рисунок 2.16 – Вкладка More

Натиснувши на «My account» пункт меню відкривається екран My account. Знімок екрану «My account» зображено на рисунку 2.17.

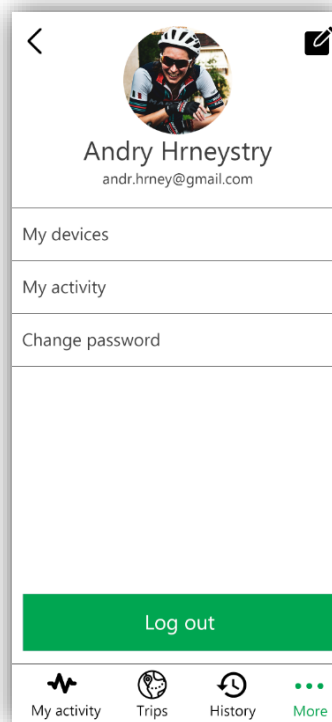


Рисунок 2.17 – «My account» screen

Натиснувши на кнопку «Edit» на екрані My account для користувача відобразиться екран редагування інформації, у якому користувач може змінити: ім'я, прізвище, номер телефону, емейл, username, аватар.

Натиснувши на пункт «My devices» на екрані My account, відобразиться екран «My devices» у якому користувач зможе підключити смарт годинник до програми.

Натиснувши на пункт «My activity» на екрані My account, відкриється екран у якому відображається вся інформація активності користувача про створення маршрутів та інше.

У екрані My account натиснувши на пункт «Change password», відкривається екран у якому користувач може змінити свій пароль.

При натисканні на пункт «Search bike rent point» на екрані More, відкривається екран пошуку найближчих місць велопрокату. На рисунку 2.18 зображено екран пошуку місць велопрокату.

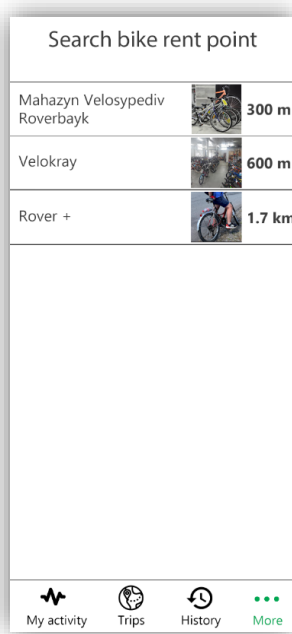


Рисунок 2.18 – Екран пошуку найближчих точок велопрокату

Пошук найближчих місць велопрокату здійснюється за допомогою геолокації, сортування результату пошуку відбувається від найменшої відстані розташування до найбільшої. При виборі елемента, відкриється інформація на Google Maps.

На екрані More натиснувши пункт меню «Settings» відкривається екран користувач матиме можливість змінити налаштування:

- увімкнути або вимкнути сповіщення;
- змінити умови використання та надсилання даних;
- змінити одиницю виміру даних
- змінити мову додатку.

Натиснувши на пункт меню «Terms of use» на екрані More відкриється PDF документ у браузері з умовами прийняття.

## **2.7 Висновки до другого розділу**

У даному розділі було складено завдання реалізації клієнт-серверного застосунку, а також було обрано архітектуру проекту – мікросервісна архітектура. Було описано реалізацію проекту, а саме:

- identity server;
- data storage service;
- databases;
- консолідація даних;
- notification service;
- SignalR – real-time communications technology
- екрани мобільних додатків.

Для identity server було наведено приклад розгортання серверу, налаштування в стартовому класі.

Для data storage service було наведено приклад налаштування Open API для поліпшення документації реалізованих методів сервера.

Для databases було розглянуто умови розгортання баз даних на стороні сервера, а також було описано зв'язки баз даних на стороні сервера.

Для SignalR – real-time communications technology було наведено приклад налаштування в стартовому класі та приклад налаштування на стороні клієнта .NET.

## **3 ЗАСТОСУВАННЯ ЗАСТОСУНКУ ТА ОПИС РЕАЛІЗОВАНИХ ФУНКЦІЙ**

### **3.1 Опис реалізованих функцій**

В результаті проектування, розробки було розроблено вело-додаток «VikeTravel», у якому реалізовано:

- функцію реєстрації нових користувачів;
  - функції авторизації та скидання паролю;
  - функції перегляду даних в реальному часі з можливістю синхронізації зі смарт годинником;
  - функції перегляду статистики поїздок;
  - функції пошуку найближчих місць велопрокату;
- функції перегляду веломаршрутів з можливістю створення власних.

### **3.2 Тестування**

Не менш важливим етапом розробки програмного забезпечення є тестування. Тестування як процес будівництва є безперервним та безкінечним, метою тестування є відшукання критичних місць програми, знаходження дефектів до процесу публікування програмного продукту (релізу). Правильно протестований програмний продукт забезпечує надійність, безпеку та високу продуктивність, що в подальшому призводить до економії часу, економічної ефективності та більшої аудиторії задоволених клієнтів [37].

Тестування програмного забезпечення – це перевірка відповідності програмного продукту, чи він відповідає поставленим вимогам та не містить дефектів. Процес тестування передбачає тестування компонентів та інтерфейсів додатку, проходження сценаріїв, створення автоматизованих скриптів, які значно полегшують регресивне тестування (повторне тестування) [38].

Тестування базується на написання тестової документації, такої як: тест кейси, баг репорти, тест план тощо.

Тест кейс – це артефакт, що описує сукупність кроків, конкретних умов і параметрів, необхідних для перевірки реалізації функції, що тестується або її частини.

В тест кейсі містяться такі основні поля:

- ID – номер тестового випадку;
- Title – заголовок, який вказує що буде тестуватися;
- Preconditions – передумова виконання тест кейсу;
- Steps – кроки виконання сценарію;
- Input – вхідні дані для виконання сценарію;
- Expected result – очікуваний результат виконання сценарію;
- Actual result (при необхідності) – фактичний результат виконання сценарію [39].

Баг репорт – документ, який повідомляє про знайдений дефект програми. В баг репорті містяться такі поля:

- Summary – короткий опис виявленого дефекту;
- Product – назва програми, у якій виявлено дефект;
- Version – версія програми;
- Build – збірка програми;
- Device – девайс на якому виникла помилка (iPhone SE);
- Details – деталі помилки, зазвичай сюди додають кроки відтворення помилки, та візуальні показники помилки (скріни роботи програми);
- Reporter – Вказується хто повідомив про помилку.

Тест план – це документ якому описується обов’язки тестувальника та план тестування. Тест план, як правило, створюється на початку циклу розробки програмного забезпечення (початок спрінта, початок фази), в документі планується, який інтерфейс, функціонал програмного продукту буде тестуватися в певному проміжку часу [40].

В процесі тестування було написано тестову документацію, а саме тест кейси. Тест кейс входу користувача наведено у таблиці 3.1.

