

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет комп'ютерних наук і технологій

(повне найменування факультету)

Кафедра програмних засобів

(повне найменування кафедри)

Пояснювальна записка

до дипломного проєкту (роботи)

бакалавр

(ступінь вищої освіти)

на тему ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ
ДЛЯ ПІДТРИМКИ РОБОТИ АВТОСТОЯНКИ
SOFTWARE FOR AN INFORMATION SYSTEM
TO SUPPORT THE OPERATION OF THE PARKING LOT

Виконав(ла): студент(ка) 4 курсу, групи КНТ-212сп
Спеціальності 122 Комп'ютерні науки
(код і найменування спеціальності)

Освітня програма (спеціалізація)
Комп'ютерні науки

САЖКО В.С.

(ПРИЗВИЩЕ та ініціали)

Керівник ІЛЬЯШЕНКО М.Б.

(ПРИЗВИЩЕ та ініціали)

Рецензент КОЦУР М.І.

(ПРИЗВИЩЕ та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет КНТ
Кафедра програмних засобів
Ступінь вищої освіти бакалавр
Спеціальність 122 Комп'ютерні науки
(код і найменування)

Освітня програма (спеціалізація) Комп'ютерні науки
(назва освітньої програми (спеціалізації))

ЗАТВЕРДЖУЮ

Завідувач кафедри ПЗ, д.т.н, проф.
Сергій СУББОТІН
“ ” 2025 року

З А В Д А Н Н Я

НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА(КИ)

САЖКА Владислава Сергійовича

(ПРИЗВИЩЕ, ім'я, по батькові)

1. Тема проєкту (роботи) Програмне забезпечення інформаційної системи для підтримки роботи автостоянки. Software for an Information System to Support the Operation of the Parking Lot

керівник проєкту (роботи) к.т.н., доцент, ІЛЬЯШЕНКО Матвій Борисович,
(науковий ступінь, вчене звання, ПРИЗВИЩЕ, ім'я, по батькові)

затверджені наказом закладу вищої освіти від “ 28 ” квітня 2025 року № 209

2. Строк подання студентом проєкту (роботи) 03 червня 2025 року

3. Вихідні дані до проєкту (роботи) рекомендована література, технічне завдання

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) 1. Аналіз предметної області. 2. Матеріали і методи. 3. Опис програми. 4. Експлуатація, тестування та експериментальне дослідження програми.

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

6. Консультанти розділів проєкту (роботи)

Розділ	ПРИЗВИЩЕ, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-4 Основна частина	ІЛЪЯШЕНКО М.Б., доцент		
Нормоконтроль	КАМІНСЬКА Ж.К., асистент		

7. Дата видачі завдання “ 28 ” квітня 2025 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітка
1	Постановка завдання роботи.	1 тиждень	Завдання, ТЗ
2	Аналіз предметної області.	1 тиждень	Розділ 1
3	Вибір мови програмування та інших технологій розробки.	2 тиждень	Розділ 2
4	Розробка структури програми.	2 тиждень	Розділ 3
5	Розробка програми.	3-4 тижні	Розділи 3, 4
6	Тестування та експериментальне дослідження програмного забезпечення.	5 тиждень	Розділ 4
7	Оформлення пояснювальної записки та документів до неї.	6 тиждень	Додатки
8	Нормоконтроль та рецензування.	7 тиждень	
9	Захист роботи.	8 тиждень	

Студент(ка)

(підпис) Владислав САЖКО
(Ім'я ПРИЗВИЩЕ)

Керівник проєкту (роботи)

(підпис) Матвій ІЛЪЯШЕНКО
(Ім'я ПРИЗВИЩЕ)

РЕФЕРАТ

Пояснювальна записка до дипломної кваліфікаційної роботи бакалавра:
94 с., 3 табл., 42 рис., 3 дод., 16 джерел.

ANDROID STUDIO, JAVA, АВТОСТОЯНКА, ІНФОРМАЦІЙНА СИСТЕМА, МОБІЛЬНИЙ ЗАСТОСУНОК, МОВА ПРОГРАМУВАННЯ.

Об'єкт дослідження – процес розробки інформаційної системи для підтримки роботи автостоянки.

Предмет дослідження – інформаційні системи для підтримки роботи автостоянки.

Мета роботи – розробка програмного забезпечення інформаційної системи для підтримки роботи автостоянки.

Матеріали, методи та технічні засоби: мова програмування Java, середовище розробки Android Studio.

Результати. Розроблено проєктні рішення для створення програмного забезпечення інформаційної системи для підтримки роботи автостоянки. Створено програмне забезпечення інформаційної системи для підтримки роботи автостоянки. Проведено тестування розробленого програмного забезпечення інформаційної системи для підтримки роботи автостоянки.

Висновки. Мету роботи досягнуто. Розроблено програмне забезпечення інформаційної системи для підтримки роботи автостоянки за допомогою мови програмування Java та середовища розробки Android Studio.

Галузь використання – надання послуг з розміщення автотранспортних засобів.

ABSTRACT

Explanatory note to the diploma qualifying work of the bachelor: 94 pages, 3 tables, 42 figures, 3 appendixes, 16 sources.

ANDROID STUDIO, JAVA, PARKING, INFORMATION SYSTEM, MOBILE APPLICATION, PROGRAMMING LANGUAGE.

The object of research is the process of developing information system for the operation of a parking lot.

The subject of the research is information systems to support the operation of a parking lot.

The purpose of this work is to develop the software of the information system to support the operation of a parking lot.

Materials, methods and technical tools: Java programming language, Android Studio development environment.

Results. Project solutions for the creation of software for the information system to support the operation of a parking lot has been designed. The software of the information system to support the operation of a parking lot has been developed. The developed software of the information system to support the operation of a parking lot has been tested.

Conclusions. The goal of the work has been achieved. The software tools of information system to support the operation of a parking lot using the Java programming language, Android Studio development environment have been developed.

Scope of use – vehicle parking services.

ЗМІСТ

	С.
Перелік скорочень та умовних позначок	8
Вступ	9
1 Аналіз предметної області	11
1.1 Інформаційні системи для підтримки роботи автостоянки	11
1.2 Аналіз інформаційних систем для підтримки роботи автостоянки	22
1.3 Висновки за розділом 1	31
2 Матеріали і методи	33
2.1 Вибір мови програмування.....	33
2.2 Вибір середовища розробки для створення програмного забезпечення інформаційної системи для підтримки роботи автостоянки	35
2.3 Висновки за розділом 2	38
3 Опис програми	39
3.1 Структура програмного забезпечення інформаційної системи для підтримки роботи автостоянки	39
3.2 Функціонування програмного забезпечення інформаційної системи для підтримки роботи автостоянки	42
3.3 Розробка бази даних програмного забезпечення інформаційної системи для підтримки роботи автостоянки.....	47
3.4 Проектування інтерфейсу програмного забезпечення інформаційної системи для підтримки роботи автостоянки	48
3.5 Висновки за розділом 3	54
4 Експлуатація, тестування та експериментальне дослідження програми.....	56
4.1 Призначення й умови застосування програми	56
4.2 Характеристики програмного забезпечення інформаційної системи для підтримки роботи автостоянки	57
4.3 Інструкція по експлуатації програми.....	58
4.3.1 Звернення до програми.....	58
4.3.2 Вхідні й вихідні дані.....	58

4.3.3 Повідомлення.....	58
4.4 Виконання програмного забезпечення інформаційної системи для підтримки роботи автостоянки	59
4.5 Тестування програмного забезпечення інформаційної системи для підтримки роботи автостоянки	66
4.6 Висновки за розділом 4	66
Висновки.....	67
Перелік джерел посилань.....	70
Додаток А Технічне завдання.....	72
Додаток Б Фрагмент тексту програми	76
Додаток В Слайди презентації	87

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК

JVM – Java Virtual Machine;

БД – база даних;

ІС – інформаційна система;

МП – мова програмування;

ПЗ – програмне забезпечення.

ВСТУП

Актуальність програмного забезпечення інформаційної системи для підтримки роботи автостоянки полягає у необхідності автоматизації основних процесів, що дозволяє спростити облік в'їзду та виїзду транспортних засобів, швидко визначати наявність вільних місць та зменшувати вплив людського фактора на роботу стоянки. Це підвищує ефективність управління, оскільки забезпечує контроль наповненості стоянки в реальному часі, автоматизоване формування звітів про фінансові показники та аналіз динаміки використання місць для паркування [1], [2].

Важливим аспектом є покращення клієнтського досвіду. Впровадження електронної оплати, безконтактних методів розрахунку та можливості онлайн бронювання місць робить використання автостоянки зручнішим. Додатково, інтеграція з мобільними додатками або вебсайтами дозволяє водіям заздалегідь дізнаватися про наявність місць [3]-[5].

Окрім цього, автоматизація допомагає скоротити витрати на персонал, оптимізувати тарифну політику відповідно до попиту та забезпечити швидку окупність інвестицій у впровадження системи [6], [7].

Проте деякі програмні засоби інформаційних систем для підтримки роботи автостоянок можуть мати високу вартість впровадження. Розробка, налаштування та інтеграція такої програмної системи потребують значних фінансових вкладень, особливо якщо вона включає відео спостереження та онлайн сервіси. Окрім того, для підтримки стабільної роботи необхідні регулярні оновлення, що також тягне за собою додаткові витрати. Крім того, можливі збої в інтеграції програмних засобів інформаційних систем для підтримки роботи автостоянок з іншими системами або платіжними платформами. Це може створювати незручності для користувачів та знижувати ефективність роботи автостоянки [1]-[7].

Тому актуальною є розробка інформаційної системи для підтримки роботи автостоянки. У дипломній кваліфікаційній роботі бакалавра

розв'язується актуальне завдання розробки програмного забезпечення інформаційної системи для підтримки роботи автостоянки.

Для досягнення поставленої мети у кваліфікаційній роботі бакалавра необхідно розв'язати такі задачі:

- виконати аналіз предметної області та інформаційних систем для підтримки роботи автостоянки;
- здійснити проєктування програмного забезпечення інформаційної системи для підтримки роботи автостоянки;
- створити програмне забезпечення інформаційної системи для підтримки роботи автостоянки;
- виконати тестування розробленого програмного забезпечення інформаційної системи для підтримки роботи автостоянки.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Інформаційні системи для підтримки роботи автостоянки

Автостоянка являє собою спеціально обладнану територія або будівля, призначена для тимчасового зберігання транспортних засобів. Вона може бути відкритою або критою, безкоштовною або платною, з персоналом або автоматизованою. Автостоянки використовуються для зберігання автомобілів на певний проміжок часу, забезпечуючи безпеку та зручність для власників транспорту [1].

На автостоянці відбувається кілька основних процесів, облік яких можна автоматизувати за допомогою відповідних програмних засобів. Спочатку здійснюється заїзд автомобіля, під час якого водій може отримати паркувальний талон або пройти автоматичну ідентифікацію. Далі транспортний засіб розміщується на вільному місці відповідно до наявності та правил паркування. В процесі перебування автомобіля система контролює його стан та час перебування, може забезпечувати відеонагляд або інші заходи безпеки [1].

Оплата за користування стоянкою може проводитися в різний спосіб, включаючи готівковий і безготівковий розрахунок, а в автоматизованих програмних системах – за допомогою мобільних додатків або абонементних карток. При виїзді з території власник транспортного засобу підтверджує оплату або виконує необхідні дії, щоб звільнити місце для наступного автомобіля [1].

Автоматизація процесів на автостоянці за допомогою відповідного програмного забезпечення дозволяє підвищити ефективність її роботи, зменшити витрати на персонал, покращити контроль за транспортними засобами та підвищити рівень безпеки. Доцільно автоматизувати процес реєстрації та в'їзду транспортних засобів, що дозволяє швидко ідентифікувати водія та автомобіль за допомогою номерних знаків або електронних карток.

Також варто впровадити автоматичний розподіл місць для паркування, що дозволяє оптимально використовувати доступний простір [1].

Оплата за паркування є ще одним важливим процесом, який слід автоматизувати, оскільки це зменшує черги та покращує зручність для користувачів. Автоматичні термінали, мобільні додатки або онлайн системи дозволяють водіям швидко здійснювати платежі без участі оператора [1].

Контроль та моніторинг автостоянки також можна здійснювати автоматично завдяки відео нагляду, сенсорам зайнятості місць та системам розпізнавання номерних знаків. Це підвищує рівень безпеки та дозволяє оперативно реагувати на порушення [1].

Скасування, продовження та редагування бронювання з використанням мобільного застосунку або вебінтерфейсу спрощує взаємодію користувачів із системою. Важливою є й автоматизація звітності, яка дозволяє власникам стоянки аналізувати завантаженість, прибутковість та ефективність використання паркомісць у режимі реального часу [1].

Програмне забезпечення для підтримки функціонування автостоянки є важливим елементом сучасної інфраструктури, що забезпечує ефективне управління паркувальними місцями, спрощує процес бронювання та оплати, а також підвищує загальний рівень комфорту як для користувачів, так і для адміністрації автостоянки [1].

Програмне забезпечення для підтримки функціонування автостоянки автоматизує процеси, пов'язані з контролем заїзду та виїзду транспортних засобів, ідентифікацією водіїв, розрахунком вартості послуг та управлінням доступом до стоянки. Таке програмне забезпечення може інтегруватися з різними зовнішніми сервісами, такими як банківські системи, мобільні додатки, відео спостереження та системи розпізнавання номерних знаків, що дозволяє значно знизити кількість ручної роботи та можливість помилок [1].

Програмне забезпечення для підтримки функціонування автостоянки також дозволяє збирати та аналізувати статистичні дані про використання паркомісць, фінансові операції, пікові навантаження та поведінку

користувачів. Це сприяє прийняттю обґрунтованих рішень щодо розширення паркінгу, зміни тарифної політики або покращення сервісу [1], [2].

Окрім стандартних функцій, програмне забезпечення інформаційних систем для підтримки роботи автостоянки може підтримувати гнучкі налаштування, що дають можливість адаптувати її до різних типів стоянок – від невеликих приватних паркінгів до великих багаторівневих комплексів та паркувальних зон у мегаполісах. Впровадження такої системи сприяє не лише оптимізації процесів, а й підвищенню рівня безпеки, зменшенню часу, витраченого на паркування, та покращенню загального користувацького досвіду [1], [2].

Розробка програмного забезпечення для підтримки функціонування автостоянки є складним процесом, що включає кілька основних етапів, спрямованих на створення ефективної та надійної системи [1], [2].

На першому етапі проводиться аналіз вимог, під час якого визначаються основні функції майбутньої системи, зокрема управління паркувальними місцями, бронювання, оплата, контроль заїзду та виїзду, а також збір статистичних даних. Важливо розуміти потреби як кінцевих користувачів, так і адміністрації автостоянки, щоб забезпечити зручність використання та ефективне управління ресурсами [1], [2].

Наступним кроком є проєктування архітектури, де визначаються основні модулі системи, взаємозв'язки між ними та принципи зберігання та обробки даних. На цьому етапі також планується користувацький інтерфейс, що має бути інтуїтивно зрозумілим для користувачів [1], [2].

Після завершення проєктування розпочинається етап розробки, під час якого програмісти створюють код системи відповідно до визначених вимог і архітектури. Цей процес може включати як створення серверної частини для обробки запитів і збереження даних, так і розробку клієнтської частини, що забезпечує взаємодію з користувачами [1], [2].

Після написання основних компонентів відбувається тестування, яке включає перевірку всіх функцій на предмет коректності роботи, безпеки та

продуктивності. Виправляються знайдені помилки, оптимізується робота системи, а також проводиться тестування з реальними користувачами для отримання зворотного зв'язку [1], [2].

Заключним етапом є впровадження та супровід, під час якого система розгортається в робочому середовищі, навчається персонал, а також здійснюється моніторинг її роботи. Регулярне оновлення та вдосконалення забезпечують відповідність змінним потребам автостоянки та покращують загальну ефективність її функціонування [1], [2].

Система паркування являє собою комплекс програмних та апаратних рішень, що забезпечує впорядковане розміщення транспортних засобів на автостоянці та оптимізує її роботу. Такі системи включають різні механізми, які дозволяють контролювати доступ, фіксувати час перебування, здійснювати оплату та формувати звітність для операторів стоянки. Основні функції таких систем можуть суттєво відрізнятися залежно від того, яким чином відбувається оплата, як ведеться моніторинг, яким способом організовано контроль в'їзду та виїзду, а також наскільки розвинена аналітична складова управління паркуванням [1], [2].

З розвитком технологій концепція організації стоянок зазнала значних змін. Колись головним завданням було лише оформлення штрафів за неправильне паркування або приймання готівки за надані послуги. Нині ж сучасні підходи передбачають впровадження програмних комплексів, автоматизованих рішень і технологій Інтернету речей, що дозволяють зробити процеси управління автостоянками значно ефективнішими [1], [2].

Спочатку контроль за паркуванням здійснювався виключно вручну. Оператор стоянки фіксував, які машини заїжджають та виїжджають, вручну виписував квитанції та слідкував за вільними місцями. У таких системах оплата найчастіше проводилася готівкою, а вся інформація про паркування зберігалася у паперових журналах або найпростішому електронному вигляді. Доступ контролювався за допомогою талонів чи перепусток, а облік вільних місць вівся за допомогою таблиць або спеціальних списків [1], [2].

Через високу трудомісткість та можливість зловживань ручні методи з часом поступилися місцем автоматизованим системам. Це дозволило мінімізувати вплив людського фактора, підвищити швидкість обробки операцій та забезпечити більш ефективний контроль за використанням паркувального простору. Сучасні технології дозволяють операторам стоянок легко керувати фінансовими потоками, підвищувати рівень безпеки та надавати зручний сервіс для водіїв [1], [2].

Протягом останніх десятиліть управління паркувальними просторами часто здійснювалося за допомогою апаратних рішень. Для організації процесу використовувалися паркомати, що встановлювалися біля паркувальних місць на вулицях, а також автоматизовані шлагбауми, які забезпечували контроль в'їзду та виїзду, дозволяючи водіям сканувати квитки та здійснювати оплату. Основні функції таких систем включали обробку платежів, контроль за транспортними засобами, управління доступом та ведення звітності. Оплата здійснювалася через спеціальні термінали, які приймали готівку або карткові розрахунки. Система відстежувала використання паркувальних місць та фіксувала порушення правил стоянки. Контроль доступу забезпечувався автоматичними шлагбаумами або іншими механізмами, що обмежували в'їзд. Важливим елементом була звітність, яка дозволяла аналізувати заповненість паркінгу та фінансові показники [1], [2].

Попри ефективність таких рішень, вони мали суттєві недоліки, зокрема високу вартість установки та необхідність регулярного технічного обслуговування. Це спонукало до пошуку альтернативних підходів, зокрема використання програмних рішень, які значно розширюють можливості управління паркувальним простором [1], [2].

Перехід до програмного забезпечення для паркування зумовлений не лише економічною вигідністю, а й розширенням функціональності. Сучасні програмні застосунки дають можливість організовувати дистанційне керування паркінгом, вести детальну аналітику та здійснювати облік вільних

місць для різних категорій користувачів, включаючи корпоративні, орендні, гостьові чи змішані зони [1], [2].

У мобільних застосунках реалізовано можливість здійснення оплати без необхідності використання фізичних терміналів. Менеджери можуть у режимі реального часу отримувати дані про завантаженість стоянки та приймати управлінські рішення. Інтеграція з апаратними системами дозволяє контролювати роботу воріт, шлагбаумів або зарядних станцій для електромобілів [1], [2].

