

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Інститут інформатики та радіoeлектроніки,
Факультет комп'ютерних наук і технологій

(повне найменування інституту, назва факультету)

Кафедра програмних засобів

(повне найменування кафедри)

Пояснювальна записка

до дипломного проєкту (роботи)

бакалавр

(ступінь вищої освіти)

на тему РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ МОНІТОРИНГУ
ПАРКОВКИ
SOFTWARE DEVELOPMENT FOR PARKING MONITORING

Виконав: студент 4 курсу, групи КНТ-219сп

Спеціальності 122 Комп'ютерні науки

забезпечення

(код і найменування спеціальності)

Освітня програма (спеціалізація)

Комп'ютерні науки

Резніченко А.С.

(прізвище та ініціали)

Керівник Корнієнко С.К.

(прізвище та ініціали)

Рецензент Казурова А.Є.

(прізвище та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»
(повне найменування закладу вищої освіти)

Інститут, факультет ФКНТ
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.
С.О. Субботін
“ ” 20 року

З А В Д А Н Н Я

НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА

Резніченка Андрія Сергійовича

(прізвище, ім'я, по батькові)

1. Тема проєкту (роботи) Розробка програмного забезпечення моніторингу парковки. Software Development for Parking Monitoring.
керівник проєкту (роботи) Корнієнко С.К., к.т.н, доцент,
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом закладу вищої освіти від “27” квітня 2022 року № 102

2. Строк подання студентом проєкту (роботи) 7 червня 2022 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. Вибір та обґрунтування структури системи, яка проєктується. 3. Основні рішення щодо реалізації компонентів системи. 4. Керівництво програміста. 5. Керівництво оператора.

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

6. Консультанти розділів проєкту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-5 Основна частина	Корнієнко С.К, к.т.н., доцент		
Нормоконтроль	Дейнега Л. Ю., ст. викладач		

7. Дата видачі завдання “09” лютого 2022 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	2-3 тижні	Розділ 1
3	Розробка архітектури програми	4 тиждень	Розділ 2
4	Розробка програми	5-6 тижні	Розділ 3
5	Тестування та експериментальне дослідження програми	7 тиждень	Розділи 4-5
6	Оформлення пояснювальної записки та документів до неї. Нормоконтроль та рецензування.	8 тиждень	Додатки
7	Захист роботи	9 тиждень	

Студент

(підпис) Резніченко А. С.
(прізвище та ініціали)

Керівник проєкту (роботи)

(підпис) Корнієнко С.К.
(прізвище та ініціали)

РЕФЕРАТ

Пояснювальна записка до дипломної кваліфікаційної роботи бакалавра:
78 с., 11 табл., 39 рис., 4 дод., 31 джерело.

ПАРКОВКА, ТРЦ, РЕЄСТРАЦІЯ ТРАНСПОРТНОГО ЗАСОБУ,
МОНІТОРИНГ, НАЯВНІСТЬ ВІЛЬНИХ МІСЦЬ, QT, C++, LARAVEL, PHP.

Об'єкт дослідження – автоматизована система моніторингу та збору статистичних даних для паркування.

Предмет дослідження – методи моніторингу та збору статистичних даних про паркування.

Мета роботи – створення автоматизованої системи для моніторингу та збору статистичних даних про паркування.

Актуальність та проблематика роботи полягає у можливості вести облік, моніторинг та збір статистичних даних про транспортні засоби на паркуваннях, зберігання, редагування та видалення даних про транспортні засоби, їх власників.

Матеріали, методи та технічні засоби: Для розробки автоматизованої системи для паркування було обрано саме технологію Qt та мову програмування C++, а для розробки клієнтського сервісу – було обрано фреймворк Laravel та мову програмування PHP, процесор не нижче Intel Core i3 під управлінням операційної системи Microsoft Windows 10.

Результати. Створено систему моніторингу та збору статистичних даних для паркування.

Висновки. Розроблено автоматизовану систему яка дозволяє проводити моніторинг та збір статистичних даних про паркування.

Галузь використання – моніторинг та збір статистичних даних про паркування.

ABSTRACT

Explanatory note to the diploma qualifying work of the bachelor: 78 pages, 11 tables, 39 figures, 4 appendixes, 31 sources.

PARKING, SHOP, VEHICLE REGISTRATION, MONITORING, AVAILABILITY, QT, C ++, LARAVEL, PHP.

The object of research is an automated system for monitoring and collecting statistical data for parking lots.

The subject of the research is methods of monitoring and collecting statistical data for parking lots.

The purpose of the work is to create an automated system for monitoring and collecting statistical data for parking lots.

The urgency and problems of the work is the ability to easily and easily keep records, monitor and collect statistics on vehicles in parking lots, storage, editing and deletion of data on vehicles, their owners.

Materials, methods and technical means: Qt technology and C ++ programming language were chosen for the development of the automated parking system, and Laravel framework and PHP programming language were chosen for the development of the client service, the processor is not lower than Intel Core i3 running the operating system Microsoft.

Results. A system for monitoring and collecting statistical data for parking lots has been created.

Conclusions. An automated system has been developed that allows monitoring and collecting statistical data for parking lots.

Area of use - monitoring and collection of statistics for parking.

ЗМІСТ

	С.
Перелік скорочень та умовних позначок	8
Вступ	9
1 Аналіз предметної області.....	10
1.1 Основні поняття	10
1.2 Опис предметної області.....	12
1.3 Огляд існуючих програмних засобів, що вирішують аналогічні завдання	15
1.3.1 Програмний продукт «Simple Parking»	16
1.4 Реєстрація та вибір домену для сайту	22
1.5 Хостинг для сайту.....	22
1.6 Висновки за розділом 1	23
2 Вибір та обґрунтування структури системи, яка проєктується	23
2.1 Вибір технології розробки	24
2.2 Фреймворк Laravel.....	25
2.3 Фреймворк Django	26
2.4 Node.js	27
2.5 Фреймворк Qt.....	27
2.6 Порівняльна таблиця існуючих фреймворків веб розробки	29
2.7 Вибір бази даних.....	29
2.7 Опис мови програмування PHP	30
2.8 Опис мови програмування C ++.....	31
2.9 Висновки за розділом 2	31
3 Основні рішення щодо реалізації компонентів системи.....	32
3.1 Постановка задачі	32
3.2 Проєктування бази даних	35
3.3 Характеристики програми	36
3.4 Висновки за розділом 3.....	37
4 Керівництво програміста.....	38
4.1 Призначення й умови використання	38

4.2 Вхідні та вихідні дані 38

4.3 Повідомлення 39

4.4 Висновки за розділом 4 40

5 Керівництво оператора..... 41

5.1 Призначення програми й умови виконання програмного продукту 41

5.2 Виконання програмного продукту 41

5.3 Висновки за розділом 5 45

Висновки 46

Перелік джерел посилань 47

Додаток А Технічне завдання 51

Додаток Б Опис програмного продукту 55

Додаток В Текст файлів програми 61

Додаток Г Слайди презентації 72

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК

API	– application Programming Interface;
БД	– база даних;
ОС	– операційна система;
ПК	– персональний комп'ютер;
ПБ	– прізвище, ім'я, по батькові;
ПП	– програмний продукт;
СКБД	– система керування базами даних;
ТЗ	– транспортний засіб;
ТРЦ	– торгово-розважальний центр;
ТЦ	– торговий центр.

ВСТУП

Автоматизована система моніторингу та збору статистичних даних про паркування – це сучасне рішення, яке дозволить великим торгівельним центрам отримувати статистичні данні з кількості відвідувачів які приїхали на власному авто. З кожним роком кількість людей, які мають власний транспортний засіб зростає, кожен з них частіше відвідує великі торгівельні центри де роблять закупівлю багатьох товарів одразу. Для цього автоводії користуються парковками біля цих торгівельних центрів. Та чим більше автомобілів, тим більше розрахунків потрібно робити, для отримання статистичних даних з кількості відвідувачів торгівельних центрів для можливості планування акцій та розпродажів. Все це є дуже ресурсо–затратно та вимагає багато часу. Також для відвідувачів ТЦ є дуже проблематично знайти вільне місце на парковці у період акцій та розпродажів, коли одразу в одному ТЦ багато магазинів проводять такі заходи [1].

Облік автомобілів на парковці для операторів чи охоронців – це доволі складний процес, який характеризується великими трудовитратами, як по кількості необхідних часових ресурсів, так і вимагає особливої уваги при підрахунках. Найбільша інтенсивність таких процесів в періоди, коли люди повертаються з роботи, у святкові дні та під час розпродажів та акцій у ТЦ [2].

Тому треба розробити автоматизовану систему моніторингу та збору статистичних даних для паркування, яка могла би вмістити у собі усі ці можливості: облік автомобілів відвідувачів, підбір вільного місця на парковці. Автоматизована система моніторингу та збору статистичних даних для паркування має бути доступна у будь-який час доби та мати доступ до мережі інтернет. На даний момент багато паркувальних майданчиків, в яких не ведеться облік автомобілів, моніторинг та збір статистичних даних про транспортні засоби на паркуваннях, зберігання, редагування та видалення даних.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Основні поняття

Парковка – це спеціальна відкрита площадка для зберігання транспортних засобів. Управління персоналом та парковкою здійснюється директором або керуючим. Він безпосередньо розпоряджається коштами в рамках бюджету, приймає всі інші рішення щодо організації парковки [1].

В'їзд (розміщення), обслуговування та вихід клієнтів контролюються операторами паркування.

Якщо парковка платна то розміщення транспортного засобу відбувається наступним чином – проводиться реєстрація клієнта (ім'я, телефон) та реєстрація транспортного засобу (ТЗ) в журналі (реєструється державний реєстраційний номер), перевіряється наявність вільних місць, присвоюється код місця для паркування та визначається певний час на який потрібно залишити автомобіль [1].

Реєстрація в системі транспортного засобу відбувається тільки в разі платного паркування. Якщо ми розглядаємо безкоштовну парковку, то абсолютно ніяких даних не зберігається і не обробляється – за результатами дослідження було виявлено, що тільки на 3-х парковках з 10 є декілька камер, які реєструють тільки момент в'їзду і виїзду автомобіля на парковку. Розпізнаванням реєстраційних знаків або визначенням завантаженості парковки і наявності вільних місць вони не займаються.

Виїзд ТЗ полягає в наступному – власник ТЗ до закінчення терміну оренди місця повинен з'явитись та надати документи, де знаходиться транспортний засіб, та документи на ТЗ [2] - [3].

Реєстрація проводиться у програмі, вводяться дані марка та модель ТЗ, далі проводиться пошук найближчої парковки та вільного місця на ньому.

В кінці тижня оператори автостоянок формують звіти про загальний дохід та статистику (кількість зайнятих місць, період оренди місця, марки та

моделі автомобілів), після чого звіт надсилається директору або керуючому паркінгу.

Основним напрямком діяльності паркінгу є відповідальність за збереження автомобіля та майна яке знаходиться всередині нього.

Директор/Керуючий парковкою володіє наступними обов'язками:

- прийом на роботу та звільнення працівників;
- контроль підлеглих;
- консультація клієнтів;
- контроль дебіторської заборгованості;
- пошук та залучення клієнтів;
- робота з обслуговуючими підприємствами;
- розвиток території;
- планування виконання цільових завдань.

Оператор/Охоронець парковки має наступні обов'язки:

- контроль в'їзду/виїзду ТЗ та реєстрація його;
- ведення вахтового журналу;
- забезпечення порядку закріпленої території;
- формування щотижневого звіту.

Основним напрямком діяльності паркінгу є відповідальність за збереження автомобіля та майна яке знаходиться всередині нього.

Загальна схема організаційної структури парковки представлена на рис.1.1.



Рисунок 1.1 – Організаційна структура паркінгу

Користувачами системи є: директор, оператор, клієнт.

Рисунок 1.1 показує, що директор/керуючий парковки є найвищою ланкою в цій організаційній структурі. Основна відповідальність директора – забезпечити максимальну ефективність парковки. Процес реєстрації транспортного засобу та реєстрації клієнтів у системі здійснюється операторами. Оператор призначає код місця транспортному засобу, на якому останній буде стояти у відведений час. Оператор знайомить клієнта зі схемою пакувального майданчика у разі надзвичайної ситуації для евакуації з місця, реєструє клієнта та транспортний засіб у системі та видає клієнту перепустку із зазначеним місцем для паркування. Клієнт, у свою чергу, сплачує послугу зберігання і ставить свій транспортний засіб на призначене місце. В кінці кожного місяця формується детальний звіт, який включає інформацію про зареєстрованих клієнтів та транспортні засоби, час перебування на парковці і надсилається директору/керуючому [3].

1.2 Опис предметної області

Аналіз предметної галузі показав основні бізнес-процеси:

- формування Бріф-анкети на клієнта та ТЗ;
- реєстрація ТЗ;
- формування та відображення звіту про зроблену реєстрацію;
- пошук вільного місця на парковці.

Для реєстрації клієнта необхідна його контактна інформація та зберегти її на сервері (рис. 1.2) та у таблиці 1.1.

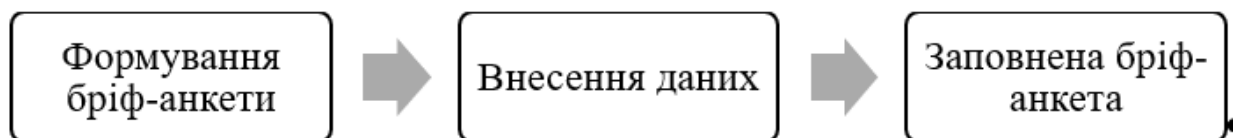


Рисунок 1.2 – Реєстрація клієнта

Таблиця 1.1 –Формування бриф – анкети

Назва характеристики	Значення характеристики
Ім'я бізнес процесу	Реєстрація клієнта
Основні учасники	Сервер, браузер, клієнт
Вхідна подія	Відправлення даних про клієнта
Вхідні документи	Немає
Вихідна подія	Відповідь про статус реєстрації
Вихідні документи	Немає
Клієнт бізнес процесу	Сервер

Реєстрація ТЗ (рис. 1.3, табл.1.2) включає наступні дані:

- власник ТЗ (фізична або юридична особа);
- номер державної реєстрації ТЗ
- час зберігання.