Таблиця 3.1 – Test case Login

ID	Steps	Input	Expected result
<b>Login when user enters incorrect creds</b>			
01	1. Enter invalid Username 2. Tap on the password field 3. Click on the Login button	<b>Preconditions:</b> Login screen is opened	“Invalid username” label is shown
02	1. Fill incorrect creds 2. Click on the Login button		“Username or password is incorrect” label is shown
03	1. Fill password field 2. Click on the Login button	Username field is empty	“username is empty” label is shown
04	1. Fill Username field 2. Click on the Login button	Password field is empty	“Password is empty” label is shown
05	1. Enter invalid Password 2. Tap on the password field 3. Click on the Login button		“Invalid password” label is shown
<b>Login workflow</b>			
01	1. Fill correct creds 2. Click on the Login button		User logged to the app, My activity screen is shown

У тест кейсі входу виконується перевірка:

- звичайної роботи функції входу;
- коли не заповнено емейл користувача;
- коли не введено пароль користувача;
- коли введено невірний пароль або логін.
- коли введено невірно username.
- коли введено невірно password.

Тест кейс реєстрації користувача наведено у таблиці 3.2.

Таблиця 3.2 – Test case Registration

ID	Steps	Input	Expected result
1	2	3	4
<b>Registration workflow</b>			
01	1. Click on the “Create account” button	<b>Preconditions:</b> Login screen is opened	Registration screen is shown
02	2. Fill correct data 3. Click on the Next button		“Confirmation registration” screen is shown
03	4. Confirm registration		“Account successful created” notification is displayed, login screen is shown
04	5. Fill correct creds 6. Click on the Login button		User logged to the app, My activity screen is shown

1	2	3	4
<b>Registration when user enters incorrect data</b>			
01	1. Click on the Next button	<b>Preconditions:</b> Registration screen is shown - First name is empty - Last name is empty - Age not selected - Email I empty - Password is empty - wage is empty	“First name is required” label is shown “Last name is required” label is shown “Age is required” label is shown “Email is required” label is shown “Password” is required” label is shown “Wage is required” label is shown
02	1. Fill fields 2. Click on the Next button	Email already existing in DB	“User with email already registered” label is shown
03	1. Fill fields 2. Click on the Next button	Easy password	“Please enter a more secure password” label is shown
04	1. Fill field 2. Click on the next button 3. Fill incorrect confirmation coded 4. Click on the Confirm button	Invalid confirmation code	“Please enter correct confirmation code” label is shown
05	1. Fill field 2. Click on the next button 3. Click on the Confirm button	Confirmation code is empty	“Confirmation code is required” label is shown

У тест кейсі реєстрації перевіряється:

- звичайна робота реєстрації;
- коли не заповнено всі обов’язкові поля:
  - first name;
  - lastName;
  - age;
  - email;
  - password;
  - wage;
- коли користувач з вказаним email вже зареєстрований;
- коли введено не заповнено код підтвердження;
- коли введено не код підтвердження.



Тест кейс активності користувача в реальному часі наведено у таблиці 3.3.

Таблиця 3.3 – Test case Real-time user activity

ID	Steps	Input	Expected result
1	2	3	4
<b>My activity workflow</b>			
01	Ride bike for one mile	<b>Preconditions:</b> - My activity screen opened - Smart watch is synchronized to the app	Miles added to mile count Actual pulse displayed Max height displayed Updated calories displayed Max speed displayed
02	Ride bike		Actual speed displayed Actual pulse displayed
03	Ride bike	Smart watch is not connected	Pulse N/A is displayed
04	Click on the Full mode button		My activity opened on the full screen
05	Click on the record button	-	Record is started, timer starts tick
06	Click on the stop record button		Record is stopped, timer stops tick, last record information is displayed

У тест кейс активності користувача в реальному часі перевіряється:

- відображення даних в реальному часі;
- відображення даних, коли не синхронізовано годинник.

Тест кейс поїздки наведено у таблиці 3.4.

Таблиця 3.4 – Test case User trips

ID	Steps	Input	Expected result
1	2	3	4
<b>Trips workflow</b>			
01	Click on the Global tab	<b>Preconditions:</b> - Trips tab opened	Global trips opened
02	Select trips Click to add to my trips button		Trip added to My trips
03	Click on the trips		Trips details screen opened
04	Click on the Get direction button		Trip routing opened on the Map app
05	Click on the Global tab Click on the Create new Routing	<b>Preconditions:</b> Trips tab opened	Trip creation screen opened
06	Fill trip title Fill descriptions Add trip image Add trip routing point Add trip category Click on the Verify my trip		Trip after one minute successful added to all global trips

У тест кейсі поїздки, робиться перевірка:

- відображення вкладки Global;
- відображення екрану Trips details;
- роботи кнопки Get direction;
- роботи функції додавання поїзди до обраних;
- роботи функції створення поїздки;
- роботи функції видалення поїздки;

Вище наведені тест кейси дозволили частково покрити функціонал програми, та відшукати не відповідності функціоналу застосунку до етапу опублікування.

### **3.3 Модифікація оновлень та розповсюдження додатку**

Додаток планується розмістити у двох популярних маркетах додатків для операційних системах Android – Play Market та iOS – AppStore [41].

Для розміщення додатків необхідно підключити програму розробника, для iOS це коштує 100\$, для Android – 25\$. Також необхідно пройти верифікацію та перевірку додатку на вимоги, які не перешкоджатимуть політиці Apple та Google [42].

Маркети Play Market та App Store підтримують налаштування додатку під певний регіони та країни, також дозволяють вставляти різні позиції оновлень для певних країн [43].

Додаток «VikeTravel» планується розповсюджуватися у першу чергу для країн у яких популярний велотранспорт – це Скандинавські країни та країни західної Європи.

Поліпшення та реалізація нових функцій додатку будуть відбуватися на основі аналітики користування додатком. Дані користування будуть збиратися з користувачів, які приймуть умови використання, а саме: «Надання даних для обробки з метою покращення функцій».

### 3.4 Висновки до третього розділу

У третьому розділі було перелічено реалізовані функції мобільного застосунку, а саме:

- функцію реєстрації нових користувачів;
- функції авторизації та скидання паролю;
- функції перегляду даних в реальному часі з можливістю синхронізації зі

смарт годинником;

- функції перегляду статистики поїздок;
- функції пошуку найближчих місць велопрокату;

функції перегляду веломаршрутів з можливістю створення власних. Також у даному розділі створено тестову документацію – тест кейси. Також описано план розміщення застосунків на етапі опублікування та поширення оновлень.

## 4 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

### 4.1 Організація робочого місця для користувача ПК

Темою дипломної роботи є розробка клієнт-серверного застосунку для систематизації та синхронізації даних геопозиціонування велосипедистів. В процесі розробки клієнт-серверного застосунку важливо забезпечити комфортне та безпечне робоче місце.