Автоматизований контроль доступу реалізується за допомогою мобільних додатків або технологій розпізнавання номерних знаків. Водії можуть відкривати шлагбауми одним натисканням у додатку або скористатися автоматизованими системами, що реагують на їхні транспортні засоби. Сучасні рішення також містять розширені аналітичні інструменти, які дозволяють відстежувати ефективність використання паркувального простору та оптимізувати його роботу [1], [2].

Рациональна організація паркувального простору відіграє важливу роль у забезпеченні безперебійного функціонування транспортних потоків, зниженні рівня заторів і покращенні загального комфорту для користувачів. Розвиток технологій призвів до поступового переходу від традиційних, заснованих на апаратних рішеннях підходів до цифрових систем управління, що дозволяють оптимізувати використання паркомісць [2], [3].

Сучасні рішення для паркування характеризуються впровадженням цифрових методів оплати, автоматизованого контролю за дотриманням правил паркування, управлінням доступом та розширеними можливостями аналітики. Застосування подібних систем дає змогу суттєво скоротити експлуатаційні витрати, підвищити ефективність управління та забезпечити зручність для користувачів [2], [3].

Автоматизовані програмні системи управління паркуванням являють собою сукупність технологій, що дозволяють ефективно контролювати розподіл паркомісць та оптимізувати транспортні потоки. Такі рішення значно

полегшують адміністрування автостоянок, дозволяючи зменшити витрати на обслуговування та підвищити загальну продуктивність. Використання програмного забезпечення для паркування поширене в різних сферах, включаючи навчальні заклади, комерційні установи, державні організації та корпоративні структури [2], [3].

Цифрові технології дозволяють реалізовувати автоматизовані платіжні механізми, ефективний контроль доступу та аналіз використання паркувального простору. Завдяки таким інструментам забезпечується покращена організація паркінгу, що відповідає сучасним вимогам щодо зручності та економічної доцільності [2], [3].

Ефективна організація паркувальних систем потребує наявності кількох ключових елементів, які забезпечують зручність, контроль і оптимальне використання місць для стоянки. Для забезпечення безперебійної роботи необхідно передбачити різні механізми оплати, які відповідають різним сценаріям використання. Працівники можуть здійснювати оплату в цифровому форматі, тоді як комерційні паркувальні простори можуть підтримувати як автоматизовані, так і ручні варіанти розрахунку. Гнучкість у виборі способів оплати сприяє кращій інтеграції системи та її ефективному функціонуванню [2], [3].

Для того щоб управління паркувальним простором було ефективним, важливу роль відіграють можливості контролю доступу, розподілу місць та оповіщення користувачів у реальному часі. Відстеження наявності вільних місць і автоматизоване бронювання дозволяють підвищити пропускну здатність парковок та інтегрувати їх із загальною транспортною системою. Крім того, системи контролю доступу можуть забезпечувати обмежений в'їзд, орієнтуючись на попередні бронювання, що дозволяє уникнути хаотичного використання простору. Такі рішення дають змогу запобігти несанкціонованому доступу та покращити загальний рівень безпеки [2], [3].

Завдяки сучасним цифровим панелям управління керування паркувальними зонами стає більш прозорим. Візуалізація статистичних даних

щодо використання паркомісць, транспортних потоків і доступності місць у реальному часі дає змогу приймати більш обґрунтовані рішення щодо оптимізації паркувальних ресурсів [2], [3].

Використання цифрових технологій відкриває можливості для створення інноваційних підходів до управління паркуванням. Компанії активно переходять від традиційних методів організації паркувального простору до автоматизованих систем, які зменшують витрати на адміністрування та дозволяють збільшити ефективність без необхідності розширення фізичних площ. Такі системи особливо актуальні для організацій із гнучкими робочими графіками, де необхідне ефективне керування бронюванням паркомісць, реєстрацією та контролем відвідуваності [2], [3].

Не тільки комерційні структури стикаються з викликами паркування. У житлових комплексах питання розподілу місць і доступу гостей може викликати значні труднощі. Власники та керівники нерухомості все частіше звертаються до автоматизованих рішень, що дозволяють легко керувати паркуванням у житлових зонах. Використання таких інформаційних технологій дозволяє усунути хаос у розподілі місць і покращити загальні умови для мешканців [2], [3].

У деяких випадках облік процесів паркування все ще здійснюється вручну, хоча сучасні автоматизовані системи керування вже довели свою ефективність і зручність. Традиційні механізми, які включають шлагбауми, квиткові термінали та платіжні пристрої, забезпечують контроль за процесом, проте їх утримання та впровадження обходиться значно дорожче, ніж сучасні цифрові рішення [2], [3].

Щоб зробити паркування комфортнішим, компанії активно інвестують у вдосконалення процесів, впроваджуючи програмне забезпечення, яке дозволяє здійснювати оплату та бронювання онлайн. Такі рішення дозволяють мінімізувати витрати, а також підвищити рівень зручності для користувачів. Серед основних переваг цифрових платформ можна виділити автоматизовану

обробку платежів, аналітику та управління процесами в режимі реального часу [2], [3].

Щодо корпоративних паркінгів, то сучасні інформаційні системи забезпечують зручність як для працівників, так і для адміністрації. Працівники отримують змогу бронювати місця, швидко реєструватися та використовувати електронні засоби доступу. У той же час управлінці мають можливість контролювати ситуацію, аналізувати використання паркомісць та коригувати правила їхнього розподілу за необхідності [2], [3].

Одним із важливих аспектів вибору програмної системи керування паркуванням є її вартість. Впровадження таких рішень допомагає не лише зменшити витрати на технічне обслуговування традиційних паркінгів, але й уникнути фінансових втрат, пов'язаних із хаотичним використанням паркомісць. Перед прийняттям рішення слід звернути увагу на функціональні можливості системи, включно з інтеграцією цифрових платежів, відео спостереженням та інструментами аналітики [2], [3].

Автоматизовані програмні системи мають велику кількість переваг, серед яких підвищений рівень безпеки, скорочення витрат часу на організацію процесу паркування, а також покращення загального користувацького досвіду завдяки оптимізованій навігації та оперативному оновленню інформації про доступність місць [2], [3].

Ефективне управління паркувальними зонами є важливим завданням для багатьох підприємств, і вибір відповідного програмного забезпечення дозволяє значно підвищити рівень їхньої ефективності. Впровадження сучасних інформаційних технологій у цю сферу дає змогу покращити контроль, оптимізувати витрати та створити зручні умови для водіїв [2], [3].

Програмні рішення для управління паркувальним простором значно змінюють процес організації паркування. Завдяки автоматизації ключових операцій, таких як розподіл місць та обробка платежів, ці технології сприяють підвищенню ефективності та покращенню взаємодії користувачів. Впровадження таких систем дозволяє значно зменшити транспортні затори та

оптимізувати пересування, що позитивно впливає на щоденну діяльність міських структур і бізнесів [3], [4].

Програмні системи управління паркуванням забезпечують комплексне вирішення завдань, пов'язаних із контролем та оптимізацією використання паркомісць. Вони не лише спрощують процес взаємодії між користувачами та операторами, а й пропонують широкий набір функцій, що допомагають покращити організацію паркувальних майданчиків [3], [4].

Серед ключових можливостей таких програмних систем можна виділити ефективний розподіл простору, що дозволяє зменшити скупчення транспорту та підвищити пропускну здатність стоянок. Автоматизовані методи обробки платежів спрощують фінансові операції, роблячи їх більш зручними та швидкими. Аналітичні інструменти дають змогу оцінювати рівень завантаженості паркувальних зон та приймати зважені рішення на основі отриманих даних. Додатково інтеграція з системами безпеки дозволяє посилити контроль за транспортними засобами та дотриманням правил паркування [5], [6].

Гнучкість і масштабованість програмних засобів роблять їх ефективними як для невеликих стоянок, так і для великих багаторівневих паркінгів. Використання подібних технологій дозволяє значно підвищити рівень організації паркування, зробивши цей процес більш структурованим та комфортним для користувачів [5], [6].

Програмні рішення, призначені для організації паркувального простору, включають широкий набір можливостей, які покликані підвищити ефективність процесів і зробити користування паркінгами більш зручним. Завдяки впровадженню таких технологій стає можливим не лише оптимізувати управління стоянками, а й створити комфортні умови для водіїв. Одним із ключових елементів є система попереднього бронювання місць, що дозволяє уникнути хаотичного пошуку вільних зон для паркування. Використання такої функції сприяє рівномірному розподілу простору та

зменшенню перевантаженості, що робить пересування містом більш впорядкованим [5], [6].

Не менш важливим аспектом є можливість здійснювати оплату за стоянку без необхідності використовувати готівку чи паперові талони. Інтеграція з цифровими платіжними сервісами дає змогу водіям швидко й безпечно проводити транзакції за допомогою мобільних додатків. Такий підхід не лише спрощує процес взаємодії з паркувальними системами, а й зменшує адміністративні витрати, пов'язані з обслуговуванням касових терміналів [5], [6].

Автоматизований збір та аналіз даних допомагає операторам приймати зважені рішення, що базуються на актуальній інформації про рівень завантаженості стоянок, фінансові показники та динаміку попиту. Завдяки таким інструментам можна виявити тенденції, що дозволяють більш гнучко адаптувати політику управління простором та впроваджувати ефективні стратегії оптимізації [5], [6].

Додаткова перевага полягає в інтеграції з системами контролю безпеки, що забезпечує захист транспортних засобів та підвищує загальний рівень безпеки на території паркінгу. Наявність механізмів спостереження та автоматизованих пропускних пунктів сприяє зменшенню ризиків несанкціонованого доступу та дозволяє швидко реагувати на будь-які інциденти [5], [6].

Сучасні рішення в цій сфері спрямовані на максимальну автоматизацію процесів, що робить управління паркуванням більш ефективним та зручним як для бізнесу, так і для користувачів [5], [6].

Використання інноваційних підходів дає змогу підвищити якість обслуговування та створити більш впорядковану й безпечну систему організації паркувального простору [5], [6].

Визначено, що програмне забезпечення для підтримки функціонування автостоянки є важливим елементом сучасної інфраструктури, що забезпечує ефективне управління паркувальними місцями, спрощує процес бронювання

та оплати, а також підвищує загальний рівень комфорту як для користувачів, так і для адміністрації автостоянки.

1.2 Аналіз інформаційних систем для підтримки роботи автостоянки

Проаналізуємо інформаційні системи для підтримки роботи автостоянки [7]-[12].

Програмне забезпечення SpotHero (рис. 1.1) розроблене для полегшення процесу керування паркуванням, як для окремих водіїв, так і для компаній. Завдяки інтеграції з мобільними пристроями та вебінтерфейсом, забезпечується швидкий доступ до актуальної інформації про наявність місць та їх вартість, що дозволяє заздалегідь здійснювати бронювання [7], [8], [10].

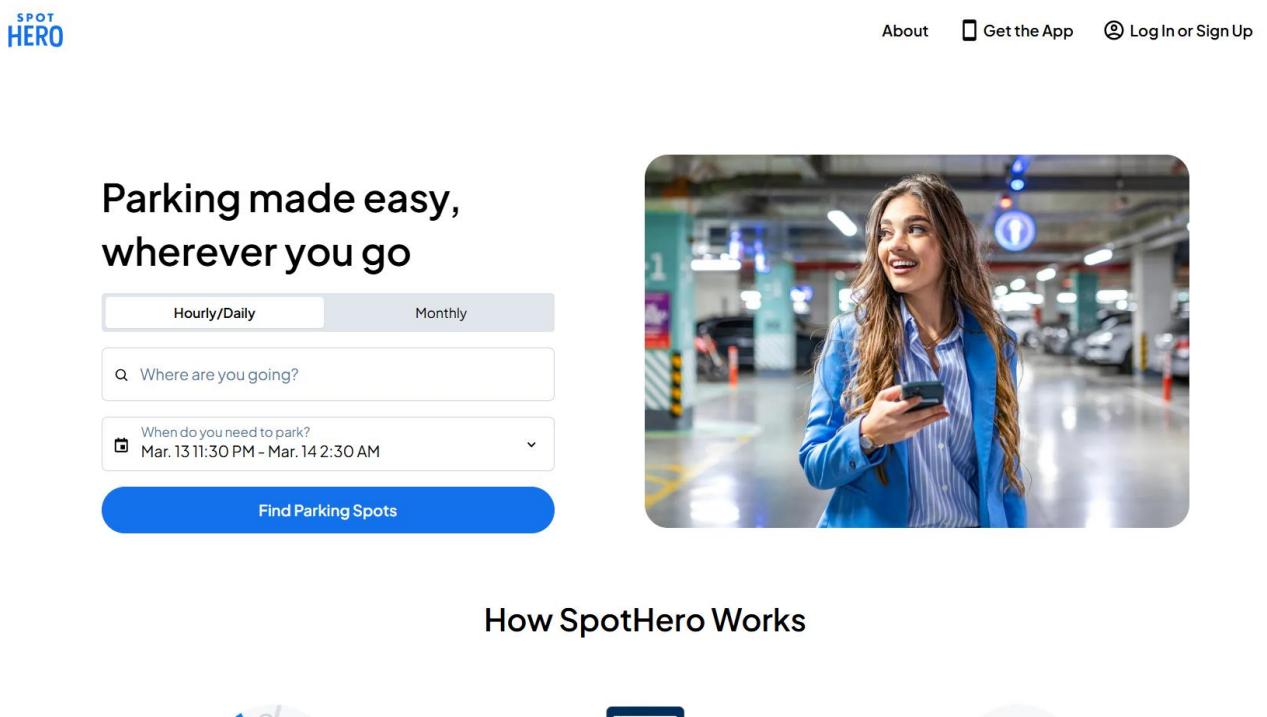


Рисунок 1.1 – Програмне забезпечення SpotHero [10]

Функціональність програмного забезпечення SpotHero спрямована на оптимізацію використання паркувального простору, що допомагає організаціям ефективно розподіляти місця для співробітників або спеціальних

подій. Після здійснення бронювання користувачі отримують детальні інструкції щодо доступу до місця, а електронні квитки забезпечують безперешкодний в'їзд та виїзд із паркінгу [7], [8], [10].

Автоматизована система оплати дозволяє миттєво здійснювати транзакції, що усуває необхідність у використанні готівки та фізичних талонів. Крім того, аналізуючи дані про завантаженість паркувальних зон, компанії можуть покращувати організацію простору та пропонувати більш вигідні умови для водіїв [7], [8], [10].

SpotHero надає можливість знаходити місця для довготривалого зберігання автомобілів, скористатися послугами паркувальника або знайти зручне паркування біля ключових транспортних вузлів. Орієнтація на високий рівень обслуговування та постійне вдосконалення сервісу дозволяє мінімізувати труднощі, пов'язані з пошуком місця для автомобіля, і зробити цей процес максимально комфортним [7], [8], [10].

Основними характеристиками програмного забезпечення SpotHero є такі [7], [8], [10]:

- зручність бронювання: користувачі можуть заздалегідь резервувати паркувальні місця через мобільний додаток або веб-інтерфейс, що гарантує наявність місця по прибуттю [7], [8], [10];

- інтеграція з мобільними пристроями: програмне забезпечення SpotHero працює на смартфонах, забезпечуючи доступ до інформації про паркування у будь-який момент [7], [8], [10];

- можливість порівняння цін: користувачі програмного забезпечення SpotHero можуть переглядати та аналізувати вартість паркування у різних локаціях для вибору найбільш вигідного варіанту [7], [8], [10];

- електронні перепустки: після оплати програмне забезпечення SpotHero надсилає користувачам цифровий абонемент, що спрощує процес в'їзду на стоянку [7], [8], [10];

- безпечні платежі: транзакції здійснюються через захищені платіжні шлюзи, що гарантує безпеку фінансових операцій [7], [8], [10];

- підтримка корпоративних клієнтів: компанії можуть керувати паркуванням для своїх співробітників або організовувати зони паркування для заходів [7], [8], [10];
- аналітика використання: програмне забезпечення SpotHero надає детальні звіти про завантаженість паркувальних зон, що допомагає оптимізувати управління простором [7], [8], [10];
- інтеграція з іншими сервісами: програмне забезпечення SpotHero надає можливість підключення до автомобільних, навігаційних та орендних платформ для зручного доступу до паркування [7], [8], [10];
- підтримка довгострокового паркування: користувачі програмного забезпечення SpotHero можуть бронювати місця на тривалий термін або користуватися додатковими послугами, такими як паркувальник [7], [8], [10].

Серед недоліків програмного забезпечення SpotHero можна відзначити залежність від доступності паркувальних місць у системі, що може обмежувати вибір у певних районах або в години підвищеного попиту. Крім того, можливі технічні збої або проблеми з інтеграцією електронних перепусток, що інколи ускладнює в'їзд на паркінг. Вартість послуг також може змінюватися залежно від локації та часу, що не завжди вигідно для користувачів. Деякі водії можуть зіткнутися з неточностями в інформації про наявність місць або з труднощами в отриманні відшкодування у випадку скасування бронювання [7], [8], [10].

Програмне забезпечення Parkopedia (рис. 1.2) представляє собою рішення для організації паркувального простору, яке допомагає водіям швидко знаходити місця для стоянки та отримувати актуальну інформацію про їхню доступність. Система дозволяє переглядати дані про розташування, вартість та заповненість паркувальних зон, що дає можливість обирати оптимальний варіант залежно від потреб користувачів. Велика база містить інформацію про стоянки в різних країнах, включаючи варіанти як на узбіччях доріг, так і на спеціалізованих майданчиках, а інтеграція з навігаційними сервісами забезпечує зручний маршрут до вибраного місця [7], [11].

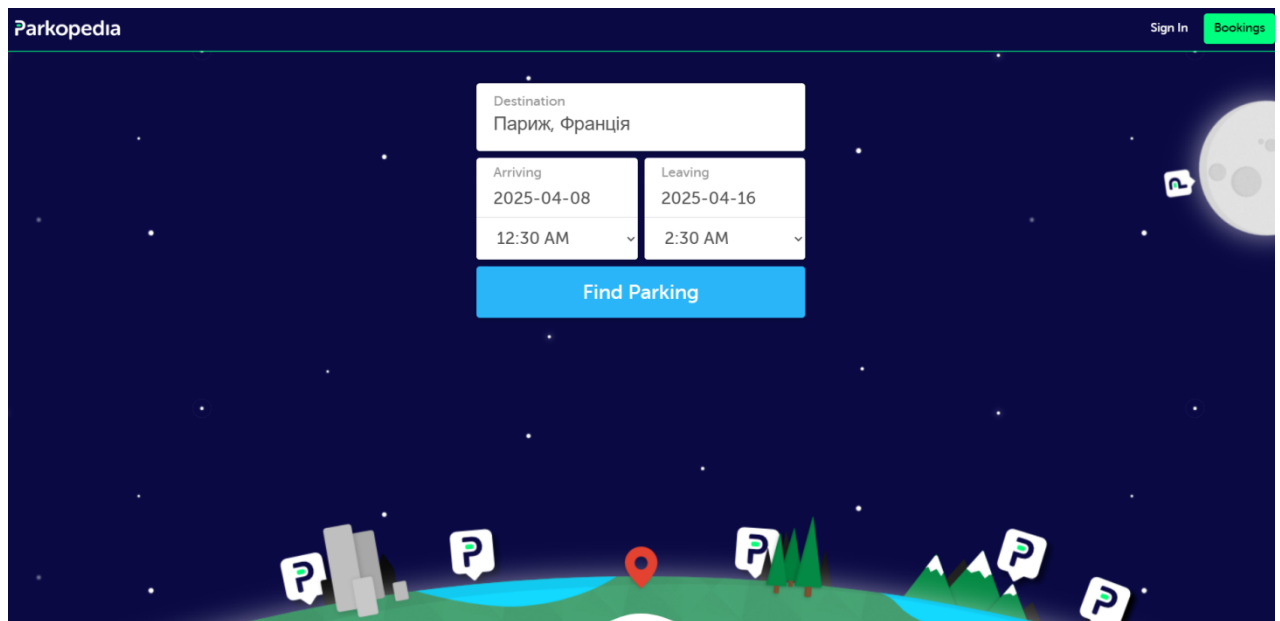


Рисунок 1.2 – Програмне забезпечення Parkopedia [11]

Програмне забезпечення Parkopedia підтримує функції попереднього резервування, здійснення платежів та отримання цифрових дозволів на паркування, що значно спрощує користування автостоянками. Для бізнесу та міських служб передбачено аналітичні інструменти, які дозволяють оцінювати рівень завантаженості паркувальних зон, коригувати тарифи та впроваджувати ефективні підходи до управління стоянками. Використання цієї платформи сприяє зменшенню дорожніх заторів, покращенню організації міського простору та створенню комфортних умов для водіїв [7], [11].