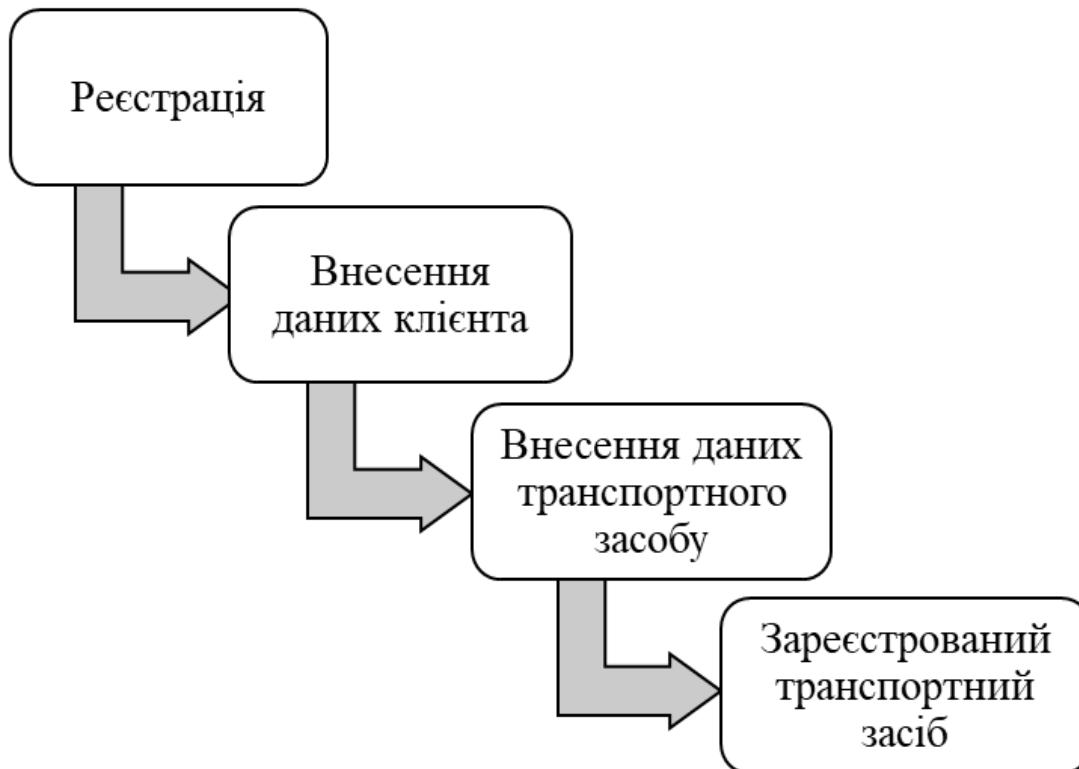


Рисунок 1.3 –Реєстрація транспортного засобу

Таблиця 1.2 –Реєстрація транспортного засобу

Назва характеристики	Значення характеристики
Ім'я бізнес процесу	Реєстрація транспортного засобу
Основні учасники	Сервер, браузер, клієнт
Вхідна подія	Відправлення даних про транспортний засіб
Вхідні документи	Свідоцтво про реєстрацію авто
Вихідна подія	Відповідь про статус реєстрації, присвоєння реєстраційного коду транспортному засобу
Вихідні документи	<u>Паркувальний талон</u> , тимчасова чи постійна перепустка або безконтактні смарт-картки
Клієнт бізнес процесу	Сервер

Бронювання місця (рис. 1.4) для паркування включає наступні дані:

- доступне місце для паркування;
- власник ТЗ (фізична або юридична особа);
- номер державної реєстрації ТЗ
- час зберігання [3].

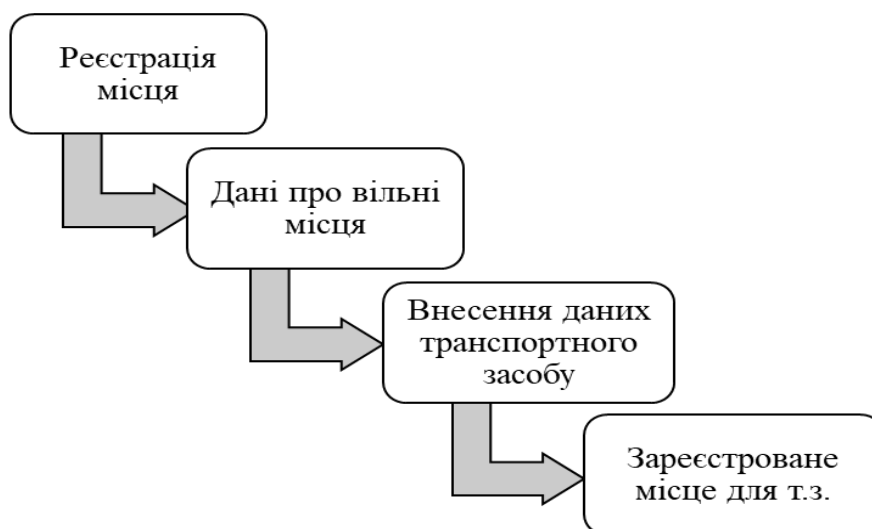


Рисунок 1.4 –Бронювання місця для паркування

Характеристики процесу представлені в табл. 1.3.

Таблиця 1.3 – Опис Бронювання місця для паркування

Назва характеристики	Значення характеристики
Ім'я бізнес процесу	Бронювання місця на автостоянці
Основні учасники	Сервер, браузер, клієнт
Вхідна подія	Відправлення даних про клієнта транспортний засіб
Вхідні документи	Немає
Вихідна подія	Відповідь про статус бронювання, присвоєння коду парко – місця транспортному засобу
Вихідні документи	Немає
Клієнт бізнес процесу	Сервер

1.3 Огляд існуючих програмних засобів, що вирішують аналогічні завдання

З бурхливим розвитком інтернет–технологій, участь комп'ютерів майже у будь–якому аспекті людського життя стає все більш розповсюдженим вільно та корисно проводити свій час, із збільшенням кількості транспортних засобів проблема моніторингу, обліку та збору статистичної інформації про кількість відвідувачів у ТЦ стає дуже затребуваною для можливості планування акцій та розпродажів.

Декілька років тому середня кількість автомобілів на 1000 населення становила приблизно 150 машин. Сьогодні кількість автомобілів перевищує 232. Таким чином, існує необхідність у створенні нових паркувальних місць та відкритті цілих нових паркувань. З такою великою кількістю автомобілів на паркувальних майданчиках надзвичайно доцільно використовувати систему обліку [4].

Сьогодні ринок таких систем дуже обмежений.

Розглянемо існуючі системи паркування та їх пошукові системи:

- «Simple Parking» [5];
- «SERVIO PARKING» [6];
- «AllStojanka» [7];
- Парковка, стоянка в Борисполі [8];
- Мікком AS101 ProPark [9].

Ці системи призначені для автоматизації процесу реєстрації та реєстрації транспортних засобів на стоянці.

1.3.1 Програмний продукт «Simple Parking»

Система «Simple Parking» (рис.1.5) має зручний інтерфейс, що дозволяє інтуїтивно зрозуміле управління. Можна переглянути схему стоянки, відобразити схему стоянки з поділом на цікаві та вільні місця, швидко зареєструвати транзакцію з розміщення та вилучення транспортного засобу з місця для паркування, показати її на макеті конкретного транспортного засобу. Він призначений для встановлення в операційних системах, таких як Windows XP, Vista та новіших версіях [5].



Рисунок 1.5 – Головне вікно програми «Simple Parking» [5]

1.3.2 Програмний продукт «SERVIO PARKING»

Основні особливості системного програмного забезпечення «SERVIO PARKING»:

- встановлення різних параметрів з точки зору часу, типу валюти, кількості місць, що використовуються для паркування, фактично використовуваних та зайнятих місць на даний момент тощо;
- встановлення тарифів погодинної оплати;
- управління системою автоматичного паркування та встановлення різних налаштувань із системного сервера;
- присвоєння логіна та пароля кожному оператору для роботи з системою, а також різні дозволи на певні дії;
- можливість отримувати та архівувати звіти за будь-який період діяльності (журнал платежів і пред'явлення абонементів і т.д.).

Загальний вигляд системи наведено на рисунках 1.6, 1.7, [6].

10.0.2.247.2000

ТВ

ITTV

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

Б

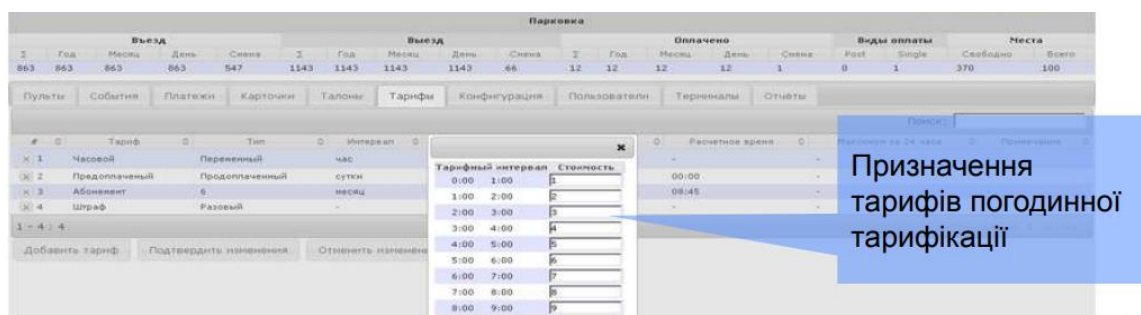
Б

Б

Б

Б

Рисунок 1.6 – Програмний продукт «SERVIO PARKING» [6]



#	Тариф	Тип	Интервал	Расчетное время
X 1	Часовой	Переменный	час	-
X 2	Предоплаченный	Предоплаченный	сутки	-
X 3	Абонемент	Б	месяц	-
X 4	Штраф	Разовый	-	-

Тарифный интервал	Стоимость
0:00 - 1:00	Б
1:00 - 2:00	Б
2:00 - 3:00	Б
3:00 - 4:00	Б
4:00 - 5:00	Б
5:00 - 6:00	Б
6:00 - 7:00	Б
7:00 - 8:00	Б
8:00 - 9:00	Б

Рисунок 1.7 – Програмний продукт «SERVIO PARKING» [6]

1.3.3 Програмний продукт «AllStojanka»

Програмний продукт AllStojanka який дозволяє [7]:

- вести записи в електронному вигляді в БД із простим та інтуїтивно зрозумілим інтерфейсом;
- отримувати достовірну та своєчасну інформацію про взаєморозрахунки;
- наочно відображати схему стоянки з поділом на зайняті та вільні місця;
- швидко оформити операцію постановки та зняття транспортного засобу з місця стоянки;
- прийняти та формалізувати отримання коштів з печаткою касового чека;
- автоматично розраховувати вартість послуг паркування з використанням різних опцій та тарифів;
- вести облік знижок;
- автоматично розраховувати та нараховувати відсотки за несвоєчасну виплату;
- контролювати дні та час затримки оплати.

1.3.4 Програмний продукт «Парковка, стоянка у Борисполі»

На рисунках 1.8 – 1.10 показано Інтернет–послугу «Паркінг, стоянка у Борисполі» [8].

Інтернет–система «Парковка, стоянка в Борисполі » має досить простий та інтуїтивно зрозумілий інтерфейс. Сайт простий, не містить жодних спеціальних функцій, загалом містить інформаційний характер про послуги на стоянці біля аеропорту Бориспіль. У " Схемі подорожі " (рис. 1.9–1.11) показано три варіанти шляху до трьох різних паркувальних майданчиків. Позначення на стоянці в відповідно до Google карти використовуються, координати GPS,

адреса та контактні номери вказані також. На наступній вкладці «Онлайн– бронювання» (рис. 1.11) відображається форма для введення даних власника автомобіля, та даних автомобіля для бронювання місця на стоянці. Наступні вкладки – дати контактів усіх стоянок та адміністрації, а також інші послуги, які надає саме паркування.

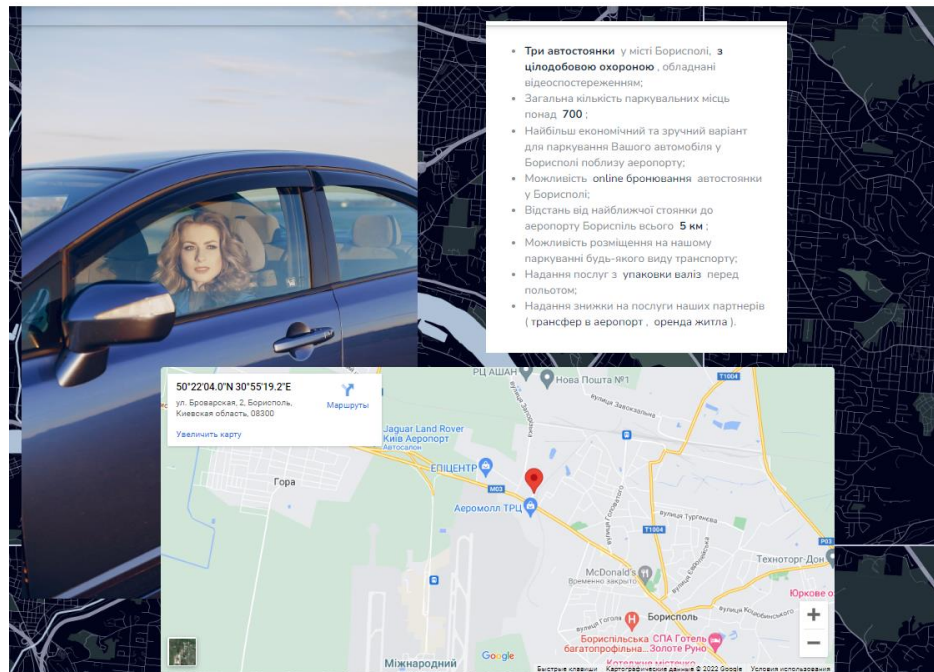


Рисунок 1.8 – Головне вікно програми «Парковка, стоянка у Борисполі» [8]

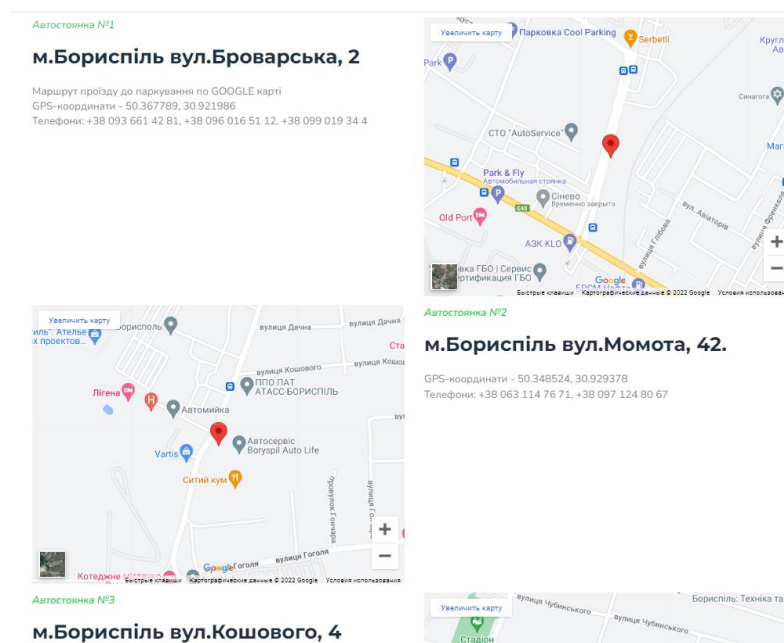


Рисунок 1.9 – Вікно програми «Парковка, стоянка у Борисполі» [8]

Рисунок 1. 10 – Вікно програми «Парковка, стоянка у Борисполі» [8]

1.3.5 Програмний продукт «Мікком AS101 ProPark»

Програмний продукт «Мікком AS101 ProPark» [9] має простий та зручний інтерфейс, все на одній сторінці. Всі дані про транспортний засіб, такі як номер державної реєстрації, марка, колір та примітка, необхідні для бронювання. Далі вводяться дані власника автомобіля. Після цього надається місце для паркування та зазначається термін зберігання автомобіля на парковці. Усі зареєстровані машини, які утримуються на стоянці, показані в таблиці нижче. Ця система не містить бронювання об'єкта, він призначений виключно для використання адміністрації паркування [8].