При розміщенні робочих місць з персональними комп'ютерами відстань між робочими столами з відеомоніторами (у напрямі тилу поверхні одного відеомонітора і екрану іншого відеомонітора) повинно бути не менше 2,0 м, а відстань між бічними поверхнями відеомоніторів – не менше 1,2 м [44].

Робочі місця з персональними комп'ютерами в приміщеннях з джерелами шкідливих виробничих факторів розміщуються в ізольованих кабінах з неорганізованим повітрообміном. При виконанні творчої роботи, що вимагає значного розумового напруження або високої концентрації уваги, робочі місця з персональними комп'ютерами рекомендується ізолювати один від одного перегородками висотою 1,5-2,0 м [44].

Екран відеомонітора повинен знаходитися від очей користувача на відстані 600-700 мм, але не ближче 500 мм з урахуванням розмірів алфавітно-цифрових знаків і символів [44].

Конструкція робочого столу повинна забезпечувати оптимальне розміщення на робочій поверхні використовуваного обладнання з урахуванням його кількості і конструктивних особливостей, характеру виконуваної роботи. При цьому допускається використання робочих столів різних конструкцій, що відповідають сучасним вимогам ергономіки. Поверхня робочого столу повинна мати коефіцієнт відбиття 0,5-0,7 [44].

Висота робочої поверхні столу для дорослих користувачів повинна регулюватися в межах 680-800 мм; при відсутності такої можливості висота робочої поверхні столу повинна складати 725 мм. Модульними розмірами робочої поверхні

столу для ПК, на підставі яких повинні розраховуватися конструктивні розміри, слід вважати: ширину - 800, 1000, 1200 і 1400 мм, глибину - 800 і 1000 мм при нерегульованій його висоті, рівній 725 мм [44].

Робочий стіл повинен мати простір для ніг висотою не менше 600 мм, шириною – не менше 500 мм, глибиною на рівні колін - не менше 450 мм, на рівні витягнутої ноги - не менше 650 мм [44].

Конструкція робочого стільця (крісла) повинна забезпечувати підтримку раціональної робочої пози під час роботи на персональному комп'ютері, дозволяти змінювати позу з метою зниження статичного напруження м'язів шийно плечевий області і спини для попередження розвитку втоми. Тип робочого стільця (крісла) слід вибирати з урахуванням зростання користувача, характеру та тривалості роботи з ПК. Робочий стілець (крісло) повинен бути підйомно-поворотним, регульованим по висоті і куту нахилу сидіння і спинки, а також відстані спинки від переднього краю сидіння, при цьому регулювання кожного параметра повинні бути незалежною, легко здійснюваною мати надійну фіксацію. Поверхня сидіння, спинки та інших елементів стільця (крісла) повинна бути напів м'якої, з нековзним, слабо електризується і повітропроникним покриттям, що забезпечує легке очищення від забруднень.

Клавіатуру слід розташовувати на поверхні столу на відстані 100-300 мм від краю, зверненого до користувача, або на спеціальній, регульованій по висоті робочій поверхні, відокремленої від основної стільниці [44].

Раціональний режим праці та відпочинку передбачає дотримання певної тривалості безперервної роботи на персональному комп'ютері і перерв, регламентованих з урахуванням тривалості робочої зміни, виду трудової діяльності.

Виділяють три види робіт, виконуваних на ПК: група А - робота з зчитування інформації з екрана ВДТ з попереднім запитом; група Б - робота з введення інформації; група В - творча робота в режимі діалогу з ПК.

При виконанні протягом робочої зміни робіт, що відносяться до різних видів трудової діяльності, за основну роботу з ПК слід сприймати таку, що займає не менше 50% часу протягом робочої зміни або робочого дня.

Для видів трудової діяльності встановлюється три категорії тяжкості і напруженості роботи з ПК, які визначаються:

- для групи А по сумарному числу прочитуються знаків за робочу зміну, але не більше 60 тис. Знаків за зміну;
- для групи Б по сумарному числу зчитувальних чи впроваджуються знаків за робочу зміну, але не більше 40 тис. Знаків за зміну;
- для групи В по сумарному часу безпосередньої роботи з ПК за робочу зміну, але не більше 6 годин за зміну.

Для попередження передчасної втомлюваності користувачів ПК рекомендується організувати робочу зміну шляхом чергування робіт з використанням персонального комп'ютера і без нього.

При виникненні у працюючих з ПК зорового дискомфорту та інших несприятливих суб'єктивних відчуттів, незважаючи на дотримання санітарно-гігієнічних і ергономічних вимог, рекомендується застосовувати індивідуальний підхід з обмеженням часу роботи з ПК [44].

У випадках, коли характер роботи вимагає постійної взаємодії з ВДТ (набір текстів або введення даних тощо) з напругою уваги та зосередженості, при виключенні можливості періодичного перемикання на інші види трудової діяльності, не пов'язані з ПК, рекомендується організація перерв на 10 -15 хвилин через кожні 45-60 хвилин роботи. Тривалість безперервної роботи з ВДТ без регламентованого перерви не повинна перевищувати однієї години [44].

Під час регламентованих перерв з метою зниження нервово-емоційного напруження, стомлення зорового аналізатора, усунення впливу гіподинамії та гіпокінезії, запобігання розвитку втоми доцільно виконувати спеціально розроблені комплекси вправ.

Користувачам ПК, виконують роботу з високим рівнем напруженості, показана психологічне розвантаження під час регламентованих перерв і в кінці робочого дня в спеціально обладнаних приміщеннях (кімната для відпочинку, ігрова кімната, спортивна кімната).

## 4.2 Електробезпека для користувачів ПК

Важливим завданням кожного роботодавця є – забезпечення безпечних умов праці. Для користувачів ПК найбільший джерелом небезпеки є електрика.

Ризики, які пов'язані з комп'ютером та ІТ-обладнанням, багато портативних пристроїв, таких як ноутбуки, планшети комп'ютери та мобільні телефони, мають дроти, які часто включаються і вимикаються, коли співробітники переходять з одного робочого місця на інше. Це може означати, що кабель живлення та штекер можуть бути пошкоджені або ослаблені, якщо їх не використовувати належним чином. Швидкий візуальний огляд може допомогти забезпечити безпеку цих пристроїв.

Щоб забезпечити безпеку обладнання, необхідно дотримуватися простих речей:

- заохочувати працівників проводити швидку перевірку обладнання та джерел живлення перед їх використанням;
- проводити регулярні візуальні перевірки - необхідна періодичність буде залежати від того, яке обладнання;
- організувати випробування портативних приладів для будь-якого обладнання класу 1 (будь-якого приладу із заземленням), яке не має подвійної ізоляції;
- зняти з експлуатації будь-яке пошкоджене або несправне обладнання та переконатися, що воно відремонтовано компетентною особою або утилізувати до того моменту отримання травми;
- планувати та вести графік технічного обслуговування з електробезпеки, для забезпечення нагляду справності устаткування, і покращити якість перевірок та тестування пристроїв;
- вести облік перевірок та випробувань електробезпеки під час роботи.
- використовувати лише пристрої та запчастини до них від перевірених, сертифікованих виробників.

