

ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
МІСЬКОГО ГОСПОДАРСТВА ІМЕНІ О. М. БЕКЕТОВА

Пояснювальна записка
до кваліфікаційної роботи бакалавра

на тему: «Розробка вебзастосунку для підтримки сервісу оренди
велосипедів»

Виконав: студент 4 курсу, групи КН-2021-1
спеціальності 122 Комп'ютерні науки

Палагін А. А.



Керівник:

к.т.н., доцент Карпенко М. Ю.



Рецензент: доц. Пахомов Ю. В.



м. Харків – 2025 рік

Харківський національний університет міського господарства імені О. М. Бекетова

(повне найменування закладу вищої освіти)

Навчально-науковий Інститут енергетичної, інформаційної

та транспортної інфраструктури

Кафедра комп'ютерних наук та інформаційних технологій

Рівень вищої освіти перший (бакалаврський)

Спеціальність 122 «Комп'ютерні науки»

(шифр і назва)

ЗАТВЕРДЖУЮ

Завідувач кафедри КНтаІТ



Марина

НОВОЖИЛОВА

« 27 » 06 2025 року

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

Палагіну Артему Андрійовичу

(прізвище, ім'я, по батькові)

1. Тема роботи Розробка вебзастосунку для підтримки сервісу оренди велосипедів

керівник роботи доцент кафедри КНтаІТ, к.т.н. Карпенко М. Ю.

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом закладу вищої освіти від «09» травня 2025 р. № 341-03

2. Термін подання студентом роботи 15.06.2025 р.

3. Вихідні дані до роботи Рекомендації щодо розробки додатку, індивідуальне завдання на розробку

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

дослідження існуючих інструментів розробки додатку для підтримки

сервісу оренди велосипедів; розгляд особливостей та класифікація

завдань; проектування та програмна реалізація модуля для підтримки

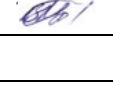
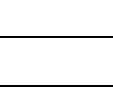
сервісу оренди велосипедів; тестування розробленого модуля.

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

Презентація – 18 аркушів, діаграми UML (IDEF0), що супроводжують етап

проектування, копії екранів інтерфейсу, копії звітів.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Розділ I	М. Ю. Карпенко, доц. каф. КНтаІТ 	26.05.2025	31.05.2025
Розділ II	М. Ю. Карпенко, доц. каф. КНтаІТ 	01.06.2025	08.06.2025
Розділ III	М. Ю. Карпенко, доц. каф. КНтаІТ 	09.06.2025	15.06.2025
Розділ IV	В. В. Малишева, доц. каф. ОПБЖД 	16.06.2025	23.06.2025

7. Дата видачі завдання 19.05.2025 р.

КАЛЕНДАРНИЙ ПЛАН

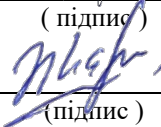
№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Вибір теми кваліфікаційної роботи	26.05.2025	
2	Затвердження тем, наукових керівників, завдань та календарного плану підготовки дипломної роботи	29.05.2025	
3	Написання I розділу	31.05.2025	
4	Написання II розділу	08.06.2025	
5	Написання III розділу	15.06.2025	
6	Написання IV розділу	19.06.2025	
7	Подання кваліфікаційної роботи керівнику	20.06.2025	
8	Робота по усуненню зауважень керівника, уточнення і доповнення практичного матеріалу, оформлення додатків до роботи	22.06.2025	
9	Подання доопрацьованого варіанту роботи керівнику	23.06.2025	
10	Захист матеріалів кваліфікаційної роботи на засіданні кафедри	23.06.2025	
11	Офіційний захист матеріалів кваліфікаційної роботи на засіданні Державної екзаменаційної комісії	28.06.2025	

Студент



(підпис)

Керівник роботи



(підпис)

А. А. Палагін

(прізвище та ініціали)

М. Ю. Карпенко

(прізвище та ініціали)

АНОТАЦІЯ

Пояснювальна записка до кваліфікаційної роботи містить: с. 63, рисунків 29, таблиць 13, додатків 2, джерел 21.

Тема кваліфікаційної роботи: «Розробка вебзастосунку для підтримки сервісу оренди велосипедів».

Метою кваліфікаційної роботи є розробка вебдодатку для організації оренди велосипедів.

Об'єктом кваліфікаційної роботи є процес організації оренди велосипедів із використанням інформаційних технологій.

Предметом кваліфікаційної роботи є методи та засоби розробки вебдодатку для автоматизації процесу оренди велосипедів.

У процесі виконання проєкту було здійснено аналіз наявних рішень у сфері оренди транспорту, розроблено концепцію вебдодатку та реалізовано функціональний вебсервіс для оренди велосипедів. Програмна реалізація додатку була здійснена з використанням технологій React, Typescript, NestJs, PostgreSQL, Docker.

Отримані результати можуть бути використані широким колом користувачів та юридичних осіб, що надають послуги щодо оренди велосипедів.

Ключові слова: ВЕБ-ДОДАТОК, ОРЕНДА ТРАНСПОРТУ, БАЗА ДАНИХ, РЕЖИМ РЕАЛЬНОГО ЧАСУ, ЕЛЕКТРОННА МАПА, UML-ДІАГРАМА, ІНТЕРФЕЙС, МОСКUP

ABSTRACT

The explanatory note to the qualification work contains: p. 63, figures 29, tables 13, appendices 2, sources 21.

Theme of the qualification work: “Development of a Web Application to Support a Bicycle Rental Service”.

The purpose of the qualification work is to develop a web application for organizing bicycle rental.

The object of the qualification work is the process of organizing bicycle rental using information technology.

The subject of the qualification work is the methods and means of developing a web application to automate the bicycle rental process.

In the course of the project, an analysis of existing solutions in the field of vehicle rental was carried out, a web application concept was developed, and a functional web service for bicycle rental was implemented. The software implementation of the application was carried out using React, Typescript, NestJs, PostgreSQL, and Docker technologies.

The obtained results can be used by a wide range of users and legal entities providing bicycle rental services.

Keywords: WEB APPLICATION, TRANSPORT RENTAL, DATABASE, REAL-TIME MODE, ELECTRONIC MAP, UML-DIAGRAM, INTERFACE, MOCKUP

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

БД – база даних;

ІК – інтерфейс користувача;

ТС – транспорта система;

ТЗ – транспортний засіб;

СУБД – система управління базою даних;

ПЗ – програмне забезпечення;

ДП – діаграма прецедентів;

ДВД – діаграма видів діяльності;

ІС – інформаційна система;

UX – User Experience – користувацький досвід;

Web – система доступу до пов’язаних між собою документів на різних комп’ютерах, підключених до Інтернету.

ЗМІСТ

ВСТУП	9
РОЗДІЛ 1 ЗАГАЛЬНІ ПОЛОЖЕННЯ	11
1.1 Аналіз діяльності та опис предметного середовища.....	11
1.1.1 Характеристика підприємства та його діяльності	11
1.1.2 Постановка задачі	14
1.1.3 Опис предметної області	16
1.2 Огляд наявних аналогів.....	18
Висновки до розділу	22
РОЗДІЛ 2 ІНФОРМАЦІЙНЕ ТА МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ	23
2.1. Глосарій	23
2.2 Аналіз предметної області	23
2.3. Специфікація функціональних та нефункціональних вимог	25
2.4 Вхідні та вихідні дані	27
2.5 Проектування системи	27
2.5.1 Вибір CASE-засобів для проектування системи	27
2.5.2 Проектування бази даних	28
2.5.3 Проектування програмного забезпечення	31
Висновки до розділу	33
РОЗДІЛ 3 ПРОГРАМНЕ ТА ТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ	34
3.1 Засоби розробки та розгортання програмного продукту	34
3.2 Вимоги до технічного та програмного забезпечення.....	34
3.3 Проектування інтерфейсу користувача.....	35
Висновки до розділу	41
РОЗДІЛ 4 ОХОРОНА ПРАЦІ	43
4.1 Регулювання питань охорони праці на законодавчому рівні	43
4.2 Виявлення потенційних небезпек стосовно об'єкту проектування.....	46
4.3 Дослідження ризику реалізації потенційних небезпек на об'єкті проектування та розробка заходів щодо їх попередження	48

	8
ЗАГАЛЬНІ ВИСНОВКИ	52
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ	53
ДОДАТОК А Специфікації варіантів використання	56
ДОДАТОК Б Лістинг основних класів програми	61

ВСТУП

У зв'язку з триваючим військовим станом в Україні значно загострилися питання мобільності населення, особливо в містах, які зазнали руйнувань або відчують обмеження у функціонуванні громадського транспорту. Зруйнована інфраструктура, перебої в роботі автобусів і електротранспорту, дефіцит пального та потреба в оперативному пересуванні змушують шукати альтернативні способи організації транспортного сполучення. У цьому контексті використання велосипедів стає не лише зручним, але й соціально значущим рішенням.

Організація оренди велосипедів дозволяє забезпечити доступний, екологічно чистий і незалежний від пального вид транспорту, що особливо актуально в умовах нестабільності. Вебдодатки для оренди двоколісного транспорту можуть стати важливим інструментом для мешканців міст і селищ, допомагаючи швидко пересуватися між районами, діставатися до укриттів, лікарень, волонтерських центрів або місць роботи. Таким чином, впровадження подібних цифрових рішень сприяє підвищенню рівня автономності громадян і покращенню логістики в умовах надзвичайної ситуації [1-2].

Метою кваліфікаційної роботи є розробка вебдодатку для організації оренди велосипедів. Реалізація такого додатку забезпечує зручний та ефективний механізм для задоволення потреб користувачів у транспортному засобі для активного відпочинку, занять спортом і мобільного пересування в межах міста. Передбачається створення сучасного цифрового рішення, яке поєднуватиме інтуїтивно зрозумілий інтерфейс, високу продуктивність і надійність обробки даних.

Об'єктом роботи є процес надання послуг оренди велосипедів у цифровому середовищі.

Предметом роботи є методи та засоби розробки вебзастосунку для автоматизації процесів оренди, обліку, управління користувачами та забезпечення зручного інтерфейсу взаємодії.

Основні завдання роботи:

- провести аналіз предметної області та вимог до сервісу оренди велосипедів;
- розробити структуру та функціональні можливості вебзастосунку.
- обрати інструменти та технології для реалізації вебзастосунку;
- спроектувати інтерфейс користувача з урахуванням принципів ui/ux-дизайну;
- реалізувати основні модулі системи: реєстрація, бронювання, управління велосипедами;
- провести тестування вебзастосунку та оцінити його ефективність.
- розробити рекомендації щодо впровадження системи у реальну експлуатацію.

Впровадження вебдодатку сприятиме покращенню транспортної доступності та мобільності населення, особливо в умовах обмеженого функціонування громадського транспорту або складної дорожньої ситуації. Завдяки автоматизації процесу оренди користувачі зможуть у зручний спосіб знайти, забронювати та оплатити велосипед у режимі реального часу, що значно підвищує ефективність використання ресурсів.

Крім того, цифровізація сервісу дозволить зменшити адміністративне навантаження на персонал, покращити облік та контроль за транспортними засобами, а також забезпечити прозору систему взаємодії з клієнтами. Вбудовані аналітичні механізми можуть бути використані для моніторингу попиту, оптимізації розміщення велосипедів та удосконалення стратегій управління міською мобільністю. Таким чином, застосунок не лише забезпечить користувачів зручним засобом пересування, але й стане інструментом підтримки сталого розвитку міського середовища.

РОЗДІЛ 1

ЗАГАЛЬНІ ПОЛОЖЕННЯ

1.1 Аналіз діяльності та опис предметного середовища

1.1.1 Характеристика підприємства та його діяльності

ТОВ «Ukraine Web Solutions» – провідна українська ІТ-компанія, що спеціалізується на розробці сучасного програмного забезпечення, вебзастосунків та хмарних сервісів для бізнесу та державного сектору. Компанія має багаторічний досвід роботи на ринку інформаційних технологій, сформовану інфраструктуру та висококваліфікований кадровий склад.

На базі «Ukraine Web Solutions» функціонує сучасний технічний парк, що включає високопродуктивні комп'ютерні системи, серверне обладнання, захищені канали зв'язку, а також інструменти для розробки, тестування та впровадження програмних рішень. У штаті компанії працюють фахівці з програмування, вебдизайну, UI/UX-дизайну, тестування, а також менеджери проєктів з досвідом реалізації ІТ-рішень на національному рівні.

Завдяки наявній матеріально-технічній базі, налагодженим процесам і професійному середовищу, на базі ТОВ «Ukraine Web Solutions» можливо ефективно апробувати та впровадити нові проєкти, зокрема вебзастосунок для підтримки сервісу оренди велосипедів. Компанія забезпечує повний цикл супроводу таких проєктів – від концепції до повноцінного впровадження з можливістю масштабування.

Організаційна структура ТОВ «Ukraine Web Solutions» представлена на рисунку 1.1, де відображено основні відділи та підрозділи, кожен з яких відповідає за конкретні напрями діяльності, забезпечуючи ефективне функціонування та поступовий розвиток організації.