Програмне забезпечення Parkopedia має такі особливості [7], [11]:

- надання інформації в реальному часі: програмне забезпечення Parkopedia дозволяє користувачам отримувати актуальні дані про наявність паркомісць, їхнє розташування та вартість, що спрощує процес пошуку зручного варіанту паркування [7], [11];

- велика база даних: програмне забезпечення Parkopedia містить відомості про велику кількість паркувальних місць у різних країнах, охоплюючи як вуличні, так і закриті автостоянки, що забезпечує широкий вибір для водіїв [7], [11];

- інтеграція з навігаційними системами: програмне забезпечення Parkopedia підтримує взаємодію з GPS та картографічними сервісами, допомагаючи спрямовувати водіїв безпосередньо до обраного місця для паркування [7], [11];
- можливість бронювання: програмне забезпечення Parkopedia надає функцію попереднього резервування паркомісць, що дозволяє уникнути зайвих затримок та забезпечує гарантовану наявність місця для авто [7], [11];
- підтримка безготівкових платежів: програмне забезпечення Parkopedia дозволяє здійснювати оплату паркування через мобільний застосунок або вебплатформу, що підвищує зручність користування [7], [11];
- інструменти для операторів паркінгів: програмне забезпечення Parkopedia забезпечує компанії та муніципалітети інструментами для аналітики, моніторингу завантаженості стоянок і оптимізації тарифної політики [7], [11];
- персоналізовані налаштування: програмне забезпечення Parkopedia дає змогу користувачам встановлювати фільтри за типом паркування, рівнем доступності, наявністю зарядних станцій для електромобілів та іншими параметрами [7], [11];
- багатоплатформний доступ: програмне забезпечення Parkopedia підтримує роботу на мобільних пристроях, планшетах та комп'ютерах, забезпечуючи зручний доступ до сервісу незалежно від використовуваного пристрою [7], [11];
- автоматичне оновлення даних: програмне забезпечення Parkopedia постійно оновлює інформацію про паркомісця, хоча можливі затримки у відображенні актуального стану зайнятості [7], [11];
- розширені можливості для бізнесу: програмне забезпечення Parkopedia пропонує компаніям рішення для управління корпоративними автопарками, включаючи бронювання місць для співробітників та аналіз витрат на паркування [7], [11].

Серед недоліків програмного забезпечення Parkopedia можна відзначити обмежену точність даних щодо реальної заповненості паркувальних місць, оскільки інформація може оновлюватися з затримкою. Деякі стоянки можуть не підтримувати автоматизоване бронювання та оплату, що ускладнює процес паркування для користувачів. Інтеграція з навігаційними системами не завжди працює коректно, що може призводити до помилок у визначенні місця призначення. У деяких регіонах база даних може містити недостатню кількість паркувальних майданчиків або застарілі відомості, що обмежує ефективність використання сервісу. Крім того, для доступу до деяких функцій може знадобитися платна підписка, що створює додаткові витрати для користувачів [7], [11].

Програмне забезпечення ParkMobile (рис. 1.3) надає засоби для ефективного управління паркувальними просторами, сприяючи організації стоянок як у комерційних зонах, так і в муніципальних структурах. Дана платформа дає змогу користувачам швидко знаходити та резервувати місця, оплачувати паркування та контролювати його тривалість за допомогою мобільних пристроїв. Вбудовані безконтактні методи оплати роблять користування послугою зручним як для водіїв, так і для адміністраторів автостоянок [7], [12].

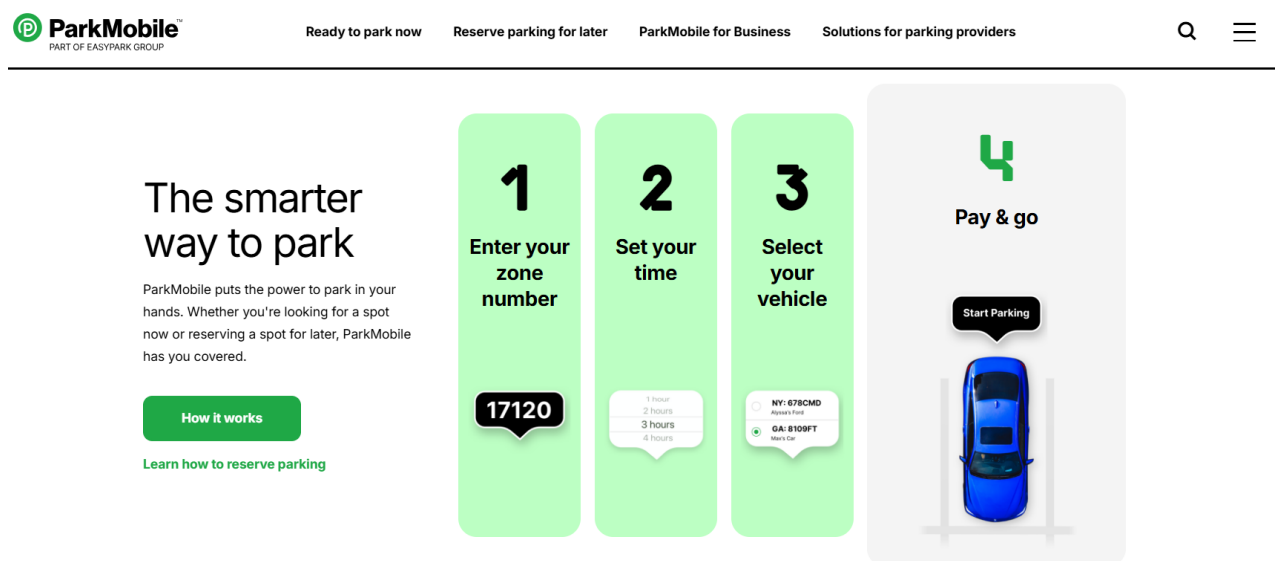


Рисунок 1.3 – Програмне забезпечення ParkMobile [12]

Для операторів паркувальних майданчиків надається можливість відстеження в реальному часі зайнятості місць, аналізу заповненості та моніторингу платіжних операцій, що дає змогу коригувати тарифи та оптимізувати використання ресурсів. Додатково передбачені функції автоматизованого виставлення рахунків, гнучкого регулювання вартості залежно від подій та детальної звітності, що допомагає мінімізувати адміністративні витрати та підвищити прибутковість [7], [12].

Рішення підходить для широкого спектра об'єктів, включаючи ділові центри, навчальні заклади та густонаселені міські райони. Використання цієї системи сприяє зменшенню навантаження на інфраструктуру, знижує рівень транспортних заторів та підвищує загальну зручність користування паркувальними послугами як для власників авто, так і для операторів [7], [12].

Програмне забезпечення ParkMobile має такі особливості [7], [12]:

- зручний мобільний застосунок: програмне забезпечення ParkMobile надає швидкий доступ до функцій паркування через мобільний телефон, дозволяючи користувачам сервісу здійснювати оплату та керувати часом стоянки дистанційно [7], [12];

- безконтактні платежі: програмне забезпечення ParkMobile підтримує безготівкові розрахунки, що підвищує комфорт користувачів і зменшує необхідність у фізичних паркоматах [7], [12];

- бронювання паркомісць: програмне забезпечення ParkMobile надає змогу заздалегідь резервувати місця, що особливо корисно в умовах обмеженої кількості паркувальних зон [7], [12];

- аналітика та звітність: програмне забезпечення ParkMobile надає операторам детальні дані про використання паркувальних місць, тенденції платежів і рівень заповнюваності [7], [12];

- інтеграція з міськими системами: програмне забезпечення ParkMobile підтримує підключення до муніципальних платформ, що спрощує управління паркувальним простором [7], [12];

- адаптивне ціноутворення: програмне забезпечення ParkMobile дозволяє встановлювати змінні тарифи залежно від часу доби, попиту або спеціальних подій [7], [12];
- автоматичне продовження часу стоянки: програмне забезпечення ParkMobile надсилає сповіщення та дозволяє подовжити паркування без необхідності повертатися до автомобіля [7], [12];
- підтримка різних способів оплати: програмне забезпечення ParkMobile підтримує оплату через банківські картки, електронні гаманці та інші платіжні методи для зручності користувачів [7], [12];
- геолокаційний пошук паркінгу: програмне забезпечення ParkMobile допомагає знайти найближчі доступні місця, використовуючи дані про поточне місцезнаходження [7], [12];
- оптимізація міського паркування: програмне забезпечення ParkMobile сприяє зменшенню заторів і підвищенню ефективності використання паркомісць у міських районах [7], [12].

Серед недоліків програмного забезпечення ParkMobile можна відзначити залежність від стабільного інтернет-з'єднання, без якого доступ до функцій системи може бути ускладненим. Деякі користувачі стикаються з технічними збоями, такими як затримки в оновленні статусу паркувальних місць або проблеми з оплатою через мобільний додаток. Крім того, можливість використання програмного забезпечення ParkMobile обмежена певними регіонами, що унеможлиблює його застосування в містах або країнах, де система не підтримується. У деяких випадках водії можуть зіткнутися з додатковими комісійними зборами за транзакції, що збільшує загальну вартість паркування. Ще одним недоліком є обмежений рівень інтеграції з деякими міськими системами паркування, що може ускладнювати управління паркомісцями для операторів та муніципалітетів [7], [12].

Порівняльну характеристику інформаційних систем для підтримки роботи автостоянки наведено у таблиці 1.1.

Таблиця 1.1 – Порівняння інформаційних систем для підтримки роботи автостоянки

Критерій порівняння	SpotHero [10]	Parkopedia [11]	ParkMobile [12]
Зручність бронювання	+	+—	+
Інформація про наявність місць на автостоянках у реальному часі	+—	+	+
Інтеграція з навігаційними системами	+—	+	+—
Автоматичне продовження паркування	—	—	+
Аналітика та звітність для операторів	+—	+	+
Підтримка різних способів оплати	+	—	+

За результатами проведеного аналізу можна зробити висновок, що у наш час існує досить багато інформаційних систем для підтримки роботи автостоянки. Проте деякі програмні засоби інформаційних систем для підтримки роботи автостоянок можуть мати високу вартість впровадження. Розробка, налаштування та інтеграція такої програмної системи потребують значних фінансових вкладень, особливо якщо вона включає відео спостереження та онлайн сервіси. Окрім того, для підтримки стабільної роботи необхідні регулярні оновлення, що також тягне за собою додаткові витрати. Крім того, можливі збої в інтеграції програмних засобів інформаційних систем для підтримки роботи автостоянок з іншими системами або платіжними платформами. Це може створювати незручності для користувачів та знижувати ефективність роботи автостоянки. Тому актуальною є розробка програмного забезпечення інформаційної системи для підтримки роботи автостоянки.

При розробці програмного забезпечення інформаційної системи для підтримки роботи автостоянки необхідно забезпечити такі функціональні вимоги:

- підтримка можливості розрахунку вартості стоянки за введеними користувачем даними;
- можливість реєстрації та авторизації користувачів;
- підтримка можливості обрання автостоянки для паркування автомобіля;
- підтримка можливості вибору додаткових опцій на стоянці;
- підтримка можливості керування (скасування) бронювань;
- підтримка можливості обрання часового періоду стоянки автомобіля на обраній парковці.

1.3 Висновки за розділом 1

Визначено, що програмне забезпечення для підтримки функціонування автостоянки є важливим елементом сучасної інфраструктури, що забезпечує ефективне управління паркувальними місцями, спрощує процес бронювання та оплати, а також підвищує загальний рівень комфорту як для користувачів, так і для адміністрації автостоянки.

За результатами проведеного аналізу зроблено висновок, що у наш час існує досить багато інформаційних систем для підтримки діяльності для підтримки роботи автостоянки. Проте деякі програмні засоби для підтримки роботи автостоянок можуть мати високу вартість впровадження. Розробка, налаштування та інтеграція такої програмної системи потребують значних фінансових вкладень, особливо якщо вона включає відео спостереження та онлайн сервіси. Окрім того, для підтримки стабільної роботи необхідні регулярні оновлення, що також тягне за собою додаткові витрати. Крім того, можливі збої в інтеграції програмних засобів інформаційних систем для підтримки роботи автостоянок з іншими системами або платіжними

платформами. Це може створювати незручності для користувачів та знижувати ефективність роботи автостоянки. Тому актуальною є розробка програмного забезпечення інформаційної системи для підтримки роботи автостоянки.

Сформульовано функціональні вимоги до програмного забезпечення інформаційної системи для підтримки роботи автостоянки.

2 МАТЕРІАЛИ І МЕТОДИ

2.1 Вибір мови програмування

При розробці програмного забезпечення інформаційної системи для підтримки роботи автостоянки було використано мову програмування Java [13], [14].

Java належить до об'єктно-орієнтованих мов програмування, що означає, що основним принципом її роботи є організація коду у вигляді взаємодіючих об'єктів. Такі об'єкти містять дані та методи, які визначають їхню поведінку, а також можуть комунікувати між собою, що робить структуру програми більш гнучкою та зручною для розширення. Використання об'єктів значно спрощує процес розробки, оскільки розробнику потрібно працювати не з окремими рядками коду, а з цілими сутностями, що взаємодіють між собою [13], [14].

Однією з ключових концепцій об'єктно-орієнтованого підходу є інкапсуляція, яка забезпечує обмежений доступ до внутрішніх даних об'єкта, що запобігає некоректним змінам та підвищує надійність програмного забезпечення. Успадкування дозволяє створювати нові класи на основі вже існуючих, що сприяє повторному використанню коду та спрощує підтримку системи. Поліморфізм дає змогу використовувати один і той самий метод для різних типів об'єктів, що значно спрощує розширення функціональності програми [13], [14].

Окрім об'єктно-орієнтованості, Java є кросплатформною мовою, що означає її здатність працювати на різних операційних системах без необхідності зміни коду. Завдяки технології Java Virtual Machine (JVM) програми, написані на Java, можуть запускатися на будь-якому пристрої, де встановлено відповідне середовище виконання [13], [14].

Java також відзначається високим рівнем безпеки, оскільки має вбудовані механізми захисту, що запобігають виконанню шкідливого коду. Управління пам'яттю здійснюється автоматично завдяки механізму збирання

сміття, який вивільняє ресурси, що більше не використовуються, що покращує продуктивність і запобігає витокам пам'яті [13], [14].

Мова Java широко застосовується у веброзробці, мобільних додатках, корпоративних системах, розподілених обчисленнях, фінансових технологіях та розробці ігор. Для створення вебзастосунків використовуються фреймворки Spring Boot та Jakarta EE, які забезпечують ефективну роботу серверної частини. Розробка мобільних додатків для Android здебільшого здійснюється саме на Java, що робить її основною мовою для створення мобільного програмного забезпечення. У корпоративному секторі Java активно застосовується у фінансових системах та банківських додатках, де важливими є безпека та надійність [13], [14].

Популярність Java пояснюється її масштабованістю, великою кількістю бібліотек та фреймворків, а також активною спільнотою розробників, що сприяє постійному розвитку мови. Завдяки своїй універсальності та стабільності вона залишається одним із найкращих виборів для створення складних та надійних інформаційних систем [13], [14].

Для розробки програмного забезпечення інформаційної системи автостоянки Java є оптимальним вибором. Завдяки підтримці різних платформ ця мова дозволяє створювати рішення, які можуть працювати на мобільних пристроях, серверних системах та вебінтерфейсах без необхідності значних змін у коді [13], [14].

Високий рівень безпеки, що забезпечується механізмами управління доступом та криптографічними API, робить її особливо придатною для роботи з фінансовими операціями та персональними даними користувачів. Автоматичне управління пам'яттю допомагає оптимізувати ресурси системи, що важливо для роботи у високонавантажених середовищах [13], [14].

Обґрунтування вибору мови програмування для розробки програмного забезпечення інформаційної системи для підтримки роботи автостоянки наведено у таблиці 2.1.

Таблиця 2.1 – Обґрунтування вибору мови програмування для розробки програмного забезпечення інформаційної системи для підтримки роботи автостоянки

Критерій порівняння мов програмування	Мова програмування		
	Java	Python	JavaScript
Кросплатформність	+	+—	+—
Продуктивність	+	—	—
Масштабованість	+	+—	+—
Підтримка об'єктно-орієнтованого програмування	+	+—	—
Екосистема та фреймворки	+	+	+—

Отже, для реалізації програмного забезпечення інформаційної системи для підтримки роботи автостоянки обрано мову програмування Java, яка є кросплатформною мовою, що дозволяє запускати програму на різних операційних системах без необхідності значних модифікацій. Крім того, Java має високу надійність та безпеку, що є критично важливим для систем, які працюють з персональними даними клієнтів, номерами автомобілів та фінансовими операціями. Вбудовані механізми управління пам'яттю та система безпеки мови Java дозволяють створити захищене програмне середовище.

2.2 Вибір середовища розробки для створення програмного забезпечення інформаційної системи для підтримки роботи автостоянки

В якості середовища розробки для створення програмного забезпечення інформаційної системи для підтримки роботи автостоянки було обрано середовище Android Studio [15], [16].

Середовище розробки Android Studio є основним середовищем для розробки мобільних застосунків на платформі Android. Це інтегроване середовище розробки, створене на основі IntelliJ IDEA, яке пропонує розробникам широкий набір інструментів для створення, тестування та оптимізації програмного забезпечення. У цьому середовищі доступні засоби для автоматизації процесу складання, діагностики потенційних проблем та аналізу продуктивності [15], [16].

Google активно підтримує Android Studio, надаючи оновлення та розширення функціональності, що дозволяє ефективно працювати з операційною системою Android та інтегрувати різноманітні сервіси, зокрема хмарні рішення. Вбудовані засоби розробки включають емулятор для тестування, редактор коду з підсвічуванням синтаксису, інструменти профілювання та аналітики [15], [16].

Серед ключових можливостей варто відзначити зручний інтерфейс проектування, що дозволяє швидко створювати та налаштовувати користувацький інтерфейс застосунків. Використання XML-редактора та бібліотеки Jetpack спрощує створення адаптивних макетів і взаємодію з компонентами системи [15], [16].