Перевірку ПК та візуальний огляд може проводити будь-хто. Крім того, електрику не потрібно проводити перевірку портативного пристрою. Однак особа, яка проводить випробування, повинна бути компетентною, пройти відповідну

підготовку та мати відповідне обладнання для тестування. Вони також повинні знати, як інтерпретувати результати випробувань, щоб підтримувати електробезпеку на роботі.

Що до законів, вимоги безпеки щодо використання комп'ютерної техніки визначають:

– ДСТУ EN 41003:2014 «Обладнання, яке підключають до телекомунікаційних мереж та/або кабельних розподільчих систем. Додаткові вимоги щодо безпеки».

– ДСТУ EN 60335-1:2015 «Прилади побутові та аналогічні електричні. Безпека. Частина 1. Загальні вимоги».

– ДСТУ EN 60950-1:2015 «Обладнання інформаційних технологій. Безпека. Частина 1. Загальні вимоги» (далі — ДСТУ EN 60950-1:2015).

– ДСТУ EN 61140:2015 «Захист проти ураження електричним струмом. Загальні аспекти щодо установок та обладнання» (далі — ДСТУ EN 61140:2015).

– ДСТУ EN 62368-1:2017 «Обладнання аудіо-, відео-, інформаційних та комунікаційних технологій. Частина 1. Вимоги щодо безпеки» [46].

НПАОП 0.00-7.15-18 не вимагає присвоєння кваліфікаційної групи з електробезпеки користувачам комп'ютерної техніки [47].

Порядок допуску до роботи з комп'ютерною технікою визначає Типове положення про порядок проведення навчання і перевірки знань з питань охорони праці (НПАОП 0.00-4.12-05), затверджене наказом Державного комітету України з нагляду за охороною праці від 26 січня 2005 р. № 15, відповідно до п. 3.1 та 3.4 якого передбачене проведення інструктажів та навчання з питань охорони праці, зокрема й електробезпеки [48].

Порядок присвоєння кваліфікаційної групи з електробезпеки на виробництві визначає НПАОП 40.1-1.21-98 Правила безпечної експлуатації електроустановок споживачів, затверджений наказом Державного комітету України по нагляду за охороною праці від 9 січня 1998 р. № 4, що поширюється на працівників, які обслуговують діючі електроустановки споживачів. Під обслуговуванням (технічним) розуміють комплекс робіт з підтримки працездатності обладнання в період його використання, до якого належать роботи з випробування обладнання, пристроїв,



огляд обладнання, підтяжка контактних з'єднань (п. 3.1 розділу III Правил технічної експлуатації електроустановок споживачів, затверджених наказом Міністерства палива та енергетики України від 25 липня 2006 р. № 258).

Під електроустановкою розуміють комплекс взаємопов'язаних машин, апаратів, ліній та допоміжного обладнання (разом з будівлями і приміщеннями, в яких їх встановлено), призначених для виробництва, трансформації, передавання, розподілу електричної енергії і перетворення її в інші види енергії (п. 1.1.3 Правил улаштування електроустановок, затверджених наказом Міністерства енергетики та вугільної промисловості України від 21 липня 2017 р. № 476 [46]).

Сучасний комп'ютер не є електроустановкою, і вимоги ПУЕ та ПТЕЕС можуть бути правомірні тільки для мережі його електроживлення, тобто на саму комп'ютерну техніку не поширюються. Вимоги безпеки електрообладнання комп'ютерної техніки регламентують державні стандарти, зокрема, серії ДСТУ EN 60335 та ДСТУ EN 60950 [49].

Технічні засоби загального (побутового) призначення не повинні використовуватися в умовах підвищеної небезпеки, тож експлуатація сучасної комп'ютерної техніки не належить до робіт підвищеної небезпеки [50].

Порядок використання комп'ютерної техніки на виробництві визначається настановою з експлуатування її заводу-виробника.

Допуск працівників до роботи з комп'ютерною технікою повинен здійснюватися шляхом проведення навчання, до програми якого мають бути включені питання безпеки під час експлуатації комп'ютерної техніки як електротехнічного пристрою.

Розробляти окрему інструкцію з охорони праці під час використання на виробництві комп'ютерної техніки недоцільно, достатньо розробити загальну інструкцію з електробезпеки, яка враховує специфіку експлуатації цього обладнання.

### 4.3 Висновки до четвертого розділу

В даному розділі розглянуто вимоги до облаштування робочого місця, а саме: відстань між робочими столами, бічна сторона – 1.2 м, фронтальна сторона – 2 м;

– розміщення перегородок між місцями роботи облаштованих ПК, висота який повинна містити – 1,5-2,0 м;

– висота робочої поверхні столу для користувачів ПК – 680-800 мм;

– розміщення моніторів;

– конструкція робочого столу;

– простір для ніг, висота – 600 мм, ширина – не менше 500;

– конструкція робочого крісла;

– розміщення периферії комп'ютера.

Також було розглянуто раціональний режим праці, розвантаження працівників після роботи, облаштування офісу та електробезпека користувачів ПК.

## ВИСНОВКИ

В процесі виконання кваліфікаційної роботи магістра була досягнута поставлена мета, а саме розроблено клієнт-серверний застосунок для систематизації та синхронізації даних геопозиціонування велосипедистів.

Виконуючи дану роботу було виконано:

- дослідження доцільності розробки застосунку;
- визначено основні функції додатку;
- визначено архітектуру проєкту;
- реалізовано сервер авторизації;
- реалізовано мікросервіс роботи з даними клієнтів;
- розроблено бази даних за допомогою технології Code-first;
- реалізовано механізм консолідації даних між відкритими базами для пошуку найближчих точок велопрокату;
- реалізовано клієнтську частину (мобільний додаток);
- реалізовано комунікацію між сервером та клієнтами в реальному часі;
- написано тестову документацію;
- протестовано функції застосунку.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Bernhard A. BBC: - The great bicycle boom of 2020 [Електронний ресурс] / Adrienne Bernhard. – 2020. – Режим доступу до ресурсу: <https://www.bbc.com/future/bespoke/made-on-earth/the-great-bicycle-boom-of-2020.html>.
2. Стійкість: до 500 євро за ваучер на пересування [Електронний ресурс] // Уряд Італії: Міністерство сталої інфраструктури та мобільності. – 2020. – Режим доступу до ресурсу: <https://www.mit.gov.it/comunicazione/news/sostenibilita-fino-500-euro-per-buono-mobilita>.
3. Guyduy M. Sporting Goods, Home Fitness, and Cycling Sales Surge in the U.S., Reports The NPD Group [Електронний ресурс] / Marissa Guyduy. – 2020. – Режим доступу до ресурсу: <https://www.npd.com/news/press-releases/2020/sporting-goods-home-fitness-and-cycling-sales-surge-in-the-us-reports-the-npd-group/>.
4. Richter F. COVID-19 Pandemic Fuels Bicycle Boom [Електронний ресурс] / Felix Richter. – 2020. – Режим доступу до ресурсу: <https://www.statista.com/chart/21794/year-over-year-change-in-bicycle-sales-in-the-us/>.
5. Hawkins A. How to keep the bike boom from fizzling out [Електронний ресурс] / Andrew Hawkins. – 2020. – Режим доступу до ресурсу: <https://www.theverge.com/22178543/bike-bicycle-boom-covid-pandemic-2020-sales-cities-infrastructure>.
6. Німців Павло вікторович - Математична модель та програмне забезпечення прогнозу генерування електроенергії вітрогенератором для геолокації. 2020.
7. Cerini D. How many calories does cycling burn? [Електронний ресурс] / Damian Cerini // 2016 – Режим доступу до ресурсу: <https://tourdevines.com.au/blog/how-many-calories-does-cycling-burn/>.
8. Веломаршрут міста Терєбовля [Електронний ресурс] – Режим доступу до ресурсу: <https://tic.terebovlia.info/veloprokat/velomarshruty/>.
9. Додаток Bike map [Електронний ресурс] – Режим доступу до ресурсу: <https://www.bikemap.net/>.

10. Документація додатку Cyclometr [Електронний ресурс] – Режим доступу до ресурсу: <https://cyclometer.com/documentation/#workoutinfo>.

11. Популярність велосипедів [Електронний ресурс] – Режим доступу до ресурсу: <https://northsearegion.eu/bits/news/bits-survey-confirms-the-growing-popularity-of-cycling/>.

12. Cycling - health benefits [Електронний ресурс] – Режим доступу до ресурсу: <https://www.betterhealth.vic.gov.au/health/healthyliving/cycling-health-benefits>.

13. Flusche D. National household travel survey - short trips analysis [Електронний ресурс] / Darren Flusche. – 2010. – Режим доступу до ресурсу: <https://www.bikeleague.org/content/national-household-travel-survey-short-trips-analysis>.

14. Додаток Strava [Електронний ресурс] – Режим доступу до ресурсу: <https://www.strava.com/>.

15. These Are the Happiest Cities in the United States [Електронний ресурс] // National Geographic. – 2017. – Режим доступу до ресурсу: [https://www.nationalgeographic.com/travel/article/happiest-cities-united-states-2017?awc=19533\\_1630439919\\_170711a048e27fd9ea14623f4b124bb9&awc=19533\\_1631081897\\_1c13a2789c7737926636917327235672](https://www.nationalgeographic.com/travel/article/happiest-cities-united-states-2017?awc=19533_1630439919_170711a048e27fd9ea14623f4b124bb9&awc=19533_1631081897_1c13a2789c7737926636917327235672).

16. Pang J. Can Hong Kong Become a Bike Friendly City? [Електронний ресурс] / J. Pang, R. Tang, S. Cai. – 2017. – Режим доступу до ресурсу: <https://jour2106group3.github.io/final-project/>.

17. Гасімов О. Мікросервісна архітектура для початківців. Частина 1 [Електронний ресурс] / Орхан Гасімов // GlobalLogic – Режим доступу до ресурсу: <https://www.globallogic.com/ua/insights/blogs/microservices-architecture-for-beginners-part-one/>.

18. Мікросервіси [Електронний ресурс] // Вікіпедія – Режим доступу до ресурсу: <https://uk.wikipedia.org/wiki/%D0%9C%D1%96%D0%BA%D1%80%D0%BE%D1%81%D0%B5%D1%80%D0%B2%D1%96%D1%81%D0%B8>.

19. IdentityServer4 - Documentation [Електронний ресурс] – Режим доступу до ресурсу: <https://identityserver4.readthedocs.io/en/latest/>.

20. IdentityServer for cloud-native applications [Електронний ресурс] // Microsoft documentation – Режим доступу до ресурсу: [https://docs.microsoft.com/en-us/dotnet/architecture/cloud-native/identity-server?ranMID=46131&ranEAID=a1LgFw09t88&ranSiteID=a1LgFw09t88-wR9HC8YKhGIS0122KA00Qg&epi=a1LgFw09t88-wR9HC8YKhGIS0122KA00Qg&irgwc=1&OCID=AID2200057\\_aff\\_7806\\_1243925&tuid=%28ir\\_\\_c16esrt1j9kftklrkk0sohz3xm2xomn0fjg9hki000%29%287806%29%281243925%29%28a1LgFw09t88-wR9HC8YKhGIS0122KA00Qg%29%28%29&irclickid=\\_c16esrt1j9kftklrkk0sohz3xm2xomn0fjg9hki000](https://docs.microsoft.com/en-us/dotnet/architecture/cloud-native/identity-server?ranMID=46131&ranEAID=a1LgFw09t88&ranSiteID=a1LgFw09t88-wR9HC8YKhGIS0122KA00Qg&epi=a1LgFw09t88-wR9HC8YKhGIS0122KA00Qg&irgwc=1&OCID=AID2200057_aff_7806_1243925&tuid=%28ir__c16esrt1j9kftklrkk0sohz3xm2xomn0fjg9hki000%29%287806%29%281243925%29%28a1LgFw09t88-wR9HC8YKhGIS0122KA00Qg%29%28%29&irclickid=_c16esrt1j9kftklrkk0sohz3xm2xomn0fjg9hki000).

21. Документація по веб-API ASP.NET Core з використанням Swagger (OpenAPI) [Електронний ресурс] // Microsoft – Режим доступу до ресурсу: <https://docs.microsoft.com/ru-ru/aspnet/core/tutorials/web-api-help-pages-using-swagger?view=aspnetcore-5.0>.

22. What is a REST API? [Електронний ресурс] // Red Hat. – 2020. – Режим доступу до ресурсу: <https://www.redhat.com/en/topics/api/what-is-a-rest-api>.

23. OAuth 2.0 - Documentation [Електронний ресурс] – Режим доступу до ресурсу: <https://oauth.net/2/>.

24. Open API [Електронний ресурс] – Режим доступу до ресурсу: [https://www.jetbrains.com/help/objc/openapi.html?source=google&medium=cpc&campaign=14298656398&gclid=Cj0KCQiA7oyNBhDiARIsADtGRZZ10rZLJcmbWrkheBD\\_LY1u7ZLEIovrqDDG5U6vcX2NFvHzCTnsdpAaAhb6EALw\\_wcB](https://www.jetbrains.com/help/objc/openapi.html?source=google&medium=cpc&campaign=14298656398&gclid=Cj0KCQiA7oyNBhDiARIsADtGRZZ10rZLJcmbWrkheBD_LY1u7ZLEIovrqDDG5U6vcX2NFvHzCTnsdpAaAhb6EALw_wcB).