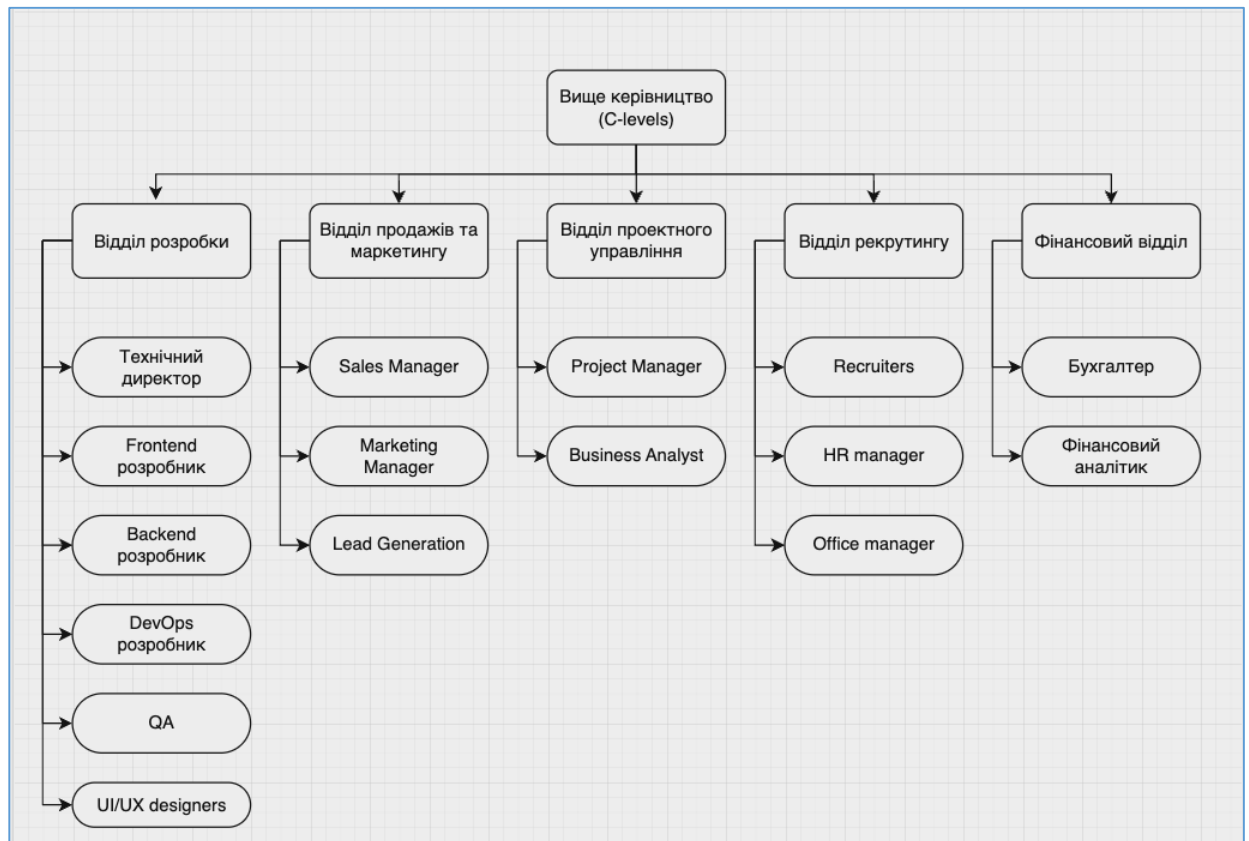


Рисунок 1.1 – Організаційна структура підприємства ТОВ «Ukraine Web Solutions»

Організаційна структура компанії має чітко виражений функціональний поділ і передбачає п'ять основних підрозділів, які підпорядковуються вищому керівництву (C-levels):

- відділ розробки включає технічного директора, frontend і backend розробників, devops-інженера, фахівця з тестування та ui/ux дизайнерів. цей підрозділ відповідає за повний цикл створення програмного забезпечення;
- відділ продажів та маркетингу об'єднує менеджерів з продажу (sales manager), маркетингу (marketing manager) та спеціаліста з генерації лідів (lead generation), забезпечуючи просування компанії та залучення клієнтів;

- відділ проєктного управління складається з проєктного менеджера (project manager) та бізнес-аналітика (business analyst), які відповідають за планування, координацію та реалізацію проєктів;
- відділ рекрутингу включає рекрутерів, HR-менеджера та офіс-менеджера. його діяльність зосереджена на підборі персоналу, управлінні людськими ресурсами та організаційній підтримці;
- фінансовий відділ охоплює бухгалтера та фінансового аналітика, які забезпечують облік, фінансове планування та аналітику.

Ця структура забезпечує ефективну координацію діяльності компанії, розмежування функціональних обов'язків і підтримує гнучкість у прийнятті управлінських рішень. На рисунку 1.2 зображено структуру відділу з розробки на ТОВ «Ukraine Web Solutions».

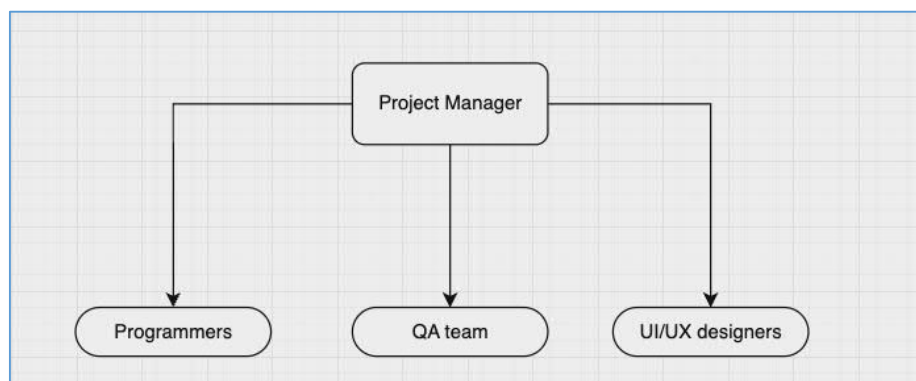


Рисунок 1.2 – Організаційна структура відділу з розробки на ТОВ «Ukraine Web Solutions».

Як видно з діаграми, керівником відділу є Project Manager (Керівник проєкту), – центральна фігура в організаційній структурі, яка відповідає за загальне керівництво проєктом, координацію дій між командами, планування, розподіл ресурсів, контроль термінів та якості виконання. В його підпорядкуванні знаходяться три основні підрозділи:

- Programmers (Програмісти), відповідають за безпосередню реалізацію програмного коду, створення функціоналу вебзастосунку, інтеграцію з базами даних, API тощо;
- QA team (Команда тестування), забезпечує контроль якості програмного продукту, тестує функціональність, перевіряє відсутність помилок і відповідність технічному завданню;
- UI/UX designers (дизайнери інтерфейсу та досвіду користувача), працюють над візуальним оформленням, зручністю використання, логікою навігації вебзастосунку, розробляють макети та прототипи інтерфейсу.

Наявність такого відділу сприяє ефективній реалізації проєктів, забезпечує їх високу якість та дозволяє вирішувати завдання автоматизації високої складності. Зокрема, здійснити успішну апробацію проєкту, що розробляється.

1.1.2 Постановка задачі

Проєкт передбачає розробку вебзастосунку для автоматизації процесу оренди велосипедів, який забезпечить швидкий доступ до інформації про наявні велосипеди, дозволить здійснити бронювання, оплату, а також відстежувати статус оренди.

Основні функції системи [1-3]:

Для користувачів:

- реєстрація та авторизація;
- перегляд доступних велосипедів за фільтрами (тип, ціна, локація);
- оформлення замовлення та оплата;
- перегляд історії оренди;
- залишення відгуків і оцінювання сервісу.

Для адміністратора:

- керування базою велосипедів (додавання, редагування, видалення);
- перегляд замовлень і керування ними;
- формування звітності;

- контроль доступності та технічного стану велосипедів.

Вхідні дані:

- персональні дані користувача;
- інформація про велосипеди;
- дані про бронювання;
- правила та умови оренди.

Вихідні дані:

- підтвердження оренди;
- повідомлення про статус замовлення;
- звіти для адміністратора.

Формат реалізації:

- вебзастосунок з адаптивним дизайном;
- інтеграція з платіжними системами;
- можливість використання на мобільних пристроях.

Очікувані результати:

- система забезпечить автоматизовану підтримку процесів оренди, підвищить зручність для користувачів і дозволить компанії ефективно керувати своїм парком велосипедів.

Характеристики бізнес-процесу оренди велосипедів наведено у таблиці 1.1.

Таблиця 1.1 – Характеристики бізнес-процесу «Оренда велосипедів»

Характеристики бізнес-процесу	Значення
Назва бізнес-процесу	Оренда велосипедів
Актори	Користувач
Вхідна подія	Реєстрація користувачів
Вхідні документи	Персональні дані користувачів
Вихідна подія	Формування даних про оренду
Вихідні документи	Параметри оренди
Клієнти	Користувачі

1.1.3 Опис предметної області

Діаграма бізнес-процесу «Оренда велосипедів» ілюструє ключові етапи взаємодії користувача із системою: від авторизації до завершення поїздки. Процес починається зі збору персональних даних, що використовуються для автентифікації та отримання токена доступу.

Після цього користувач поповнює баланс, що дає змогу перейти до вибору велосипеда на мапі. Після підтвердження бронювання система надає інформацію про активну поїздку. Завершення оренди фіксується у базі даних.

Основним учасником процесу є користувач, який послідовно проходить усі етапи для отримання послуги оренди. Контекстна діаграма бізнес-процесу наведена на рисунку 1.3.

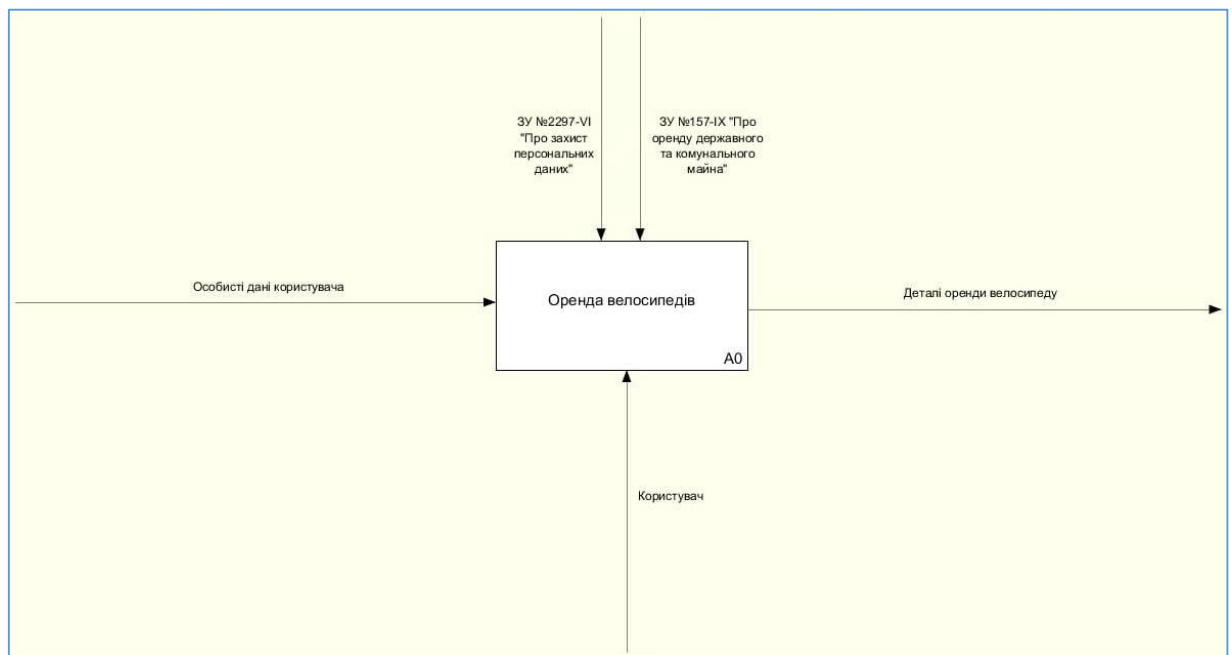


Рисунок 1.3 – Контекстна діаграма процесу «Оформлення оренди велосипедів»»

Результат декомпозиції бізнес-процесу [11-13] наведено на рисунку 1.4. Як видно на діаграмі, процес оренди велосипедів включає чотири основні етапи (A1–A4), кожен з яких виконує окрему функцію, а саме.

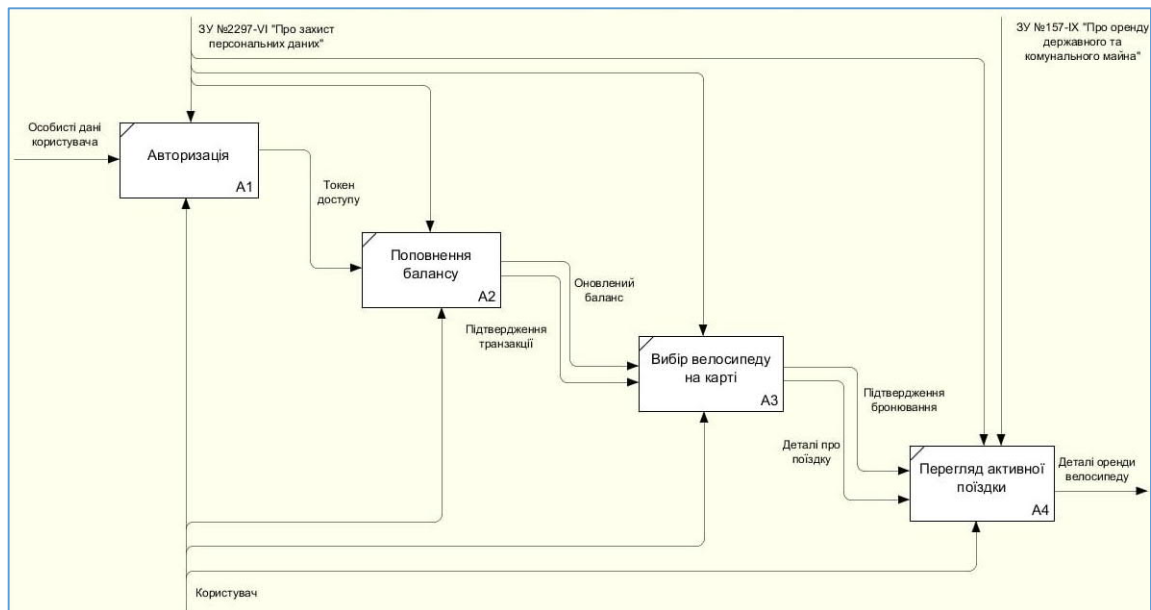


Рисунок 1.4 – Декомпозиція контекстної діаграми процесу «Оформлення оренди велосипедів»

A1 – Авторизація

- Вхідні дані: особисті дані користувача.
- Керування: ЗУ №2297-VI «Про захист персональних даних».
- Вихідні дані: токен доступу.
- Виконавець: користувач.

Цей процес забезпечує автентифікацію користувача в системі перед подальшими діями.

A2 – Поповнення балансу.

- Вхідні дані: токен доступу.
- Керування: підтвердження транзакції.
- Вихідні дані: оновлений баланс.
- Виконавець: користувач.

Користувач поповнює рахунок для можливості здійснення оренди.

A3 – Вибір велосипеда на карті.

- Вхідні дані: оновлений баланс.
- Керування: деталі поїздки, доступні велосипеди.
- Вихідні дані: підтвердження бронювання.
- Виконавець: користувач.

Після поповнення балансу користувач здійснює вибір велосипеда через карту інтерфейсу.

A4 – Перегляд активної поїздки

- Вхідні дані: підтвердження бронювання.
- Керування: ЗУ №157-IX «Про оренду державного та комунального майна».
- Вихідні дані: деталі оренди велосипеда.
- Виконавець: користувач.

Користувач має змогу переглядати інформацію про свою поточну оренду та маршрут.

Декомпозиція забезпечує чітке уявлення про послідовність дій у системі та сприяє структурованій реалізації функціональних компонентів додатку.

1.2 Огляд наявних аналогів

Для обґрунтування доцільності розробки та оцінки ключових вимог до майбутньої системи у кваліфікаційній роботі було виконано огляд систем, що мають схожий функціонал.

Сервіс компанії Lime (рис. 1.5) організує послуги оренди велосипедів у містах по всьому світу. Діяльність компанії та її сервісів спрямована на підвищення мобільності та зниження транспортного навантаження внаслідок впровадження доступного сервісу для короткострокових поїздки.

Для пошуку велосипедів використовуються GPS-технології. Адаптація до конкретних потреб користувачів здійснюється через низку тарифних планів.