1.3.6 Порівняльна таблиця існуючих програм

Основні показники можливостей розглянутих вище програмних продуктів для паркування наведені у табл. 1.5.

Таблиця 1.5 – Основні показники сучасних програмних продуктів для паркування

	Simple Parking	SERVIO PARKING	AllStojanka	Парковка, стоянка в Борисполі	Мікком AS101 ProPark
1	2	3	4	5	6
Інтерфейс користувача	Desktop–система	Desktop–система	Desktop–система	Web–система	Desktop–система
Облік	+	+	+	+	+
Відображення схеми автостоянки	+	+	+	–	+
Приймання та оформлення коштів	+	+	–	–	+
Відображення конкретного ТЗ	+	+	+	–	+
Перевірка клієнтів	+	–	+	–	–
Реєстрація клієнтів	+	+	+	+	+
Контроль прострочення	+	–	+	–	–

Порівнявши програмні продукти зазначені вище, можна виділити їх основні переваги та недоліки. Simple Parking система має зручний, приємний на перший погляд і зрозумілий призначений для користувача інтерфейс. Система настільна, що не дозволить клієнтам дивитись на безкоштовні місця для паркування в Інтернеті. У SERVIO PARKING [6] системи, це настільна система, в відміну від попереднього, він має функцію відображення вільних паркувальних місць, а функція оплати послуг також є суттєвою перевагою. Недоліком такої системи є складний інтерфейс, який може бути не відразу зрозумілим для нових користувачів системи. Системи AllStojanka та Мікком AS101 ProPark , як і попередні, є настільними системами, на відміну від попередніх систем, ці системи мають набагато більше додаткових функцій (обчислення штрафних санкцій, облік знижок, розрахунок вартості послуг). Також тут простий на перший погляд користувацький інтерфейс без

красивого та яскравого дизайну. Веб– система «Парковка, стоянка в Борисполі», містить лише більш інформативний характер із інформацією про те, як дістатися до стоянки та контактною інформацією. Ця система має лише одну функцію бронювання місця для паркування.

Під час виконання дипломної кваліфікаційної роботи бакалавра будуть враховані всі переваги та недоліки розглянутих систем та веб–системи моніторингу про паркування та стоянок.

1.4 Вибір доменного імені та його реєстрація

Щоб вибрати доменне ім'я, необхідно використовувати наступні рекомендації:

- коротке та легко вимовне доменне ім'я, яке чітко описує суть сайту або послуги без використання складних символів, дефісів і цифр;
- розширення доменного імені (необхідно вибирати відповідно до спрямованості сайту (.com - домени реєструються комерційними організаціями, .ua - український хостінг, .com.ua - домени реєструються комерційними організаціями з України, .net - домени реєструються для інтернет магазинів, .org - загальний домен верхнього рівня та інші));
- необхідно перевірити доменне ім'я, щоб воно не співпадало з відомими торговими марками (прав на товарний знак).

Щоб отримати доменне ім'я, необхідно знайти реєстратора домену потім перевірити, чи доступний даний домен чи ні. Якщо доступний, то зареєструвати його [10] - [14].

1.5 Хостинг для сайту

Веб-хостинг — це онлайн-сервіс, який дозволяє публікувати файли веб-сайту в Інтернеті. Отже, кожен, хто має доступ до Інтернету, має доступ до

сайту. На практиці зазвичай це стосується послуги, яку ви отримуєте від постачальника веб-хостингу [12] - [16].

Види веб-хостингу:

- спільний хостинг;
- хостинг WordPress;
- VPS хостинг;
- виділений хостинг;
- хмарний хостинг.

Веб-хостинг має різноманітні специфікації та пакети:

- має обмеження пропускної здатності та трафіку;
- вбудований конструктор веб-сайтів;
- обсяг пам'яті;
- хостинг електронної пошти;

1.6 Висновки за розділом 1

У розділі були розглянуті основні проблеми, що виникають при зберіганні транспортних засобів на парковці та можливі варіанти їх вирішення. Також були розглянуті провідні програмні продукти на ринку паркування та проаналізовано їх можливості. На підставі розглянутого було визначені цілі випускної кваліфікаційної роботи бакалавра.

2 ВИБІР ТА ОБҐРУНТУВАННЯ СТРУКТУРИ СИСТЕМИ, ЯКА ПРОЄКТУЄТЬСЯ

2.1 Вибір технології розробки

Розробку програмного забезпечення потрібно виконувати у вигляді веб-додатку, до якого зможе через мережу інтернет мати доступ користувач.

Веб-додаток (Web app) — це прикладна програма, яка зберігається на віддаленому сервері та доставляється через Інтернет через інтерфейс браузера. Веб-сервіси за визначенням є веб-програмами, і багато веб-сайтів, хоча й не всі, містять веб-програми. За словами редактора Web.AppStorm Джарела Реміка, будь-який компонент веб-сайту, який виконує певну функцію для користувача, кваліфікується як веб-програма [17] - [19].

Веб-додатки можуть бути розроблені для широкого спектру використання і можуть використовуватися будь-ким; від організації до окремої особи з багатьох причин. Зазвичай використовувані веб-додатки можуть включати веб-пошту, онлайн-калькулятори або магазини електронної комерції. До деяких веб-програм можна отримати доступ лише через певний браузер; однак більшість із них доступні незалежно від браузера [19].

Веб-програми не потрібно завантажувати, оскільки доступ до них здійснюється через мережу. Користувачі можуть отримати доступ до веб-програм через веб-браузер, такий як Google Chrome, Mozilla Firefox або Safari .

Щоб веб-програма працювала, їй потрібні веб-сервер, сервер додатків і база даних. Веб-сервери керують запитами, які надходять від клієнта, а сервер додатків виконує запитане завдання. Базу даних можна використовувати для зберігання будь-якої необхідної інформації [20].

Веб-додатки зазвичай мають короткі цикли розробки і можуть бути створені невеликими командами розробників. Більшість веб-програм написані на JavaScript, HTML5 або каскадних таблицях стилів (CSS). Програмування на стороні клієнта зазвичай використовує ці мови, які допомагають створювати інтерфейс програми. Програмування на стороні сервера виконується для

створення сценаріїв, які використовуватиме веб-додаток. Такі мови, як Python, Java і Ruby, зазвичай використовуються в програмуванні на стороні сервера.

2.2 Фреймворк Laravel

Laravel — це безкоштовний веб-фреймворк PHP з відкритим кодом, призначений для розробки веб-додатків за архітектурним шаблоном MVC model-view-controller. Це потужний фреймворк MVC PHP, розроблений для розробників, яким потрібен простий інструмент для створення повнофункціональних веб-додатків [21].

Laravel спрощує звичайні завдання, які використовуються в більшості веб-проектів, такі як;

- аутентифікація;
- кешування та маршрутизація.

Laravel був створений, щоб зробити процес розробки простим і приємним для розробників, не жертвуючи жодною функціональністю веб-додатка. Використовуючи PHP Framework, ви в кінцевому підсумку заощадите купу часу, позбавитеся необхідності створювати повторюваний код та зможете швидко створювати програми (RAD) [21].

Переваги використання фреймворку Laravel:

- висока безпека;
- підвищена продуктивність кешування;
- аутентифікація;
- відкритий код і велике ком'юніті;
- шаблонизатор blade;
- міграції баз даних;
- MVC-архітектура;
- об'єктно-орієнтовані бібліотеки;
- юніт-тести;
- перспективні додатки;

- дружній код;
- багатомовність;
- інтеграція з mail-сервісом;
- робота з помилками і винятками;
- система відкладених завдань;
- запуск завдань за розкладом;
- швидкість розробки) [22].

2.3 Фреймворк Django

Django це високорівневий веб-фреймворк Python. Створений досвідченими розробниками, він вирішує велику частину праці веб-розробки, тому ви можете зосередитися на написанні свого додатка без необхідності винаходити велосипед. Це безкоштовний веб-фреймворк РНР з відкритим кодом [23].

Переваги використання фреймворку Django:

- ORM, API доступ до БД з підтримкою транзакцій;
- вбудований інтерфейс адміністратора, з уже наявними перекладами на більшість мов;
- диспетчер URL на основі регулярних виразів;
- розширювана система шаблонів з тегами та наслідуванням;
- система кешування;
- інтернаціоналізація;
- архітектура застосунків, що підключаються, які можна встановлювати на будь-які Django-сайти;
- «generic views» - шаблони функцій контролерів;
- аутентифікація;
- система фільтрів («middleware»);
- бібліотека для роботи з формами [24].

2.4 Node.js

Node.js: - це кросплатформне середовище JavaScript з відкритим вихідним кодом, зосереджене на серверних і мережевих додатках [25].

Node.js дозволяє розробникам створювати швидкі, масштабовані мережеві програми за допомогою легкого для розуміння коду. Працює на ОС Windows, Mac OSX, Linux, Unix та інших операційних системах. Підтримує процесори ARM, такі як Raspberry Pi або Beaglebone Black.

Node.js оптимізовано для веб-додатків у режимі реального часу. Таким чином, надаючи йому перевагу над іншими мовами та фреймворками. Деякі фреймворки не можуть добре працювати при обробці одночасних підключень [25].

2.5 Фреймворк Qt

Qt – це графічний C++ –інструментарій, який розробляється з 1994 року Trolltech. Він доступний для Windows (будь-якої версії), Unix (будь-якої версії), MacOS X. Безсумнівно, інструментарій Qt повністю об'єктно-орієнтований [26].

Фреймворк може функціонувати на різних типах програмного та апаратного забезпечення. Спочатку вона була створена як бібліотека класів, і з роками ліцензія зазнала численні трансформації. Користувачі можуть вибирати між двома доступними типами ліцензій: комерційною платною ліцензією та безкоштовною ліцензією з відкритим кодом [26].

Основу фреймворка утворюють модулі Qt Essentials наведено у табл. 2.1.

Qt — популярне рішення, яке використовується в проектах, які зосереджені на створенні вбудованих пристроїв і програмного забезпечення Інтернету речей . Він також використовується для розробки настільних і мобільних пристроїв. Одним із найвідоміших проектів Qt є KDE Plasma , який часто оцінюють як найкраще середовище робочого столу. Qt використовують

такі компанії, як LG, Tesla, Microsoft, Samsung, BMW, Siemens, HP, Philips тощо[26].

Таблиця 2.1 – Модулі Qt Essentials

Модулі Qt Essentials	Особливості
Qt Core	Ці неграфічні основні класи не вимагалися всіма іншими модулями
Qt GUI	Базові класи, що використовуються для дизайну графічного інтерфейсу, включаючи OpenGL
Мультимедіа Qt	Набір типів QML і класів C++ для обробки мультимедіа
Мультимедійні віджети Qt	Класи на основі віджетів для реалізації мультимедійних функцій
Мережа Qt	API для програм покладається на мережі TCP/IP
Qt QML	Структура та типи мови розмітки Qt QML
Qt Quick	Framework для створення динамічних програм із користувацьким інтерфейсом QML
Швидкі елементи керування Qt 2	Легкі, дуже ефективні типи QML, які спрощують формулювання інтерфейсу
Швидкі діалоги Qt	Типи для створення та взаємодії з системними діалогами
Швидкі макети Qt	Типи QML для компоновання об'єктів у розробленому інтерфейсі
Швидкий тест Qt	Тестовий фреймворк для QML-додатків
Qt SQL	Класи для інтеграції баз даних SQL
Тест Qt	Класи для детального тестування програм і бібліотек Qt
Віджети Qt	Набір елементів інтерфейсу користувача для створення класичного інтерфейсу

2.6 Порівняльна таблиця існуючих фреймворків веб розробки

Популярною технологією веб розробки на даний момент є розробка на фреймворкі Laravel [17]-[26].

Результати порівняння фреймворку веб розробки наведено у табл. 2.2.

Таблиця 2.2 – Порівняння фреймворку веб розробки

Параметр	Laravel	Django	Node.js	Qt
1	2	3	4	5
Об'єктно–орієнтовані бібліотеки	+	+	+	+
Мова програмування	Python	Python	JavaScript	C ++
Юніт–тести	+	+	-	-
Проблеми безпекою	-	-	+/-	+/-
Вбудовані шаблони	+	+	-	+
Швидкість розробки	+	+	-	-
Інтеграція з mail–сервісом	+	—	-	-
Інтернаціоналізація	—	+	-	-
Розширювана система шаблонів з тегами та наслідуванням	+	+	-	-

Для розробки клієнтського сервісу – було обрано фреймворк Laravel

2.7 Вибір бази даних

Для розробки моделі та бази знань було вирішено використовувати MySQL [20].

MySQL — це система управління реляційною базою даних з відкритим вихідним кодом. Як і інші реляційні бази даних, MySQL зберігає дані в таблицях, що складаються з рядків і стовпців. Користувачі можуть визначати, маніпулювати, контролювати та запитувати дані за допомогою мови структурованих запитів, більш відомої як SQL. Ім'я MySQL є комбінацією «Му», імені дочки творця MySQL Майкла Віденіуса та «SQL» [20].

Гнучка і потужна програма MySQL є найпопулярнішою системою баз даних з відкритим вихідним кодом у світі. Як частина широко використовуваного технологічного стеку LAMP (який складається з операційної системи на базі Linux, веб-сервера Apache, бази даних MySQL і PHP для обробки), він використовується для зберігання та отримання даних у широкому спектрі популярних програм, веб-сайти та послуги. MySQL простий у використанні [20].

2.7 Опис мови програмування PHP

PHP це скриптова мова програмування створена для генерації HTML-сторінок на стороні веб-сервера [27]. PHP технологія застосовується у сфері веб додатків. Працює з наступними мовами програмування:

- Ruby;
- .NET;
- Java;
- Python;
- Perl.

Особливості технології PHP:

- вбудовані бібліотеки до БД (MySQL, PostgreSQL, SQLite, mSQL, Oracle, dbm, Hyperware, Informix, InterBase, Sybase) [28];
- традиційність. Конструкція мови запозичена з C, Perl;
- наявність сирцевого коду та безкоштовність;
- ефективність.

2.8 Опис мови програмування C ++

Мова C++ вдало поєднала в собі достоїнства об'єктно-орієнтованого програмування. З моменту свого створення C++ став однією з широко використовуваних мов програмування у світі. Добре написані програми на C++ працюють швидко та ефективно. Ця мова більш гнучка, ніж інші мови: вона може працювати на найвищих рівнях абстракції. C++ надає високооптимізовані стандартні бібліотеки. Він забезпечує доступ до низькорівневих апаратних функцій, щоб максимізувати швидкість та мінімізувати вимоги до пам'яті. C++ може створювати практично будь-які програми: ігри, драйвери пристроїв, НРС, хмарні, настільні, вбудовані та мобільні програми та багато іншого. Навіть бібліотеки та компілятори інших мов програмування пишуться на C++. Порівняння мов програмування наведено у таблиці 2.3 [26].