Розробка на платформі Android Studio може здійснюватися як мовою Java, так і Kotlin, що забезпечує високу продуктивність і гнучкість. Крім того, підтримується інтеграція з такими сервісами, як Firebase, Google Maps API та Google Pay, що значно розширює можливості застосунків [15], [16].

Емулятор, вбудований у середовище розробки, дозволяє тестувати застосунки без необхідності використовувати реальний пристрій, моделюючи різні сценарії роботи. Також Android Studio містить інструменти для публікації застосунків у Google Play [15], [16].

Постійне вдосконалення Android Studio забезпечує розробникам доступ до новітніх технологій і методів роботи, що дозволяє створювати якісні, стабільні та продуктивні мобільні застосунки [15], [16].

Обґрунтування вибору середовища розробки для створення програмного забезпечення інформаційної системи для підтримки роботи автостоянки наведено у таблиці 2.2.

Таблиця 2.2 – Обґрунтування вибору середовища розробки для створення програмного забезпечення інформаційної системи для підтримки роботи автостоянки

Критерій порівняння середовищ розробки	Середовища розробки		
	Android Studio	Visual Studio	PyCharm
Продуктивність та швидкість роботи	+	+	+—
Велика кількість бібліотек та плагінів	+	+	+—
Оптимізація для мобільних застосунків	+	+—	—
Гнучкість у роботі з базами даних	+	+	+—
Вбудований емулятор для тестування	+	+—	—

Таким чином, для створення програмного забезпечення інформаційної системи для підтримки роботи автостоянки обрано середовище розробки Android Studio, що завдяки вбудованим інструментам дозволяє легко розроблювати інтуїтивно зрозумілий інтерфейс користувача, який зробить роботу з програмою швидкою та зручною. Крім того, Android Studio пропонує потужний емулятор, інструменти профілювання продуктивності, що дозволяє створювати швидкі та стабільні застосунки.

2.3 Висновки за розділом 2

Для реалізації програмного забезпечення інформаційної системи для підтримки роботи автостоянки обрано мову програмування Java, яка є кросплатформною мовою, що дозволяє запускати програму на різних операційних системах без необхідності значних модифікацій. Крім того, Java має високу надійність та безпеку, що є критично важливим для систем, які працюють з персональними даними клієнтів, номерами автомобілів та фінансовими операціями. Вбудовані механізми управління пам'яттю та система безпеки мови Java дозволяють створити захищене програмне середовище.

Для створення програмного забезпечення інформаційної системи для підтримки роботи автостоянки обрано середовище розробки Android Studio, що завдяки вбудованим інструментам дозволяє легко розроблювати інтуїтивно зрозумілий інтерфейс користувача, який зробить роботу з програмою швидкою та зручною. Крім того, Android Studio пропонує потужний емулятор, інструменти профілювання продуктивності, що дозволяє створювати швидкі та стабільні застосунки.

3 ОПИС ПРОГРАМИ

3.1 Структура програмного забезпечення інформаційної системи для підтримки роботи автостоянки

Структуру програмного забезпечення інформаційної системи для підтримки роботи автостоянки наведено на рис. 3.1.



Рисунок 3.1 – Структура програмного забезпечення інформаційної системи для підтримки роботи автостоянки

Програмне забезпечення інформаційної системи для управління автостоянкою базується на шести функціональних модулях, кожен з яких відповідає за окремий аспект роботи сервісу. Центральне місце в архітектурі

займає головний клас, який визначає глобальний стан системи. Його ініціалізація відбувається одразу при запуску, і він залишається активним протягом усього часу функціонування програми. Саме тут зберігаються змінні, доступні з будь-якої частини коду.

Для взаємодії з користувачем реалізовано кілька інтерфейсних компонентів. Спершу відображається екран аутентифікації, де виконується перевірка користувача та демонструється логотип. Далі пропонується інтерфейс швидкого замовлення паркувальних послуг, що дозволяє вводити мінімальну кількість даних. Передбачений також окремий розділ для авторизації, що містить поля введення облікових даних та кнопку переходу до бронювання. Після входу користувач потрапляє в основне меню, де надається доступ до перегляду поточних, майбутніх, завершених чи скасованих бронювань. Окрім цього, є функціонал для повідомлення про повернення, а також можливість редагувати активні та майбутні замовлення.

Процес бронювання реалізовано через спеціальний компонент, що містить набір фрагментів, кожен з яких відповідає за окремий етап реєстрації бронювання. Збереження платіжної інформації відбувається в окремому модулі, де користувач може додавати чи редагувати картки. Управління особистими даними та інформацією про компанію також винесено в окремий фрагмент, що дозволяє змінювати, додавати або видаляти відповідні відомості. У спеціальному інтерфейсі відображаються всі бронювання користувача з можливістю їхнього перегляду, а аналогічний розділ для транспортних засобів забезпечує роботу з даними про автомобілі.

Для коректної роботи з мережею передбачено компонент, що визначає статус підключення. Окремий механізм відповідає за модифікацію запитів, дозволяючи вносити необхідні заголовки. Реалізовано інтерфейс, який описує об'єкти, що генеруються через систему впровадження залежностей. Робота з базою даних здійснюється через абстрактний клас, який містить опис усіх необхідних сутностей.

Здійснення запитів до серверної частини реалізовано за допомогою сервісного інтерфейсу, що містить визначення основних запитів до API. Окремий компонент відповідає за вихід користувача з системи, видаляючи всі дані та надсилаючи запит про завершення сесії. Таким чином, структура програмного забезпечення автостоянки забезпечує чітке розділення функціональності, що дозволяє ефективно керувати всіма процесами, пов'язаними з бронюванням, оплатою та адмініструванням послуг.

Програмне забезпечення інформаційної системи для управління автостоянкою складається з шести ключових модулів (модуль керування життєвим циклом програми, модуль аутентифікації та роботи з користувачами, модуль оформлення бронювання, модуль фінансових операцій, модуль моніторингу та звітності, модуль мережевої взаємодії), кожен з яких виконує окрему функціональну роль у загальній інформаційній системі.

Модуль керування життєвим циклом програми відповідає за ініціалізацію глобального стану системи та підтримку основних параметрів протягом усього часу роботи програми. Головний клас цього модуля створюється при запуску програми та зберігає змінні, які використовуються в інших частинах додатка.

Модуль аутентифікації та роботи з користувачами відповідає за перевірку особи користувача, надання йому доступу до системи та керування персональними даними. Він містить екрани автентифікації, реєстрації та авторизації, а також інтерфейс для редагування особистої інформації та налаштувань профілю.

Основна функція модулю оформлення бронювання полягає у забезпеченні процесу резервування паркувального місця, починаючи з вибору місця, введення необхідної інформації та підтвердження замовлення. Цей модуль також дозволяє переглядати деталі активних, минулих та майбутніх бронювань, а також їх зміну або скасування.

Модуль фінансових операцій містить усі механізми, пов'язані з платежами, включаючи збереження інформації про платіжні карти, здійснення транзакцій, перевірку статусу оплат та історію фінансових операцій.

Модуль моніторингу та звітності призначений для збору, аналізу та відображення інформації щодо використання автостоянки. Він містить інструменти для перегляду звітів про завантаженість паркінгу, історію бронювань та статистику фінансових операцій.

Модуль мережевої взаємодії забезпечує зв'язок з віддаленими серверами, перевіряє наявність Інтернет з'єднання, обробляє запити до API та додає необхідні заголовки для коректної роботи сервісу. Також цей модуль містить механізми виходу з системи та очищення локальних даних користувача при завершенні сесії.

Кожен із цих модулів взаємодіє з іншими, формуючи комплексну систему управління автостоянкою, що поєднує в собі ефективне бронювання та зручність користування програмою.

3.2 Функціонування програмного забезпечення інформаційної системи для підтримки роботи автостоянки

Функціонування програмного забезпечення інформаційної системи для підтримки роботи автостоянки подамо за допомогою діаграми прецедентів, зображеної на рис. 3.2.

Функціонування ПЗ ІС для управління автостоянкою передбачає взаємодію користувача з системою через набір основних дій. Першим кроком є авторизація, яка дозволяє користувачеві увійти в систему для подальшого виконання дій. Після успішного входу користувач отримує можливість заповнити дані бронювання, що включає вибір місця для паркування, час перебування та інші необхідні параметри.



Рисунок 3.2 – Схема функціонування ПЗ ІС для підтримки роботи автостоянки у вигляді діаграми прецедентів

Коли всі дані введені, система дозволяє підтвердити платіж. На цьому етапі відбувається взаємодія між користувачем і системою: система перевіряє правильність платежу, після чого ініціює отримання коштів за бронювання.

Окрім базових операцій, користувач також має змогу редагувати власні дані або змінювати параметри бронювання. Якщо бронювання більше не є актуальним, у системі передбачена можливість його скасування.

Завдяки такій організації роботи, програмне забезпечення забезпечує зручний і послідовний процес бронювання місця на автостоянці, надаючи користувачам можливість швидко виконувати необхідні дії, а системі – ефективно обробляти бронювання та платежі.

При розробці програми створено різні алгоритми, що забезпечують підтримку процесів функціонування автостоянки. На рис. 3.3 наведено алгоритм передавання даних про бронювання користувача.

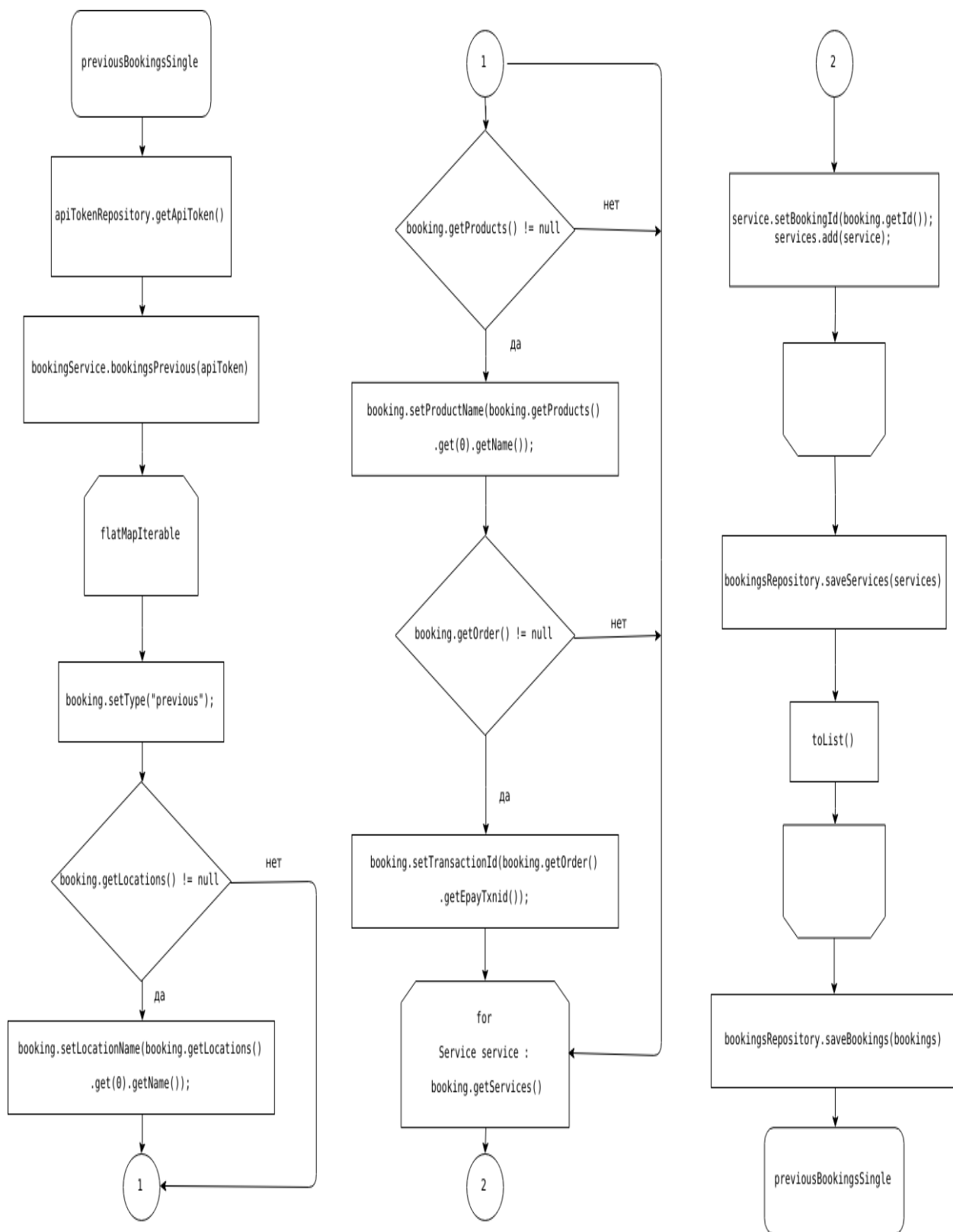


Рисунок 3.3 – Алгоритм передавання даних про бронювання користувача

Алгоритм отримання даних про бронювання користувача наведено на рис. 3.4.

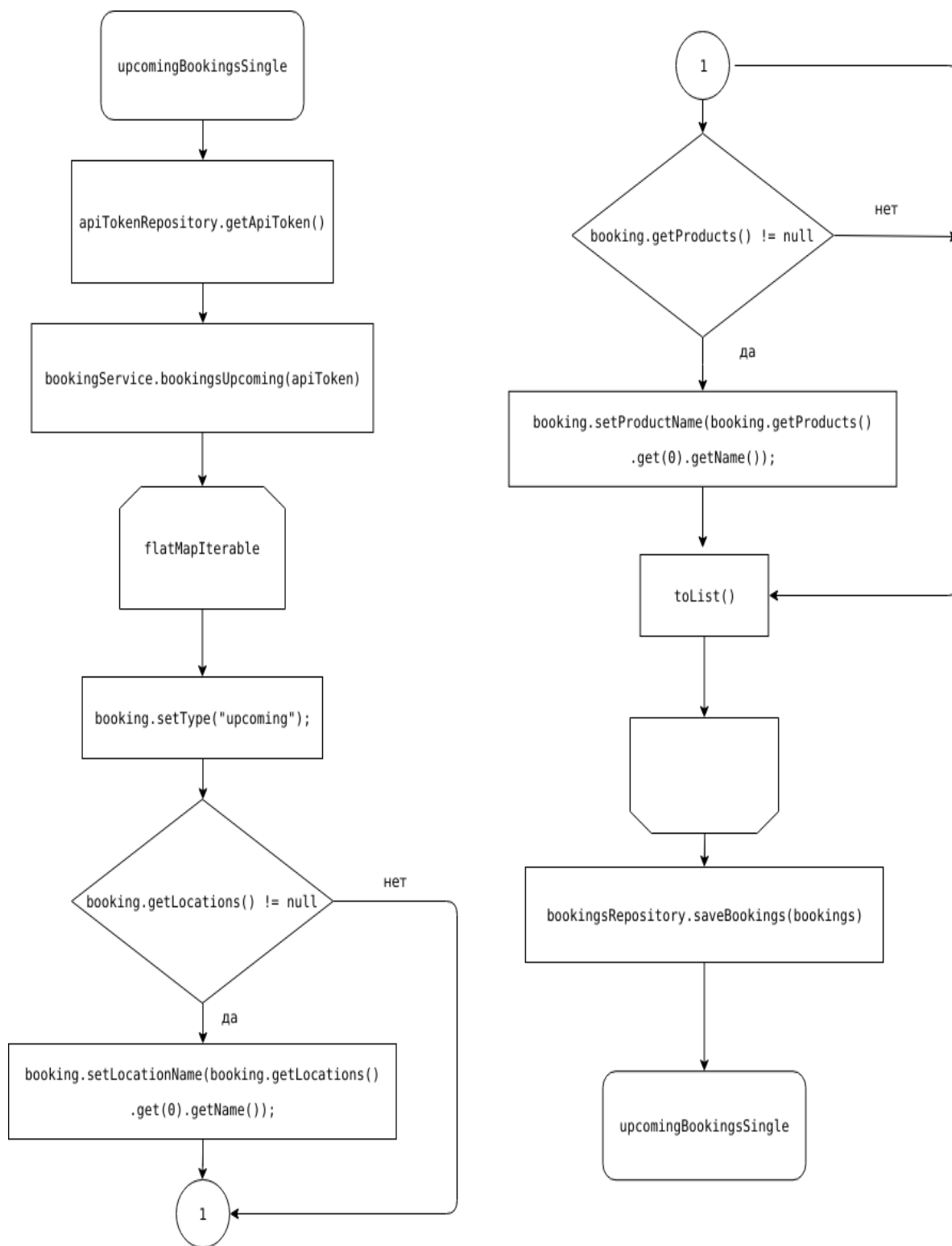


Рисунок 3.4 – Алгоритм отримання даних про бронювання користувача

Алгоритм відправлення запиту при вході до програмного забезпечення інформаційної системи для підтримки роботи автостоянки наведено на рис. 3.5.

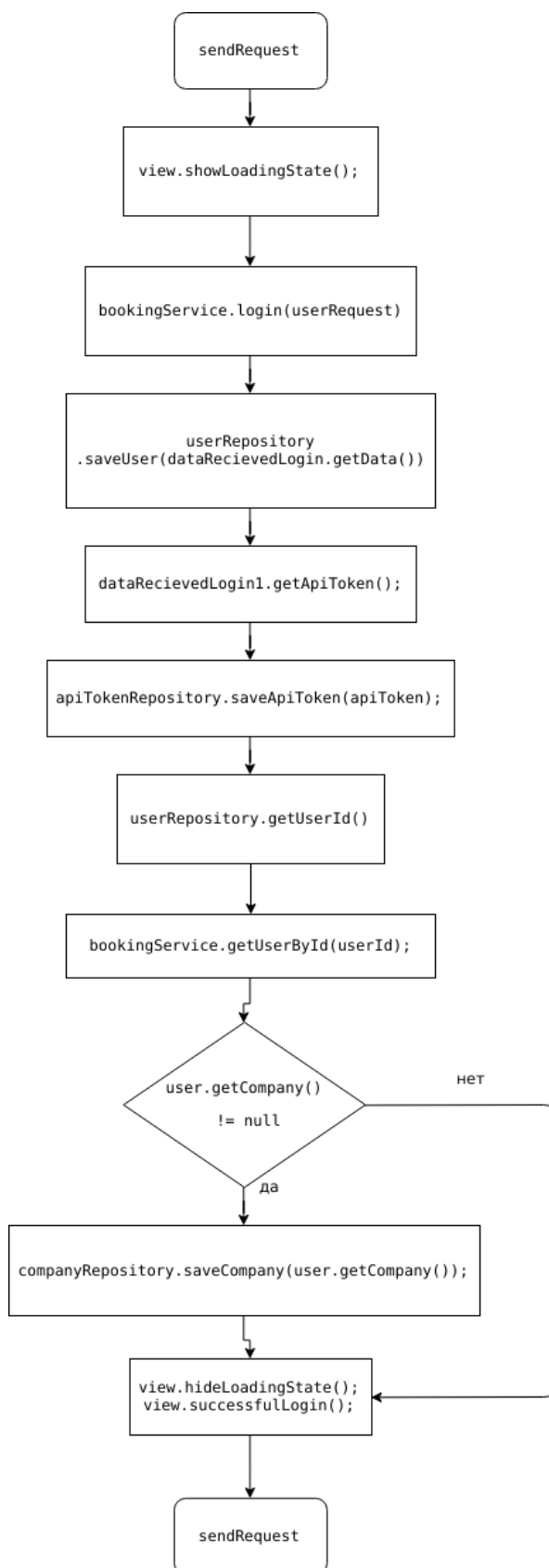


Рисунок 3.5 – Алгоритм відправлення запиту при вході до програмного забезпечення інформаційної системи для підтримки роботи автостоянки

3.3 Розробка бази даних програмного забезпечення інформаційної системи для підтримки роботи автостоянки

База даних у програмному забезпеченні інформаційної системи для підтримки роботи автостоянки виконує ключову роль у зберіганні, управлінні та обробці даних, необхідних для функціонування системи. Вона забезпечує надійне збереження інформації про користувачів, бронювання паркомісць, фінансові операції та додаткові послуги.