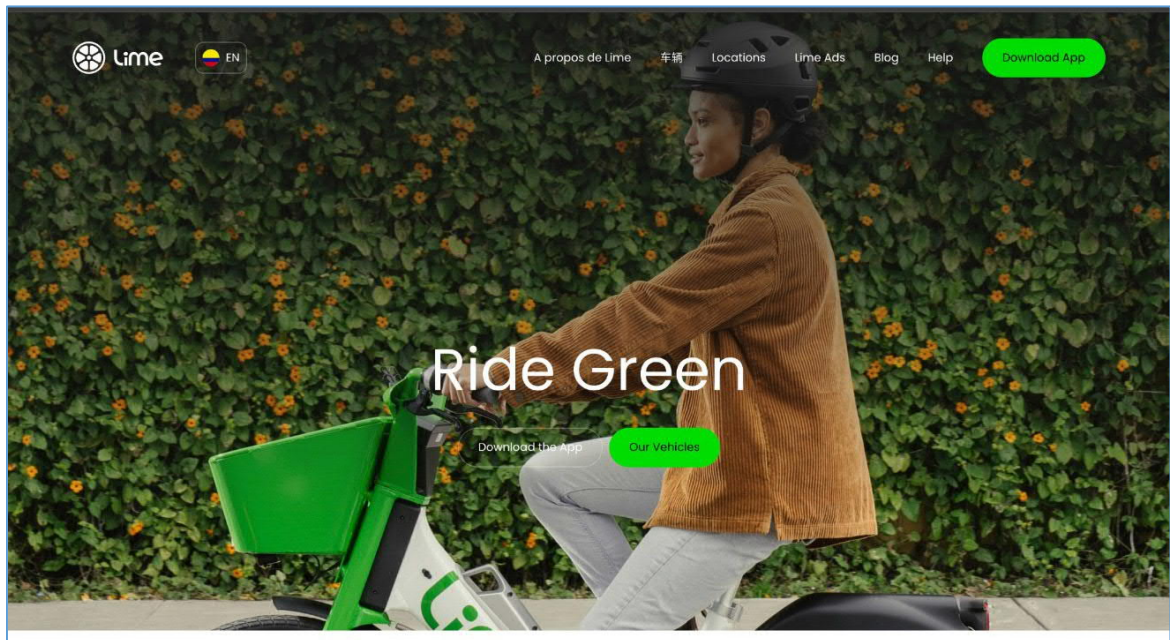


Рисунок 1.5 – Головне вікно сервісу компанії Lime

Оренда здійснюється безстанційного, залишати велосипеди можна в довільному місці.

До переваг системи Lime можна віднести:

- зручність мобільного додатку, – він має інтуїтивно зрозумілий інтерфейс, дозволяє швидко знаходити доступні велосипеди на мапі та здійснювати бронювання у декілька кліків;
- гнучка тарифна політика, – це дозволяє оптимізувати витрати залежно від характеру поїздки;
- наявність електричних велосипедів, – суттєво підвищує комфорт користування.

Щодо недоліків системи Lime можна вказати на такі:

- висока вартість коротких поїздок;
- недостатній контроль за технічним станом велосипедів;

- обмежене географічне покриття;
- недосконала взаємодія з іншими видами транспорту, це ускладнює створення комбінованих маршрутів.

Водночас Lime є прикладом сучасного рішення у сфері оренди велосипедів з широким функціоналом, однак має низку обмежень, що потребують удосконалення.

Сервіс «Bird» (рис. 1.6) є ще одним з провідних на ринку прокату велосипедів, електросамокатів та електровелосипедів. Основний інструмент користування сервісом – мобільний застосунок, він дозволяє швидко знаходити, бронювати та розблоковувати транспортний засіб (ТЗ). Додаток доступний для Android та iOS, відрізняється простим інтерфейсом, забезпечує зручний пошук та бронювання ТЗ. Користувач має змогу переглядати доступні ТЗ, їх розташування, заряд батареї та доступність.

Активація та розблокування ТЗ робиться через сканування QR-коду. Після завершення поїздки ТЗ блокується через додаток з фіксацією місця паркування.

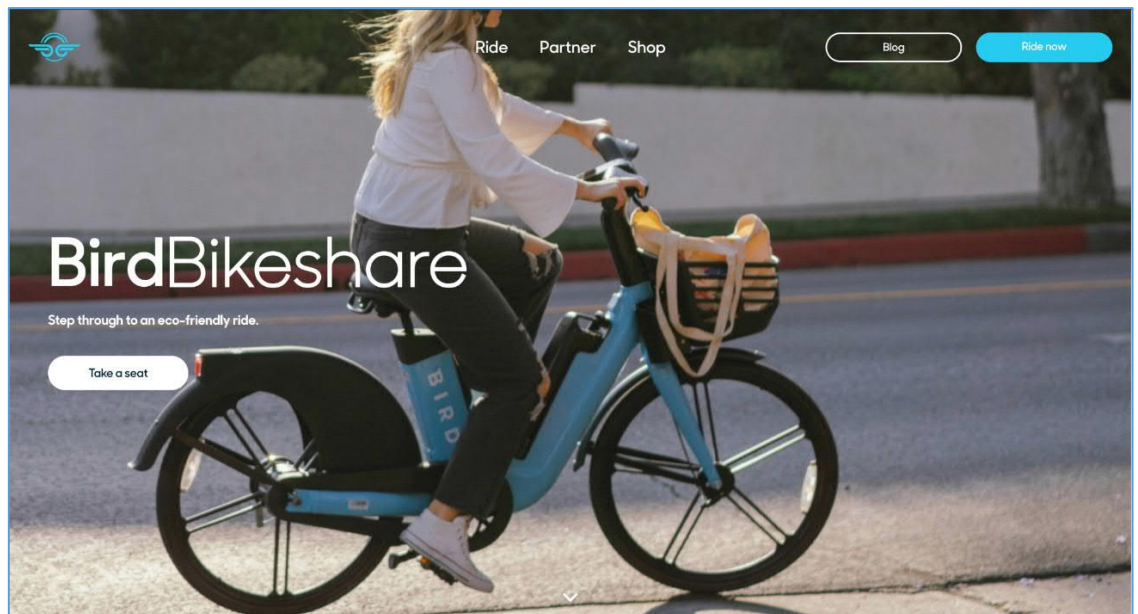


Рисунок 1.6 – Головне вікно сервісу Bird

До переваг сервісу можна знайти такі:

- зручний інтерфейс;
- широке географічне покриття;
- висока надійність роботи.

Серед недоліків сервісу можна вказати:

- Висока вартість.
- Обмеження щодо часу оренди.

Порівняльні характеристики розглянутих сервісів наведені у таблиці 1.2.

Таблиця 1.2 – Порівняльна характеристика систем-аналогів

Назва продукту	LimeBike	Bird
Інтерфейс	Зручний інтерфейс з інтерактивною картою	Зрозумілий інтерфейс
Підтримка користувачів	Підтримка через додаток та або веб-сайт,	Система підтримки через додаток
Функціональність	Оренда ТЗ, геолокація, вибір тарифного плану	Оренда через QR-коди
Платформи	Android, iOS	Android, iOS
Наявність зворотного зв'язку	Працює в межах додатку та на веб-сайті	Працює через додаток
Оплата	Через додаток згідно до тарифів	Через додаток згідно до тарифів

Проведений аналіз аналогів дозволяє дійти висновку, що обидва додатки забезпечують ефективний, зручний механізм оренди, але потребують певної адаптації до вітчизняних умов та удосконалення інтерфейсу. Тому виглядає доцільним розробка спеціалізованого сервісу з організації оренди велосипедів, який буде:

- адаптований до регіональних особливостей та соціально-економічних умов;
- доступнішим за вартістю завдяки локальній орієнтації;

- інтегрованим із місцевою транспортною інфраструктурою (наприклад, враховуючи розклад маршруток чи наявність укриттів поблизу);
- гнучким у налаштуванні тарифів і функціоналу, відповідно до потреб цільової аудиторії.

Реалізація такого сервісу дозволить суттєво покращити транспортне забезпечення населення та може бути впроваджене в більшості регіонів України.

Висновки до розділу

У першому розділі кваліфікаційної роботи здійснено аналіз предметної області, що становить основу для розробки програмного продукту. Наведено організаційну структуру підприємства, на базі якого планується проведення апробації розробленого рішення. Також розглянуто та проаналізовано ключові бізнес-процеси, що пов'язані з діяльністю підприємства у сфері оренди транспортних засобів.

Окрему увагу приділено аналізу існуючих аналогів програмних рішень, що дало змогу визначити їх сильні та слабкі сторони. На основі цього сформульовано обґрунтування доцільності створення вебдодатку, адаптованого до локальних умов і потреб користувачів.

РОЗДІЛ 2

ІНФОРМАЦІЙНЕ ТА МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ

2.1. Глосарій

Глосарій є важливим елементом документації, що містить ключові терміни предметної області та сприяє уніфікації термінології між замовником і розробником [4-8]. Глосарій для нашого проєкту наведено в таблиці 2.1.

Таблиця 2.1 – Глосарій проєкту оренди велосипедів

Термін	Опис терміну
1. Поняття проєкту та предметної області	
Оренда ТЗ	Можливість тимчасово користуватися ТЗ.
Тарифи	Перелік можливих варіантів оплати.
Стан велосипеда	Опис технічного стану велосипеда.
Замовлення	Запит на оренду ТЗ.
Резервування	Резервування велосипеда перед подальшою орендою.
Ідентифікація користувача	Підтвердження персональних даних користувача.
Історія замовлень	Перелік попередніх замовлень користувача.
Вхідні дані	Персональні дані користувача
Вихідні дані	Умови оренди

Термінологія, представлена в глосарії, використовуватиметься на подальших етапах проєктування системи для забезпечення послідовності та однозначного тлумачення ключових понять.

2.2 Аналіз предметної області

Опис предметної області буде здійснено з використанням нотацій UML та IDEF0, що дозволяють формалізувати логіку бізнес-процесів системи. Аналіз розпочинається з побудови діаграми варіантів використання [11-13].

Діаграма використання для нашого проєкту наведена на рисунку 2.1.

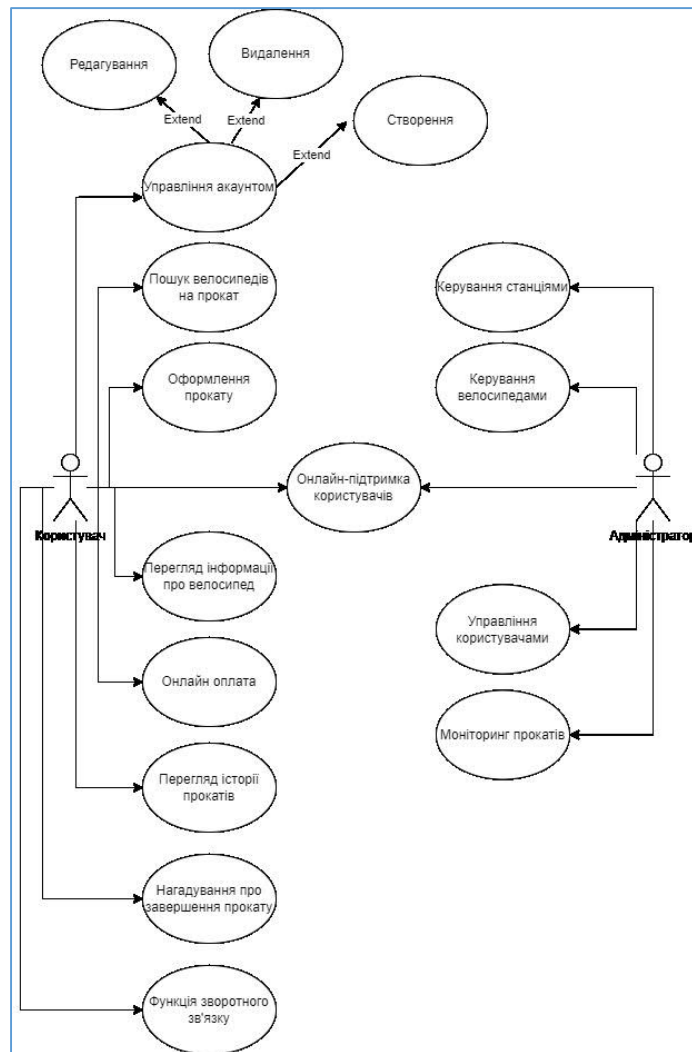


Рисунок 2.1 – Діаграма використання вебдодатку

Діаграма ілюструє взаємодію двох основних акторів – Користувача та Адміністратора – з різними функціями системи.

Основні функції користувача:

- реєстрація та керування акаунтом (створення, редагування, видалення);
- пошук велосипедів на прокат;
- оформлення прокату;
- перегляд інформації про велосипед;
- онлайн оплата;

- перегляд історії прокатів;
- нагадування про завершення прокату;
- функція зворотного зв'язку;
- онлайн-підтримка користувачів;

Основні функції адміністратора:

- керування станціями прокату;
- керування велосипедами;
- управління користувачами;
- моніторинг прокатів;
- онлайн-підтримка користувачів;

Деякі функції, як-от «Редагування» та «Видалення», є розширеннями (Extend) до базової операції «Управління акаунтом».

Діаграма демонструє повноцінний функціонал системи підтримки оренди велосипедів – від реєстрації користувачів до управління прокатом і надання технічної підтримки, що забезпечує ефективну роботу як для клієнтів, так і для адміністраторів.

2.3. Специфікація функціональних та нефункціональних вимог

Важливим етапом проєкту є формулювання переліку функціональних та нефункціональних вимог. Перелік функціональних вимог додатку наведено у таблиці 2.13.

Таблиця 2.13 – Перелік функціональних вимог

Ідентифікатор вимоги	Назва вимоги (варіанта використання)	Атрибути вимог		
		Пріоритет	Трудність	Виконавець
ФВ-01	Авторизація	Високий	Середня	Розробник
ФВ-02	Реєстрація	– “ –	– “ –	– “ –
ФВ-03	Управління акаунтом	Середній	– “ –	– “ –
ФВ-04	Пошук велосипедів для оренди	Високий	– “ –	– “ –
ФВ-05	Оформлення оренди	– “ –	– “ –	– “ –
ФВ-06	Перегляд параметрів велосипеда	– “ –	– “ –	– “ –
ФВ-07	Онлайн оплата	– “ –	Висока	– “ –

Перелік нефункціональних вимог описує, яким має бути програмний додаток, його властивості. Специфікація нефункціональних вимог до вебдодатку, що проєктується, надано у таблиці 2.14.

Таблиця 2.14 – Перелік нефункціональних вимог

Ідентифікатор	Назва вимоги	Атрибути		
		Пріоритет	Трудність	Виконавець
1	2	3	4	5
1. Інтерфейс				
НФВ-01	Зручність управління	Високий	Низька	Дизайнер
НФВ-02	Якість дизайну	– “ –	– “ –	– “ –
НФВ-03	Масштабованість, гнучкість	– “ –	– “ –	– “ –
2. Надійність				
НФВ-04	Стабільність роботи додатку	Високий	Висока	Розробник
НФВ-05	Захист даних	– “ –	– “ –	– “ –
НФВ-06	Обробка помилок	– “ –	– “ –	– “ –
3. Безпека				

1	2	3	4	5
НФВ-07	Захист персональних даних	Високий	Середня	Інженер з безпеки
НФВ-08	Перешкода кібератакам	– “ –	– “ –	– “ –
НФВ-09	Аудит безпеки	– “ –	– “ –	– “ –

2.4 Вхідні та вихідні дані

Вхідними даними для роботи додатку є інформація про велосипеди, їх характеристики та параметри геолокації. Також вхідними даними є персональні дані користувачів та здійснені ними оренди велосипедів. Платформою для інформаційного забезпечення системи є база даних реляційного типу..