Таблиця 2.3 – Порівняння мов програмування [20] - [31]

Мова програмування	Швидкість обробки	Читабельність	Простота	GUI (графічний інтерфейс користувача)	Графіка (2D)	Графіка (3D)	Кроссплатформеність	Спец. процесор
C	8	6	2	3	5	8	7	10
C++	8	6	5	4	6	8	7	7
C#	7	7	5	6	6	7	2	—
Java	6	7	6	7	7	6	10	—
Python	2	5	10	8	10	1	10	—
VB.net	6	10	8	10	5	2	2	—
Assembler	10	—	—	—	—	—	—	5

2.9 Висновки за розділом 2

Для розробки системи було обрано технологію Qt та мову програмування C++, а для розробки клієнтського сервісу – було обрано фреймворк Laravel та мову програмування PHP.

3 ОСНОВНІ РІШЕННЯ ЩОДО РЕАЛІЗАЦІЇ КОМПОНЕНТІВ СИСТЕМИ

3.1 Постановка задачі

Автоматизована система – це програма клієнт–сервер для моніторингу паркінгів, парковок та стоянок на наявність вільних місць, відображення паркувальних місць на електронному табло та зберігання статистики з кількості автомобілів що відвідували паркування.

Необхідно розробити систему моніторингу стоянок, яка повинна складатися з адміністративної панелі та панелі оператора. Адміністративна панель повинна містити всі функції, за допомогою яких ви можете керувати системою, тобто керувати операторами, паркінгами, транспортними засобами, реєструвати користувачів та транспортні засоби самим адміністратором. Панель оператора повинна складатися з реєстрації користувачів, реєстрації транспортного засобу, пошуку місця для паркування. Сама автоматизована система повинна включати дані про схеми паркування, кількість місць на парковці та наявність вільних місць.

Список вимог до автоматизованої системи.

Система повинна:

- реєструвати клієнтів та їх транспортні засоби;
- зберігати отриманні данні до бази даних на сервері;
- мати високу швидкодію;
- мати інтерфейс для оператора та адміністратора;
- відображати кількість вільних місць на табло при в'їзді до паркування (рис. 3.1).



Рисунок 3.1 – Електронне табло з відображенням кількості вільних місць [9]

Після аналізу системних вимог виділено наступні випадки використання. Вони зображені у вигляді діаграми використання (рис. 3.2). Діаграма випадків використання дозволяє швидко побачити основні функції, які автоматизована система буде виконувати.

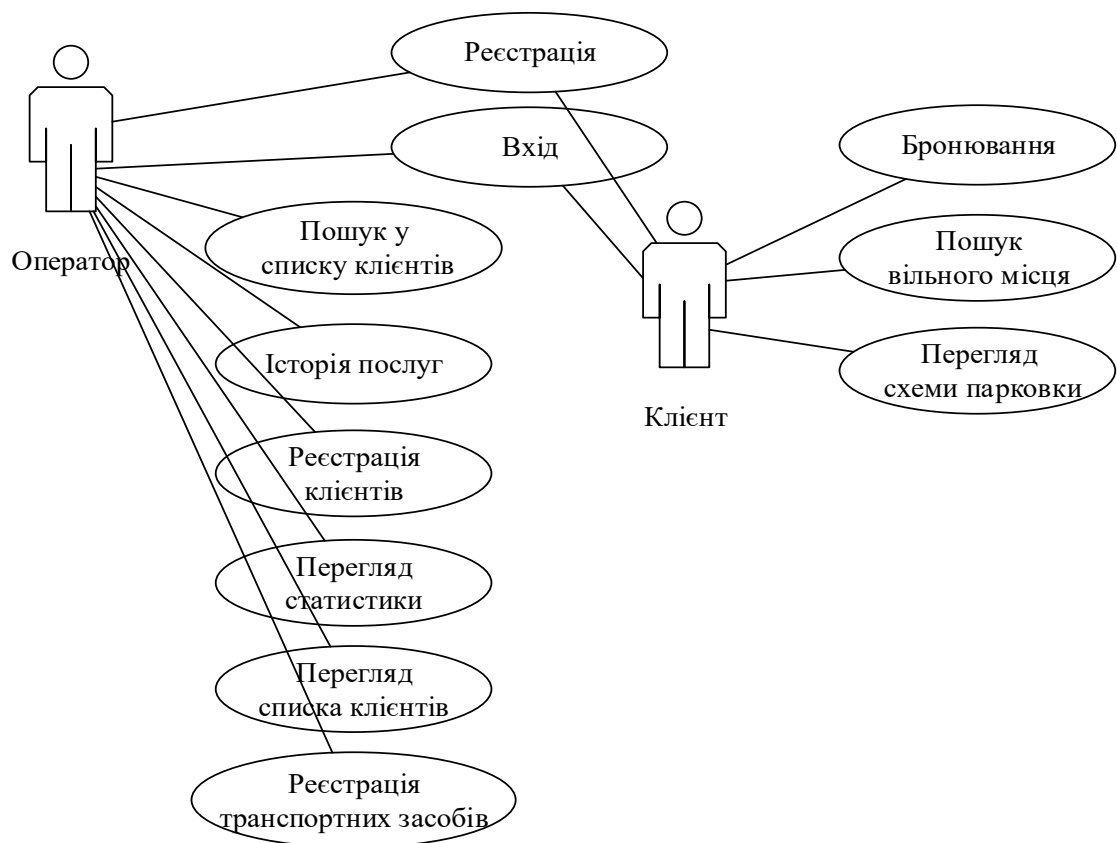


Рисунок 3.2 – Діаграма варіантів використання

Для реєстрації клієнта з сайту необхідно виконати такі дії;

– відкрити у браузері офіційний сайт сервісу <https://www.park.com.ua/parking/>:

– натисніть кнопку "Зареєструйтесь" - вона розташована на кожній сторінці:

– вкажіть дані на транспортний засіб:

– введіть адресу електронної пошти та перевірочний капча-код:

– активуйте обліковий запис, пройшовши за посиланням з листа від noreply@park.com.ua.

У випадку коли оператор/охоронець вже перебуває у системі, заходить у меню «Реєстрація транспортного засобу» реєстрація відбувається за спрощеною процедурою (рис. 3.3).

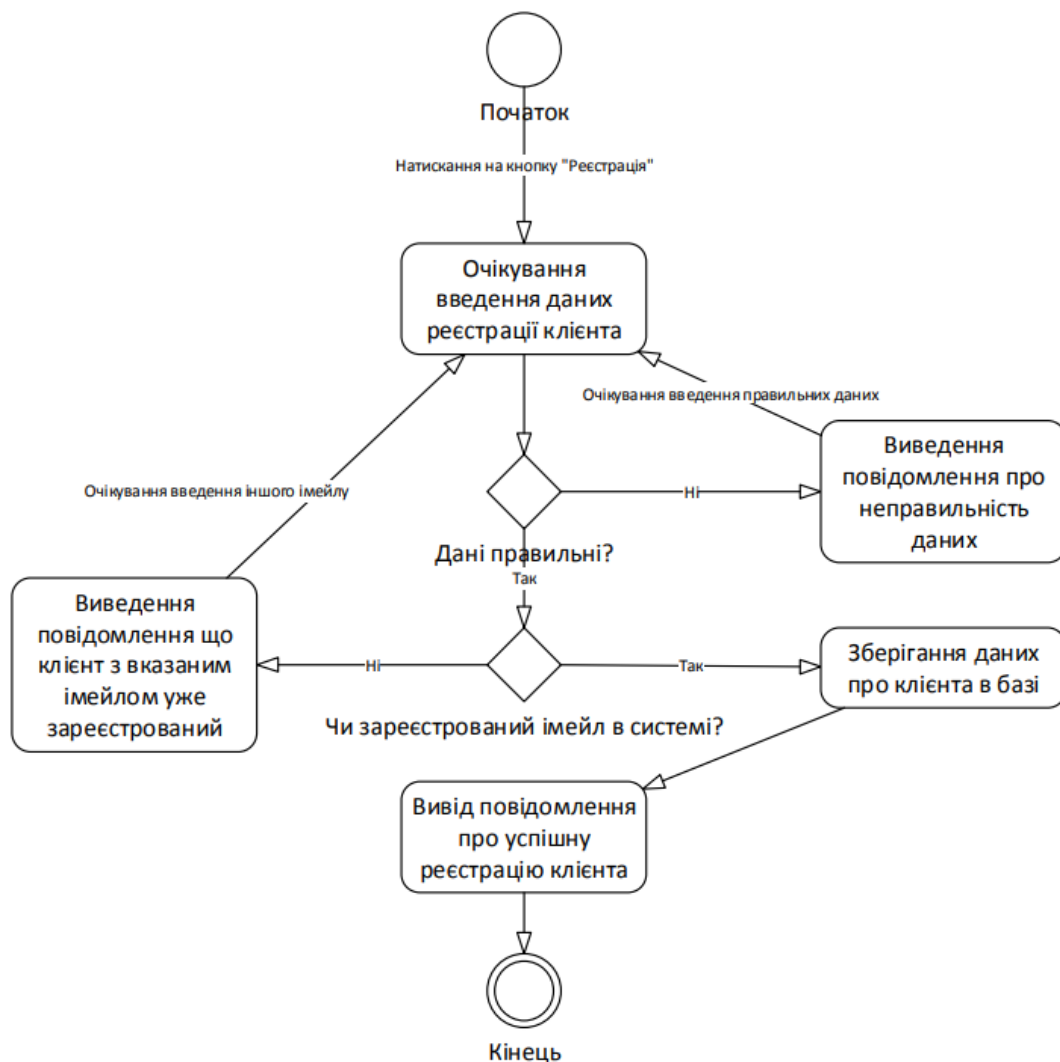


Рисунок 3.3 – Діаграма варіантів «Реєстрація транспортного засобу»

У випадку коли оператор/охоронець вже перебуває у системі, заходить у меню «Транспортні засоби» (рис. 3.4) може дивитися наявні ТЗ, які зареєстровані у програмі. Також оператор може редагувати інформацію про ТЗ, щоб видалити інформацію про ТС, необхідно натиснути на кнопку «Видалити». Щоб редагувати дані про ТС, необхідно натиснути на кнопку «Редагувати» та ввести потрібні дані та підтвердити редагування.

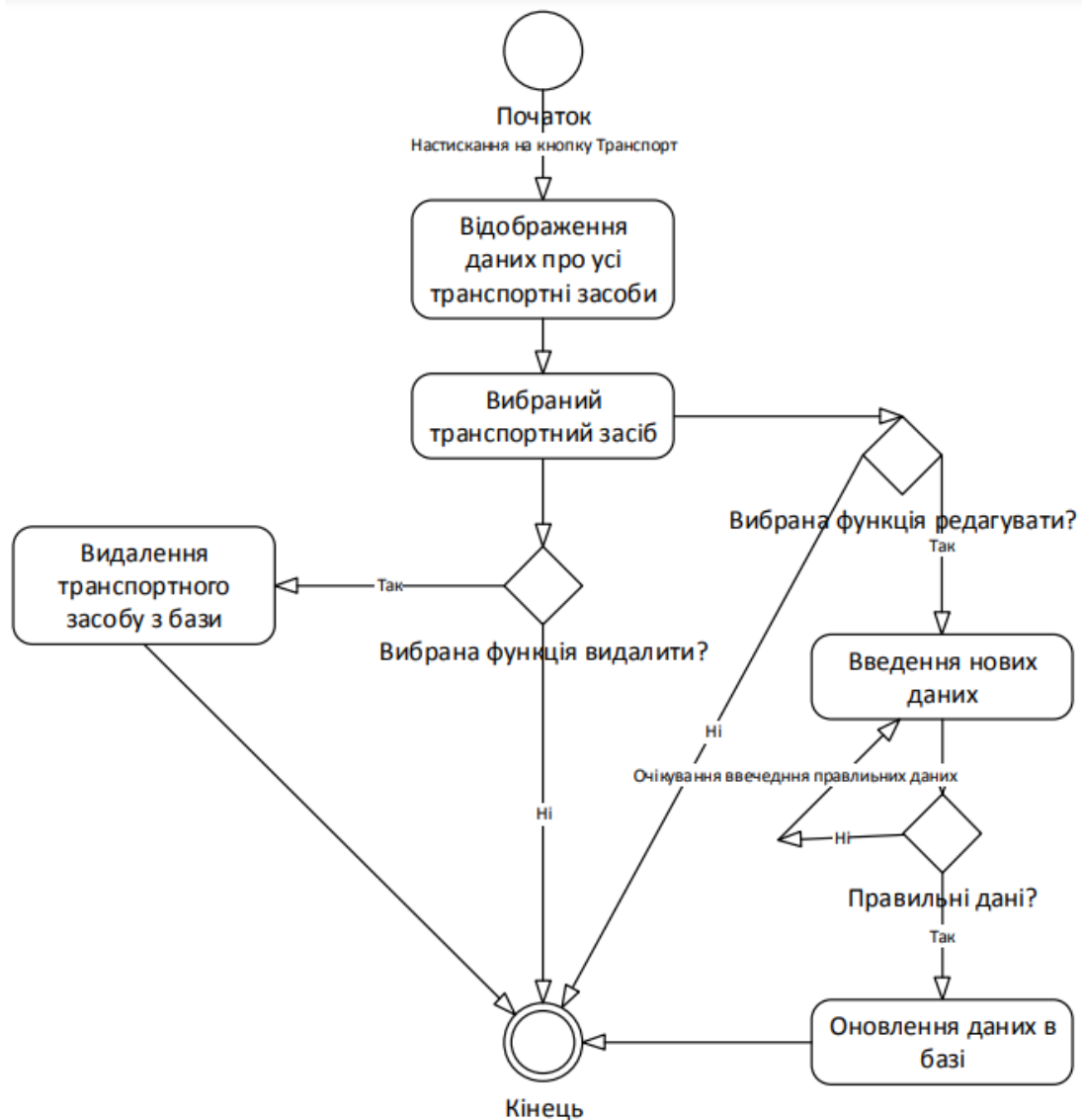


Рисунок 3.4 – Діаграма варіантів «Транспортні засоби»

3.2 Проєктування бази даних

Була розроблена схема даних автоматизованої системи паркування (рис.3.5).