У розробленій інформаційній системі для управління автостоянкою використовується клієнт-серверна архітектура, що передбачає збереження даних у базі даних. База даних містить дві основні таблиці: Booking і Service.

Схему бази даних програмного забезпечення інформаційної системи для підтримки роботи автостоянки наведено на рис. 3.6.

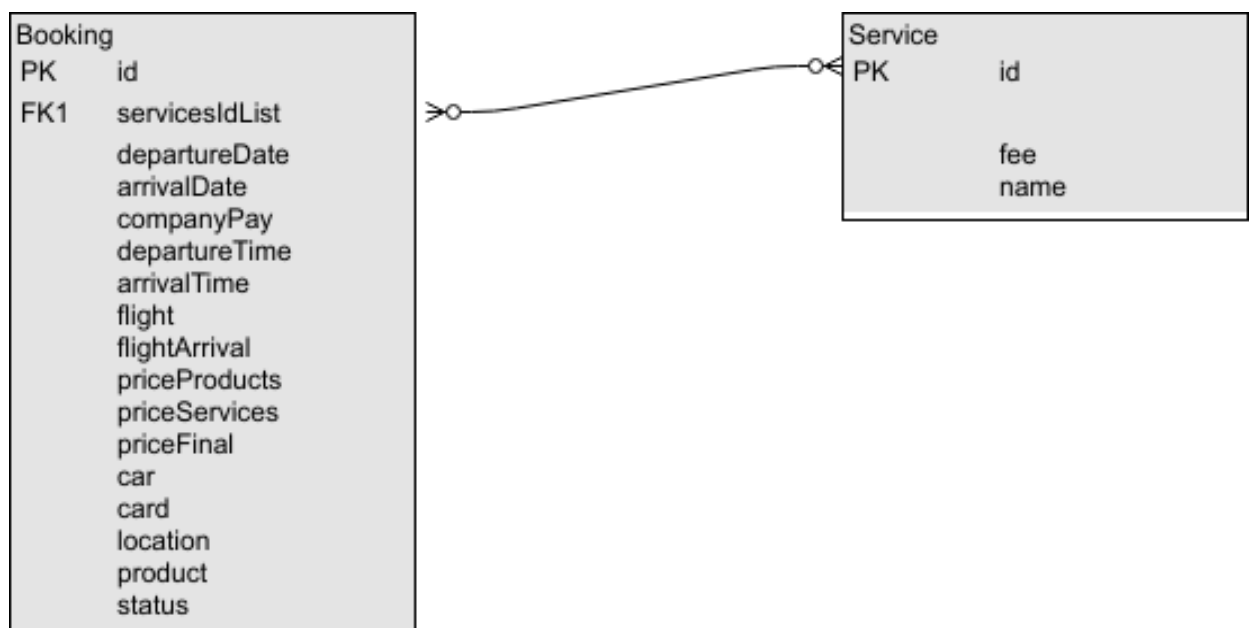


Рисунок 3.6 – Схема бази даних програмного забезпечення інформаційної системи для підтримки роботи автостоянки

Таблиця Booking зберігає інформацію про бронювання місць на автостоянці. У ній фіксуються такі параметри, як унікальний ідентифікатор бронювання, дані користувача, час початку та завершення бронювання, статус

(активне, скасоване, завершене), а також посилання на відповідні записи у таблиці Service.

Таблиця Service містить інформацію про додаткові послуги, які користувач може замовити разом із бронюванням. Це можуть бути, наприклад, мийка авто, зарядка електромобіля або інші послуги.

Основне призначення бази даних – це організація централізованого доступу до даних, що дозволяє різним модулям системи працювати з актуальною та узгодженою інформацією. Вона підтримує швидкий пошук і вибірку потрібних записів, наприклад, активних або минулих бронювань, дозволяє редагувати інформацію та оновлювати статуси бронювань у режимі реального часу.

Окрім цього, база даних сприяє автоматизації процесів оплати, зберігаючи деталі транзакцій, підтвердження платежів та інформацію про користувацькі платіжні методи. Завдяки цьому система може коректно обробляти фінансові операції та надавати користувачам доступ до історії їхніх платежів.

Таким чином, база даних забезпечує ефективну взаємодію всіх компонентів інформаційної системи, сприяє її стабільній роботі, зменшує ризик втрати даних і підвищує рівень обслуговування користувачів.

3.4 Проєктування інтерфейсу програмного забезпечення інформаційної системи для підтримки роботи автостоянки

Проєктування інтерфейсу взаємодії користувача з програмним забезпеченням інформаційної системи для підтримки роботи автостоянки виконано з урахуванням вимог технічного завдання. При розробленні програмного забезпечення інформаційної системи для підтримки роботи автостоянки враховані функціональні вимоги до програми:

- підтримка можливості розрахунку вартості стоянки за введеними користувачем даними;

- можливість реєстрації та авторизації користувачів;
- підтримка можливості обрання автостоянки для паркування автомобіля;
- підтримка можливості вибору додаткових опцій на стоянці;
- підтримка можливості обрання часового періоду стоянки автомобіля на обраній парковці;
- підтримка можливості керування (скасування) бронювань.

Інтерфейс головного екрану програмного забезпечення інформаційної системи для підтримки роботи автостоянки наведено на рис. 3.7.

New booking

park one

Choose

Bln

Departure: 10-05-2025

Arrival: 15-05-2025

114€

Next

Рисунок 3.7 – Інтерфейс головного екрану програмного забезпечення інформаційної системи для підтримки роботи автостоянки

Екран входу користувача до програмного забезпечення інформаційної системи для підтримки роботи автостоянки наведено на рис. 3.8.

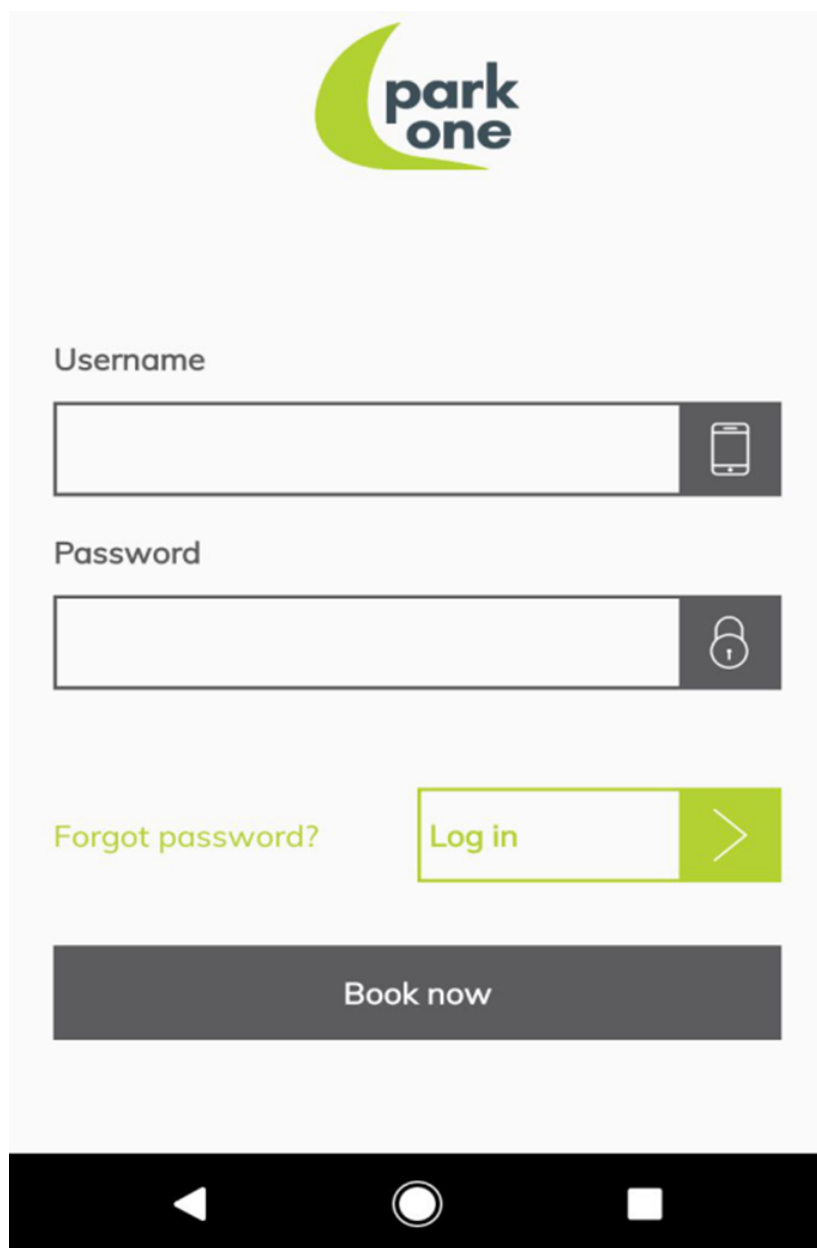


Рисунок 3.8 – Екран входу користувача до програмного забезпечення інформаційної системи для підтримки роботи автостоянки

Екран підтвердження даних про дати бронювання автостоянки наведено на рис. 3.9.

Екран вибору додаткових опцій на стоянці наведено на рис. 3.10.

New booking

Choose date

1 2 3 4 5 6 7

Select departure date Select arrival date

10-05-2025 17 15-05-2025 17

Departure time Arrival time

08:00 17:00

Рисунок 3.9 – Екран підтвердження даних про дати бронювання автостоянки

New booking

Select additional services

1 2 3 4 5 6 7

Click and add services you would like

☒ Outside Hand Car wash	35€
☒ Inside Windows Cleaning	25€
☐ Tire Pressure Check	8€
☐ Tire Pressure	8€
☐ Fuel service	5€
Total in services	60€

Рисунок 3.10 – Екран вибору додаткових опцій на стоянці

Екран введення даних про власника автомобіля наведено на рис. 3.11.

Екран введення даних про автомобіль наведено на рис. 3.12.

Menu icon | New booking

Provide/confirm details

1 2 3 4 5 6 7

Title
Ms. [person icon]

First name*
Vlad [person icon]

Last name*
Sagko [person icon]

Job Title
FOP [person icon]

Рисунок 3.11 – Екран введення даних про власника автомобіля

Menu icon | New booking

Confirm/add car

1 2 3 4 5 6 7

Existing Cars:

License plate: Qw123ee
Car color: White
Car manufacture: Germany

Edit car [checkbox] Use car

Car manufacture
[car icon]

Car color
[car icon]

Рисунок 3.12 – Екран введення даних про автомобіль

Екрани введення даних про банківську картку та підтвердження даних бронювання наведено на рис. 3.13 та 3.14, відповідно.

New booking

Confirm/add billing

1 2 3 4 5 6 7

Use existing billing

3000 33XX XXXX 3000
/

VISA

☐ Use card

Add New card >

Рисунок 3.13 – Екран введення даних про банківську картку

New booking

Parking summary

1 2 3 4 5 6 7

Mobile phone no. +38050773977880

Billing details

3000 33XX XXXX 3000
/

VISA

☒ I agree to the [terms of agreement](#)

Confirm and go to payment >

Рисунок 3.14 – Екран підтвердження даних бронювання

Екран зі списком бронювань користувача місць на стоянках наведено на рис. 3.15.

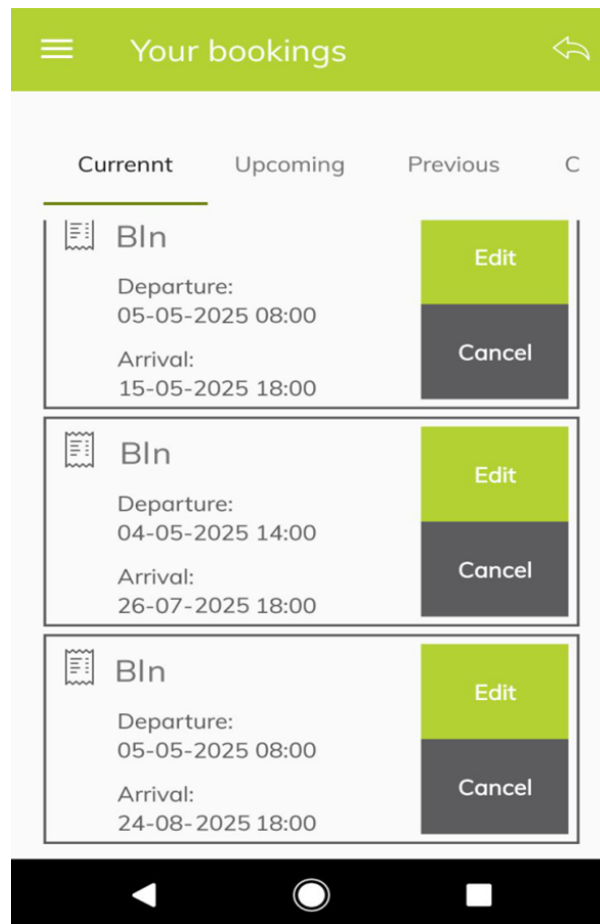


Рисунок 3.15 – Список бронювань користувача місць на стоянках

3.5 Висновки за розділом 3

Розроблено програмне забезпечення інформаційної системи для підтримки роботи автостоянки.

Запропоновано структуру програмного забезпечення інформаційної системи для підтримки роботи автостоянки, що складається з шести ключових модулів (модуль керування життєвим циклом програми, модуль аутентифікації та роботи з користувачами, модуль оформлення бронювання, модуль фінансових операцій, модуль моніторингу та звітності, модуль

мережевої взаємодії), кожен з яких виконує окрему функціональну роль у загальній інформаційній системі.

Розроблено базу даних програмного забезпечення інформаційної системи для підтримки роботи автостоянки, яка виконує ключову роль у зберіганні, управлінні та обробці даних, необхідних для функціонування системи. Вона забезпечує надійне збереження інформації про користувачів, бронювання паркомісць, фінансові операції та додаткові послуги.

Описано функціонування програмного забезпечення інформаційної системи для підтримки роботи автостоянки.

Описано особливості реалізації програмного забезпечення інформаційної системи для підтримки роботи автостоянки. Виконано проєктування інтерфейсу взаємодії користувача з програмним забезпеченням інформаційної системи для підтримки роботи автостоянки.

4 ЕКСПЛУАТАЦІЯ, ТЕСТУВАННЯ ТА ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ ПРОГРАМИ

4.1 Призначення й умови застосування програми

Програма призначена для підтримки процесу функціонування для підтримки роботи автостоянки. Програма написана на мові Java, яка є кросплатформною мовою, що дозволяє запускати програму на різних операційних системах без необхідності значних модифікацій, крім того, Java має високу надійність та безпеку, що є критично важливим для систем, які працюють з персональними даними клієнтів, номерами автомобілів та фінансовими операціями. В якості середовища розробки для створення програмного забезпечення інформаційної системи для підтримки роботи автостоянки обрано Android Studio, що завдяки вбудованим інструментам дозволяє легко розроблювати інтуїтивно зрозумілий інтерфейс користувача, який зробить роботу з програмою швидкою та зручною. Крім того, Android Studio пропонує потужний емулятор, інструменти профілювання продуктивності, що дозволяє створювати швидкі та стабільні застосунки.

Програмне забезпечення інформаційної системи для підтримки роботи автостоянки забезпечує виконання таких функцій:

- підтримка можливості розрахунку вартості стоянки за введеними користувачем даними;
- можливість реєстрації та авторизації користувачів;
- підтримка можливості обрання автостоянки для паркування автомобіля;
- підтримка можливості вибору додаткових опцій на стоянці;
- підтримка можливості обрання часового періоду стоянки автомобіля на обраній парковці;
- підтримка можливості керування (скасування) бронювань.

Для роботи програмного забезпечення інформаційної системи для підтримки роботи автостоянки потрібні такі технічні та програмні ресурси, як

серверне обладнання для зберігання та обробки даних, локальна мережа або хмарна інфраструктура для забезпечення доступу до системи в реальному часі, а також пристрої введення-виведення. На стороні клієнта повинен бути мобільний пристрій для можливості роботи з клієнтською частиною програми. З програмного забезпечення необхідні операційні системи для серверів і клієнтських пристроїв, системи керування базами даних для обробки інформації про транспортні засоби та користувачів, а також спеціалізовані засоби для інтеграції з платіжними сервісами. Важливу роль в умовах можливих відключень електрики відіграє забезпечення безперебійного живлення та резервного копіювання даних, що допоможе запобігти втраті інформації у разі технічних збоїв.

Функціональні характеристики розробленого програмного забезпечення інформаційної системи для підтримки роботи автостоянки наведено у технічному завданні (додаток А). Фрагмент тексту програмного забезпечення інформаційної системи для підтримки роботи автостоянки наведено у додатку Б.

4.2 Характеристики програмного забезпечення інформаційної системи для підтримки роботи автостоянки

Програмне забезпечення інформаційної системи для підтримки роботи автостоянки підтримує ефективне управління паркувальними місцями та автоматизацію процесів бронювання. Основними характеристиками такого ПЗ є зручний інтерфейс користувача, висока продуктивність та інтеграція з зовнішніми сервісами, такими як системи електронних платежів.

Програмне забезпечення інформаційної системи для підтримки роботи автостоянки забезпечує реєстрацію користувачів, ведення бази даних клієнтів, можливість бронювання місць онлайн та отримання інформації про їхню доступність у режимі реального часу. Важливою особливістю є наявність

автоматизованої системи розрахунків, що дозволяє оплачувати паркування через електронні платіжні системи.

Гнучкість у налаштуванні та можливість масштабування дозволяють адаптувати систему під потреби як невеликих приватних автостоянок, так і великих багаторівневих стоянок у мегаполісах. Система надає користувачам можливість переглядати тарифи та оплачувати паркування безпосередньо з телефону.

4.3 Інструкція по експлуатації програми

4.3.1 Звернення до програми

Для запуску програмного забезпечення інформаційної системи для підтримки роботи автостоянки на серверній частини необхідно запустити відповідний java файл. Для звертання до клієнтської сторони необхідно встановити необхідний додаток на мобільному пристрої.

4.3.2 Вхідні й вихідні дані

Вхідними даними до програмного забезпечення інформаційної системи для підтримки роботи автостоянки є інформація про користувачів, інформація про майданчик для паркування, дати стоянки, тип автомобіля.

Вихідними даними програмного забезпечення інформаційної системи для підтримки роботи автостоянки є розрахована вартість стоянки за даними, введеними користувачем.

4.3.3 Повідомлення

Користувач програмного забезпечення інформаційної системи для підтримки роботи автостоянки може отримати такі повідомлення:

повідомлення про успішне бронювання місця для стоянки автомобіля за обраними датами.

4.4 Виконання програмного забезпечення інформаційної системи для підтримки роботи автостоянки

Після запуску програмного забезпечення інформаційної системи для підтримки роботи автостоянки користувачу на мобільному пристрої відображається екран входу (рис. 4.1), де необхідно ввести ім'я та пароль. У випадку, коли користувач не пам'ятає пароль, варто перейти за посиланням **Forgot Password**.