Вихідними даними додатку є звіти про орендуванні велосипедів, відомості про оплати, відмовим від оренди, виявлені порушення умов оренди. Передбачено можливість експортувати результати роботи додатку до інших програмних засобів через формат "*.xml".

2.5 Проектування системи

З метою забезпечення ефективного моделювання та документування системних процесів у роботі було використано CASE-засоби [1-3, 9,10]. При цьому першим етапом цієї роботи стало обґрунтоване обрання відповідного інструментарію для подальшого використання на всіх етапах проектування.

2.5.1 Вибір CASE-засобів для проектування системи

У межах роботи розглядалися такі CASE-засоби: Microsoft Visio та Ramus Educational. З огляду на функціональність, зручність і безкоштовну доступність було обрано Microsoft Visio для універсального моделювання та Ramus Educational – для створення IDEF0-моделей.

2.5.2 Проектування бази даних

У кваліфікаційній роботі було проведено розробку логічної та фізичної моделі даних [4-8] для онлайн-сервісу оренди велосипедів.

2.5.2.1 Концептуальне інфологічне проектування

Інформаційне забезпечення вебдодатку складається з трьох основних таблиць: User, Bikes та Transaction, які забезпечують зберігання ключових даних системи.

Таблиця «User» містить інформацію про користувачів, включаючи унікальний ідентифікатор, контактні дані, ім'я, пароль та статус підтвердження електронної пошти.

Таблиця «Bikes» зберігає дані про велосипеди: ідентифікатор, модель, координати, доступність і вартість оренди за годину.

Таблиця «Transaction» фіксує орендні операції, містить суму, час дії, а також зовнішні ключі bikeId і userId, що пов'язують транзакцію з відповідним велосипедом і користувачем. Деталізований опис атрибутів подано в таблиці 2.15.

Таблиця 2.15 – Словник бази даних додатку

№ з/п	Найменування елемента	Ідентифікатор	Тип і довжина
1	2	3	4
1	Унікальний код користувача	id_	SERIAL
2	Час реєстрації кабінету	cre_Ated_At	TIMESTAMP
3	Час оновлення запису	upd_Ated_At	TIMESTAMP
4	e-mail користувача	email	character
5	Телефон користувача	phoneNumber	character
6	Ім'я користувача	fullName	character
7	Підтвердження e-mail	isEmailConfirmed	boolean
8	Пароль користувача	password	character
9	Ідентифікатор ТЗ	id_	BIGSERIAL
10	Дата створення запису про ТЗ	cre_Ated_At	TIMESTAMP
11	Дата оновлення запису про ТЗ	upd_Ated_At	TIMESTAMP
12	Модель ТЗ	model	character
	Географічна широта	l_Atitude	

1	2	3	4
13	розташування ТЗ		FLOAT
14	Географічна довгота розташування ТЗ	longitude	FLOAT
15	Доступність ТЗ для оренди	available	boolean
16	Вартість оренди ТЗ	rentalPricePerHour	FLOAT
17	Ідентифікатор транзакції	id_	BIGSERIAL
18	Час створення запису	cre_Ated_At	TIMESTAMP
19	Час оновлення запису	upd_Ated_At	TIMESTAMP
20	Сума транзакції	amount	double
21	Час початку оренди	expiresIn	character
22	Час закінчення оренди	expiresOut	character
23	Ідентифікатор орендованого ТЗ	bikeId_	integer
24	Ідентифікатор особи, що орендувала ТЗ	userId_	integer

Обмеження атрибутів для означених елементів словника БД наведена у таблиці 2.16.

Таблиця 2.16 – Обмеження атрибутів

№	Ім'я атрибуту	Допустимі значення	Структура	Умова	За замовчуванням
1	id_	>= 0	–	NOT NULL	–
2	cre_Ated_At	–	дд.мм.пппп гг:хх:сс	– “ –	now()
3	upd_Ated_At	–	– “ –	– “ –	now()
4	email	–	–	NOT NULL UNIQUE	–
5	phoneNumber	–	–	– “ –	–
6	fullName	–	–	– “ –	–
7	isEmailConfirmed	–	–	– “ –	–
8	password	–	–	NOT NULL	–
9	id_	>= 0	–	– “ –	–
10	cre_Ated_At	–	дд.мм.пппп гг:хх:сс	– “ –	now()
11	upd_Ated_At	–	– “ –	– “ –	now()
12	model	–	–	– “ –	–

13	l_Atitude	–	–	– “ –	–
14	longitude	–	–	– “ –	–
15	available	–	–	– “ –	–
16	rentalPricePerHo	>0	–	– “ –	–
17	id	>= 0	–	– “ –	–
18	createdAt	–	дд.мм.рррр гг:хх:сс	–	now()
19	updatedAt	–	– “ –	–	– “ –
20	amount	>0	–	– “ –	–
21	expiresIn	–	–	– “ –	true
22	expiresOut	–	–	– “ –	– “ –
23	bikeId	>= 0	–	– “ –	–
24	userId	>= 0	–	– “ –	–

2.5.2.2 Проектування логічної та фізичної моделі даних

Виходячи з результатів попереднього аналізу було сформовано логічну Модель містить 3 основних таблиці і розроблена з урахуванням принципів реляційної моделі даних, що забезпечує ефективну реалізацію в сучасних СУБД. Фізична модель бази даних додатку наведена на рисунку 2.2. Модель виконана з використанням програми MS Visio і орієнтована на використання БД реляційного типу.

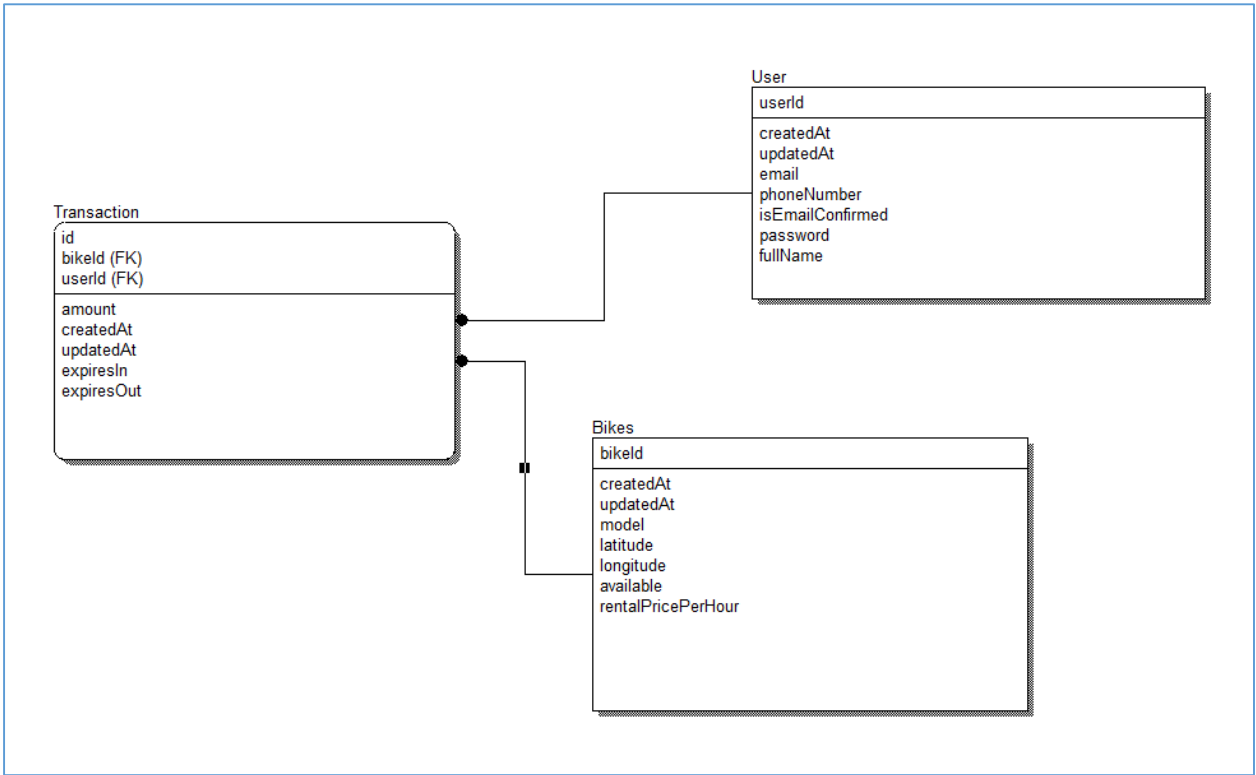


Рисунок 2.2 – Схема бази даних додатку

2.5.3 Проектування програмного забезпечення

Діаграми активності додатку для кожного процесу наведено на рисунках 2.3–2.11.

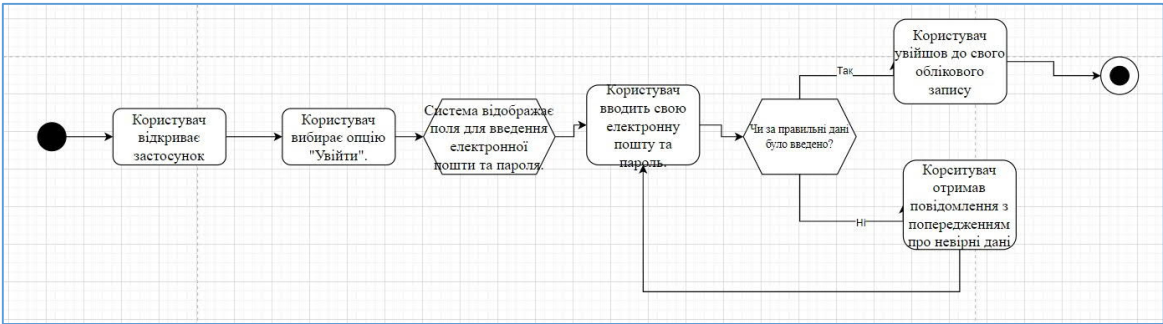


Рисунок 2.3 – Діаграма діяльності варіанта використання «Авторизація»

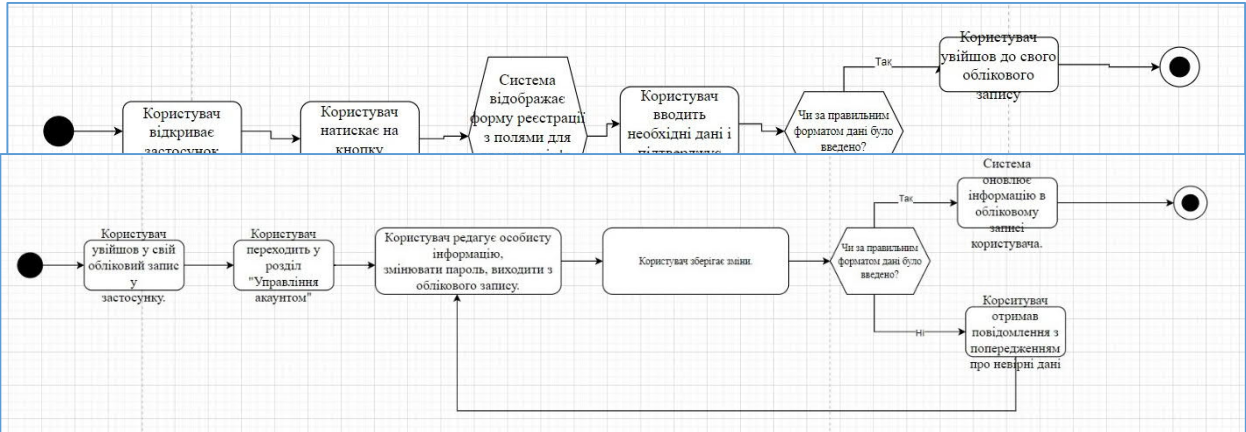


Рисунок 2.4 – Діаграма діяльності варіанта використання «Реєстрація»

Рисунок 2.5 – Діаграма діяльності варіанта «Управління акаунтом»

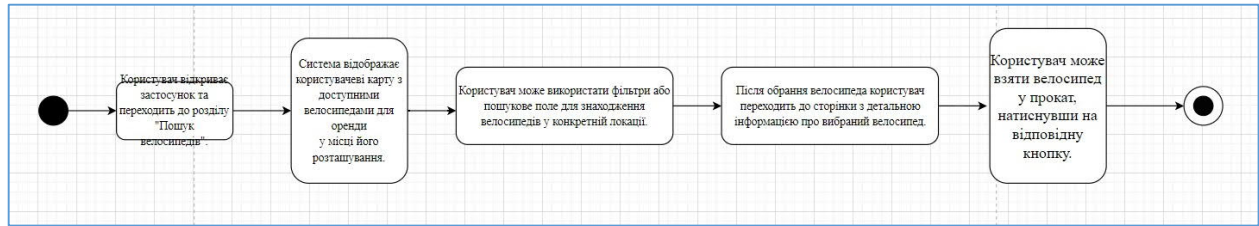


Рисунок 2.6 – Діаграма діяльності варіанта використання «Пошук ТЗ для оренди»



Рисунок 2.7 – Діаграма діяльності варіанта використання «Оформлення оренди»

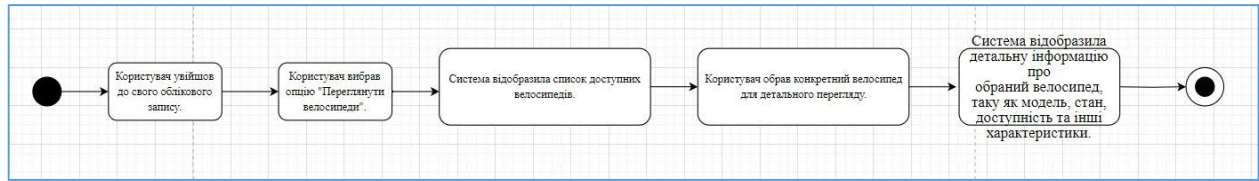


Рисунок 2.8 – Діаграма діяльності варіанта використання «Перегляд інформації про ТЗ»



Рисунок 2.9 – Діаграма діяльності варіанта використання «Оплата»

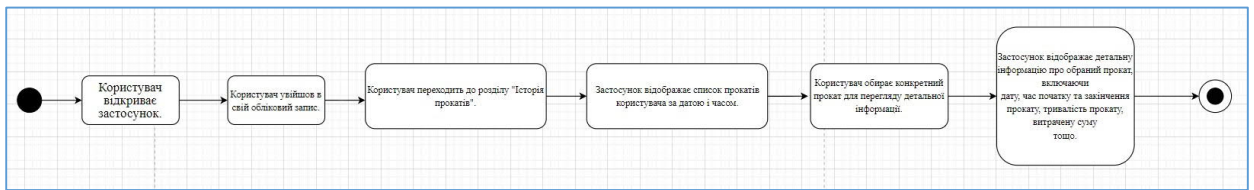


Рисунок 2.10 – Діаграма діяльності варіанта використання «Історія оренд»

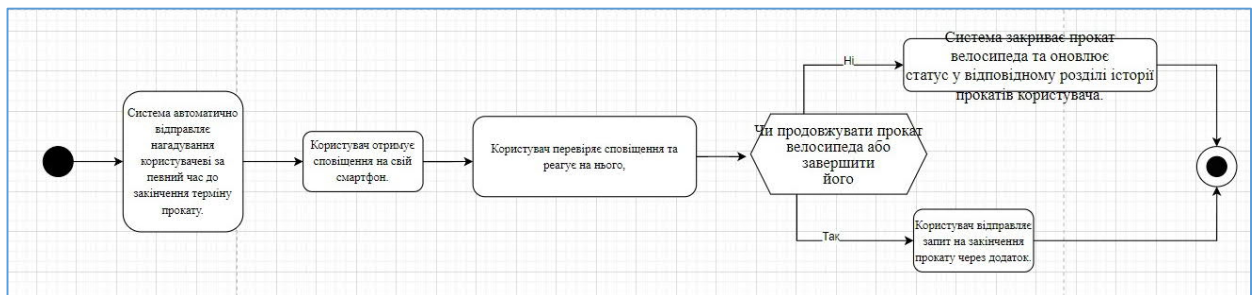


Рисунок 2.11 – Діаграма діяльності варіанта використання «Нагадування про завершення оренди»

Висновки до розділу

Другий розділ кваліфікаційної роботи присвячено всебічному аналізу предметної області, на основі якого сформовано концептуальну модель системи та побудовано IDEF0-діаграми, що відображають основні бізнес-процеси. У розділі також визначено функціональні можливості вебдодатку, розроблено структуру інформаційного забезпечення, а також представлено логічну та фізичну моделі даних, необхідні для реалізації системи в реляційному середовищі.