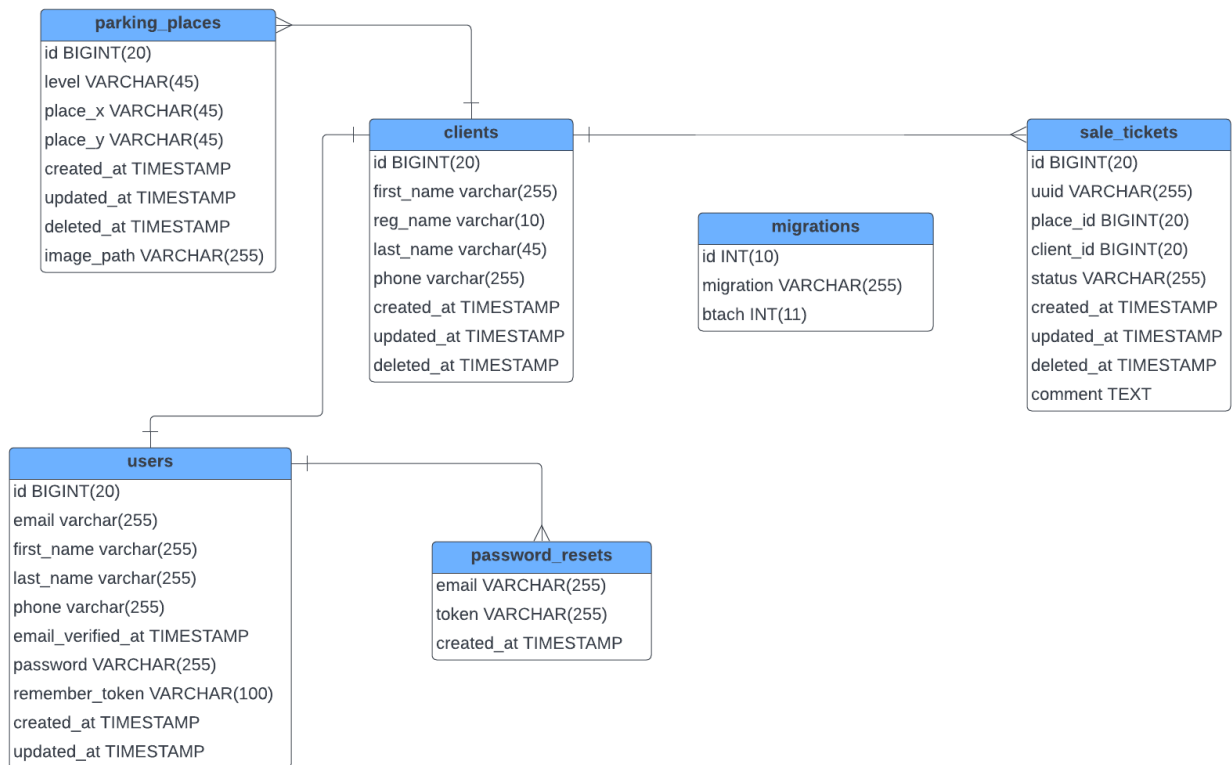


Рисунок 3.5 – Схема бази даних

Parking_places – це список справжніх місць на парковці, які визначаються за рівнями (column – level) та за координатами x та y (column – place_x & place_y).

Sale_tickets – це список існуючих сесій парковок, де place_id – foreign key, а client_id – nullable foreign key (клієнт може бути не зареєстрований у системі).

Clients – це список зареєстрованих клієнтів парковки.

Users – це список операторів або адміністраторів системи.

Таблиці migrations та password_resets – це службові таблиці системи.

3.3 Характеристики програми

Розроблена автоматизована система моніторингу та збору статистичних даних призначена для використання на паркуваннях в торгівельних центрах та на звичайних паркуваннях або автостоянках. При використанні цього програмного забезпечення спрощується облік автомобілів на паркуванні для

операторів чи охоронців, та підбір вільного місця на парковці. Програмний продукт повинен бути простим в наданні послуг та повинен бути доступним із будь-яких пристроїв із можливістю виходу в мережу інтернет.

Програма призначена для запуску в середовищі Windows на ПК з характеристиками наведеними у табл. 3.1.

Таблиця 3.1 – Мінімальні апаратні вимоги до ПК

Характеристика	Значення
CPU	не нижче Intel Core i3
RAM	4 ГБ
HDD	Не нижче 30 ГБ

Систему моніторингу та збору статистичних даних можна викликати будь-якими доступними засобами операційної системи за допомогою веб-браузеру.

3.4 Висновки за розділом 3

У даному розділі дана постановка задачі також розглянута розробка моделі реляційної бази даних програмної системи моніторингу та збору статистичних даних. Описано характеристики програми, вхідні та вихідні дані розробленої системи.

4 КЕРІВНИЦТВО ПРОГРАМІСТА

4.1 Призначення й умови використання

Розроблена автоматизована система моніторингу та збору статистичних даних призначена для використання на парковках в торгівельних центрах та на звичайних парковках або автостоянках. При використанні цього програмного забезпечення спрощується облік автомобілів на парковці для операторів чи охоронців, та підбір вільного місця на парковці. Програмний продукт повинен бути простим в надані послуг та повинен бути доступним із будь-яких пристроїв із можливістю виходу в мережу інтернет.

Програма призначена для запуску в середовищі Windows на ПК з характеристиками наведеними у табл. 4.1.

Таблиця 4.1 – Мінімальні апаратні вимоги до ПК

Характеристика	Значення
CPU	не нижче Intel Core i3
RAM	4 ГБ
HDD	Не нижче 30 ГБ

4.2 Вхідні та вихідні дані

Розроблена автоматизована система моніторингу та збору статистичних даних призначена для використання на парковках в торгівельних центрах та на звичайних парковках або автостоянках. У якості вхідних даних виступають дані, необхідні для реєстрації клієнта та його ТЗ (бріф-анкета на клієнта, номер державної реєстрації ТЗ, час зберігання який планується), ця інформація зберігається в базі даних. Вихідними даними є інформація стосовно наявного місця для паркування відповідно до вимог користувача, час зберігання який є фактичним, знижка на використання паркування, час затримки оплати,

статистичні данні (кількість відвідувачів на ТЗ у день акційних пропозицій та у звичайний день).

4.3 Повідомлення

Для забезпечення коректної роботи програми користувачеві у разі будь-яких некоректних дій видаються попередження або повідомлення про помилки. Усі типи повідомлень, що видаються, представлені в табл. 4.2.

Таблиця 4.2 – Діалогові вікна

Повідомлення	Умова появи	Реакція користувача
Закінчити роботу з програмою?	Виникає під час спроби закрити програму	Натиснути “Yes”, якщо потрібно закінчити роботу з програмою, інакше натиснути “No” або закрити діалогове вікно
Помилка доступу до файлу!	Виникає при спробі запису вже відкритий файл.	Переконайтеся, що файли, призначені для запису програми, не відкриті в середовищі Windows, переконається, що директорія, в якій проводиться запис файлу, не захищена від запису.
Server error	Помилка серверу»	
Server started on port	Сервер запущений під портом	

4.4 Висновки за розділом 4

У четвертому розділі були розглянуті основні призначення та умови застосування автоматизованої системи. Також було розглянуто виняткові ситуації, які можуть виникати як при заповненні форм, так і при виконанні програм.

5 КЕРІВНИЦТВО ОПЕРАТОРА

5.1 Призначення програми й умови виконання програмного продукту

Розроблена автоматизована система моніторингу та збору статистичних даних призначена для використання на парковках в торгівельних центрах та на звичайних парковках або автостоянках. При використанні цього програмного забезпечення спрощується облік автомобілів на парковці для операторів чи охоронців, та підбір вільного місця на парковці. Програмний продукт повинен бути простим в надані послуг та повинен бути доступним із будь-яких пристроїв із можливістю виходу в мережу інтернет.

Програма призначена для запуску в середовищі Windows на ПК з характеристиками наведеними у табл. 5.1.

Таблиця 5.1 – Мінімальні апаратні вимоги до ПК

Характеристика	Значення
CPU	не нижче Intel Core i3
RAM	4 ГБ
HDD	Не нижче 30 ГБ

5.2 Виконання програмного продукту

Головний екран входу (рис. 5.1) призначений для авторизації та реєстрації користувачів які користуються програмною системою моніторингу та збору статистичних даних.

Для авторизації необхідно ввести email або моб. телефон та пароль та натиснути кнопку «Увійти».

Також на початковому екрані можливо перейти до відновлення паролю, або якщо ви новий клієнт то можете зареєструватися.

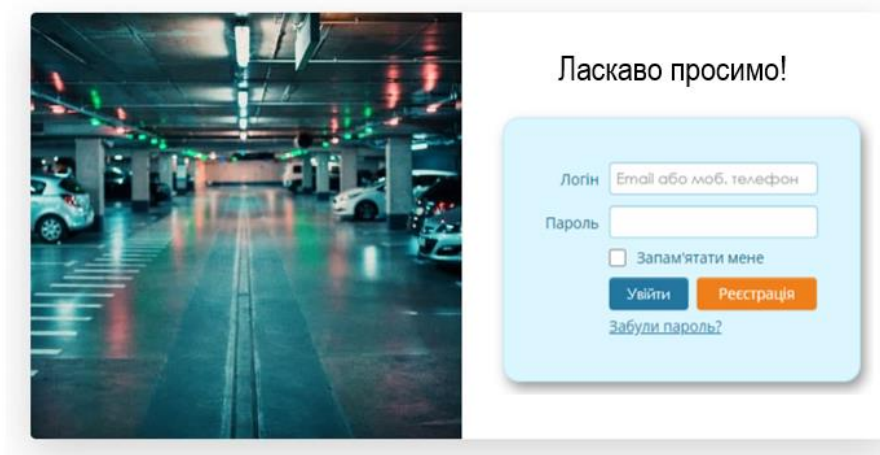


Рисунок 5.1 – Головний екран входу

Для реєстрації у системі (рис. 5.2) треба ввести у відповідні поля «Ім'я», «Прізвище», «електронну адресу», «Пароль» і «Номер мобільного телефону» та натиснути кнопку «Зареєструватися».

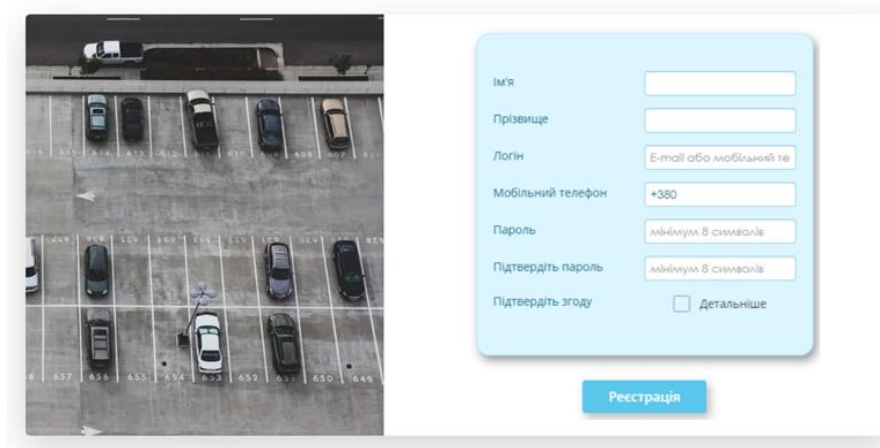


Рисунок 5.2 – Форма реєстрації нового користувача у системі

Після завершення реєстрації потрібно буде зробити наступне:

- якщо при реєстрації Ви вказали свій E-mail, то на Вашу електронну пошту, вказану при реєстрації, прийде лист з посиланням для активації. Потрібно натиснути на посилання у тексті листа для завершення реєстрації у кабінеті;

Рисунок 5.4 – Діаграма кількості відвідувачів по днях

Також у програмному продукті можна переглянути вартість години паркування, якщо вона не безкоштовна (рис. 5.5), та її сплатити. З правилами паркування та стоянки транспортних засобів можна також ознайомитись у ПП (рис. 5.6).

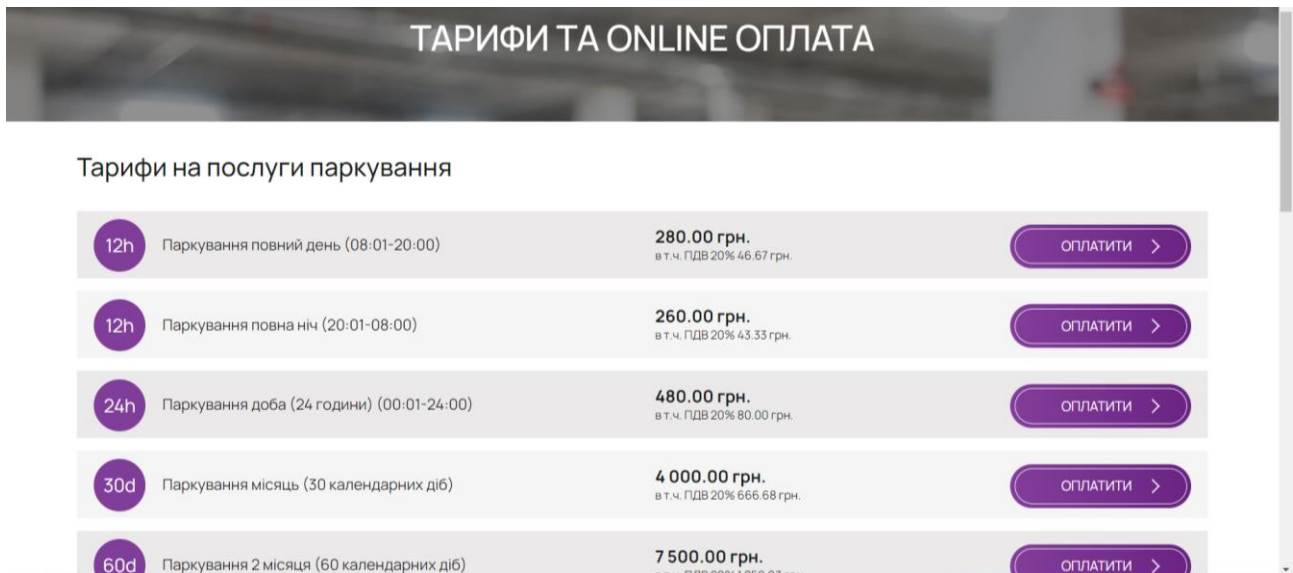


Рисунок 5.5 – Інтерфейсне вікно "Тарифи та Online оплата"

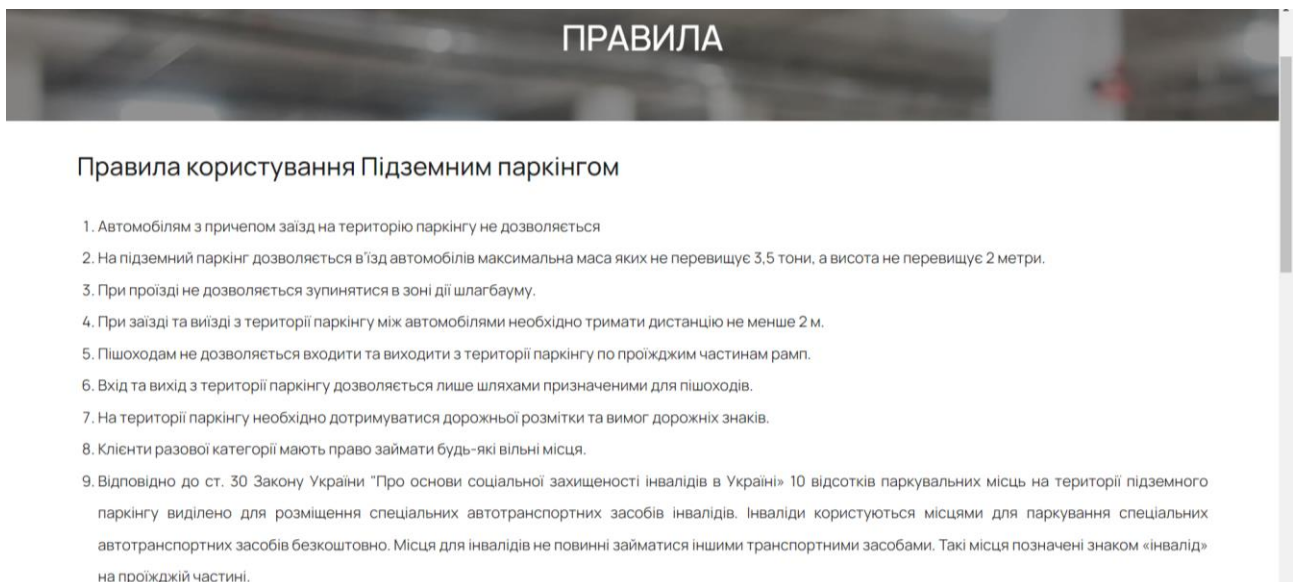


Рисунок 5.6 – Інтерфейсне вікно "Правила"

5.3 Висновки за розділом 5

У даному розділі наведено коротке керівництво оператора по роботі з графічним інтерфейсом програми, надано приклади екранних форм програми. Також наведено мінімальні вимоги до обладнання, на якому буде виконуватись програма.