25. MS Azure - Notification hub [Електронний ресурс] – Режим доступу до ресурсу: <https://azure.microsoft.com/en-us/services/notification-hubs/#overview>.

26. Firebase documentation [Електронний ресурс] – Режим доступу до ресурсу: <https://firebase.google.com/docs>.

27. Pattern: Database per service [Електронний ресурс] // Microservice architecture – Режим доступу до ресурсу: <https://microservices.io/patterns/data/database-per-service.html>.

28. EF 6 Code-first [Електронний ресурс] // Entity framework tutorial – Режим доступу до ресурсу: <https://www.entityframeworktutorial.net/code-first/what-is-code-first.aspx>.

29. MongoDB [Електронний ресурс] // Вікіпедія – Режим доступу до ресурсу: <https://en.wikipedia.org/wiki/MongoDB>.

30. MongoDB Realm Sync - The Cloud-Native Database. – С. [https://www.mongodb.com/realm/lp?utm\\_source=google&utm\\_campaign=gs\\_footprint\\_row\\_search\\_nbnon\\_re](https://www.mongodb.com/realm/lp?utm_source=google&utm_campaign=gs_footprint_row_search_nbnon_re).

31. Консолідація [Електронний ресурс] // Вікіпедія – Режим доступу до ресурсу: <https://uk.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BD%D1%81%D0%BE%D0%BB%D1%96%D0%B4%D0%B0%D1%86%D1%96%D1%8F>.

32. SignalR [Електронний ресурс] // Вікіпедія – Режим доступу до ресурсу: <https://en.wikipedia.org/wiki/SignalR>.

33. .Net core SignalR project [Електронний ресурс] – Режим доступу до ресурсу: <https://github.com/cihatsolak/netcore-signalr>.

34. Популярні операційні системи для смартфонів. Мобільні операційні системи: Опис і порівняння [Електронний ресурс] // [passportbdd.ru](https://passportbdd.ru). – 2020. – Режим доступу до ресурсу: <https://passportbdd.ru/uk/browsing-the-internet/populyarnye-operacionnyye-sistemy-dlya-smartfonov-mobilnye/>.

35. Kotlin документація [Електронний ресурс] – Режим доступу до ресурсу: <https://kotlinlang.org/docs/home.html>.

36. Apple - Swift [Електронний ресурс] – Режим доступу до ресурсу: <https://developer.apple.com/swift/>.

37. Software testing [Електронний ресурс] // Wikipedia – Режим доступу до ресурсу: [https://en.wikipedia.org/wiki/Software\\_testing](https://en.wikipedia.org/wiki/Software_testing).

38. Sachedina F. The Ultimate Guide to Software Testing [Електронний ресурс] / Fahim Sachedina // Global app testing – Режим доступу до ресурсу: <https://www.globalapptesting.com/blog/software-testing>.

39. Введення в тестування програмного забезпечення [Електронний ресурс] – Режим доступу до ресурсу: <https://qalearning.com.ua/theory/lectures/material/testing-intro/>.

40. Test Plan [Електронний ресурс] // Software testing – Режим доступу до ресурсу: <https://softwaretestingfundamentals.com/test-plan/>.

41. Mobile Application Market by Marketplace (Apple iOS Store, Google Play Store, and Other Marketplaces), App Category (Gaming, Entertainment & Music, Health & Fitness, Travel & Hospitality, Retail & E-Commerce, Education & Learning and O [Електронний ресурс] – Режим доступу до ресурсу: <https://www.marketresearch.com/Allied-Market-Research-v4029/Mobile-Application-Marketplace-Apple-iOS-12862471/>.

42. What to Expect from the App Store and Google Play Store When You Launch Your First App [Електронний ресурс] // Yalantis – Режим доступу до ресурсу: <https://yalantis.com/blog/apple-app-store-and-google-play-store/>.

43. Uploading Apps to the Apple App Store and Google Play [Електронний ресурс] // Expo – Режим доступу до ресурсу: <https://docs.expo.dev/distribution/uploading-apps/>.

44. Гігієнічні вимоги до організації роботи з візуальними дисплейними терміналами електронно-обчислювальних машин [Електронний ресурс] // МІНІСТЕРСТВО ОХОРОНИ ЗДОРОВ'Я УКРАЇНИ Головне санітарно-епідеміологічне управління – Режим доступу до ресурсу: <https://zakon.rada.gov.ua/rada/show/v0007282-98#Text>.

45. Правила безпечної експлуатації електроустановок [Електронний ресурс] // ДЕРЖАВНИЙ КОМІТЕТ УКРАЇНИ ПО НАГЛЯДУ ЗА ОХОРОНОЮ ПРАЦІ – Режим доступу до ресурсу: <https://zakon.rada.gov.ua/laws/show/z0011-98#Text>.

46. Наказ про прийняття національних стандартів України, гармонізованих з європейськими стандартами, та скасування національних стандартів України [Електронний ресурс] // МІНІСТЕРСТВО ЕКОНОМІЧНОГО РОЗВИТКУ І ТОРГІВЛІ УКРАЇНИ – Режим доступу до ресурсу: <https://zakon.rada.gov.ua/rada/show/v0869731-14#Text>.



47. Наказ про затвердження Вимог щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями [Електронний ресурс] // МІНІСТЕРСТВО СОЦІАЛЬНОЇ ПОЛІТИКИ УКРАЇНИ. – 2018. – Режим доступу до ресурсу: <https://zakon.rada.gov.ua/laws/show/z0508-18#Text>.

48. Наказ про затвердження Типового положення про порядок проведення навчання і перевірки знань з питань охорони праці та Переліку робіт з підвищеною небезпекою [Електронний ресурс] // ДЕРЖАВНИЙ КОМІТЕТ УКРАЇНИ З НАГЛЯДУ ЗА ОХОРОНОЮ ПРАЦІ – Режим доступу до ресурсу: <https://zakon.rada.gov.ua/laws/show/z0231-05#Text>.

49. Наказ про прийняття національних стандартів, прийняття змін та поправок до національних стандартів [Електронний ресурс] // ДЕРЖАВНЕ ПІДПРИЄМСТВО "УКРАЇНСЬКИЙ НАУКОВО-ДОСЛІДНИЙ І НАВЧАЛЬНИЙ ЦЕНТР ПРОБЛЕМ СТАНДАРТИЗАЦІЇ, СЕРТИФІКАЦІЇ ТА ЯКОСТІ". – 2019. – Режим доступу до ресурсу: <https://zakon.rada.gov.ua/rada/show/v0491774-19#Text>.

50. Електробезпека комп'ютерного обладнання [Електронний ресурс] – Режим доступу до ресурсу: <https://studfile.net/preview/5211197/page:11/>.