РОЗДІЛ 3

ПРОГРАМНЕ ТА ТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ

3.1 Засоби розробки та розгортання програмного продукту

Для реалізації вебдодатку було обрано операційну систему Windows, враховуючи її продуктивність, надійність і сумісність із необхідним програмним забезпеченням.

Серед розглянутих СУБД – Microsoft Access, MS SQL Server та PostgreSQL. З огляду на функціональні можливості та економічну доцільність, оптимальними варіантами визначено Microsoft Access або PostgreSQL.

3.2 Вимоги до технічного та програмного забезпечення

Апаратне забезпечення:

- Процесор: 2 ядра (мінімум)
- Оперативна пам'ять: 8 ГБ RAM і більше
- Вільне місце на диску: щонайменше 1 ГБ.

Програмне забезпечення:

- Операційна система: Windows 11;
- Мова програмування: Python / PHP / Node.js;
- Система керування базами даних: MySQL;
- HTML5, CSS3, JavaScript;
- Середовище: VS Code / PhpStorm / WebStorm;
- Браузери: Google Chrome, Mozilla Firefox, Microsoft Edge, Safari (останні версії).

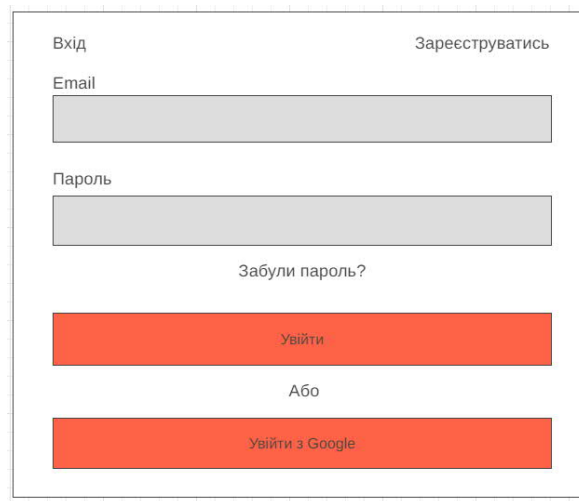
Додаткове ПЗ:

- Платіжна система: інтеграція з Stripe / LiqPay / WayForPay;
- Карта для пошуку: Google Maps API / OpenStreetMap;
- Засоби тестування: Postman, Jest, Cypress;

- Система керування версіями: Git + GitHub/GitLab;

3.3 Проектування інтерфейсу користувача

Інтерфейс додатку організовано через низку вікон, кожна з яких відповідає за певний функціонал застосунку. Для розробки інтерфейсу були створені відповідні варфрейми (мокапи), – схематичні макети елементів інтерфейсу. При створенні варфреймів було використано вебзастосунок wireframe.cc. Результати роботи наведені на рисунках 3.1–3.6.



The wireframe shows a login interface within a rectangular frame. At the top left is the text 'Вхід' (Login) and at the top right is 'Зареєструватись' (Register). Below these are two input fields: the first is labeled 'Email' and the second is labeled 'Пароль' (Password). Below the password field is a link that says 'Забули пароль?' (Forgot password?). At the bottom of the form are two red buttons: the top one is labeled 'Увійти' (Login) and the bottom one is labeled 'Увійти з Google' (Login with Google), preceded by the word 'Або' (Or).

Рисунок 3.1 – Мокап сторінки авторизації

Реєстрація

Вхід

Прізвище та ім'я

Email

Номер телефону

Пароль

Зареєструватись

Або

Зареєструватись з Google

Рисунок 3.2 – Мокап сторінки реєстрації

Balance: X

Ім'я Прізвище

Інтерактивна гугл-мапа

Рисунок 3.3 – Мокап головної сторінки

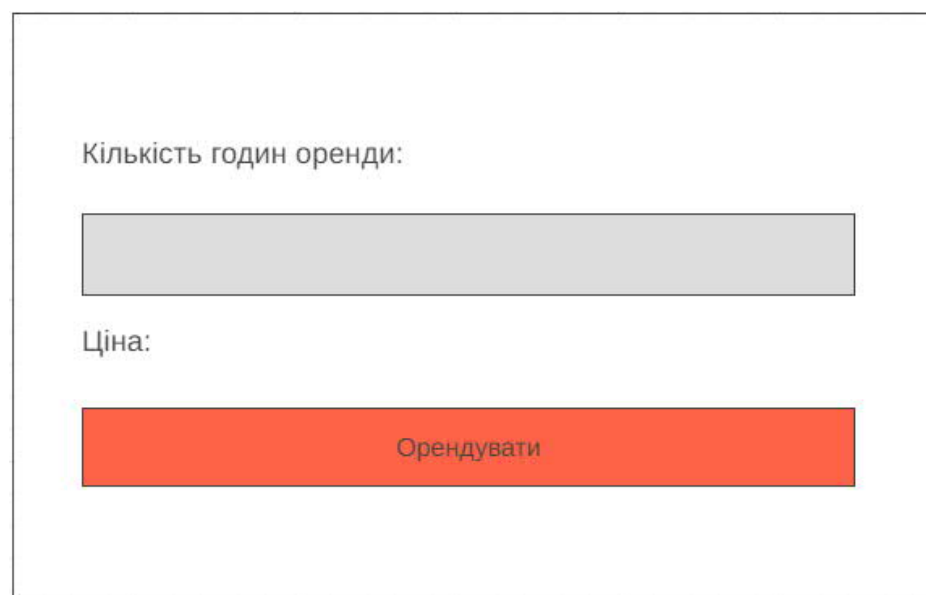


Зображення велосипеда

Модель велосипеда
Ціна за годину

Орендувати

Рисунок 3.4 – Мокап сторінки оформлення оренди

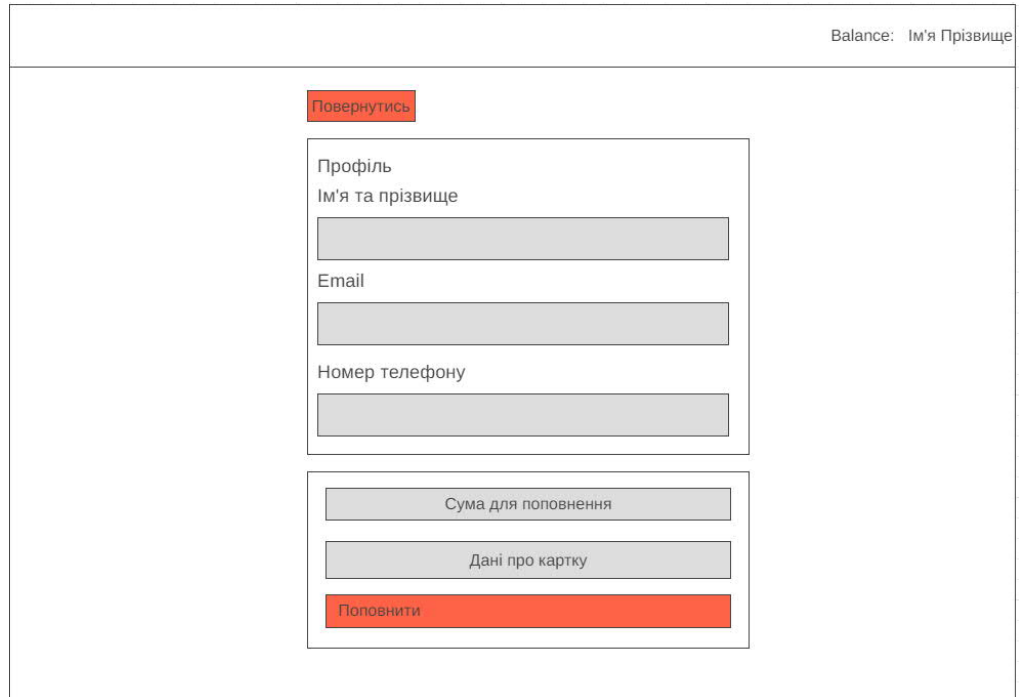


Кількість годин оренди:

Ціна:

Орендувати

Рисунок 3.5 – Мокап сторінки оплати



Balance: Ім'я Прізвище

[Повернутись](#)

Профіль

Ім'я та прізвище

Email

Номер телефону

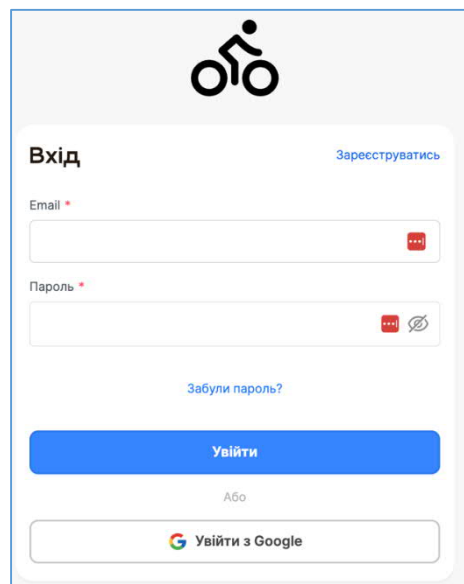
Сума для поповнення

Дані про картку

[Поповнити](#)

Рисунок 3.6 – Мокап сторінки профілю користувача

Приклади вікон інтерфейсу додатку, що розробляється, наведено на рисунках 3.7–3.10.





Вхід [Зареєструватись](#)

Email *

Пароль *

[Забули пароль?](#)

[Увійти](#)

Або

 [Увійти з Google](#)

Рисунок 3.7 – Сторінка входу в акаунт



Реєстрація

Вхід

Прізвище та ім'я *

Вкажіть ваше прізвище та ім'я

Email *

example@gmail.com

Номер телефону *



+38 (000) 000-00-00

Пароль *

Вигадайте надійний пароль

☒ Я підтверджую, що ознайомився(-лася) з [Політикою конфіденційності](#) та погоджуюся з [Умовами користування](#).