ВИСНОВКИ

В результаті виконання кваліфікаційної роботи бакалавра було розроблено автоматизовану систему моніторингу та збору статистичних даних котра призначена для використання на паркуваннях в торгівельних центрах та на звичайних паркувальних майданчиках або автостоянках. Подальший розвиток системи – створення мобільного додатку для клієнтів, платежі у системі, розширення клієнтського та операторського функціонала у існуючих програмних продуктах.

Мною було вивчено механізм роботи паркувальних систем, покращено навички проектування та програмування великих і складних систем, реалізована автоматизована система моніторингу та збору статистичних даних про паркування.

Розроблена автоматизована система є кросплатформеною, надійною та продуктивною, максимально економічною з точки зору необхідного апаратного забезпечення, для створення використано новітні технології розробки ПЗ.

Програма має простий та досить зручний інтерфейс. Систему можна уніфікувати у залежності від основних цілей клієнта та зробити більш спрямованим на результат під умови використання через велику гнучкість програмного продукту.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Мобільне паркування [Електрон. ресурс]. – Режим доступу: <https://ktps.kyiv.ua/services/pay/mobile-parking>.
2. Автоматизация парковок [Електрон. ресурс]. – Режим доступу: <https://expertsolution.com.ua/avtomatizacija-parkovok>.
3. Парковочное оборудование и парковочные системы производства компании СЭА [Електрон. ресурс]. – Режим доступу: <https://www.sea.com.ua/parkovochnoe-oborudovanie/>.
4. Parkinglogix [Electronic resource]. – Access mode: <https://parkinglogix.com/>.
5. Simple Parking [Electronic resource]. – Access mode: <http://www.simpleairportparking.com/>.
6. SERVIO PARKING для автоматизації парковок [Електрон. ресурс]. – Режим доступу: <https://expertsolution.com.ua/uk/servio-parking>.
7. AllStojanka [Електрон. ресурс]. – Режим доступу: <https://studwood.net/2339644/informatika/allstojanka>.
8. Парковка, стоянка в Борисполі [Електрон. ресурс] / — Режим доступу: <https://bac.in.ua/>.
9. Групповое табло системы AS101 ProPark - Автомат-Паркинг [Електрон. ресурс]. – Режим доступа: <https://automat-parking.ru>.
10. Граф Х. Создание веб-сайтов с помощью Joomla! 1.5 [Текст] / Х. Граф. – М. : Вильямс, 2009. – 304 с.
11. R commander (Rcmdr) [Electronic resource]. – Access mode: <https://www.rcommander.com/>.
12. Субботін С. О. Інтелектуальні системи : навчальний посібник / С. О. Субботін, А. О. Олійник ; під заг. ред. проф. С. О. Субботіна. – Запоріжжя : ЗНТУ, 2014. – 216 с.
13. Олійник, А. О. Інтелектуальний аналіз даних / А. О. Олійник, С. О. Субботін, О. О. Олійник. – Запоріжжя : ЗНТУ, 2012. – 278 с.

14. Маркотт И. / Отзывчивый веб-дизайн [Текст] / Итан Маркотт. – М.: Манн, Иванов и Фербер, 2012. – 176 с.
15. Нильсен Я.Х. Веб-дизайн [Текст] / Якоб Х. Нильсен. – М.: Символ-Плюс, 2006. – 512 с.
16. Нильсен Я.Х. Дизайн Web-страниц. Анализ удобства и простоты использования 50 узлов [Текст] / Я. Нильсен, М. Тахир. – М. : Издательский дом «Вильямс», 2002. – 326 с.
17. Томлинсон Т. CMS Drupal 7 : руководство по разработке системы управления веб-сайтом [Текст] / Т. Томлинсон. – М. : Вильямс, 2011. – 560 с.
18. Гото К. Веб-редизайн: книга Келли Гото и Эмили Котлер [Текст] / Келли Гото, Эмили Котлер. – 2-е изд. – СПб: Символ-Плюс, 2006. – 416 с.
19. Пономаренко В. С. Основы технологий Internet : учебное пособие [Текст] / В. С. Пономаренко, С. В. Минухин, И. А. Торохтий. – Х. : Изд. ХГЭУ, 2001. – 256 с.
20. MySQL [Электрон. ресурс]. – Режим доступа: <http://www.mysql.ru/>.
21. Laravel [Electronic resource]. – Access mode: <https://laravel.com/docs/5.1/releases>.
22. Laravel. Blade Part 1 [Electronic resource]. – Access mode: https://www.youtube.com/watch?v=jJKM2rc_pO8.
23. Django makes it easier to build better web apps more quickly and with less code [Electronic resource]. – Access mode: <https://www.djangoproject.com/>.
24. Django [Electronic resource]. – Access mode: <https://habr.com/hub/django/>.
25. Node.JS [Electronic resource]. – Access mode: <https://habr.com/hub/nodejs/>.
26. Шлее М. Qt 5.10 Профессиональное программирование на C++. /Макс Шлее. – СПб.: БХВ–Петербург, 2018. – 1072 с.
27. Зандстра М. PHP. Объекты, шаблоны и методики программирования [Текст] / М. Зандстра. – К. : Apress, 2011. – 560 с.

28. Хьюгс С. РНР. Руководство разработчика [Текст] / С. Хьюгс, А. Змиевский. – К. : ДиаСофт, 2004.–384 с.
29. Плюсы и минусы программирования на Java [Электрон. ресурс]. – Режим доступа: <https://cutt.ly/JbKj3pq>.
30. Сьерра, К. Изучаем Java. Лекции и упражнения / К. Сьерра, Б. Бэйтс. – М. : Эксмо, 2017. – 720 с.
31. JetBrains .NET IDE hits RTM [Electronic resource] – Access mode: <https://blog.jetbrains.com/dotnet/2017/08/03/rider-2017-1-jetbrains-net-ide-hits-rtm/>

ДОДАТОК А
Технічне завдання

Вступ

Автоматизована система моніторингу та збору статистичних даних призначена для використання на парковці в торгівельних центрах та на звичайних паркувальних майданчиках або автостоянках.

A.1 Підстави для розробки

Підставою для розробки є завдання на дипломний проєкт бакалавра на тему “Розробка програмного забезпечення моніторингу парковки”, затверджене наказом Національного університету «Запорізька політехніка» № 102 від 27 квітня 2022.

A.2 Призначення розробки

Автоматизована система призначена для ведення обліку, моніторингу та збору статистичних даних про транспортні засоби на паркувальних майданчиках, зберігання, редагування та видалення даних про транспортні засоби та їх власників.

A.3 Постановка задачі

Необхідно розробити автоматизовану систему моніторингу парковок, яка повинна складатися з адміністративної панелі та панелі оператора. Адміністративна панель повинна містити всі функції, за допомогою яких ви можете керувати системою, тобто керувати операторами, паркінгами, транспортними засобами, реєструвати користувачів та транспортні засоби самим адміністратором. Панель оператора повинна складатися з реєстрації користувачів, реєстрації транспортного засобу, пошуку місця для паркування.

Сама автоматизована система повинна включати дані про схеми парковок, кількість місць на парковці та наявність вільних місць.

A.4 Основні вимоги до програми

A.4.1 Вимоги до функціональних характеристик

Вхідною інформацією є відомості про моніторинг та збір статистичних даних про паркування в торговельних центрах та на звичайних паркувальних майданчиках або автостоянках.

Основними функціями системи є:

- ведення електронної БД клієнтів та їх транспортних засобів;
- зберігання отриманих даних до БД на сервері;
- формування статистики;
- генерація звітної документації;

A.4.2 Вимоги до надійності

Автоматизована система моніторингу парковок повинна відповідати таким вимогам до надійності:

- захищати автоматизовану систему від несанкціонованого доступу;
- захищати автоматизовану систему від помилкових дій користувача;
- здійснювати обробку виняткових ситуацій.

A.4.3 Умови експлуатації

Експлуатація автоматизованої системи відбувається згідно експлуатаційної документації на розроблену систему, яка відповідає стандартам і містить інформацію, необхідну для освоєння і експлуатації системи.

.

A.5 Вимоги до програмної документації

Склад програмної документації повинен включати в себе:

- технічне завдання;
- керівництво програміста;
- керівництво оператора;
- опис програми.

A.6 Основні обмеження на установку і використання програмного продукту

Автоматизована система моніторингу та збору статистичних даних призначена для використання на паркуваннях в торгівельних центрах та на звичайних паркуваннях або автостоянках. При використанні цього програмного забезпечення спрощується облік автомобілів на паркуванні для операторів чи охоронців, та підбір вільного місця на парковці. Програмний продукт повинен бути простим в наданні послуг та повинен бути доступним із будь-яких пристроїв із можливістю виходу в мережу інтернет.

Програма призначена для запуску в середовищі Windows 10 на ПК з характеристиками наведеними у табл. А.1.

Таблиця А.1 – Мінімальні апаратні вимоги до ПК

Характеристика	Значення
CPU	не нижче Intel Core i3
RAM	4 ГБ
HDD	Не нижче 30 ГБ

Систему моніторингу та збору статистичних даних можна викликати будь-якими доступними засобами операційної системи за допомогою веб-браузеру.

ДОДАТОК Б
Опис програмного продукту

Б.1 Повідомлення

Для забезпечення коректної роботи програми користувачеві у разі будь-яких некоректних дій видаються попередження або повідомлення про помилки. Усі типи повідомлень, що видаються, представлені в табл. Б.1.

Таблиця Б.1 – Діалогові вікна

Повідомлення	Умова появи	Реакція користувача
Закінчити роботу з програмою?	Виникає під час спроби закрити програму	Натиснути “Yes”, якщо потрібно закінчити роботу з програмою, інакше натиснути “No” або закрити діалогове вікно
Помилка доступу до файлу!	Виникає при спробі запису вже відкритий файл.	Переконайтеся, що файли, призначені для запису програми, не відкриті в середовищі Windows, переконається, що директорія, в якій проводиться запис файлу, не захищена від запису.
Server error	Помилка серверу»	
Server started on port	Сервер запущений під портом	

Б.4 Структурна схема програмного продукту

Була розроблена модель реляційної бази даних програмної системи контролю за паркуванням. Модель бази даних показана на рисунку Б.1

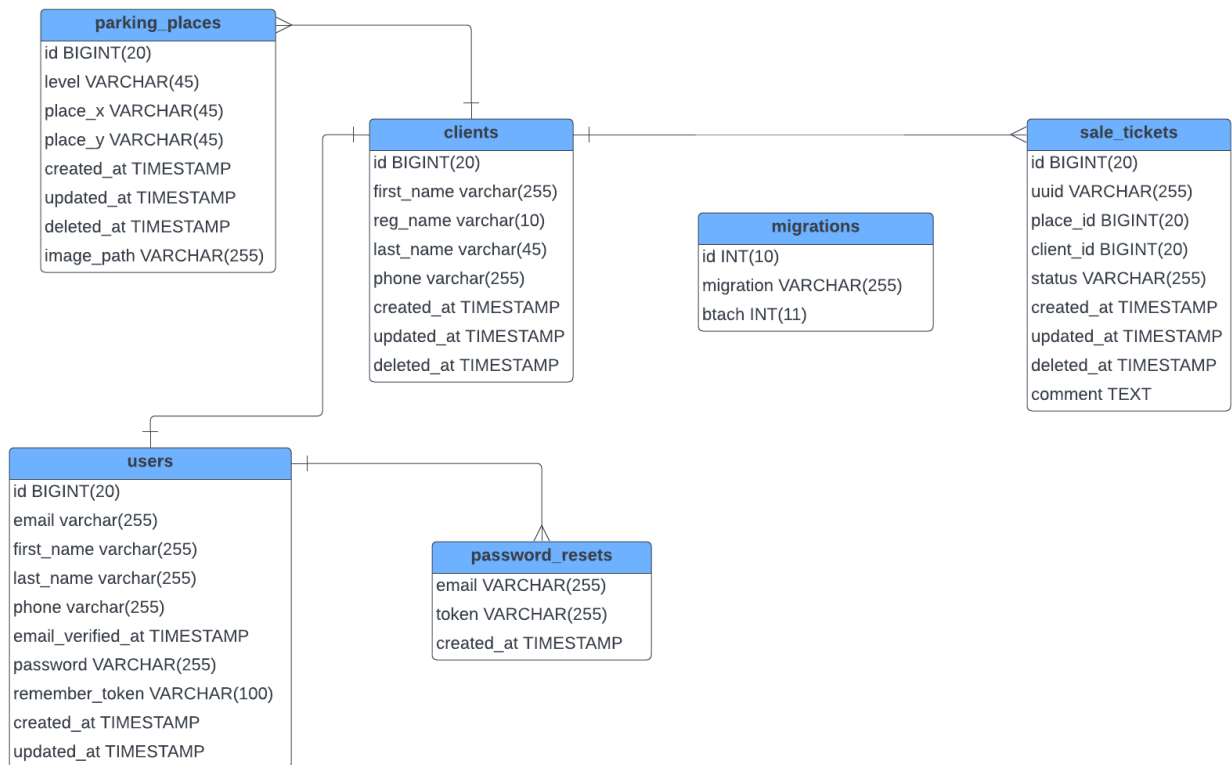


Рисунок Б.1 – Схема бази даних

Parking_places – це список справжніх місць на парковці, які визначаються за рівнями (column – level) та за координатами x та y (column – place_x & place_y).

Sale_tickets – це список існуючих сесій парковок, де place_id – foreign key, а client_id – nullable foreign key (клієнт може бути не зареєстрований у системі).

Clients – це список зареєстрованих клієнтів парковки.

Users – це список операторів або адміністраторів системи.

Таблиці migrations та password_resets – це службові таблиці системи.

Б.6 Приклади реалізації

Б.6.1 Головний екран входу

Головний екран входу (рис. Б.2) призначений для авторизації та реєстрації користувачів які користуються програмною системою моніторингу та збору статистичних даних.

Для авторизації необхідно ввести email або моб. телефон та пароль та натиснути кнопку «Увійти».

Також на початковому екрані можливо перейти до відновлення паролю, або якщо ви новий клієнт то можете зареєструватися.