# Додатки

ER-Діаграма



Тези конференції

---

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ТЕРНОПІЛЬСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ  
ІМЕНІ ІВАНА ПУЛЮЯ

МАТЕРІАЛИ

ІХ НАУКОВО-ТЕХНІЧНОЇ КОНФЕРЕНЦІЇ

**«ІНФОРМАЦІЙНІ МОДЕЛІ,  
СИСТЕМИ ТА ТЕХНОЛОГІЇ»**



8–9 грудня 2021 року

ТЕРНОПІЛЬ  
2021

<b>А.І. Одарич, М.І. Яворська, М.І. Паламар</b> ДО ОЦІНКИ СТАТИЧНОГО НАВАНТАЖЕННЯ ГОРИЗОНТАЛЬНОГО ВІТРОВОГО ПОТОКУ НА ДЗЕРКАЛО АНТЕНИ <b>I. Odarych, M.I. Yavorska, M.I. Palamar</b> BEFORE ESTIMATING THE STATIC LOAD OF THE HORIZONTAL AIR FLOW ON THE ANTENNA MIRROR	15
<b>М. Фриз, Б. Млинко</b> АНАЛІЗ ВЛАСТИВОСТЕЙ УМОВНОГО ЛІНІЙНОГО ВИПАДКОВОГО ПРОЦЕСУ МЕТОДОМ ХАРАКТЕРИСТИЧНИХ ФУНКЦІЙ <b>M. Fryz, B. Mlynko</b> CONDITIONAL LINEAR RANDOM PROCESS PROPERTIES ANALYSIS USING CHARACTERISTIC FUNCTION METHOD	17
<b>Я. Литвиненко, Г. Шимчук</b> ОГЛЯД МАТЕМАТИЧНИХ МОДЕЛЕЙ ДЛЯ ЗАДАЧІ ПРОГНОЗУВАННЯ ВИКОРИСТАННЯ ПРИРОДНОГО ГАЗУ <b>Ia. Lytvunenko, G. Shymchuk</b> OVERVIEW OF MODELS FOR FORECASTING NATURAL GAS USE	19
<b>О.К. Шкодзінський, М.С. Дзюмак</b> РОЗРОБКА МАТЕМАТИЧНОЇ МОДЕЛІ ДЛЯ ДОСЛІДЖЕННЯ ПОПЕРЕЧНИХ КОЛИВАНЬ ВАЛА З ДИСКОМ <b>O. Shkodzinsky, M. Dziurak</b> DEVELOPMENT OF A MATHEMATICAL MODEL FOR INVESTIGATION OF TRANSVERSE OSCILLATIONS OF A SHAFT WITH A DISC	21
<b>СЕКЦІЯ 2. ІНФОРМАЦІЙНІ СИСТЕМИ ТА ТЕХНОЛОГІЇ, КІБЕРБЕЗПЕКА</b>	
<b>В. Антонів, С. Дейдей, В. Дунець</b> ЗАДАЧА ЦИФРОВОЇ ФІЛЬТРАЦІЇ СИГНАЛІВ В РАДІОТЕХНІЦІ <b>V. Antoniv, Ye. Deidei, V. Dunets</b> THE PROBLEM OF DIGITAL FILTRATION OF SIGNALS IN RADIOTECHNICS	22
<b>В.М. Ахтемійчук</b> МОДЕЛЮВАННЯ АВТОМАТИЗОВАНОЇ СИСТЕМИ ВЕНТИЛЮВАННЯ СКЛАДСЬКИХ ПРИМІЩЕНЬ <b>V.M. Akhtemiychuk</b> MODELING OF THE AUTOMATED SYSTEM OF VENTILATION OF WAREHOUSES	23
<b>Б.М. Баняк</b> МЕТОДИ ФОРМУВАННЯ ПСЕВДОВИПАДКОВИХ ЧИСЕЛ В КРИПТОГРАФІЧНИХ ЗАСОБАХ ЗАХИСТУ БАНКІВСЬКИХ ІНФОРМАЦІЙНИХ СИСТЕМ <b>B M</b>	25
<b>Ю.А. Береза, Д.І.Матик, В.В. Никитюк</b> РЕАЛІЗАЦІЯ КОМУНІКАЦІЇ У КЛІЄНТ-СЕРВЕРНОМУ ЗАСТОСУНКУ ЗА ДОПОМОГОЮ SIGNALR <b>Y. Bereza, D. Matsyk, V. Nykytyuk</b> IMPLEMENTATION OF COMMUNICATION IN THE CLIENT-SERVER APPLICATION USING SIGNALR	28

<b>Ю.З. Лещини, В.Є. Петрусь</b> ПОБУДОВА МУЛЬТИКАНАЛЬНОГО СЕРВЕРА В СИСТЕМІ «РОЗУМНИЙ БУДИНОК» <b>Yu. Leshchynshyn, V. Petrus</b> THE MULTI-CHANNEL SERVER DEVELOPMENT IN THE SYSTEM «SMART HOME»	139
<b>С.В. Соленко, Р.О. Жаровський</b> ВИКОРИСТАННЯ SMART-КОНТРАКТІВ НА БАЗІ БЛОКЧЕЙНА CARDANO В ЕЛЕКТРОННІЙ КОМЕРЦІЇ <b>S. Solenko, R. Zharovskyi</b> USE OF SMART-CONTRACTS BASED ON CARDANO BLOCKCHAIN IN ELECTRONIC COMMERCE	140
<b>А.М. Луцків, Д.А. Цісарук, В.В. Шуптарський</b> АНАЛІЗ ЖИТТЄВОГО ЦИКЛУ ПРОЦЕСУ ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ КОМП'ЮТЕРНИХ СИСТЕМ <b>A.M. Lutskiy, D.A. Tsisaruk, V.V. Shuptarskyi</b> ANALYSIS OF SOFTWARE TESTING LIFE CYCLE PROCESS IN COMPUTER SYSTEMS	142
<b>Ю.В. Шевчук, Н.Б. Стадник</b> АЛГОРИТМ ІДЕНТИФІКАЦІЇ ВІДВІДУВАЧА В ДОМОФОННІЙ СИСТЕМІ ЗА ЗОБРАЖЕННЯМ ОСОБИ <b>Yu.V. Shevchuk, N.B. Stadnyk</b> VISITOR IDENTIFICATION ALGORITHM IN THE INTERCOM SYSTEM BY PERSONAL IMAGE	143
<b>В.В. Яцишин, Х.В. Яворська</b> ВІДМІНОСТІ LOW-CODE/NO-CODE РОЗРОБКИ <b>V.V. Yatsyshyn, K.V. Yavorska</b> DIFFERENCES IN LOW-CODE/NO-CODE DEVELOPMENT	144
<b>СЕКЦІЯ 4. ПРОГРАМНА ІНЖЕНЕРІЯ ТА МОДЕЛЮВАННЯ СКЛАДНИХ РОЗПОДІЛЕНИХ СИСТЕМ</b>	
<b>І.В.Бендера, Г.Б. Цуприк</b> РОЗРОБКА СИСТЕМИ АНАЛІЗУ ТА ПРОГНОЗУВАННЯ ПОДІЙ З ВИКОРИСТАННЯМ ТЕХНОЛОГІЇ C#/NET <b>I.V.Bendera, H.B.Tsupryk</b> DEVELOPMENT OF AN ANALYSIS AND EVENT FORECASTING SYSTEM USING C # /. NET TECHNOLOGIES	145
<b>Ю.А. Береза, В.В. Никитюк</b> НАЛАШТУВАННЯ СЕРВЕРА АВТОРИЗАЦІЇ IDENTITY4 ДЛЯ РОЗРОБЛЕННЯ ДОДАТКУ ГЕОПОЗИЦІОНУВАННЯ ВЕЛОСИПЕДИСТІВ <b>Y. Bereza, V. Nykytyuk</b> SETTING UP THE IDENTITY 4 AUTHORIZATION SERVER FOR DEVELOPING APPLICATIONS WITH GEOPOSITIONING CYCLISTS	146
<b>Н. Базюк, А. Флейтута</b> ІНЖЕНЕРІЯ ВИМОГ ДО ПРОГРАМНОГО ПРОДУКТУ В ГНУЧКИХ ТЕХНОЛОГІЯХ РОЗРОБКИ <b>N. Baziuk, A. Fleituta</b> SOFTWARE REQUIREMENTS ENGINEERING IN AGILE DEVELOPMENT	147