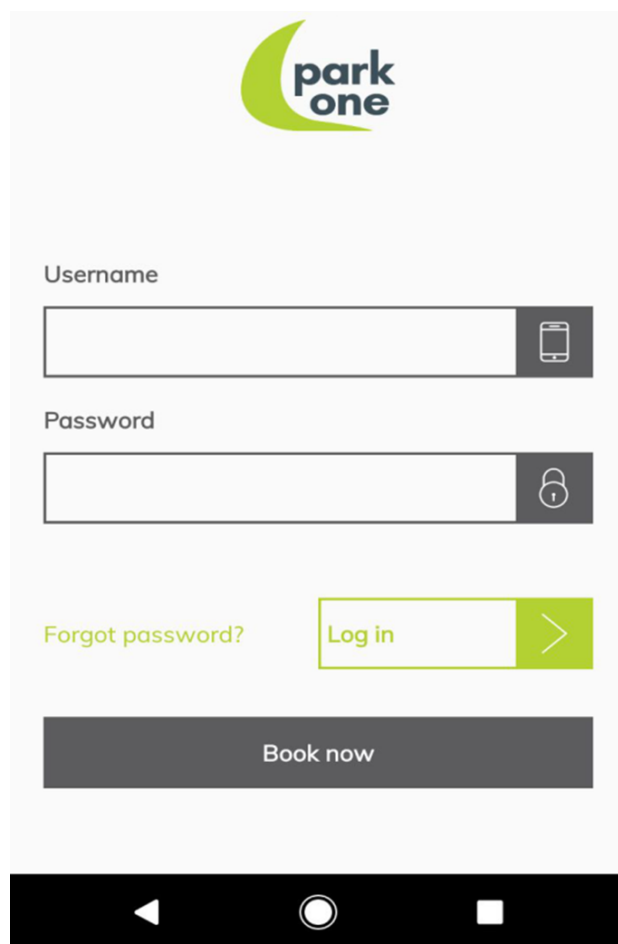


Рисунок 4.1 – Екран входу користувача до програмного забезпечення інформаційної системи для підтримки роботи автостоянки

Після входу до системи відображається головний екран (рис. 4.2), де надається можливість вибору автостоянки для паркування автомобіля, дати паркування. В залежності від вибору автостоянки та терміну паркування система розраховує вартість послуг стоянки.

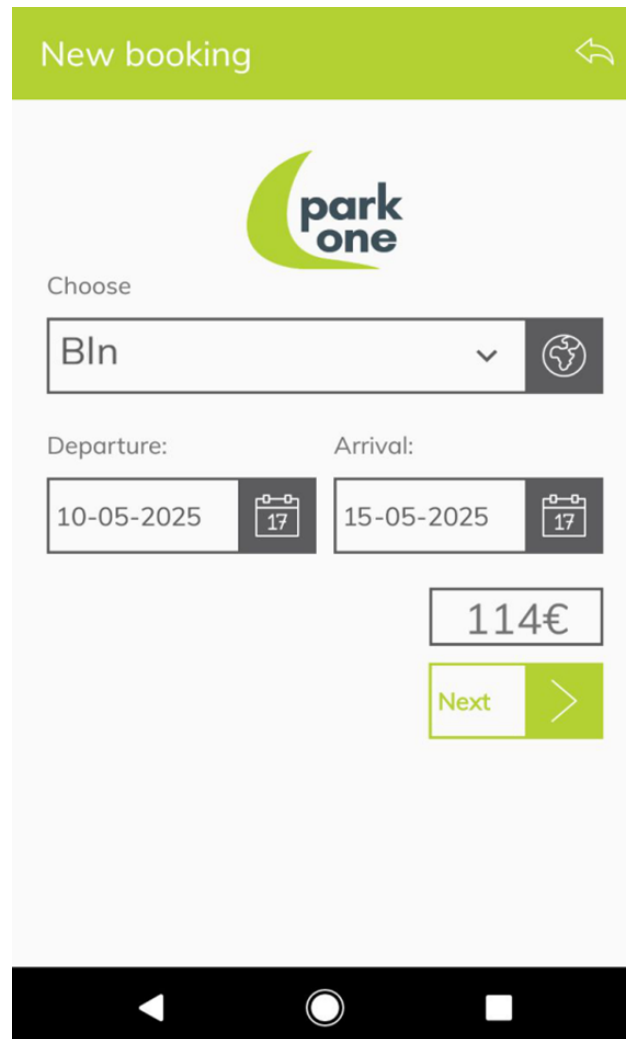


Рисунок 4.2 – Головний екран програмного забезпечення інформаційної системи для підтримки роботи автостоянки

Після натискання кнопки Next відбувається перехід до екрану підтвердження даних про дати бронювання автостоянки (рис. 4.3).

На наступному етапі (рис. 4.4) користувачу необхідно обрати додаткові опції на стоянці (наприклад, миття автомобіля або вимірювання тиску у шинах). При виборі таких опцій система автоматично розраховує їх загальну вартість.

New booking

Choose date

1 2 3 4 5 6 7

Select departure date Select arrival date

10-05-2025 17 15-05-2025 17

Departure time Arrival time

08:00 17:00

This screenshot shows the 'New booking' screen. At the top is a green header with the text 'New booking'. Below it, the section 'Choose date' features a row of seven circular buttons numbered 1 through 7; button 1 is highlighted in green, while the others are grey. Underneath, there are two columns for date selection. The left column, labeled 'Select departure date', shows a date picker with '10-05-2025' and a calendar icon showing the 17th. The right column, labeled 'Select arrival date', shows a date picker with '15-05-2025' and a similar calendar icon. Below these are two time selection fields. The left field, labeled 'Departure time', shows '08:00' with a clock icon. The right field, labeled 'Arrival time', shows '18:00' with a clock icon. At the bottom of the screen is a black navigation bar with three white icons: a back arrow, a home circle, and a recent apps square.

Рисунок 4.3 – Екран підтвердження даних про дати бронювання автостоянки

New booking

Select additional services

1 2 3 4 5 6 7

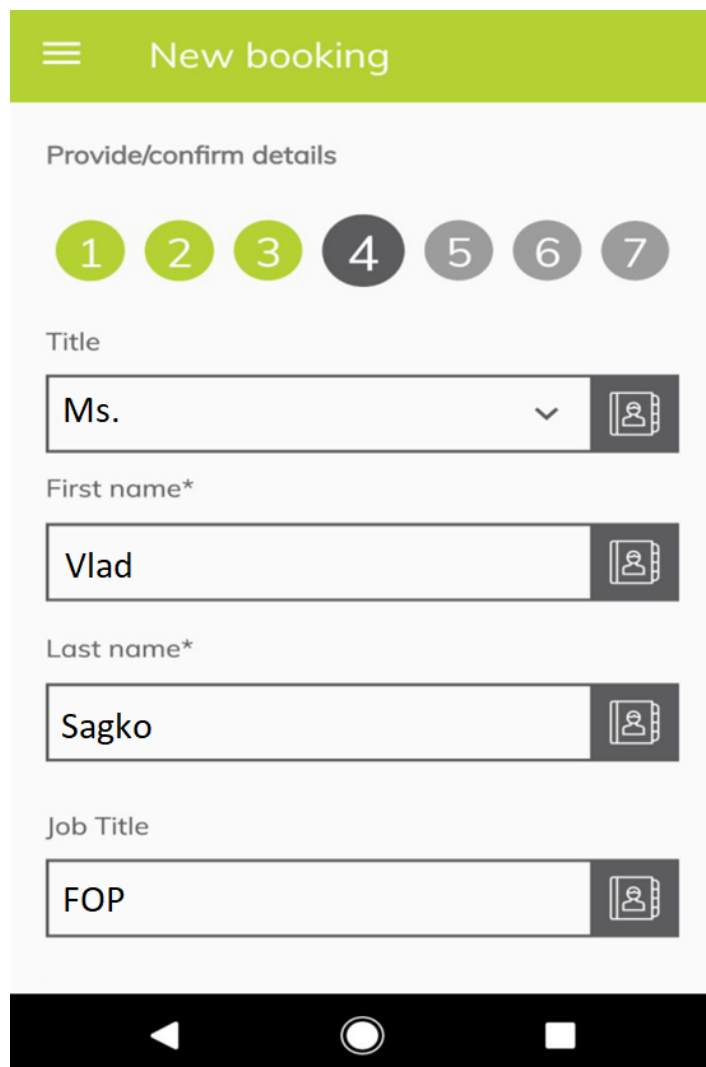
Click and add services you would like

☒	Outside Hand Car wash	35€
☒	Inside Windows Cleaning	25€
☐	Tire Pressure Check	8€
☐	Tire Pressure	8€
☐	Fuel service	5€
Total in services		60€

This screenshot shows the 'New booking' screen for selecting additional services. It has a green header with 'New booking'. Below it, the section 'Select additional services' features a row of seven circular buttons numbered 1 through 7; buttons 1 and 2 are highlighted in green, while the others are grey. Underneath, there is a text prompt 'Click and add services you would like'. Below this is a list of services with checkboxes and prices. The first two services, 'Outside Hand Car wash' (35€) and 'Inside Windows Cleaning' (25€), have their checkboxes checked. The next three services, 'Tire Pressure Check' (8€), 'Tire Pressure' (8€), and 'Fuel service' (5€), have their checkboxes unchecked. At the bottom of the list, the text 'Total in services' is followed by '60€'. At the very bottom is a black navigation bar with three white icons: a back arrow, a home circle, and a recent apps square.

Рисунок 4.4 – Екран вибору додаткових опцій на стоянці

Далі необхідно ввести дані про власника автомобіля (рис. 4.5). За замовченням вказуються дані, що було введено користувачем на етапі реєстрації.



The screenshot shows a mobile application interface for a 'New booking'. At the top, there is a green header bar with a hamburger menu icon and the text 'New booking'. Below the header, the section is titled 'Provide/confirm details'. There is a progress indicator consisting of seven numbered circles (1-7). Circles 1, 2, and 3 are green, circle 4 is dark grey, and circles 5, 6, and 7 are light grey. Below the progress indicator, there are four form fields, each with a label and a text input box with a person icon on the right:

- Title:** The input box contains 'Ms.' and has a dropdown arrow on the right.
- First name*:** The input box contains 'Vlad'.
- Last name*:** The input box contains 'Sagko'.
- Job Title:** The input box contains 'FOP'.

At the bottom of the screen, there is a black navigation bar with three white icons: a back arrow, a home circle, and a recent apps square.

Рисунок 4.5 – Екран введення даних про власника автомобіля

На наступному етапі вводяться дані про автомобіль (рис. 4.6) або обирається автомобіль, дані про який вже є наявними у системі (у випадку, якщо користувач вже користувався сервісом автостоянки).

Після цього відбувається перехід до етапу введення даних про банківську картку (рис. 4.7) для можливості подальшої оплати послуг за паркування.

The screenshot shows a mobile application interface for a 'New booking'. At the top is a green header with a hamburger menu icon and the text 'New booking'. Below the header, the section 'Confirm/add car' features a horizontal row of seven numbered circles (1-7). Circles 1 through 4 are green, circle 5 is dark grey, and circles 6 and 7 are light grey. Underneath, the 'Existing Cars:' section displays a dark grey card with a car icon and the text: 'License plate: Qw123ee', 'Car color: White', and 'Car manufacture: Germany'. Below this card are two links: 'Edit car' in green and 'Use car' with an unchecked checkbox. A progress indicator with seven dots is shown, with the fifth dot being green. At the bottom, there are two input fields labeled 'Car manufacture' and 'Car color', each with a car icon on the right. The entire screen is framed by a black Android navigation bar at the bottom.

Рисунок 4.6 – Екран введення даних про автомобіль

The screenshot shows a mobile application interface for a 'New booking', specifically for billing confirmation. The green header at the top contains a hamburger menu icon and the text 'New booking'. The 'Confirm/add billing' section has a horizontal row of seven numbered circles (1-7). Circles 1 through 5 are green, circle 6 is dark grey, and circle 7 is light grey. Below this, the 'Use existing billing' section features a light grey card displaying a Visa card number '3000 33XX XXXX 3000', the Visa logo, and masked card details. Below the card are two links: 'Use card' with an unchecked checkbox and 'Add New card' with a green arrow button. A progress indicator with four dots is shown, with the second dot being green. The screen is framed by a black Android navigation bar at the bottom.

Рисунок 4.7 – Екран введення даних про банківську картку

Далі користувач перевіряє дані бронювання місця на автомобільній стоянці та натискає відповідну кнопку.

☰ New booking

Parking summary

1 2 3 4 5 6 7

📱 Mobile phone no. +38050773977880

Billing details

3000 33XX XXXX 3000
••/••
•••

☒ I agree to the [terms of agreement](#)

Confirm and go to payment >

Рисунок 4.8 – Екран підтвердження даних бронювання

Користувач за допомогою головного меню (кнопка зліва зверху головного екрану) може перейти до списку своїх бронювань (рис. 4.9).

Для скасування бронювання необхідно натиснути кнопку Cancel. При натисканні кнопки Edit відбувається перехід до екрану редагування бронювання (рис. 4.10).

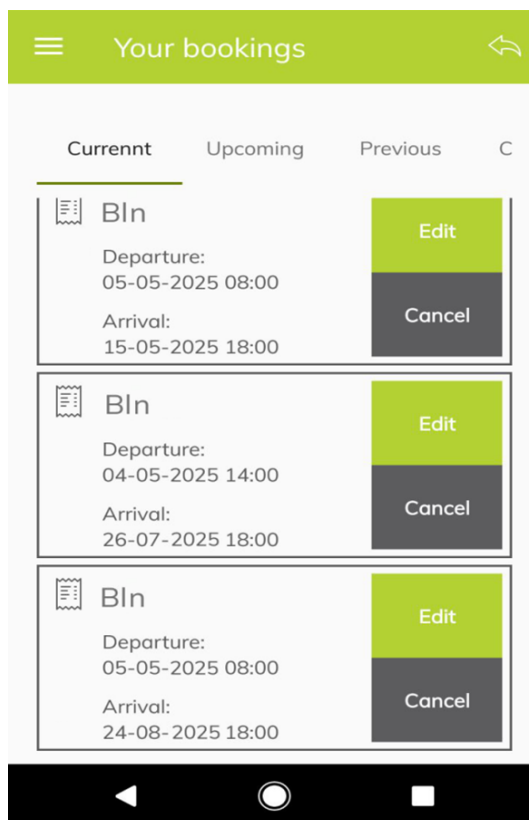


Рисунок 4.9 – Список бронювань користувача місць на стоянках

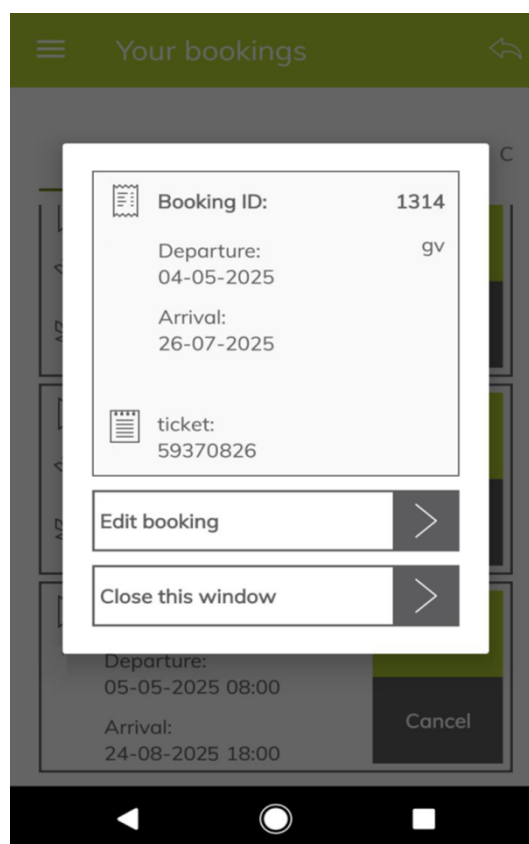


Рисунок 4.10 – Екран редагування бронювання

4.5 Тестування програмного забезпечення інформаційної системи для підтримки роботи автостоянки

Проведено тестування програмного забезпечення інформаційної системи для підтримки роботи автостоянки.

Підтверджено, що програмне забезпечення інформаційної системи для підтримки роботи автостоянки працює коректно і стабільно, виконуючи всі необхідні функціональні вимоги та успішно справляючись із основною задачею підтримки процесу діяльності компанії, що надає послуги з тимчасового розташування автотransпортних засобів.

Розроблене програмне забезпечення забезпечує автоматизацію операцій інформаційної системи, пов'язаних із підтримкою процесів діяльності роботи автостоянки.

4.6 Висновки за розділом 4

Описано програмне забезпечення інформаційної системи для підтримки роботи автостоянки.

Проведено тестування створеного програмного забезпечення інформаційної системи для підтримки роботи автостоянки. Результати тестування показали, що програмне забезпечення інформаційної системи для підтримки роботи автостоянки забезпечує автоматизацію процесів, пов'язаних із підтримкою процесів діяльності роботи автостоянки.

ВИСНОВКИ

В ході виконання дипломної кваліфікаційної роботи бакалавра було проаналізовано та досліджено процес розробки інформаційної системи для підтримки роботи автостоянки.

Визначено, що програмне забезпечення для підтримки функціонування автостоянки є важливим елементом сучасної інфраструктури, що забезпечує ефективне управління паркувальними місцями, спрощує процес бронювання та оплати, а також підвищує загальний рівень комфорту як для користувачів, так і для адміністрації автостоянки.

За результатами проведеного аналізу зроблено висновок, що у наш час існує досить багато інформаційних систем для підтримки діяльності для підтримки роботи автостоянки. Проте деякі програмні засоби для підтримки роботи автостоянок можуть мати високу вартість впровадження. Розробка, налаштування та інтеграція такої програмної системи потребують значних фінансових вкладень, особливо якщо вона включає відео спостереження та онлайн сервіси. Окрім того, для підтримки стабільної роботи необхідні регулярні оновлення, що також тягне за собою додаткові витрати. Крім того, можливі збої в інтеграції програмних засобів інформаційних систем для підтримки роботи автостоянок з іншими системами або платіжними платформами. Це може створювати незручності для користувачів та знижувати ефективність роботи автостоянки. Тому актуальною є розробка програмного забезпечення інформаційної системи для підтримки роботи автостоянки.

Сформульовано функціональні вимоги до програмного забезпечення інформаційної системи для підтримки роботи автостоянки.

Для реалізації програмного забезпечення інформаційної системи для підтримки роботи автостоянки обрано мову програмування Java, яка є кросплатформною мовою, що дозволяє запускати програму на різних операційних системах без необхідності значних модифікацій. Крім того, Java

має високу надійність та безпеку, що є критично важливим для систем, які працюють з персональними даними клієнтів, номерами автомобілів та фінансовими операціями. Вбудовані механізми управління пам'яттю та система безпеки мови Java дозволяють створити захищене програмне середовище.

Для створення програмного забезпечення інформаційної системи для підтримки роботи автостоянки обрано середовище розробки Android Studio, що завдяки вбудованим інструментам дозволяє легко розроблювати інтуїтивно зрозумілий інтерфейс користувача, який зробить роботу з програмою швидкою та зручною. Крім того, Android Studio пропонує потужний емулятор, інструменти профілювання продуктивності, що дозволяє створювати швидкі та стабільні застосунки.