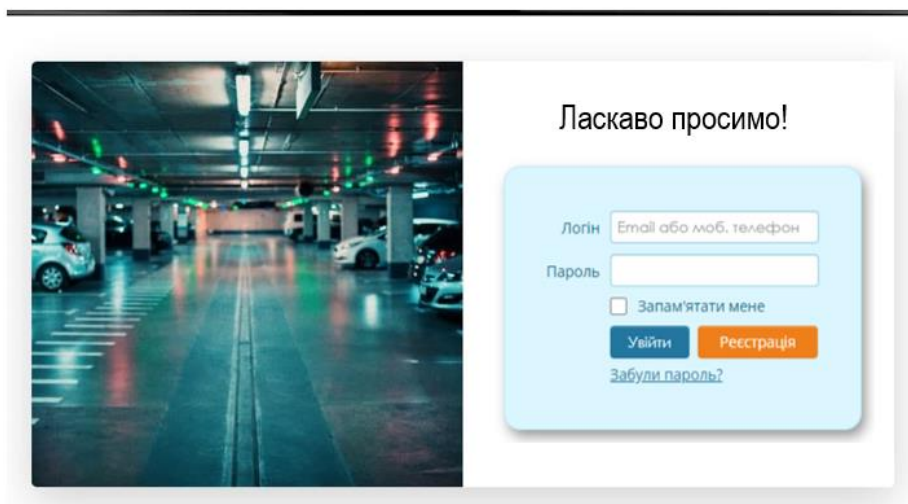


Рисунок Б.2 – Головний екран входу

Для реєстрації у системі (рис. Б.3) треба ввести у відповідні поля «Ім'я», «Прізвище», «електронну адресу», «Пароль» і «Номер мобільного телефону» та натиснути кнопку «Зареєструватися».

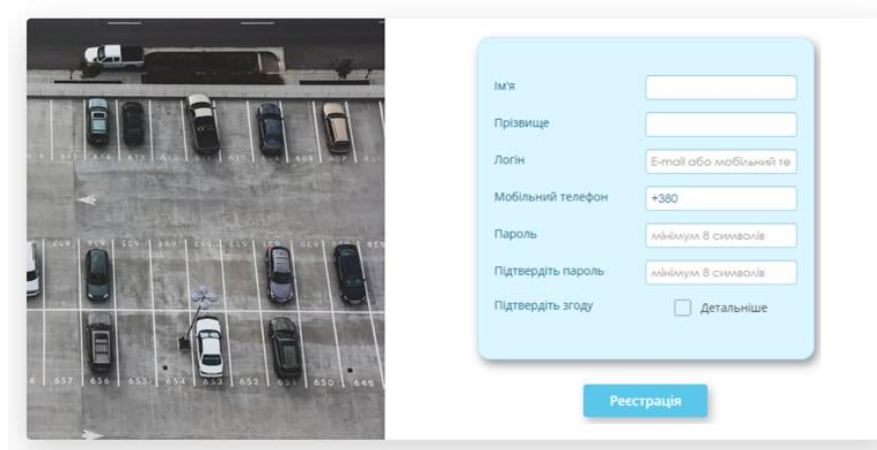


Рисунок Б.3 – Форма реєстрації нового користувача у системі

Після завершення реєстрації потрібно буде зробити наступне:

– якщо при реєстрації Ви вказали свій E-mail, то на Вашу електронну пошту, вказану при реєстрації, прийде лист з посиланням для активації.

Потрібно натиснути на посилання у тексті листа для завершення реєстрації у кабінеті;

– якщо при реєстрації Ви вказали свій номер мобільного телефону, то після першого входу в кабінет, з'явиться поле для вводу коду, який прийде вам у SMS.

На сторінці статистики оператор або керуючий може переглядати дату (рис. Б.4), час та кількість автомобілів які заїхали на парковку. На основі цих даних керуючий ТРЦ або ТЦ може зробити висновок про доцільність проведення рекламних акцій, спонсорських показів, розпродажів та проконтролювати пікові завантаження парковок (рис. Б.5).

СІЧЕНЬ 2022						
ПОНЕДІЛОК	ВІВТОРОК	СЕРЕДА	ЧЕТВЕР	П'ЯТНИЦЯ	СУБОТА	НЕДІЛЯ
					1 1/364 Новий рік	2 2/363
3 3/362	4 4/361	5 5/360	6 6/359	7 7/358 Різдво Крестове	8 8/357	9 9/356
10 10/355	11 11/354	12 12/353	13 13/352	14 14/351	15 15/350	16 16/349
17 17/348	18 18/347	19 19/346	20 20/345	21 21/344	22 22/343	23 23/342
24 24/341	25 25/340	26 26/339	27 27/338	28 28/337	29 29/336	30 30/335
31 31/334						

Рисунок Б.4 – Календар



Рисунок Б.5 – Діаграма кількості відвідувачів по днях

Також у програмному продукті можна переглянути вартість години паркування, якщо вона не безкоштовна (рис. Б.6), та її сплатити. З правилами

паркування та стоянки транспортних засобів можна також ознайомитись у ПП (рис. Б.7).

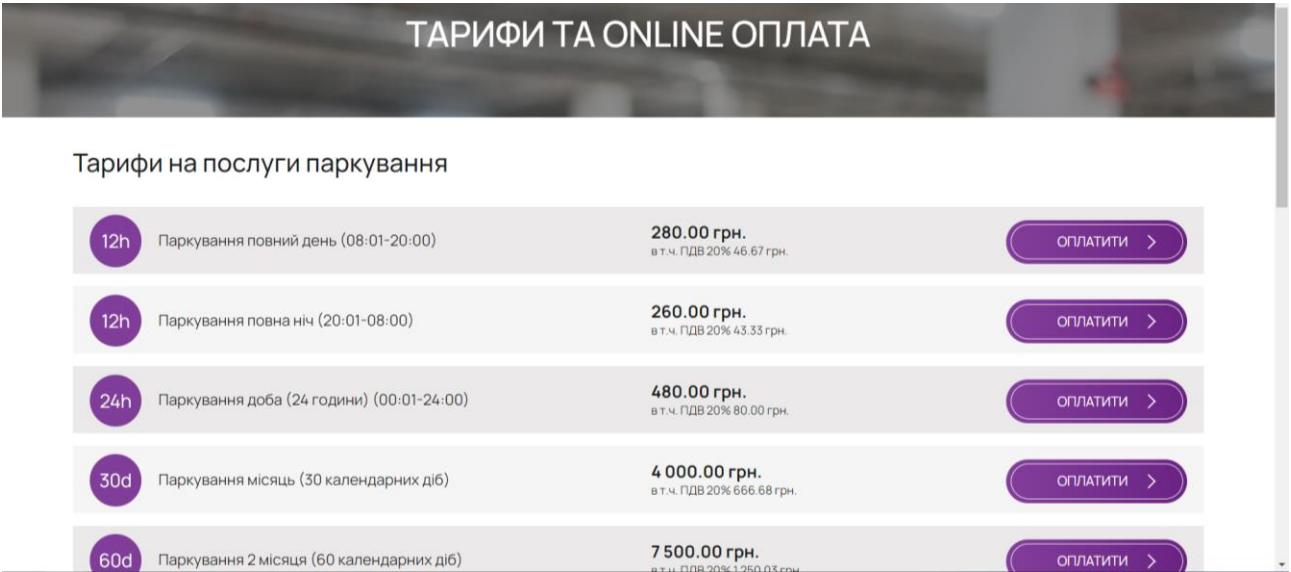


Рисунок Б.6 – Інтерфейсне вікно "Тарифи та Online оплата"

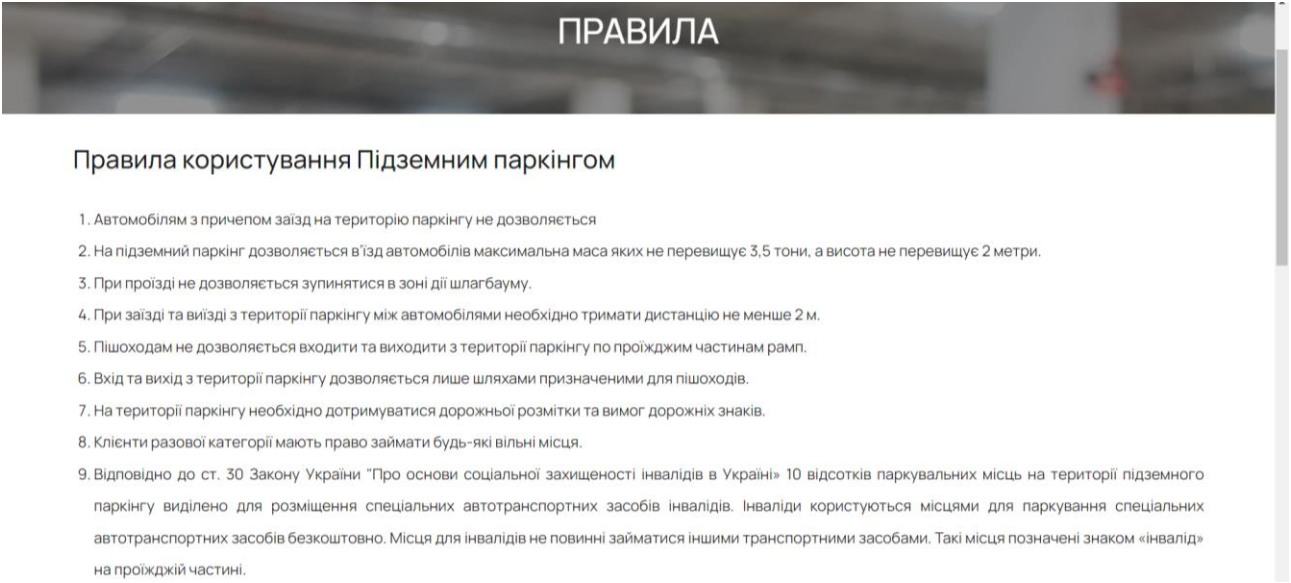


Рисунок Б.7 – Інтерфейсне вікно "Правила"

ДОДАТОК В
Текст програми

B.1 AnalyticController

```

<?php
namespace App\Http\Controllers;
use App\Models\Sale;
use App\Models\ParkingPlace;
use Illuminate\Support\Facades\Auth;
use Illuminate\Support\Facades\DB;
class AnalyticController extends Controller
{
    private $chartColors = [
        'red' => 'rgb(255, 99, 132)',
        'orange' => 'rgb(255, 159, 64)',
        'yellow' => 'rgb(255, 205, 86)',
        'green' => 'rgb(75, 192, 192)',
        'blue' => 'rgb(54, 162, 235)',
        'purple' => 'rgb(153, 102, 255)',
        'grey' => 'rgb(201, 203, 207)'
    ];
    public function topParkingPlaces()
    {
        $parkingPlaces = ParkingPlace::query()
            ->toBase()
            ->select([
                'parking_places.id',
                DB::raw('SUM(parking_places.amount) as cnt'),
            ])
            ->groupBy('parking_places.id')
            ->get();
        $chartData = [
            'labels' => [],
            'datasets' => [
                [
                    'data' => [],

```

```

        'backgroundColor' => [],
        'label' => 'Самые популярные парковочные
места за все время',
    ]
],
];
foreach ($parkingPlaces as $product) {
    $chartData['labels'][] = $product->product_name;
    $chartData['datasets'][0]['data'][] = $product-
>cnt;

    $chartData['datasets'][0]['backgroundColor'][] =
$this->randomColor();
}
return view('analytic.topSales', [
    'chartData' => $chartData,
]);
}
public function moneyFlow()
{
    $parkingPlaces = Sale::query()
        ->toBase()
        ->select([
            DB::raw('DATE_FORMAT(sale_at, "%Y-%m-%d") as
sale_day'),
            DB::raw('SUM(total_parking_currency) as
money'),
        ])
        ->where('shop_id', Auth::user()->shop->id)
        ->whereBetween('sale_at', [now()->firstOfMonth(),
now()->lastOfMonth()])
        ->groupBy('sale_day')
        ->get()
        ->keyBy('sale_day');
    $chartData = [
        'labels' => [],

```

```

        'datasets' => [
            [
                'data' => [],
                'backgroundColor' => $this->chartColors['red'],
                'borderColor' => $this->chartColors['red'],
                'label' => 'Количество посетителей в день',
                'fill' => false,
            ]
        ],
    ];
    $iterator = now()->firstOfMonth();
    while (!$iterator->isLastOfMonth()) {
        $chartData['datasets'][0]['data'][] =
            $parkingPlaces[$iterator->format('Y-m-d')->money ?? 0;
        $chartData['labels'][] = $iterator->format('Y-m-d');
        $iterator->addDay();
    }
    return view('analytic.moneyFlow', [
        'chartData' => $chartData,
    ]);
}
private function randomColor()
{
    return $this->chartColors[array_rand($this->chartColors)];
}
}

```

B.2 ClientController

```

<?php
namespace App\Http\Controllers;
use App\Models\Client;
use App\Models\SaleTicket;
use Illuminate\Database\Eloquent\Builder;
use Illuminate\Http\RedirectResponse;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Auth;
use Illuminate\Support\Facades\DB;
use Illuminate\Validation\ValidationException;
use Throwable;

class ClientController extends Controller
{
    public function byPhone(Request $request)
    {
        $this->validate($request, [
            'phone' => 'required|string',
        ]);
        $client = Client::query()
            ->where('phone', $request->get('phone'))
            ->where('shop_id', Auth::user()->shop->id)
            ->first();
        return response()->json(['client' => $client]);
    }

    public function index()
    {
        if (request()->expectsJson()) {
            $query = Client::query();
            $query->where(function(Builder $query) {
                $query->where('name', 'like', '%'
request()->get('text') . '%')
                ->orWhere(

```

```

        DB::raw('REPLACE(phone, " ", "'')'),
        'like',
        '%' . str_replace(' ', '',
\request()->get('text') . '%')
    )
    ->orWhere('created_at', 'like', '%' .
request()->get('text') . '%');
    });
    $query->where('shop_id', Auth::user()->shop->id);
    $query->orderBy('created_at', 'DESC');
    $query = $query->toBase();
    $query->addSelect([
        'sales_amount' => SaleTicket::query()
            ->select(DB::raw('count(*)'))
            ->where(['client_id' =>
DB::raw('clients.id')])
    ]);
    $paginator = $query->paginate();
    return response()->json([
        'rows' => $paginator->items(),
        'pagination' => pagesList($paginator),
    ]);
}
return view('client.index', [
]);
}
public function create(Request $request)
{
    return view('client.create');
}
/**
 * @param Request $request
 * @return RedirectResponse
 * @throws ValidationException
 * @throws Throwable

```

```

*/
public function store(Request $request)
{
    $this->validate($request, [
        'name' => 'required|string|min:2|max:255',
        'phone' => ['required', 'string', 'regex:/^\+380
\d{2} \d{3} \d{2} \d{2}$/'],
    ]);
    DB::transaction(function() use ($request) {
        $client = new Client();
        $client->forceFill([
            'shop_id' => Auth::user()->shop->id,
            'name' => $request->get('name'),
            'phone' => $request->get('phone'),
        ]);
        $client->save();
    });
    return redirect()
        ->route('clients.index')
        ->with('success', 'Клиент добавлен');
}
public function edit(Client $client)
{
    if ($client->shop_id != Auth::user()->shop->id) {
        abort(403);
    }
    return view('client.edit', [
        'client' => $client,
    ]);
}
public function update(Client $client, Request $request)
{
    if ($client->shop_id != Auth::user()->shop->id) {
        abort(403);
    }
}