Зареєструватись

Або

 Зареєструватись з Google

Рисунок 3.8 – Сторінка реєстрації

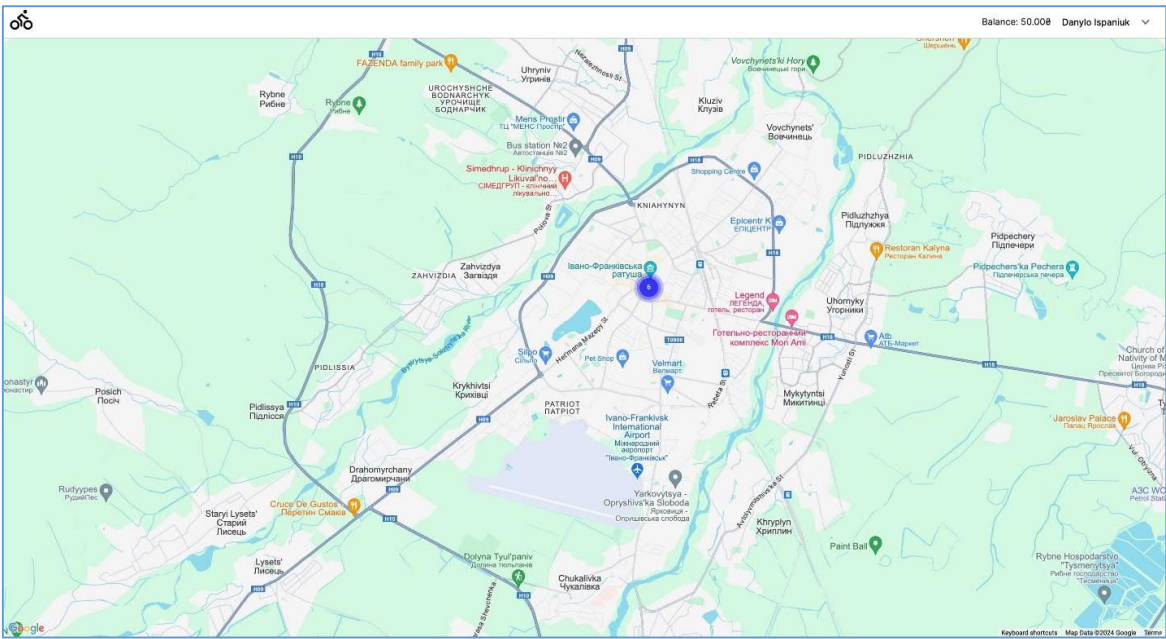


Рисунок 3.9 – Основна сторінка додатку

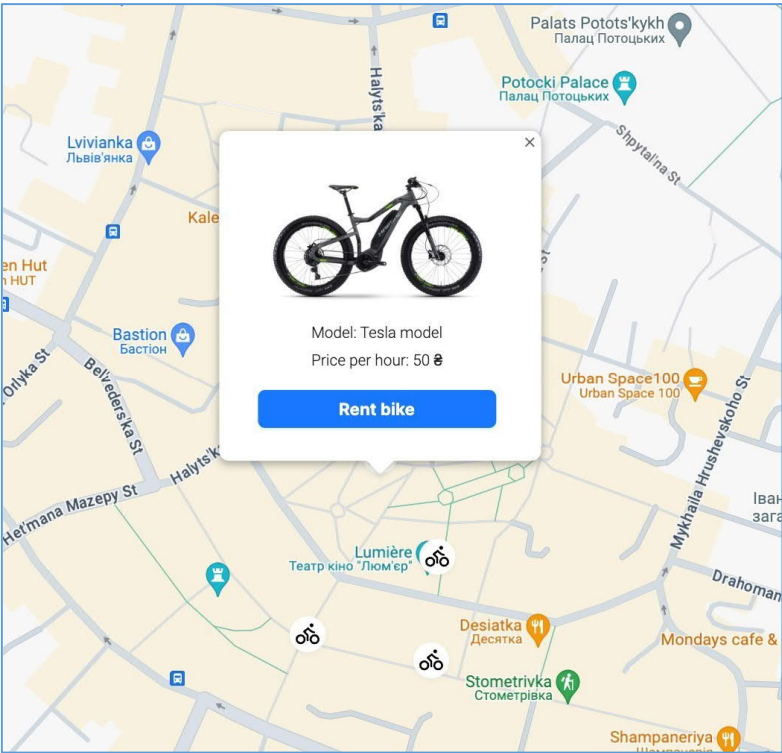


Рисунок 3.10 – Вікно оренди велосипедів

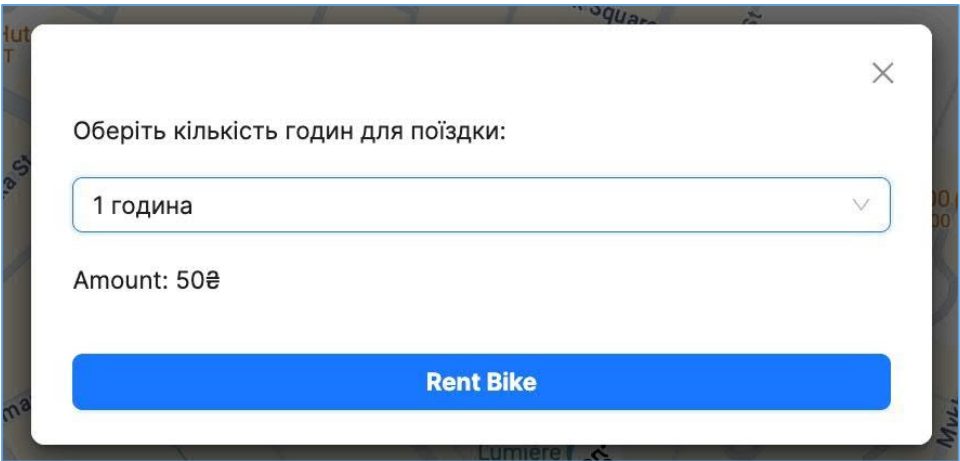


Рисунок 3.11 – Вікно оплати оренди

The image shows a user profile window with two sections. The top section, titled 'Profile', contains fields for 'Full name' (Danylo Ispaniuk), 'Email' (ispanyuk13@gmail.com), and 'Phone Number' (+380997516903) with a Ukrainian flag icon. The bottom section, also titled 'Profile', contains a card number field with a '101' icon and 'грн.' currency, and a 'Card number' field with 'MM / YY CVC' placeholder. A blue 'Pay' button is at the bottom.

Рисунок 3.12 – Вікно профіля користувача

Висновки до розділу

У третьому розділі кваліфікаційної роботи проведено обґрунтування вибору програмного та технічного забезпечення, необхідного для реалізації вебдодатку та його розгортання. На основі аналізу характеристик було обрано операційну систему Windows як оптимальне середовище розробки. Серед систем управління базами даних рекомендовано Microsoft Access або PostgreSQL з огляду на функціональність та економічну доцільність.

Визначено мінімальні вимоги до апаратного і програмного забезпечення для клієнтської частини, що гарантують стабільну роботу застосунку.

Також виконано проектування інтерфейсу користувача шляхом створення варфреймів у середовищі wireframe.cc, що дало змогу візуалізувати

структуру основних екранів вебдодатку. Представлено макети ключових сторінок, які демонструють логіку користувацької взаємодії з системою.

РОЗДІЛ 4

ОХОРОНА ПРАЦІ

4.1 Регулювання питань охорони праці на законодавчому рівні

Охорона праці являє собою систему заходів, спрямованих на забезпечення та створення безпечного робочого місця для працівників усіх організацій – незалежно від форми власності та виду діяльності [20, 21].

Законодавство України про охорону праці спирається, перш за все, на закон «Про охорону праці» від 1992 року, який визначає гарантії прав на охорону праці, особливості організації охорони праці тощо та підкреслює роль роботодавця у створенні безпечних та нешкідливих умов праці та профілактиці виробничого травматизму та професійної захворюваності. Для реалізації цих та інших завдань роботодавець повинен забезпечити функціонування системи управління охороною праці та здійснювати регулярний контроль та оцінку її дієвості.

Моніторинг показників ефективності функціонування системи управління охороною праці відповідно до заздалегідь визначених планів та стандартів повинен бути обов'язком керівництва. Моніторинг посилює відданість керівництву цілям безпеки та охорони праці в цілому та допомагає розвивати позитивну культуру безпеки та охорони праці, винагороджуючи позитивну роботу, виконану для контролю ризику. Потрібні два типи моніторингу [20, 21]:

- активні системи, які контролюють проектування, розробку, встановлення та експлуатацію механізмів управління, систем безпеки та запобіжних заходів на робочому місці;
- реактивні системи, які відслідковують нещасні випадки, погіршення стану здоров'я, інциденти та інші свідчення про неналежне виконання вимог охорони праці та виробничої безпеки.

1. Активний моніторинг.

Кожна організація повинна збирати інформацію для адекватного розслідування причин неналежної роботи чи умов. Повинні бути встановлені та підтримуватись документовані процедури для виконання цих заходів на регулярній основі для ключових операцій. Система моніторингу повинна включати:

- визначення відповідних даних, які необхідно зібрати, та необхідної точності результатів;
- контроль за досягненням конкретних планів, встановлення критеріїв ефективності та цілей;
- встановлення необхідного контрольно-вимірювального обладнання та оцінка його точності та надійності;
- калібрування та регулярне технічне обслуговування цього обладнання разом з документованими записами як процедур, що використовуються, так і отриманих результатів;
- аналіз та запис зібраних даних моніторингу та документовані дії, які необхідно вжити у разі порушення результатів критеріями ефективності;
- оцінка всіх даних у рамках огляду управління безпекою та охороною праці;
- документовані процедури для перевірки моніторингу та наслідків для безпеки та охорони праці майбутніх змін у робочих системах.

Методи, які слід використовувати для активного вимірювання системи управління охороною праці, включають [20, 21]:

- систематичні перевірки робочих процесів або послуг для моніторингу конкретних цілей, наприклад, щотижневі, щомісячні або щоквартальні звіти;
- систематичний огляд оцінок ризиків організації для визначення того, чи функціонують вони так, як передбачалося, чи їх необхідно оновити, а також чи впроваджуються необхідні поліпшення;

- інспекції підприємств або обладнання, наприклад, обов'язкові інспекції та сертифікація підприємств;
- відбір проб навколишнього середовища на наявність пилу, хімічної пари, шуму або біологічних агентів;
- аналіз записів системи управління охороною праці.

2. Реактивний моніторинг.

Необхідно створити систему внутрішньої звітності з усіх нещасних випадків (включаючи випадки захворювань) та інцидентів недотримання системи управління безпекою та охороною праці, щоб накопичений досвід можна було використовувати для покращення системи управління. Організація повинна заохочувати відкритий та позитивний підхід до звітності та подальших дій, а також запровадити систему, яка гарантує виконання вимог до звітності.

Організація повинна встановити процедури розслідування нещасних випадків та інцидентів для виявлення їх причин, включаючи можливі недоліки у системі управління безпекою та охороною праці. Особи, відповідальні за розслідування нещасних випадків та інцидентів, повинні бути визначені, а розслідування має включати плани коригуючих дій, які включають заходи щодо:

- якнайшвидшого відновлення відповідності встановленим вимогам;
- запобігання повторення аналогічних небезпечних подій та ситуацій;
- оцінка та пом'якшення будь-яких несприятливих наслідків для безпеки та здоров'я;
- розгляд оцінок ризику;
- оцінка наслідків запропонованих заходів щодо підвищення ефективності функціонування системи управління охороною праці.

4.2 Виявлення потенційних небезпек стосовно об'єкту проектування

У дипломній роботі бакалавра розглянуто завдання із розробки веб-застосунку для підтримки сервісу оренди велосипедів. Структура працівників сервісу оренди велосипедів може включати наступне [20, 21]:

- оператори (офіс/колцентр): здійснюють роботу з клієнтами, оформлення заявок, техпідтримку, робота переважно пов'язана із використанням персонального комп'ютера;
- техніки/механіки: займаються обслуговуванням та ремонтом велосипедів, зарядкою електровелосипедів, заміною деталей;
- працівники виїзних бригад/кур'єри: здійснюють переміщення велосипедів, доставку, розвантаження/навантаження;
- контролери на точках прокату: здійснюють прийом/видачу велосипедів, огляд, первинну перевірку стану.

Нижче наведено результати аналізу умов праці працівників сервісу оренди велосипедів.

Таблиця 4.1 – Аналіз умов праці працівників сервісу оренди велосипедів

Фактор	Хто наражається на небезпеку	Наслідки впливу
1	2	3
Метеоумови (спека, холод, дощ, вітер)	Виїзні працівники	Переохолодження, тепловий удар, простудні захворювання
Загрози у дорожньому середовищі	Кур'єри, бригади	ДТП, травми під час падіння
Робота з інструментами та рухомими частинами	Механіки	Порізи, затискання, уколи, забиття
Підвищені фізичні навантаження	Усі, хто вручну переміщує велосипеди	Болі у спині, суглобові захворювання, перевантаження м'язів

1	2	3
Шум від обладнання (насосів, інструментів)	Механіки, ремонтники	Підвищена стомлюваність, дратівливість
Вібрації (при використанні електроінструменту)	Механіки	Вібраційна хвороба при тривалій експозиції
Мастильні та засоби для чищення	Техніки	Подразнення шкіри, алергії, контактні дерматити
Акумулятори електровелосипедів (літій-іонні)	Електромеханіки	Ризик займання, хімічний опік при пошкодженні
Контакт із біологічними забруднювачами (бруд, слина, пил, продукти життєдіяльності птахів/тварин)	Виїзні працівники	Алергії, інфекційні захворювання, кишкові інфекції при випадковому занесенні в організм
Монотонність роботи (у операторів)	Колл-центр, диспетчери	Втрата уваги, втома, дратівливість
Високий темп праці	Кур'єри, техпідтримка	Стрес, нервова напруга
Робота з незадоволеними клієнтами	Усі, хто контактує з клієнтами	Емоційне вигорання, тривожність
Робота у темний час доби	Виїзні працівники	Порушення сну, стомлюваність, підвищений ризик травм

Ці та інші фактори та їх наслідки обумовлюють потребу у розробці та впровадженні заходів із покращення умов праці працівників сервісу оренди велосипедів.

4.3 Дослідження ризику реалізації потенційних небезпек на об'єкті проектування та розробка заходів щодо їх попередження

Оцінка ризиків є тим інструментом (процедурою), що допомагає організації визначитися із переліком небезпек, присутніх на робочих місцях працівників підприємства або організації, які представляють собою найбільшу загрозу та потребують першочергового впровадження заходів реагування.

Методи оцінки ризиків варіюються в залежності від ресурсів організації, цілей, які вона перед собою ставить, наявності достатньо кваліфікованих фахівців тощо. Одним із методів, відомих своєю простотою та зручністю, є матриця оцінки ризиків, яка являє собою таблицю, що ураховує два критерії оцінювання: серйозність небезпеки та ймовірність її виникнення.

Переваги використання матриці ризиків включають [20, 21]:

- дані представлені у наочній та доступній для розуміння формі;
- її можна легко змінити чи скоригувати залежно від ситуації;
- метод забезпечує стандартизований підхід до оцінки ризиків, що спрощує порівняння та визначення пріоритетності ризиків;
- метод також дозволяє організаціям ефективніше розподіляти ресурси. Розуміючи рівень ризику, пов'язані з різними подіями, організації можуть розподіляти ресурси відповідним чином.

Недоліки використання матриці ризиків включають [20, 21]:

- часто надмірно спрощений метод. Надмірне спрощення може не відображати складність певних ризиків, що призводить до неповної оцінки ризиків;
- статичність: матриці ризиків не враховують, як ризики можуть змінюватися або розвиватися з часом. Щоб підтримувати їхню ефективність, їх слід регулярно переглядати та оновлювати;
- суб'єктивність: оцінка ймовірності та впливу передбачає певний ступінь суб'єктивності. Різні фахівці можуть надавати значення по-різному, що призводить до непослідовних результатів;

- якість даних: їхня точність багато в чому залежить від доступності відповідних даних;
- у деяких випадках категорії можуть бути недостатньо конкретними для точної оцінки ризиків.

Таблиця 4.2 – Матриця оцінювання ризиків

Хімічний опік при обслуговуванні акумуляторів електровелосипедів				
Визначення категорії серйозності небезпеки		Визначення рівня ймовірності небезпеки		Індекс ризику небезпеки
Вид, категорія	Опис	Вид, рівень	Опис	
III – гранична	Ушкодження шкіри та прилеглих тканин	C	Випадкова подія, яка може трапитися під час обслуговування акумуляторів	III C – небажаний (гранично допустимий) рівень ризику

Робота з акумуляторами електровелосипедів – це діяльність, пов’язана з підвищеною пожежо- та електронебезпекою, впливом хімічних речовин, а також фізичними ризиками, що потребує відповідних рекомендацій та заходів безпеки.

1. Організація безпечного робочого простору.

Робота повинна здійснюватися у спеціально призначеному для цього приміщенні: окрема вентильована зона (з витяжкою) для зарядки, діагностики та утилізації акумуляторів.

Температурний режим: зберігання та робота з батареями при температурі від +10 до +25°C.

2. Засоби індивідуального захисту (ЗІЗ).

Рекомендовані ЗІЗ: діелектричні рукавички, захисні окуляри від розбризкування електроліту.

3. Поводження з акумуляторами та зарядними пристроями.

Перевірка акумуляторів на цілісність, здуття, корозію, замикання перед підключенням.

Використання тільки оригінальних зарядних пристроїв із захистом від перезаряду, перегріву та короткого замикання.

Заборона зберігання/заряджання батарей поблизу джерел тепла, води та на сонці.

Встановлення датчиків диму та температурних сенсорів.

Контроль процесів зарядки – не залишати без нагляду.

4. Запобігання пожежам та коротким замиканням.

Обов'язкова наявність вогнегасника класу D або вуглекислотного.

Заземлення робочого місця.

Використання силового обладнання з пристроєм захисного відключення та автоматичним відключенням живлення.

Обов'язкова перевірка кабелів та роз'ємів на наявність пошкоджень.

5. Безпека при ремонті та утилізації.

Заборонено розбирання герметичних елементів (особливо літій-іонних), якщо немає відповідного досвіду.

Пошкоджені чи несправні АКБ обов'язково потрібно здавати в ліцензовану організацію.

Тимчасове зберігання – у металевій ящику з наповнювачем (наприклад, піском).

6. Підготовка та навчання персоналу.

Обов'язкове навчання з поводження з літійовими акумуляторами; протипожежної безпеки; надання першої допомоги при хімічних опіках, короткому замиканні; утилізації небезпечних відходів.

Проведення інструктажів та тренувань з надзвичайних ситуацій: що робити при загорянні, короткому замиканні, пошкодженні акумуляторних батарей.