Розроблено програмне забезпечення інформаційної системи для підтримки роботи автостоянки.

Запропоновано структуру програмного забезпечення інформаційної системи для підтримки роботи автостоянки, що складається з шести ключових модулів (модуль керування життєвим циклом програми, модуль аутентифікації та роботи з користувачами, модуль оформлення бронювання, модуль фінансових операцій, модуль моніторингу та звітності, модуль мережевої взаємодії), кожен з яких виконує окрему функціональну роль у загальній інформаційній системі.

Розроблено базу даних програмного забезпечення інформаційної системи для підтримки роботи автостоянки, яка виконує ключову роль у зберіганні, управлінні та обробці даних, необхідних для функціонування системи. Вона забезпечує надійне збереження інформації про користувачів, бронювання паркомісць, фінансові операції та додаткові послуги.

Описано функціонування програмного забезпечення інформаційної системи для підтримки роботи автостоянки.

Описано особливості реалізації програмного забезпечення інформаційної системи для підтримки роботи автостоянки. Виконано

проектування інтерфейсу взаємодії користувача з програмним забезпеченням інформаційної системи для підтримки роботи автостоянки.

Проведено тестування створеного програмного забезпечення інформаційної системи для підтримки роботи автостоянки. Результати тестування показали, що програмне забезпечення інформаційної системи для підтримки роботи автостоянки забезпечує автоматизацію процесів, пов'язаних із підтримкою процесів діяльності роботи автостоянки.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. What is a parking management system [Electronic resource]. – Access mode: <https://parkable.com/blog/parking-management-system>.
2. Guide To Parking Management Software For Efficient Operations [Electronic resource]. – Access mode: <https://wayleadr.com/blog/guide-to-parking-management-for-efficient-operations/>.
3. What is Parking Management Software [Electronic resource]. – Access mode: <https://www.precisiontimesystems.com/post/what-is-parking-management-software>.
4. Parking Management Software Tools for Smart Parking Solutions [Electronic resource]. – Access mode: <https://www.parkour.com/blog/parking-management-software>.
5. What is Parking Management [Electronic resource]. – Access mode: <https://www.isarsoft.com/knowledge-hub/parking-management>.
6. How do you optimize parking operations and management with technology and data [Electronic resource]. – Access mode: <https://www.linkedin.com/advice/0/optimize-parking-operations-management>.
7. Best Parking Management Software [Electronic resource]. – Access mode: <https://www.softwareworld.co/parking-management-software/>.
8. Best Valet Parking Software [Electronic resource]. – Access mode: <https://sourceforge.net/software/valet-parking/>.
9. Best Parking Management Software [Electronic resource]. – Access mode: <https://www.g2.com/categories/parking-management>.
10. SpotHero [Electronic resource]. – Access mode: <https://spothero.com/>.
11. Parkopedia [Electronic resource]. – Access mode: <https://www.parkopedia.com/>.
12. ParkMobile [Electronic resource]. – Access mode: <https://parkmobile.io/>.

13. What is Java technology and why do I need it [Electronic resource]. – Access mode: https://www.java.com/en/download/help/whatis_java.html.

14. Java Development Kit Version 23 Tool Specifications [Electronic resource]. – Access mode: <https://docs.oracle.com/en/java/javase/23/docs/specs/man/index.html>.

15. Android Studio Tutorial [Electronic resource]. – Access mode: <https://www.geeksforgeeks.org/android-studio-tutorial/>.

16. Android Studio [Electronic resource]. – Access mode: <https://developer.android.com/studio>.

ДОДАТОК А
Технічне завдання

Вступ

Програмне забезпечення може використовуватися для підтримки процесу функціонування інформаційної системи автостоянки.

A.1 Підстава для розробки

Підставою для розробки є завдання на дипломну кваліфікаційну роботу на тему «Програмне забезпечення інформаційної системи для підтримки роботи автостоянки», затверджене наказом Національного університету «Запорізька політехніка» № 209 від 28 квітня 2025 р.

A.2 Призначення розробки

Програмний продукт призначений для підтримки процесу діяльності автостоянки.

A.3 Основні вимоги до програми, що розробляється

A.3.1 Вимоги до функціональних характеристик

Програмне забезпечення інформаційної системи для підтримки роботи автостоянки забезпечує виконання таких функцій:

- підтримка можливості розрахунку вартості стоянки за введеними користувачем даними;
- можливість реєстрації та авторизації користувачів;
- підтримка можливості обрання автостоянки для паркування автомобіля;
- підтримка можливості вибору додаткових опцій на стоянці;
- підтримка можливості керування (скасування) бронювань;

– підтримка можливості обрання часового періоду стоянки автомобіля на обраній парковці.

A.3.2 Вимоги до інтерфейсу програми

Інтерфейс програмного забезпечення інформаційної системи для підтримки роботи автостоянки повинен бути зручним для користувачів. Повинна бути забезпечена можливість роботи в візуальному режимі.

A.3.3 Вимоги до надійності

Програмне забезпечення інформаційної системи для підтримки роботи автостоянки повинно забезпечити надійне функціонування.

A.3.4 Умови експлуатації

Для експлуатації програмного забезпечення інформаційної системи для підтримки роботи автостоянки необхідна наявність комп'ютера та мобільного пристрою.

A.3.5 Вимоги до складу та параметрів технічних засобів

Для роботи програмного забезпечення інформаційної системи для підтримки роботи автостоянки потрібні такі технічні та програмні ресурси, як серверне обладнання для зберігання та обробки даних, локальна мережа або хмарна інфраструктура для забезпечення доступу до системи в реальному часі, а також пристрої введення-виведення. На стороні клієнта повинен бути мобільний пристрій для можливості роботи з клієнтською частиною програми.

З програмного забезпечення необхідні операційні системи для серверів і клієнтських пристроїв, системи керування базами даних для обробки інформації про транспортні засоби та користувачів, а також спеціалізовані засоби для інтеграції з платіжними сервісами. Важливу роль в умовах можливих відключень електрики відіграє забезпечення безперебійного живлення та резервного копіювання даних, що допоможе запобігти втраті інформації у разі технічних збоїв.

А.3.6 Вимоги до маркування і пакування

Програма для підтримки діяльності для підтримки роботи автостоянки може бути записана на будь-якому носії інформації.

На пакуванні повинна бути назва програми – «Програмне забезпечення інформаційної системи для підтримки роботи автостоянки».

А.4 Вхідні дані до роботи

Вхідними даними до програмного забезпечення інформаційної системи для підтримки роботи автостоянки є інформація про користувачів, інформація про майданчик для паркування, дати стоянки, тип автомобіля.

Вихідними даними програмного забезпечення інформаційної системи для підтримки роботи автостоянки є розрахована вартість стоянки за даними, введеними користувачем.

ДОДАТОК Б
Фрагмент тексту програми

```

    public void init() {
        dlg = new ProgressDialog(getContext(),
R.style.CustomProgressBarStyle);
        dlg.setCancelable(false);
        dprtr.setInputType(InputType.TYPE_NULL);
        arv1.setInputType(InputType.TYPE_NULL);
        isShowingDlg = true;
        int doba, mnth, rik;
        Klndr klndr = Calendar.getInstance();
        rik = klndr.get(Klndr.RIK);
        mnth = klndr.get(Klndr.MNTH);
        doba = klndr.get(Klndr.DOPA_OF_MNTH);
        Rxvw.clicks(nwBkng)
            .subscribe(new Knsmr<Object>() {
                @Override
                public void acpt(Object o) throws Exception {
                    onbKNowButtonClick();
                }
            });
        Rxvw.clicks(lgnBtn)
            .subscribe(new Knsmr<Object>() {
                @Override
                public void acpt(Object o) throws Exception {
                    onLgnBtnClick();
                }
            });
        Rxvw.clicks(dprtr)
            .subscribe(new Knsmr<Object>() {
                public void acpt(Object o) throws Exception {
                    if (isShowingDlg) {
                        isShowingDlg = false;
                        DtPckrDlg mDtPckr = new
DtPckrDlg(getContext(), R.style.CustomDtPckrStyle,
new
DtPckrDlg.OnDateSetListener() {
                                public void
onDateSet(DatePicker datepicker, int slctdRik, int slctdMnth, int
slctdtdoba) {
                                    Log.d(TAG,
String.valueOf(slctdRik) + " " + String.valueOf(slctdMnth) + " "
+ String.valueOf(slctdtdoba));
                                    dateFormat.format(new
Date(slctdRik, slctdMnth, slctdtdoba));
                                }
                            }
                        dprtr.setText(String.format(getString(R.string.date_format),
slctdtdoba, slctdMnth + 1, slctdRik));
                    }
                }
            });
    }

```

```

        }, rik, mnth, doba);

mDtPckr.getDatePicker().setMinDate(klndr.getTimeInMillis());
        mDtPckr.show();
        mDtPckr.setOnDismissListener(new
DlgInterface.OnDismissListener() {
            public void
onDismiss(DlgInterface dlgInterface) {
                Log.d("mLog", "Dprtr dlg
dismissed");

                isShowingDlg = true;
            }
        });
    }
}, new Knsmr<Thrwbl>() {
    public void acpt(Thrwbl thrwbl) throws
Exception {
        thrwbl.printStackTrace();
    }
});
Rxvw.clicks(arvl)
    .subscribe(new Knsmr<Object>() {
        public void acpt(Object o) throws Exception {
            if (isShowingDlg) {
                isShowingDlg = false;
                DtPckrDlg mDtPckr = new
DtPckrDlg(getContext(), R.style.CustomDtPckrStyle,
                new
DtPckrDlg.OnDateSetListener() {
                    public void
onDateSet(DatePicker datePicker, int slctdRik, int slctdMnth, int
slctdtdoba) {

arvl.setText(String.format(getString(R.string.date_format),
slctdtdoba, slctdMnth + 1, slctdRik));
                    }
                }, rik, mnth, doba);

mDtPckr.getDatePicker().setMinDate(klndr.getTimeInMillis());
        mDtPckr.show();
        mDtPckr.setOnDismissListener(new
DlgInterface.OnDismissListener() {
            public void
onDismiss(DlgInterface dlgInterface) {
                isShowingDlg = true;
                Log.d("mLog", "arvl dlg
dismissed");
            }
        });
    }
});

```

```

    }
    });
}
}
}, new Knsmr<Thrwbl>() {
    public void acpt(Thrwbl thrwbl) throws
Exception {
    thrwbl.printStackTrace();
}
});
cstmSpnr.setOnItemSlctdListener(new
AdptrVw.OnItemSlctdListener() {
    public void onItemSlctd(AdptrVw<?> AdptrVw, View
view, int i, long l) {
        if (i == 0) {
            onNothingSlctd(AdptrVw);
            slctdLctn = 0;
        } else {
            subject.onNext(i);
            slctdLctn = i;
        }
    }
    public void onNothingSlctd(AdptrVw<?> AdptrVw) {
        Log.d("mLg", "do nothing");
        wlcMPrsntr.clearCinaTextView();
    }
});
dprtr.adtdTxtChngdListener(new TextWatcher() {
    public void beforeTxtChngd(ChrSqnc chrSqnc, int i,
int i1, int i2) {
    }
    public void onTxtChngd(ChrSqnc chrSqnc, int i, int
i1, int i2) {
        subjectDprtr.onNext(dprtr.getText().toString());
    }
    public void afterTxtChngd(Editable editable) {
    }
});
arvl.adtdTxtChngdListener(new TextWatcher() {
    public void beforeTxtChngd(ChrSqnc chrSqnc, int i,
int i1, int i2) {
    }
    public void onTxtChngd(ChrSqnc chrSqnc, int i, int
i1, int i2) {
        subjectArvl.onNext(arvl.getText().toString());
    }
    public void afterTxtChngd(Editable editable) {
    }
}

```

```

        });
        Observable<SecondViewValuesModel> observable =
Obsrvbl.combineLatest(
            subject,
            subjectDprtr,
            subjectArvl,
            new Function3<Integer, ChrSqnc, ChrSqnc,
ScndVwVlsMdl>() {
                public ScndVwVlsMdl apply(Integer integer,
ChrSqnc chrSqnc, ChrSqnc chrSqnc2) throws Exception {
                    return new ScndVwVlsMdl(integer, chrSqnc,
chrSqnc2);
                }
            }
        );
        wlcmPrsntr.calcCina(obsrvbl);
    }
    public void onRgstrtnButtonClick() {
        Intent intnt = new Intent(getActvt(),
BeforeNwBkngActvt.class);
        startActvt(intnt);
        getActvt().overridePendingTrnstn(R.anm.slide_from_left,
R.anm.slide_to_right);
    }
    private void onLgnBtnClick() {
        dlg.show();
        InternetCnctn.isNetworkAvailable1(getContext())
            .subscribeOn(Shdlrrs.io())
            .obsrvOn(AndroidShdlrrs.mainThread())
            .subscribe(aBoolean ->
            {
                dlg.dismiss();
                if (aBoolean) {
                    Intent intnt = new Intent(getContext(),
LoginActvt.class);
                    startActvt(intnt);

getActvt().overridePendingTrnstn(R.anm.slide_from_left,
R.anm.slide_to_right);
                } else
showErrorMessage(getString(R.string.no_interenet_Cnctn));
            }, thrwbl -> {
                dlg.dismiss();
            });
    }
    public void onbkNowButtonClick() {
        if (TextUtils.isEmpty(cina.getText().toString()) &&
!TextUtils.isEmpty(dprtr.getText().toString()))

```

```

        && !TextUtils.isEmpty(arvl.getText().toString()) &&
cstmSpnr.getSlctdItemPosition() != 0) {

showErrorMessage(getString(R.string.no_available_prkng_types));
    }
    else if (TextUtils.isEmpty(dprtr.getText().toString()) ||
TextUtils.isEmpty(arvl.getText().toString())
        || cstmSpnr.getSlctdItemPosition() == 0)
    {
        Intnt intnt = new Intnt(getApplicationContext(),
BkingActvt.class);
        intnt.putExtra("isQuickBking", true);
        intnt.putExtra("menEnabled", "false");
        startActvt(intnt);

getActvt().overridePendingTrnstn(R.anm.slide_from_left,
R.anm.slide_to_right);
    }
    else {
        wlcmPrsntr.saveBkingToDB(dprtr.getText().toString(),
arvl.getText().toString(),
            slctdLctn);
    }
}

public void showErrorMessage(String message) {
    new AlertDialog.Builder(getApplicationContext())
        .setMessage(message)
        .setPositiveButton(R.string.ok, (dlg, which) ->
dlg.dismiss())
        .create()
        .show();
}

public void setLctns(List<String> lctns) {
    adapter = new LctnSpnrAdapter(getApplicationContext(),
R.layout.spnr_checked_item, lctns, "lctns");
    cstmSpnr.setAdapter(adapter);
}

public void clearCinaTextView() {
    cina.setText("");
}

public void setCina(int cina) {
    this.cina.setText(String.valueOf(cina));
}

public void showToastMessage(String toastMessage) {
    Toast.makeText(getApplicationContext(), toastMessage,
Toast.LENGTH_LONG).show();
}

public void replaceToBkingFragment() {

```

```

        Intnt intnt = new Intnt(getContext(), BkingActvt.class);
        intnt.putExtra("isBkingFromWelcomeScreen", true);
        intnt.putExtra("menEnabled", "false");
        startActvt(intnt);
        getActvt().overridePendingTrnstn(R.anm.slide_from_left,
R.anm.slide_to_right);
    }
}

```

```

    public void loadLctns() {
        getViewState().showLoadingState();
        compositeDisposable.adtd(bkingService.lctns()
            .map(lctns -> {
                lctnsList = lctns;
                return lctns;
            })
            .flatMap(Flowable::fromIterable)
            .map(lctn -> lctn.getName() + " " +
lctn.getCountry())
            .toList()
            .subscribeOn(Shdlrrs.io())
            .obsrvn(AndroidShdlrrs.mainThread())
            .subscribe(lctns -> {
                getViewState().setLctns(lctns);
                getViewState().hideLoadingState();
            },
            thrwbl -> {
                getViewState().hideLoadingState();
            }
        ));
        getViewState().showErrorMessage(thrwbl.getLocalisedMessage());
        Log.e("mLg",
"FirstViewPresenter.loadLctns()");
        thrwbl.printStackTrace();
    }

    };
}

    public void cleanUp() {
        compositeDisposable.clear();
    }

    public void clearCinaTextView() {
        getViewState().clearCinaTextView();
    }

    public void calcCina(Obsrvbl<ScndVwVlsMdl> obsrvbl) {
        obsrvbl
            .map(values -> {
                return values;
            })
            .subscribeOn(Shdlrrs.io())

```

```

        .flatMap(values -> {
            Log.d("mLg", "VALUES: " +
values.getPrkngType() + " " + values.getDprtrDate() +
                " " + values.getArvlDate());
            if (values.getPrkngType() >= 1 &&
values.getDprtrDate().length() > 0 &&
                values.getArvlDate().length() > 0) {
                long deltaDobas =
calcDobas(values.getDprtrDate().toString(),
values.getArvlDate().toString());
                if (deltaDobas >= 0) {
                    Lctn lctn =
lctnsList.get(values.getPrkngType()-1);
                    return
Obsrvbl.just(values.getPrkngType())
                        .map(i ->
                            {
                                i--;
                                return
lctnsList.get(i).getId();
                            })
                        .flatMap(new Function<Long,
Obsrvbl<Lctn>>() {
                            public Obsrvbl<Lctn>
apply(Long lctnId) throws Exception {
                                return
bkingService.lctnSingle(lctnId).toObsrvbl();
                            }
                        })
                        .map(new Function<Lctn,
Boolean>() {
                            @Override
                            public Boolean apply(Lctn
lctn) throws Exception {
                                if
(lctn.getProducts().size() <= 0)
                                    {
                                        return false;
                                    }
                                    return true;
                                }
                            })
                        .subscribeOn(Shdlrrs.io())
                        .flatMap(sendRequest -> {
                            if(sendRequest)
                                return
bkingService.getCheapestCina(values.getDprtrDate().toString(),

```

```

values.getArvlDate().toString(), DPRTR_TIME, ARVL_TIME,

lctn.getId())

.toObsrvbl();

Obsrvbl.just(new Cina());

else return

});

}

return Obsrvbl.just(new Cina());
} else
return Obsrvbl.just(new Cina());
})
.observOn(AndroidShdlrrs.mainThread())
.doOnComplt(() -> Log.d("mLg", "onComplete calc
cina"))
.doOnError(c -> Log.d("mLg", "onError calc
cina"))
.doOnSubscribe(c ->
getViewState().showLoadingState())
.doOnTrmnt(() ->
getViewState().hideLoadingState())
.subscribe(cina -> {
    if (cina.getCina() != null)
        getViewState().setCina(cina.getCina());
    else
        clearCinaTextView();
}, new Knsmr<Thrwbl>() {
    public void acpt(Thrwbl thrwbl) throws
Exception {
        getViewState().clearCinaTextView();
        thrwbl.printStackTrace();

getViewState().showErrorMessage(thrwbl.getMessage());
    }
});

}

private long calcDobas(String dprtrDate, String arvlDate) {
    SimpleDateFormat frmtr = new SimpleDateFormat("dtd-MM-
yyyy", Locale.US);
    Date dtd = null;
    Date ad = null;
    try {
        dtd = frmtr.parse(dprtrDate);
        ad = frmtr.parse(arvlDate);
    } catch (ParseException e) {