```

```

        $this->validate($request, [
            'name' => 'required|string|min:2|max:255',
            'phone' => ['required', 'string', 'regex:/^\+380
\d{2} \d{3} \d{2} \d{2}$/'],
        ]);
        DB::transaction(function() use ($request, $client) {
            $client->forceFill([
                'shop_id' => Auth::user()->shop->id,
                'name' => $request->get('name'),
                'phone' => $request->get('phone'),
            ]);
            $client->save();
        });
        return redirect()
            ->route('client.index')
            ->with('success', 'Информация сохранена');
    }

    public function destroy(Client $client)
    {
        $client->delete();
        return redirect()->back()->with('success', 'Клиент
успешно удален');
    }
}

```

B.3 RegisterController

```

<?php
namespace App\Http\Controllers\Auth;
use App\Http\Controllers\Controller;
use App\Models\User;
use App\Providers\RouteServiceProvider;
use Illuminate\Foundation\Auth\RegistersUsers;

```

```

use Illuminate\Support\Facades\Hash;
use Illuminate\Support\Facades\Validator;

class RegisterController extends Controller
{
    /**
     |-----
-----
    | Register Controller
    |-----
-----
    |
    | This controller handles the registration of new users
as well as their
    | validation and creation. By default this controller
uses a trait to
    | provide this functionality without requiring any
additional code.
    |
    */
    use RegistersUsers;
    /
    * Where to redirect users after registration.
    *
    * @var string
    */
    protected $redirectTo = RouteServiceProvider::HOME;
    /
    * Create a new controller instance.
    *
    * @return void
    */
    public function __construct()
    {
        $this->middleware('guest');
    }

```

```

    }
    /
    * Get a validator for an incoming registration request.
    *
    * @param array $data
    * @return \Illuminate\Contracts\Validation\Validator
    */
protected function validator(array $data)
{
    return Validator::make($data, [
        'first_name' => ['required', 'string', 'max:255',
'min:2'],
        'last_name' => ['required', 'string', 'max:255',
'min:2'],
        'email' => ['required', 'string', 'email',
'max:255', 'unique:users'],
        'password' => ['required', 'string', 'min:8',
'confirmed'],
        'phone' => ['required', 'string', 'regex:/^\+380
\d{2} \d{3} \d{2} \d{2}$/'], 'unique:users,phone',]
    ], [
        'phone.regex' => 'Введите полностью ваш номер
телефона',
        'phone.unique' => 'Данный номер телефона уже
используется',
    ]);
}

/
* Create a new user instance after a valid registration.
*
* @param array $data
* @return \App\Models\User
*/
protected function create(array $data)

```

```
{  
  return User::create([  
    'first_name' => $data['first_name'],  
    'last_name' => $data['last_name'],  
    'email' => $data['email'],  
    'password' => Hash::make($data['password']),  
    'phone' => $data['phone'],  
  ]);  
}  
}
```

ДОДАТОК Г
Слайди презентації

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЗАПОРІЗЬКА ПОЛІТЕХНІКА»

Кафедра програмних засобів

ДИПЛОМНА КВАЛІФІКАЦІЙНА РОБОТА

РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ МОНІТОРИНГУ ПАРКОВКИ
SOFTWARE DEVELOPMENT FOR PARKING MONITORING

Виконав

ст. гр. КНТ-219сп

Керівник

к.т.н., доцент

А. С. Резніченко

С. К. Корнієнко

Рисунок Г.1 – Слайд № 1

Об'єкт дослідження – система моніторингу та збору статистичних даних для парковок.

Предмет дослідження – методи моніторингу та збору статистичних даних для парковок.

Мета роботи – створення програмної системи для моніторингу та збору статистичних даних для парковок.

Рисунок Г.2 – Слайд № 2

Основні показники сучасних програмних продуктів для парковки

	<u>Simple Parking</u>	SERVIO PARKING	<u>AllStojanka</u>	Парковка, стоянка в Борисполі	<u>Мікком AS101 ProPark</u>
1	2	3	4	5	6
Інтерфейс користувача	<u>Desktop-система</u>	<u>Desktop-система</u>	<u>Desktop-система</u>	<u>Web-система</u>	<u>Desktop-система</u>
Облік	+	+	+	+	+
Відображення схеми автостоянки	+	+	+	-	+
Приймання та оформлення коштів	+	+	-	-	+
Відображення конкретного ТЗ	+	+	+	-	+
Перевірка клієнтів	+	-	+	-	-
Реєстрація клієнтів	+	+	+	+	+
Контроль прострочення	+	-	+	-	-

3

Рисунок Г.3 – Слайд № 3

Порівняння фреймворків веб розробки

Параметр	<u>Laravel</u>	<u>Django</u>	Node.js	<u>Qt</u>
1	2	3	4	5
Об'єктно-орієнтовані бібліотеки	+	+	+	+
Мова програмування	<u>Python</u>	<u>Python</u>	<u>JavaScript</u>	C ++
<u>Юніт-тести</u>	+	+	-	-
Проблеми безпекою з	-	-	+/-	+/-
Вбудовані шаблони	+	+	-	+
Швидкість розробки	+	+	-	-
Інтеграція з <u>mail-сервісом</u>	+	-	-	-
Інтернаціоналізація	-	+	-	-
Розширювана система шаблонів з тегами та наслідуванням	+	+	-	-

4

Рисунок Г.4 – Слайд № 4

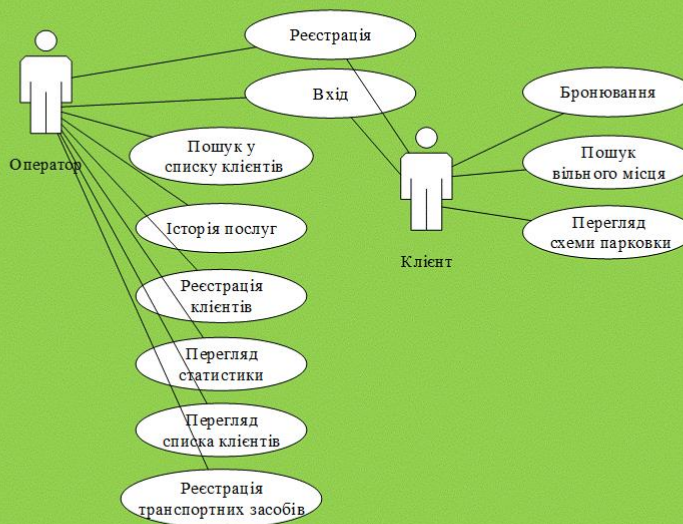
Порівняння мов програмування

Мова програмування	Швидкість обробки	Читабельність	Простота	GUI (графічний інтерфейс користувача)	Графіка (2D)	Графіка (3D)	Кроссплатформеність	Спец. процесор
C	8	6	2	3	5	8	7	10
C++	8	6	5	4	6	8	7	7
C#	7	7	5	6	6	7	2	—
Java	6	7	6	7	7	6	10	—
Python	2	5	10	8	10	1	10	—
VB.net	6	10	8	10	5	2	2	—
Assembler	10	—	—	—	—	—	—	5

5

Рисунок Г.5 – Слайд №5

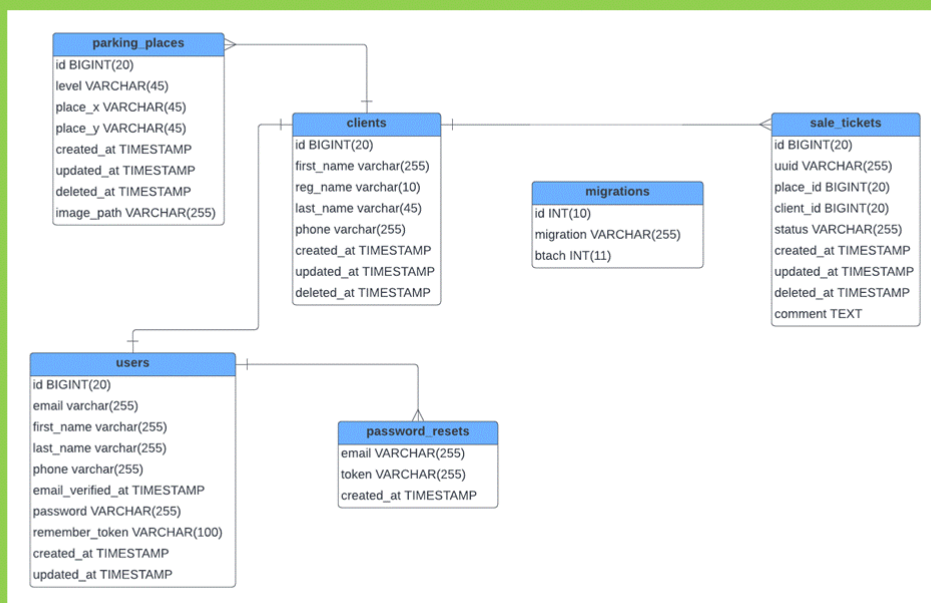
Діаграма варіантів використання



6

Рисунок Г.6 – Слайд № 6

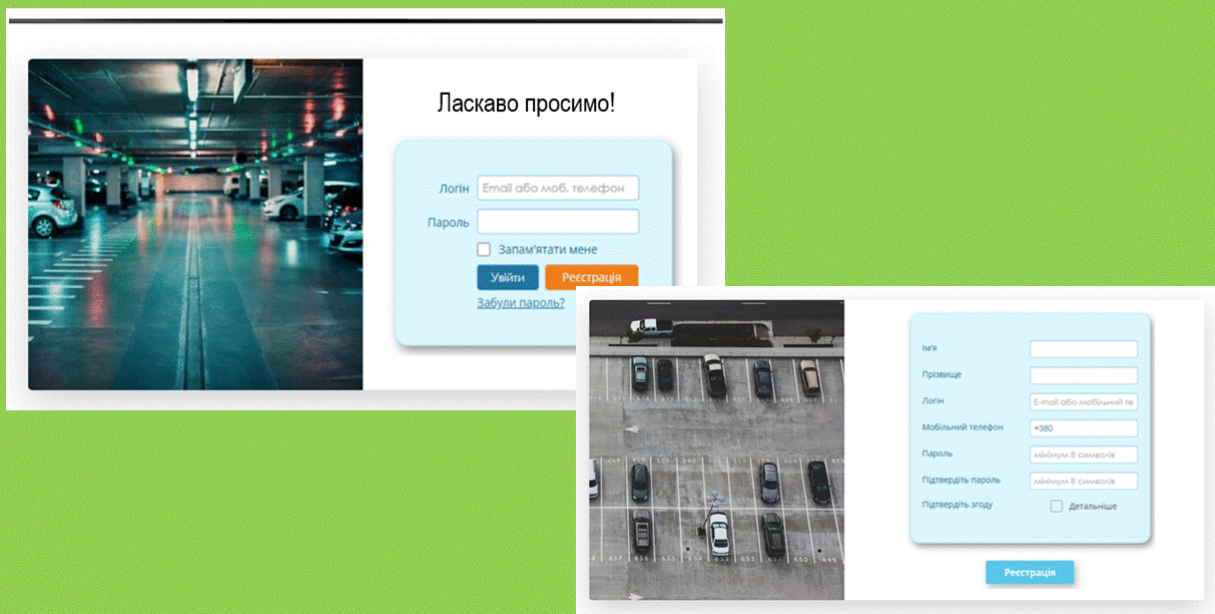
Схема бази даних



7

Рисунок Г.7 – Слайд № 7

Приклад роботи з програмою



8

Рисунок Г.8 – Слайд № 8

Приклад роботи з програмою



СІЧЕНЬ 2022						
ПОНЕДІЛОК	ВІВТОРОК	СЕРЕДА	ЧЕТВЕР	П'ЯТНИЦЯ	СУБОТА	НЕДІЛЯ
					1 12/04	2 12/05
3 12/06	4 12/07	5 12/08	6 12/09	7 12/10	8 12/11	9 12/12
10 12/13	11 12/14	12 12/15	13 12/16	14 12/17	15 12/18	16 12/19
17 12/20	18 12/21	19 12/22	20 12/23	21 12/24	22 12/25	23 12/26
24 12/27	25 12/28	26 12/29	27 12/30	28 12/31	29 12/31	30 12/31
31 12/31						

9

Рисунок Г.9 – Слайд № 9

Модель бази даних

ТАРИФИ ТА ONLINE ОПЛАТА

Тарифи на послуги паркування

12h	Паркування повний день (08:01-20:00)	280.00 грн. в т.ч. ПДВ 20% 46.67 грн.	ОПЛАТИТИ >
12h	Паркування повна ніч (20:01-08:00)	260.00 грн. в т.ч. ПДВ 20% 41.67 грн.	ОПЛАТИТИ >
24h	Паркування доба (24 години) (00:01-24:00)	480.00 грн. в т.ч. ПДВ 20% 76.67 грн.	ОПЛАТИТИ >
30d	Паркування місяць (30 календарних днів)	4 000.00 грн. в т.ч. ПДВ 20% 666.67 грн.	ОПЛАТИТИ >
60d	Паркування 2 місяця (60 календарних днів)	7 500.00 грн. в т.ч. ПДВ 20% 1 250.00 грн.	ОПЛАТИТИ >

ПРАВИЛА

Правила користування Підземним паркінгом

- Автомобілям з причепом заїзд на територію паркінгу не дозволяється
- На підземний паркінг дозволяється в'їзд автомобілів максимальна маса яких не перевищує 3.5 тонн, а висота не перевищує 2 метри.
- При проїзді не дозволяється зупинятися в зоні дії шлагбауму.
- При заїзді та виїзді з території паркінгу між автомобілями необхідно тримати дистанцію не менше 2 м.
- Пішоходам не дозволяється входити та виходити з території паркінгу по проїжджій частині рампи.
- Вхід та вихід з території паркінгу дозволяється лише шляхами призначеними для пішоходів.
- На території паркінгу необхідно дотримуватися дорожньої розмітки та вимог дорожніх знаків.
- Клієнти разової категорії мають право займати будь-які вільні місця.
- Відповідно до ст. 30 Закону України "Про основи соціальної захищеності інвалідів в Україні" 10 відсотків паркувальних місць на території підземного паркінгу виділено для розміщення спеціальних автотранспортних засобів інвалідів. Інваліди користуються місцями для паркування спеціальних автотранспортних засобів безкоштовно. Місця для інвалідів не повинні займатися іншими транспортними засобами. Такі місця позначені знаком «інвалід» на проїжджій частині.

10

Рисунок Г.10 – Слайд № 10

ВИСНОВКИ

В результаті виконання кваліфікаційної роботи бакалавра було розроблено автоматизовану систему моніторингу та збору статистичних даних котра призначена для використання на паркуваннях в торгівельних центрах та на звичайних паркувальних майданчиках або автостоянках. Подальший розвиток системи – створення мобільного додатку для клієнтів, платежі у системі, розширення клієнтського та операторського функціонала у існуючих програмних продуктах.

Мною було вивчено механізм роботи паркувальних систем, покращено навички проектування та програмування великих і складних систем, реалізована автоматизована система моніторингу та збору статистичних даних про паркування.

Розроблена автоматизована система є кросплатформеною, надійною та продуктивною, максимально економічною з точки зору необхідного апаратного забезпечення, для створення використано новітні технології розробки ПЗ.

Програма має простий та досить зручний інтерфейс. Систему можна уніфікувати у залежності від основних цілей клієнта та зробити більш спрямованим на результат під умови використання через велику гнучкість програмного продукту.

11

Рисунок Г.11 – Слайд № 11