7. Психофізіологічні та організаційні заходи.

Ротація завдань для зниження стомлюваності та підвищення концентрації.

Переривання роботи у разі виявлення ознак нагрівання батарей, запаху або диму.

Обов'язковий регламент ведення журналу обслуговування акумуляторів.

Висновки по розділу

У розділі проведено аналіз законодавства України з питань охорони праці та визначено особливості діяльності, яку повинен провадити в організації роботодавець для реалізації задач із створення безпечного робочого середовища.

Сфера поводження з літій-іонними акумуляторними батареями, яка розглядається у розділі, є потенційно високоризикованою, особливо при порушенні умов експлуатації. Розглянуті рекомендації спрямовані на запобігання пожежам та вибухам, захисту здоров'я та життя працівників, забезпечення відповідності стандартам електробезпеки та охорони праці.

ЗАГАЛЬНІ ВИСНОВКИ

У кваліфікаційній роботі було проведено комплексний аналіз предметної області вебдодатку для оренди велосипедів, потреб користувачів і вимог до програмного забезпечення, що дозволило сформулювати чіткий перелік функціональних і нефункціональних вимог. Результатом стала розробка зручного, ефективного та інтуїтивно зрозумілого інструменту, спрямованого на популяризацію активного способу життя й екологічно орієнтованого транспорту.

Аналіз предметної області та існуючих рішень забезпечив визначення оптимального функціоналу, який відповідає актуальним запитам цільової аудиторії та створює конкурентні переваги. Впроваджено функції, що підвищують зручність використання й ефективність взаємодії з системою, що в цілому сприяє якісному користувацькому досвіду.

У роботі було спроектовано інформаційне забезпечення системи, структура БД, розроблено інтерфейс додатку.

Також у роботі було здійснено програмну реалізацію додатку та проведено її тестування.

Результати кваліфікаційної роботи відповідають сучасним технічним вимогам, отриманий додаток може використовуватись для організації процесу оренди велосипедів широким колом фізичних та юридичних осіб.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Николайчук Я. М. Проектування спеціалізованих комп'ютерних систем. Навч. посіб. / Я. М. Николайчук. – Львів: Лира-К, 2010. – 216 с.
2. Литвин В.В. Проектування інформаційних систем. Навч. посіб / В. В. Литвин. – Львів: Лира-К, 2011. – 188 с.
3. Авраменко В.С., Авраменко А.С. Проектування інформаційних систем: навчальний посібник / В.С. Авраменко, А.С. Авраменко. - Черкаси: Черкаський національний університет ім. Б. Хмельницького, 2017. - 434 с.
4. Анісімов А.В. Інформаційні системи та бази даних: Навчальний посібник для студентів факультету комп'ютерних наук та кібернетики. / Анісімов А.В., Кулябко П.П. – Київ. – 2017. – 110 с.
5. Боровик В. М. Автоматизоване робоче місце. Проектування інформаційних систем і баз даних. Навч.посіб. / В. М. Боровик, В. П. Гамаюн. – К.: Видавнича група ВНУ, 2010. – 321 с.
6. Пасічник В. В. Організація баз даних та знань / В. В. Пасічник, В. А. Резниченко. – К.: Видавнича група ВНУ, 2006. – 384 с.
7. Хомоненко А. Д. Бази даних. Підручник для вузів / А. Д. Хомоненко, В.М. Циганков, М. Г. Мальцев. – Київ: Корона-принт, 2004. – 737 с.
8. Золотухіна О.А. Якість та тестування інформаційних систем: навч. посіб. / О.А. Золотухіна. – Київ: ННІТ ДУТ, 2020. – 128 с.
9. ДСТУ 3008:2015 Інформація та документація. Звіти у сфері науки і техніки. Структура та правила оформлювання. – Київ : ДП УкрНДНЦ, 2015. – 28 с.
10. ДСТУ 8302:2015. Інформація та документація. Бібліографічне посилання. Загальні положення та правила складання / Нац. стандарт України. – Київ : ДП «УкрНДНЦ», 2016. – 18 с.

11. Створення діаграми варіантів використання [Електронний ресурс] – Режим доступу: https://dut.edu.ua/ua/news-1-626-7758-zastosuvannya-uml-v-diplomnih-robotah_kafedra-kompyuternih-nauk-ta-informaciy-nih-tehnologiy
12. IDEF0. Знайомство з нотацією та приклади використання. [Electronic resource]. Access mode: <https://www.trinion.org/blog/idef0-znakomstvo-s-notaciey-i-primer-ispolzovaniya>
13. Методологія IDEF0 [Electronic resource]. Access mode: https://stud.com.ua/87184/ekonomika/metodologiya_idef0
14. Золотухіна О.А. Якість та тестування інформаційних систем: навч. посіб. / О.А. Золотухіна. – Київ: ННІТ ДУТ, 2020. – 128 с.
15. Як тестувати веб-сайт: основні етапи і поради. [Electronic resource]. Access mode: <https://brainlab.com.ua/uk/blog-uk/yak-testuvati-veb-sayt-osnovn-etapi-poradi>
16. Використання Docker для розгортання web-додатка. [Electronic resource]. Access mode: <https://avivi.pro.ua/blog/vikoristovu-mo-docker-dlya-rozgortannya-web-dodatka-bitrix/>
17. Керівництво користувача з розробки діаграми активності UML. [Electronic resource]. Access mode: <https://www.mindonmap.com/uk/blog/uml-activity-diagram/>
18. Умови прокату та правила безпеки при оренді велотранспорту. [Electronic resource]. Access mode: <https://velokratia.com.ua/terms/>
19. Основні завдання та етапи роботи для backend розробки. [Electronic resource]. Access mode: <https://careers.easternpeak.com/blog/guide-to-a-profession-of-backend-developer/>
20. Закон України «Про охорону праці». – Офіційний сайт Верховної Ради України. – Режим доступу: <https://zakon.rada.gov.ua/laws/show/2694-12>.
21. Безпека під час роботи з акумуляторами: як запобігти аваріям і псуванню обладнання. – Режим доступу: <https://vinur.com.ua/ua/aboutus/usefull-info/articles/bezpeka-pid-chas-roboty-z-akumulyatoramy-yak-zapobigty-avariyam-i-psuvannyu->

[obladnannya?srsltid=AfmBOopx3dg-KaPV0qSex3uEDW3wF3oqbEUP_B8GOrsEWhdyG1rDNTJp.](#)

ДОДАТОК А

Специфікації варіантів використання

Таблиця А.1 – Специфікація варіанту «Авторизація»

Назва варіанта використання	Авторизація
Опис варіанта використання	Користувач виконує процедуру авторизації для отримання доступу до особистого облікового запису.
Актори	Користувач
Передумова	Користувач вже має обліковий запис у застосунку.
Основний сценарій	1. Користувач відкрив додаток 2. Користувач вибрав «Увійти». 3. Система запитує логін та пароль. 4. Користувач вводить логін та пароль. 5. Система перевіряє логін та пароль. 6. Перевірка введених даних 7. Дані коректні: користувач отримує доступ до кабінету. 8. Дані некоректні: користувач отримує повідомлення про помилку.
Альтернативний сценарій	Крок 6.2: користувач повторно ввести персональні дані або використовує опцію «Відновити пароль».
Виключення	Якщо користувач не має облікового запису, йому надається можливість зареєструватися в системі.

Таблиця А.2 – Специфікація варіанту «Реєстрація»

Назва варіанта використання	Реєстрація
1	2
Опис варіанта використання	Користувач реєструється в системі.
Актори	Користувач
Передумова	Користувач має вебдодаток та доступ до мережі Інтернет застосунку.
Основний сценарій	1. Користувач відкриває додаток. 2. Користувач натискає «"Реєстрація» . 3. Відкривається форма реєстрації. 4. Користувач вводить дані. 5. Користувач підтверджує персональні дані. 6. Система перевіряє дані користувача. 5.1. Дані коректні: система формує новий обліковий запис. 6.2. Дані некоректні користувач отримує повідомлення про помилку. 7. Повідомлення про успішну реєстрацію.
Альтернативний сценарій	Користувач має обліковий запис і натискає «Увійти».. Користувач закриває додаток.
Виключення	Дані користувача некоректні: система пропонує ввести коригування

Таблиця А.3 – Специфікація варіанту «Управління акаунтом»

Назва варіанта використання	Управління акаунтом
1	2
Опис варіанта використання	Управління акаунтом, редагування профіля користувача.
Актори	Користувач
Передумова	Користувач має пройти авторизацію.
Основний сценарій	1. Користувач зайшов до персонального кабінету. 2. Користувач натискає «Управління акаунтом». 3. Користувач редагує дані. 4. Користувач зберігає дані. 5. Система оновлює персональні дані користувача.
Альтернативний сценарій	Зміна пароля., підтвердження змін. Перевірка пароля. Збереження пароля в обліковому записі. Повідомлення про зміну пароля. Вихід з облікового запису. Натиснути кнопку «Вийти». Завершення сеансу користувача.
Виключення	Управління акаунтом
Опис варіанта використання	Незареєстрований користувач: заборонено управляти акаунтом. Невідповідність пароля: повідомлення про помилку.

Таблиця А.4 – Специфікація варіанту «Пошук велосипедів на прокат»

Назва варіанта використання	Пошук велосипедів на прокат
Опис варіанта використання	Користувач шукає велосипед для оренди.
Актори	Користувач
Передумова	Користувач має бути авторизованим
Основний сценарій	1. Користувач натискає «Пошук велосипедів». 2. Додаток виводить мапу з доступними велосипедами. 3. Користувач включає пошукові фільтри, обирає велосипед. 4. Користувач переходить до сторінки описом велосипеда. 5. Користувач орендє велосипед.
Альтернативний сценарій	Пропозиції оренди відсутні: пропозиція зробити інший вибір.
Виключення	Користувач не авторизований: обмеження доступу.

Таблиця А.5 – Специфікація варіанту «Пошук велосипедів на прокат»

Назва варіанта використання	Оформлення прокату
Опис варіанта використання	Користувач оформлює прокат велосипеда через застосунок.
Актори	Користувач
Передумова	Користувач повинен бути авторизованим в системі.
Основний сценарій	1. Користувач відкриває додаток. 2. Натискає «Оформити прокат». 3. Додаток виводить перелік велосипедів. 4. Користувач обирає велосипед. 5. Користувач вводить тривалість оренди, робить оплату. 6. Система активізує оренду, розблоковує велосипед. 7. Користувач отримує велосипед. 8. По завершенню користування користувач повертає велосипед.
Альтернативний сценарій	Велосипед недоступний: пропозиція обрати інший варіант. Порушено термін оренди: блокування оренди, перерахунок оплати.
Виключення	Користувач не авторизований: обмеження доступу. Оплату не здійснено: повідомлення користувачу та повторна спроба.

Таблиця А.6 – Специфікація варіанту «Перегляд інформації про велосипед»

Назва варіанта використання	Перегляд інформації про велосипед
1	2
Опис варіанта використання	Користувач переглядає дані про доступні велосипеди.
Актори	Користувач
Передумова	Користувач авторизований.
Основний сценарій	1. Користувач увійшов до персонального кабінету. 2. Вибрав опцію «Переглянути велосипеди». 3. Додаток показав перелік доступних велосипедів. 4. Користувач обрав велосипед для перегляду параметрів. 5. Додаток показує інформацію про велосипед.
Подія	Пропозиції велосипедів відсутні: повідомлення помилку.
Альтернативний сценарій	Перегляд інших пропозицій.
Виключення	Користувач не має аккаунту: пропозиція реєстрації.

Таблиця А.7 – Специфікація варіанту «Онлайн оплата»

Назва варіанта використання	Онлайн оплата
Опис варіанта використання	Користувач робить оплату
Актори	Користувач
Передумова	Користувач авторизований
Основний сценарій	1 .Обрано пункт «Оплатити». 2 .Активізується сторінка оплати. 3 .Введення даних платежу. 4 .Перевірка даних, здійснення платежу. 5 .Підтвердження оплати. 6 .Розблокування велосипеду.
Альтернативний сценарій	Дані платежу неправильні: повідомлення про помилку, повторне введення. Платіж не авторизований: повідомлення, повторна спроба.
Виключення	Платіжна система недоступна: повідомлення, повторна спроба

Таблиця А.8 – Специфікація варіанту «Перегляд історії прокатів»

Назва варіанта використання	Перегляд історії прокатів
Опис варіанта використання	Користувач бачить історію оренди.
Актори	Користувач
Передумова	Користувач авторизований, є історія оренд.
Основний сценарій	1. Користувач заходить до кабінету. 2. Користувач натискає «Історія прокатів». 3. Відображається список оренд. 4. Користувач вибирає факт оренди. 5. Додаток виводить дані про обраний факт оренди.
Альтернативний сценарій	-
Виключення	Історія орендування відсутня: повідомлення про помилку.

Таблиця А.9 – Специфікація варіанту «Нагадування про завершення прокату»

Назва варіанта використання	Нагадування про завершення прокату
1	2
Опис варіанта використання	Користувач отримує повідомлення про завершення оренди.
Актори	Користувач
Передумова	Користувач повинен бути

1	2
Основний сценарій	1 .Додаток відправляє нагадування про закінчення терміну оренди. 2 .Користувач отримує повідомлення. 3 .Користувач реагує на повідомлення. 4 .Вибрано закінчення оренди: користувач відправляє запит на закінчення оренди. 5 .Додаток завершує оренду, оновлює історію.
Альтернативний сценарій	Користувач може ігнорувати сповіщення та продовжити користуватися велосипедом.
Виключення	-

Таблиця А.10 – Специфікація варіанту «Функція зворотного зв'язку»

Назва варіанта використання	Функція зворотного зв'язку
Опис варіанта використання	Користувачу запропоновано надіслати відгук.
Актори	Користувач
Передумова	Користувач авторизований.
Основний сценарій	1. Користувач відкриває додаток. 2. Користувач натискає «Зворотній зв'язок». 3. Користувач натискає «Надіслати відгук». 4. Користувач вводить текст відгуку. 5. Користувач відправляє відгук.
Альтернативний сценарій	Скасування операції.
Виключення	Відсутній зв'язок: повідомлення про помилку.

Таблиця А.11 – Специфікація варіанту «Онлайн-підтримка користувачів»

Назва варіанта використання	Онлайн-підтримка користувачів
1	2
Опис варіанта використання	Користувачі отримують довідку про використання додатку.
Актори	Користувач
Передумова	Користувач авторизований.
Основний сценарій	1 .Користувач увійшов до застосунку. 2 .Користувач вибрав «Підтримка». 3 .Додаток відкриває форму зв'язку. 4 .Користувач надає опис проблеми. 5 .Підтримка надає відповідь. 6 .Користувач отримує відповідь. 7 .Користувач закриває форму.
Альтернативний сценарій	Відсутні вільні оператори: повідомлення.
Виключення	Відсутній доступ до інтернету: повідомлення про помилку.