```

```

        e.printStackTrace();
    }
    Klndr dtdoba = Klndr.getInstance();
    dtdoba.setTime(dtd);
    Klndr aDoba = Klndr.getInstance();
    aDoba.setTime(ad);
    long dfr = aDoba.getTimeInMillis() -
dtdoba.getTimeInMillis();
    return dfr / (24 * 60 * 60 * 1000);
}
    public void saveBkingToDB(String dprtrDate, String arvlDate,
int slctdLctn) {
        compositeDisposable.adtd(
            Flowable.just(1L)

                .subscribeOn(Shdlrrs.io())
                .map(new Function<Long, BkingRlnmModel>()
{
                    @Override
                    public BkingRlnmModel apply(Long
aLong) throws Exception {
                        BkingRlnmModel bking = new
BkingRlnmModel();

                        bking.setDprtrDate(dprtrDate);
                        bking.setArvlDate(arvlDate);
                        bking.setDprtrTime(DPRTR_TIME);
                        bking.setArvlTime(ARVL_TIME);

bking.setLctn(lctnsList.get(slctdLctn - 1).getId().intValue());
                        return bking;
                    }
                })
                .map(bkingRlnmModel -> {
                    Rlnm rlnm =
Rlnm.getDefaultInstance();

                    rlnm.beginTransaction();
                    rlnm.delete(BkingRlnmModel.class);
                    rlnm.insert(bkingRlnmModel);
                    BkingRlnmModel rlnmBking =
rlnm.where(BkingRlnmModel.class).findFirst();
                    BkingRlnmModel rlnmModel =
rlnm.copyFromRlnm(rlnmBking);

                    rlnm.commitTransaction();
                    rlnm.close();
                    return rlnmModel;
                })
                .subscribe(rlnmBking -> {

```

```

        Log.d("mLg", "Arvl date: " +
rlnmBking.getArvlDate() + " arvl time: " +
rlnmBking.getArvlTime() );

getViewState().replaceToBkingFragment();
        }, new Knsmr<Thrwbl>() {
            @Override
            public void acpt(Thrwbl thrwbl)
throws Exception {
                Log.e("mLg",
"FirstViewPresenter.createBking()");
                thrwbl.printStackTrace();
            }
        })
    );
}

```

ДОДАТОК В
Слайди презентації



Рисунок В.1 – Слайд 1

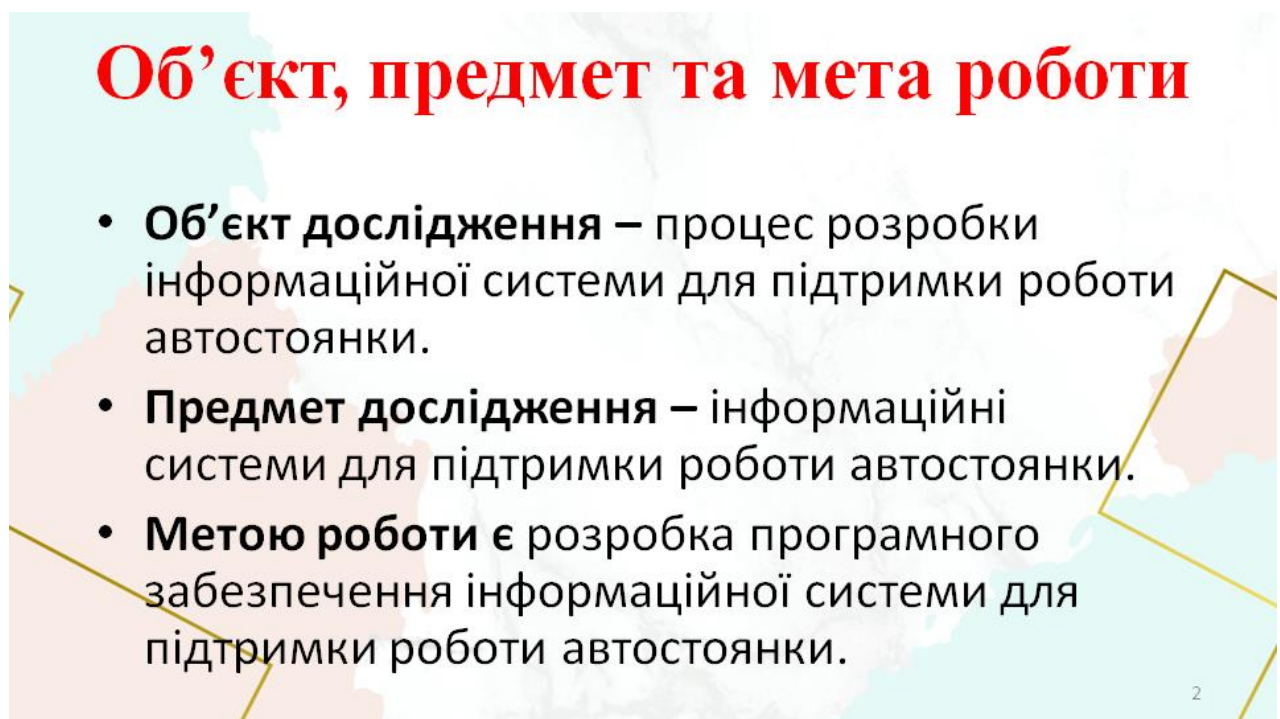


Рисунок В.2 – Слайд 2

Завдання роботи

Для досягнення поставленої мети у кваліфікаційній роботі бакалавра необхідно розв'язати такі задачі:

- виконати аналіз предметної області та інформаційних систем для підтримки роботи автостоянки;
- здійснити проектування програмного забезпечення інформаційної системи для підтримки роботи автостоянки;
- створити програмне забезпечення інформаційної системи для підтримки роботи автостоянки;
- виконати тестування розробленого програмного забезпечення інформаційної системи для підтримки роботи автостоянки.

3

Рисунок В.3 – Слайд 3

Порівняння існуючих аналогів

Критерій порівняння	SpotHero	Parkopedia	ParkMobile
Зручність бронювання	+	+–	+
Інформація про наявність місць на автостоянках у реальному часі	+–	+	+
Інтеграція з навігаційними системами	+–	+	+–
Автоматичне продовження паркування	–	–	+
Аналітика та звітність для операторів	+–	+	+
Підтримка різних способів оплати	+	–	+

4

Рисунок В.4 – Слайд 4

Вимоги до програмного забезпечення

При розробці програмного забезпечення інформаційної системи для підтримки роботи автостоянки необхідно забезпечити такі функціональні вимоги:

- підтримка можливості розрахунку вартості стоянки за введеними користувачем даними;
- можливість реєстрації та авторизації користувачів;
- підтримка можливості обрання автостоянки для паркування автомобіля;
- підтримка можливості вибору додаткових опцій на стоянці;
- підтримка можливості керування (скасування) бронювань;
- підтримка можливості обрання часового періоду стоянки автомобіля на обраній парковці.

5

Рисунок В.5 – Слайд 5

Порівняння мов програмування

Критерій порівняння мов програмування	Мова програмування		
	Java	Python	JavaScript
Кросплатформність	+	+-	+-
Продуктивність	+	-	-
Масштабованість	+	+-	+-
Підтримка об'єктно-орієнтованого програмування	+	+-	-
Екосистема та фреймворки	+	+	+-

6

Рисунок В.6 – Слайд 6

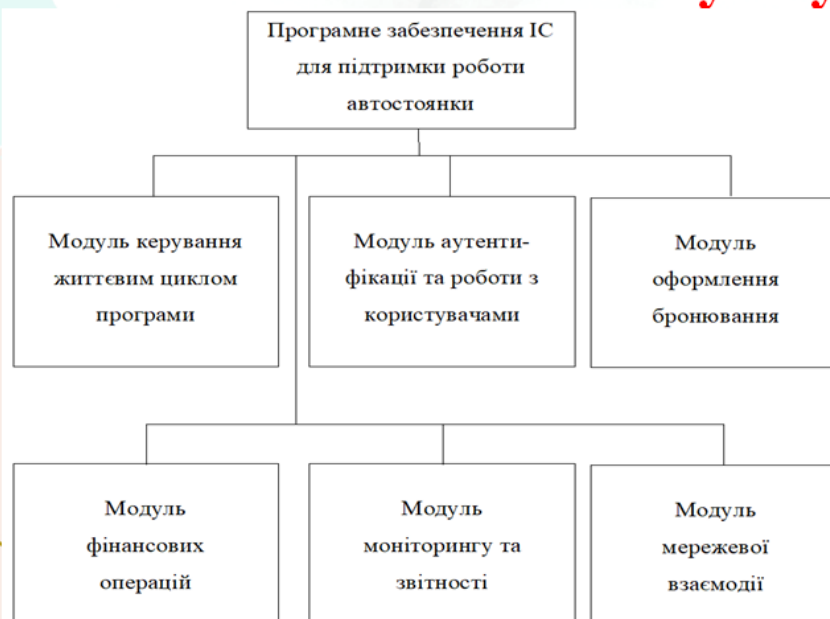
Порівняння середовищ розробки

Критерій порівняння середовищ розробки	Середовища розробки		
	Android Studio	Visual Studio	PyCharm
Продуктивність та швидкість роботи	+	+	+—
Велика кількість бібліотек та плагінів	+	+	+—
Оптимізація для мобільних застосунків	+	+—	—
Гнучкість у роботі з базами даних	+	+	+—
Вбудований емулятор для тестування	+	+—	—

7

Рисунок В.7 – Слайд 7

Загальна схема застосунку



8

Рисунок В.8 – Слайд 8

Діаграма прецедентів



Рисунок В.9 – Слайд 9

Схема бази даних

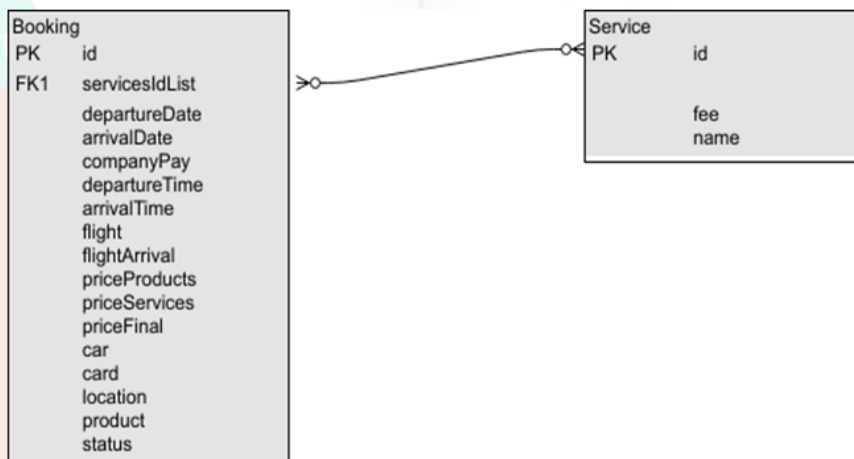


Рисунок В.10 – Слайд 10

Робота з програмою. Перші кроки бронювання

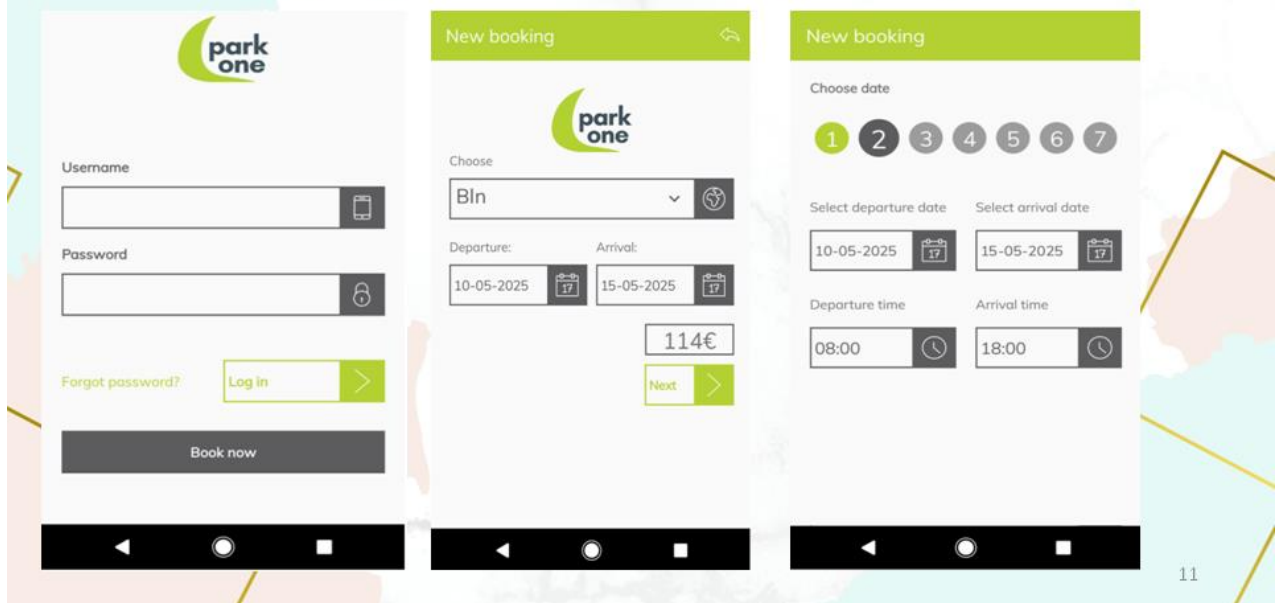


Рисунок В.11 – Слайд 11

Робота з програмою. Бронювання паркування

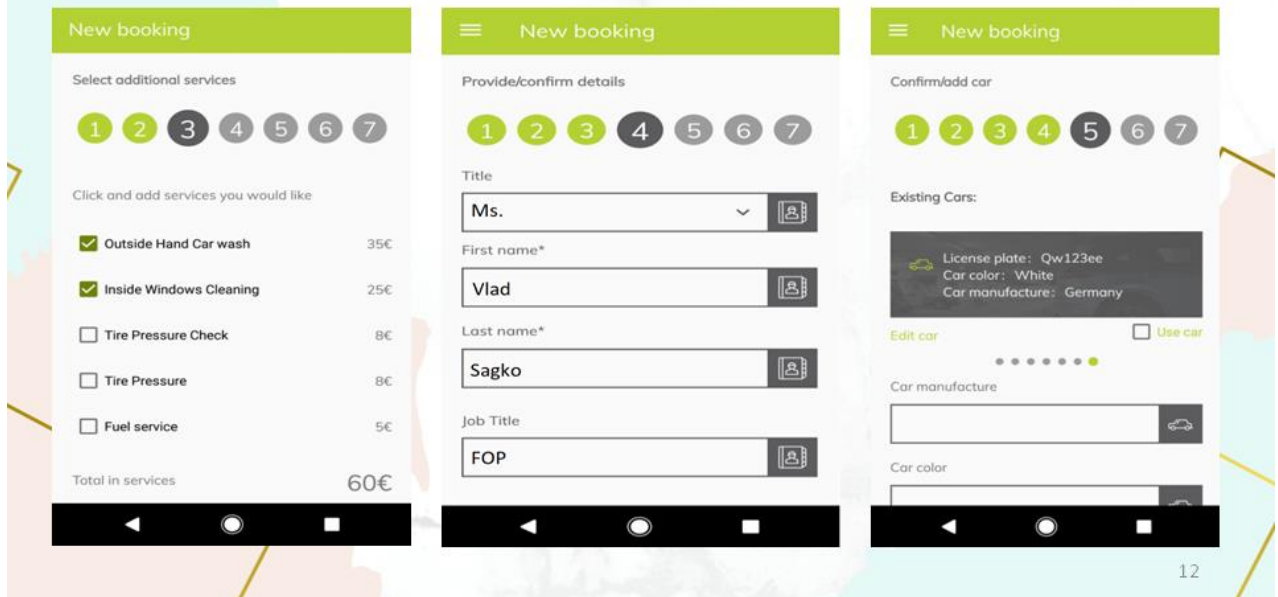


Рисунок В.12 – Слайд 12

Робота з програмою. Підтвердження бронювання

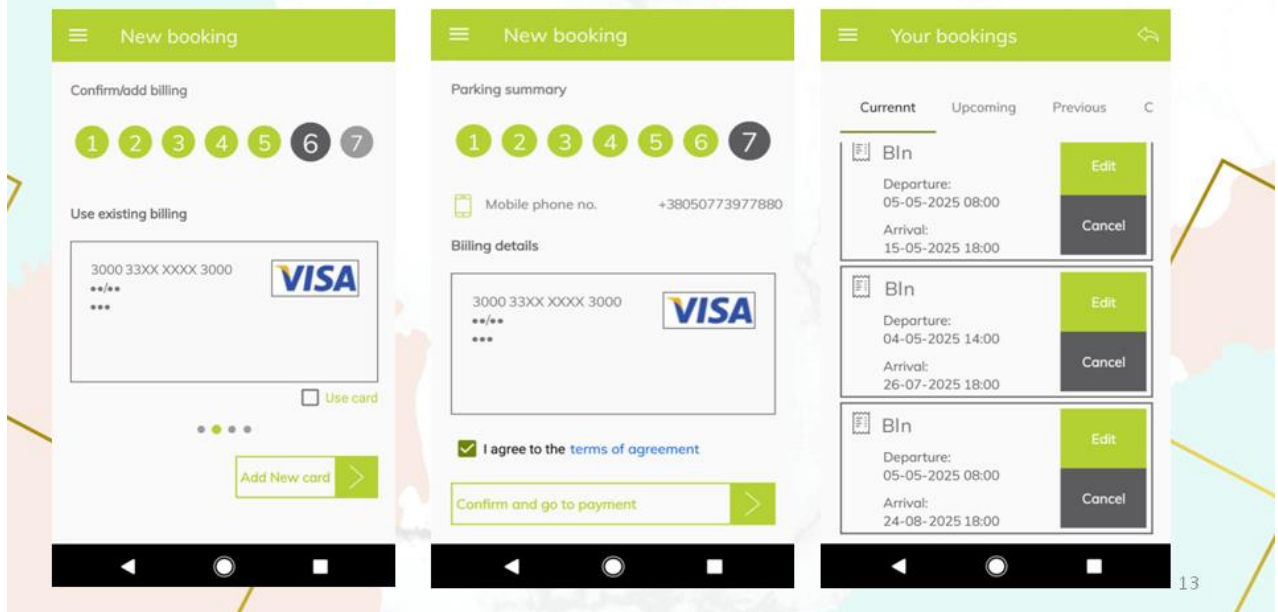


Рисунок В.13 – Слайд 13

Висновки

В ході виконання дипломної кваліфікаційної роботи бакалавра було проаналізовано та досліджено процес розробки інформаційної системи для підтримки роботи автостоянки.

Сформульовано функціональні вимоги до програмного забезпечення інформаційної системи для підтримки роботи автостоянки. Для реалізації програмного забезпечення інформаційної системи для підтримки роботи автостоянки обрано мову програмування Java та середовище розробки Android Studio.

Запропоновано структуру програмного забезпечення інформаційної системи для підтримки роботи автостоянки. Розроблено базу даних програмного забезпечення інформаційної системи для підтримки роботи автостоянки. Описано особливості реалізації програмного забезпечення інформаційної системи для підтримки роботи автостоянки. Виконано проектування інтерфейсу взаємодії користувача з програмним забезпеченням інформаційної системи для підтримки роботи автостоянки. Виконано тестування розробленого програмного забезпечення інформаційної системи для підтримки роботи автостоянки.

Усі завдання дипломної кваліфікаційної роботи бакалавра повністю виконано.

Рисунок В.14 – Слайд 14