УДК 004

**Ю.А. Береза, студент групи САМ-61; Д.І.Мацик, студент групи СНМ-61;  
В.В. Никитюк, канд. техн. наук, доц. каф. КН**  
(Тернопільський національний технічний університет імені Івана Пулюя, Україна)

## **РЕАЛІЗАЦІЯ КОМУНІКАЦІЇ У КЛІЄНТ-СЕРВЕРНОМУ ЗАСТОСУНКУ ЗА ДОПОМОГОЮ SIGNALR**

UDC 004

**Y. Bereza, student; D. Matsyk, student; V. Nykytyuk, Ph.D.**

## **IMPLEMENTATION OF COMMUNICATION IN THE CLIENT-SERVER APPLICATION USING SIGNALR**

Інформаційні технології на сьогоднішній день охоплюють багато напрямків, в тому числі і медицину. Поєднавши комунікацію між лікарем та пацієнтом, розробимо клієнт-серверний застосунок. Для реалізації SignalR у проєкті спершу потрібно налаштувати в стартовому класі сервера. Налаштування в стартовому класі наведено у лістингу 1.

Лістинг 1 – Конфігурація SignalR в стартовому класі на стороні сервера

```
public void ConfigureServices(IServiceCollection services) =>
services.AddSignalR();
public void Configure(IApplicationBuilder app, IWebHostEnvironment env,
ServerDbContext dataContext)
{ app.UseEndpoints(endpoints => {
    endpoints.MapDefaultControllerRoute();
    endpoints.MapHub<UpdateHub>("/update-hub"); });}
```

Далі необхідно створити клас UpdateHub, який буде наслідувати клас Hub. Клас UpdateHub буде реалізовувати у собі обробку даних, та надсилатиме їх у певні групи клієнтам. У лістингу 2 наведено реалізацію UpdateHub.

Лістинг 2 – Реалізація UpdateHub

```
[Authorize]
public class UpdatesHub : Hub {
private readonly IRepository _repository;
public UpdatesHub(IRepository repository){ _repository = repository;}
public async override Task OnConnectedAsync(){ var user = await
_repository.GetUserById(Guid.Parse(Context.User.Claims.First(p => p.Type ==
"Id").Value));
    await Groups.AddToGroupAsync(Context.ConnectionId,
user.CustomerCompanyId.ToString()); }}
```

SignalR чудовий інструмент для оновлення баз даних в режимі реального часу. Використання SignalR у проєкті значно заощаджує час реалізації комунікації клієнтів зі сервером.

УДК 004

**Ю.А. Береза, студент групи САМ-61, В.В. Никитюк, канд. техн. наук, доц. каф. КН**

(Тернопільський національний технічний університет імені Івана Пулюя, Україна)

## **НАЛАШТУВАННЯ СЕРВЕРА АВТОРИЗАЦІЇ IDENTITY4 ДЛЯ РОЗРОБЛЕННЯ ДОДАТКУ ГЕОПОЗИЦІОНУВАННЯ ВЕЛОСИПЕДИСТІВ**

UDC 004

**Y. Bereza, student, V. Nykytyuk, Ph.D.**

### **SETTING UP THE IDENTITY 4 AUTHORIZATION SERVER FOR DEVELOPING APPLICATIONS WITH GEOPOSITIONING CYCLISTS**

Майже кожен додаток містить в собі реалізацію аутентифікації та авторизації. Ці функції забезпечують певну ідентифікацію користувача та надають доступ до ресурсів.

Розглянемо приклад розгортання сервера identity4. Для початку потрібно налаштувати конфігурації (оголосити та вказати ресурси сервера). У лістингу 1 наведено налаштування конфігурації сервера.

Лістинг 1 – Конфігурація сервера авторизації

```
public void ConfigureServices(IServiceCollection services) {
    var builder = services.AddIdentityServer()
        .AddDeveloperSigningCredential()
        .AddInMemoryApiScopes(Config.ApiScopes)
        .AddInMemoryClients(Config.Clients); }
public static class Config {
public static IEnumerable<ApiScope> ApiScopes =>
    new List<ApiScope>{
        new ApiScope("api1", "BikeTravel
API")}; }
```

Після завершення конфігурації, необхідно вказати клієнти сервера авторизації (Mobile, Desktop, Web або інші) [1].

На цьому моменті, налаштування SignalR на стороні сервера можна вважати завершеним. Далі необхідно у класах контролерів, реалізувати методи, по яким користувач зможе комунікувати з сервером (отримувати, оновлювати дані), сервер буде опрацьовувати дані та відправляти оновлення користувачам. Для відправки оновлення користувачам використовуються методи Client.SendAsync(), Groups("GroupName").SendAsync().

Client.SendAsync() – використовується для надсилання всім, або конкретному клієнту.

Groups("GroupName").SendAsync() – використовується для надсилання групам клієнта.

Використання SignalR у проекті значно заощаджує час реалізації комунікації клієнта зі сервером. Також слід зазначити, що SignalR реалізація не потребує великих затрат на тестування, оскільки більшість frameworks та бібліотек випускаються вже відтестовані інструменти розробки, для яких з часом виходять все більш стабільні збірки.

#### **Література.**

1. IdentityServer4 – Documentation [Електронний ресурс] – Режим доступу до ресурсу: <https://identityserver4.readthedocs.io/en/latest/>.