ДОДАТОК Б

Лістинг основних класів програми

```
File: ActiveRent.tsx import React from "react";
import { useSelector } from "react-redux"; import {
Button } from "antd";
import dayjs from "dayjs";

import { BlockContent } from
"@/components/BlockContent/BlockContent";

import { Loader } from "@/elements";

import { getUserTransactionActive } from
"@/react-queries/transaction/useGetUserTransaction";
import { updateUserTransaction } from "@/react-queries/transaction/useUpdateTransaction";

import styles from "@/ActiveRent.module.scss";
import { RootState } from "@/store";
export const ActiveRent = () => {
const user = useSelector((state: RootState) =>
state.user.main);
const { data: transactionData, isLoading } =
getUserTransactionActive(user?.id || 0, {
enabled: !!user?.id
});

const { mutate: updateTransaction, isLoading:
isLoading } = useUpdateTransaction();

const handleCancelBooking = () => { if
(transactionData?.length) {
updateTransaction({
id: transactionData[0]?.id,
expiresIn: transactionData[0]?.expiresIn, expiresOut:
dayjs().toISOString()
});
}
};

if (isLoading) {
return <Loader isFullScreen />;
}

return (
<div className={styles.wrapper}>
<BlockContent title="Активна оренда" content={
<div className={styles.container}>
<h1>Ви маєте активний велосипед</h1>
<p>
Час початку:{" "}
{transactionData?.length &&
dayjs(transactionData[0]?.expiresIn).format("DD/MM
/YYYY HH:mm")}
```

```
</p>
<p>
Час закінчення:{" "}
{transactionData?.length &&
dayjs(transactionData[0]?.expiresOut).format("DD/MM
/YYYY HH:mm")}
</p>
<Button onClick={handleCancelBooking}
loading={isLoading} type="primary">
Закінчити поїздку
</Button>
</div>
}
/>
</div>
);
};
```

File: Dashboard.tsx

```
import React from "react";
import MapProvider from
"@/components/Map/MapProvider";
import styles from "@/Dashboard.module.scss";

export const Dashboard = () => { return (
<div className={styles.dashboard wrapper}>
<div className={styles.dashboard container}>
<MapProvider />
</div>
</div>
);
};
```

File: Profile.tsx

```
import React from "react";
import { useNavigate } from "react-router-dom";
import { Button } from "antd";

import { BlockContent } from
"@/components/BlockContent/BlockContent";
import { Payment } from
"@/components/Payment/Payment"; import {
ProfileUserView } from
"@/components/ProfileUserView/ProfileUserView";
import { ROUTES } from "@/constants"; import styles
from "@/Profile.module.scss"; const { DASHBOARD } =
ROUTES;
export const Profile = () => { const navigate =
useNavigate();

const handleNavigateToMap = () => {
navigate(DASHBOARD);
```

```

};

return (
  <div className={styles.container}>
    <p className={styles.header}>
      <Button onClick={handleNavigateToMap} ghost
        className={styles.header button}>
        Back to Map

    </p>
    <BlockContent title="Profile"
      content={<ProfileUserView

    <BlockContent title="Profile" content={<Payment />}
    />
    </div>
  );
};

File: SignIn.tsx
import { useTranslation } from "react-i18next";
import { Link } from "react-router-dom";

import { AuthSignInForm } from "@components";
import { ROUTES } from "@constants";
export const LogIn = () => { const { t } =
  useTranslation(); const { INDEX } = ROUTES;

  return (
    <div>
      <Link to={INDEX}>{t("home")}</Link>
      <AuthSignInForm />
    </div>
  );
};

File: SignUp.tsx
import { useTranslation } from "react-i18next";
import { Link } from "react-router-dom";

import { AuthSignUpForm } from
"@components/AuthSignUpForm/AuthSignUpForm"
;

import { ROUTES } from "@constants"; const { INDEX
} = ROUTES;
export const SignUp = () => { const { t } =
  useTranslation();

  return (
    <div>
      <Link to={INDEX}>{t("home")}</Link>

      <AuthSignUpForm />
    </div>
  );
};

```

```

File: AppRouter.tsx
import { FC } from "react";
import { Navigate, Route, Routes } from "react-
router-dom"; import { ROUTES } from "@constants";
import { AuthRoutes } from "../AuthRoutes"; import {
  MainRoutes } from "../MainRoutes"; import {
  PrivateRoutes } from "../PrivateRoutes";

const { DASHBOARD, AUTH } = ROUTES;
export const AppRouter: FC = () => { return (
  <Routes>
    <Route index element={<Navigate to={DASHBOARD}
      replace
    />} />

    <Route element={<PrivateRoutes />}>
      <Route path="/" element={<MainRoutes />} />
    </Route>
    <Route path={`/${AUTH}/*`} element={<AuthRoutes
    />} />
  </Routes>
);
};

File: AuthRoutes.tsx
import { FC } from "react";
import { Route, Routes } from "react-router-dom";
import { AuthLayout } from "@layouts";
import { AuthSignInForm, AuthSignUpForm } from
"@components";

import { ROUTES } from "@constants";

import { ResetPassword } from
"../ResetPassword/ResetPassword";
import { SetNewPassword } from
"../SetNewPassword/SetNewPassword";

const { SIGN_IN, SIGN_UP, RESET_PASSWORD,
  SET_NEW_PASSWORD }
= ROUTES;

export const AuthRoutes: FC = () => { return (
  <AuthLayout>
    <Routes>
      <Route path={SIGN_IN} element={<AuthSignInForm
      />} />
      <Route path={SIGN_UP} element={<AuthSignUpForm
      />} />
      <Route path={RESET_PASSWORD}
        element={<ResetPassword
      />} />

      <Route path={SET_NEW_PASSWORD}
        element={<SetNewPassword />} />
    </Routes>
  </AuthLayout>
);
};

```

File: MainRoutes.tsx

```
import { FC } from "react";
import { Route, Routes } from "react-router-dom";
import { MainLayout } from "@layouts";
import { ROUTES } from "@constants";

import { ActiveRent } from
"../ActiveRent/ActiveRent"; import { Dashboard }
from "../Dashboard/Dashboard"; import { Profile }
from "../Profile/Profile";

const { DASHBOARD, PROFILE, ACTIVE_RENT } =
ROUTES;

export const MainRoutes: FC = () => { return (
<MainLayout>
<Routes>

<Route path={DASHBOARD} element={<Dashboard
/>} />
<Route path={PROFILE} element={<Profile />} />
<Route path={ACTIVE_RENT} element={<ActiveRent
/>} />
</Routes>
</MainLayout>
);
}

File: PrivateRoutes.tsx import { FC } from "react";
import { Navigate, Outlet } from "react-router-dom";
import { getJWTToken } from "@helpers";
import { ROUTES } from "@constants"; const { AUTH,
SIGN_IN } = ROUTES;
export const PrivateRoutes: FC = () => { const token =
getJWTToken();

return token ? <Outlet /> : <Navigate
to={`/${AUTH}/${SIGN_IN}`} replace />;
};

File: AuthSignInForm.tsx
import { FC, useEffect } from "react";
import { useTranslation } from "react-i18next";
import { useDispatch } from "react-redux"; import {
useLocation } from "react-router-dom"; import {
Form, Typography } from "antd";
import { FormProps } from "antd/lib";

import { Button, FormItem, Input } from
"@elements"; import { useSignIn } from "@react-
queries";
import { useYupSync } from "@hooks";
import { useFormErrors } from
"@hooks/form/useFormErrors"; import { ROUTES }
from "@constants";
import { ISignIn } from "@types";
import styles from "../AuthSignInForm.module.scss";
import { signInSchema } from "../AuthSignInSchema";
```

```
import { setAuthActivePageInfo } from
"@store/auth/auth.slice";
```

```
const INITIAL_VALUES = {
remember: true
};
```

```
const { Link } = Typography; const {
RESET_PASSWORD } = ROUTES
```

```
export const AuthSignInForm: FC<FormProps> =
(props) => { const { pathname } = useLocation();
const dispatch = useDispatch();
const [form] = Form.useForm<ISignIn>(); const { t } =
useTranslation();
const onError = useFormErrors(form);

});
```

```
const { mutate: signIn, isLoading } = useSignIn({
onError
```

```
const yupSync = useYupSync(signInSchema);
```

```
useEffect(() => { dispatch(setAuthActivePageInfo({
titleForm: t("sign-in.title"), linkForm: t("sign-up.btn-
text"),
googleTitle: t("auth-form.google-sign-in")
}));
}, [pathname])
```

```
return (
<Form
form={form} onFinish={signIn}
className={styles["sign-in-form"]} layout="vertical"
initialValues={INITIAL_VALUES}
{...props}
>
<div className={styles["sign-in-inputs-wrapper"]}>
<FormItem label="Email" name="email" required
rules={[yupSync]}>
<Input type="email" />
</FormItem>
```

```
<FormItem label={t("auth-form.password")}
name="password" required rules={[yupSync]}>
<Input type="password" />
</FormItem>
```

```
<Link className={styles["forgot-password"]}
href={`/${AUTH}/${RESET_PASSWORD}`}>
{t("auth-form.forgot-password")}
</Link>
</div>
```

```
<Button size="large" htmlType="submit"
type="primary"
disabled={isLoading} loading={isLoading}
>
```

```

{t("sign-in.btn-text")}
</Button>
</Form>
);
};

File: AuthSignUpForm.tsx
import { FC, useEffect } from "react";
import { Trans, useTranslation } from "react-i18next";
import { useDispatch } from "react-redux";
import { useLocation } from "react-router-dom";
import { Form, FormProps } from "antd";

import { Button, Checkbox, FormItem, Input,
notification } from "@elements";
import { useSignUp } from "@react-queries";
import { useYupSync } from "@hooks";
import { useFormErrors } from
"@hooks/form/useFormErrors"; import {
ISignUpForm } from "@types";
import styles from "../AuthSignUpForm.module.scss";
import { signUpSchema } from
"./AuthSignUpSchema";
import { setAuthActivePageInfo } from
"@store/auth/auth.slice";

const INITIAL_VALUES = {
  agreement: true
};

export const AuthSignUpForm: FC<FormProps> =
(props) => { const { t } = useTranslation();
const { pathname } = useLocation(); const dispatch =
useDispatch();
const [form] = Form.useForm<ISignUpForm>(); const
onError = useFormErrors(form);

});

const { mutate: signUp, isLoading } = useSignUp({
  onError
const yupSync = useYupSync(signUpSchema);
const onFinish = ({ agreement, ...values }:
ISignUpForm) =>
{
  if (!agreement) { notification.error(
    <span className={styles["error-message"]}>{t("auth-
form.agreement-error")}</span>
  );
}

return;
}
signUp(values);
});

useEffect(() => { dispatch(setAuthActivePageInfo({
  titleForm: t("sign-up"), linkForm: t("sign-in.title"),
  googleTitle: t("auth-form.google-sign-up")
})));

```

```

}, [pathname])

return (
  <Form
    name="sign-up" className={styles["sign-up-form"]}
    form={form}
    onFinish={onFinish} initialValues={INITIAL_VALUES}
    layout="vertical"
    {...props}
  >
    <div className={styles["sign-up-inputs-wrapper"]}>
      <FormItem name="fullName" label={t("auth-
form.name")}
        required rules={[yupSync]}>
        <Input placeholder={t("auth-form.fullName-
placeholder")} />
      </FormItem>

      <FormItem name="email" label="Email" required
        rules={[yupSync]}>
        <Input placeholder={t("auth-form.email-
placeholder")} />
      </FormItem>

      <FormItem name="phoneNumber" label={t("auth-
form.phone")} required rules={[yupSync]}>
        <Input
          type="phone"
          placeholder={t("auth-form.phoneNumber-
placeholder")}
        />
      </FormItem>

      <FormItem name="password" label={t("auth-
form.password")} required rules={[yupSync]}>
        <Input type="password" placeholder={t("auth-
form.password-placeholder")} />
      </FormItem>

      <FormItem name="agreement"
        valuePropName="checked">
        <Checkbox className={styles.conditionContent}
          defaultChecked>
          <Trans
            i18nKey="auth-form.agreement" values={{
              privacy: t("auth-form.privacy"), conditionUse:
                t("auth-form.conditionUse")
            }}
            components={{ a: <a
              className={styles.conditionLink} href="#" /> }}
          />
        </Checkbox>
      </FormItem>
    </div>

    <Button size="large" htmlType="submit"
      type="primary"

```

```

disabled={isLoading} loading={isLoading}
className={styles["sign-up-btn"]}
>
{t("sign-up.btn-text")}
</Button>
</Form>
);
};

```

File: MapProvider.tsx

```

import React from "react";
import { APIProvider } from "@vis.gl/react-google-maps";
import { CustomMap } from "./Map";
export default function MapProvider() { return (
<APIProvider
  apiKey={process.env.REACT_APP_MAP_API_KEY as string}>

```

```

<CustomMap />
</APIProvider>
);
}

```

File: CustomMap.tsx

```

import React from "react";
import { Map } from "@vis.gl/react-google-maps";

```

```

import { useGetBikes } from "@react-queries/bikes/useGetMarkers";

import styles from "./Map.module.scss";
import { Markers } from "../Markers/Markers"; export
const CustomMap = () => {
  const { data: bikesData } = useGetBikes();
  return (
    <div className={styles.container}>
      <Map
        style={{ borderRadius: "20px" }} defaultZoom={13}
        defaultCenter={{
          lat: 48.912511,
          lng: 24.692757
        }}
        gestureHandling={"greedy"} disableDefaultUI
        mapId={process.env.REACT_APP_MAP_ID_KEY as string}
      >
        {bikesData && <Markers data={bikesData} />}
      </Map>
    </div>
  );
};

```

File: Markers.tsx

```

import React, { useEffect, useRef } from "react";
import { Marker, MarkerClusterer } from
"@googlemaps/markerclusterer";
import { useMap } from "@vis.gl/react-google-maps";
import { IBike } from
"@types/bike/bike.type";

```

```

import { CustomAdvancedMarker } from
"../AdvancedMarker/AdvancedMarker";

export const Markers = ({ data }: { data: IBike[] }) => {
  const map = useMap();
  const [markers, setMarkers] = React.useState<{ [key:
    string]: Marker }>({});
  const [activeMarker, setActiveMarker] =
    React.useState<number | null>(null);
  const clusterer = useRef<MarkerClusterer |
    null>(null);

  const handleActiveMarker = (key: number | null) => {
    if (activeMarker === key) {
      return setActiveMarker(null);
    }

    return setActiveMarker(key);
  };

  useEffect(() => { if (!map) return;
    if (!clusterer.current) {
      clusterer.current = new MarkerClusterer({ map });
    }, [map]);

    useEffect(() => { clusterer.current?.clearMarkers();
      clusterer.current?.addMarkers(Object.values(marker
        s));
    }, [markers]);

    => {

      const setMarkerRef = (marker: Marker | null, key:
        string)

        if (marker && markers[key]) return; if (!marker &&
          !markers[key]) return;

        setMarkers((prev) => { if (marker) {
          return { ...prev, [key]: marker };
        }

        const newMarkers = { ...prev }; delete
          newMarkers[key];
          return newMarkers;
        });

        return (
          <>
            {data.map((bike) => { return (
              <CustomAdvancedMarker
                activeMarker={activeMarker}
                handleActiveMarker={handleActiveMarker}
                setMarkerRef={setMarkerRef} key={bike.id}
                bike={bike}
              />
            );

```