



Хмельницький національний університет  
 Факультет програмування та комп'ютерних і телекомунікаційних систем  
 Кафедра комп'ютерних наук та інформаційних технологій


## КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

на тему Автоматизоване робоче місце адміністратора автомобільної парковки

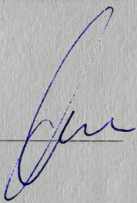
Галузь знань 12 – Інформаційні технології  
Шифр і назва галузі знань  
 Спеціальність 122 – Комп'ютерні науки  
Шифр і назва спеціальності  
 Освітня програма Комп'ютерні науки  
Назва освітньої програми

Виконав: студент 4 курсу, група КН-17-1  Б.С. Лавренюк  
Курс, група виконавця Підпис Ініціали, прізвище

Керівник: старший викладач кафедри КНІТ  Т.К. Скрипник  
Науковий ступінь, посада Підпис Ініціали, прізвище

Нормоконтроль: к.т.н., доцент кафедри КНІТ  Р.О. Багрій  
Науковий ступінь, посада Підпис Ініціали, прізвище

До захисту допускаю:

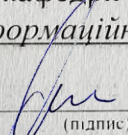
Зав. кафедри КНІТ, д.т.н., професор  О.В. Бармак  
Підпис Ініціали, прізвище

08 червня 2021 р.

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
 Факультет програмування та комп'ютерних і телекомунікаційних систем  
 Кафедра комп'ютерних наук та інформаційних технологій  
 Освітній ступінь бакалавр  
 Галузь знань 12 – Інформаційні технології  
 Спеціальність 122 – Комп'ютерні науки

ЗАТВЕРДЖУЮ

Завідувач кафедри комп'ютерних наук та інформаційних технологій

  
 (підпис)

д.т.н., професор О.В. Бармак

« 08 » лютого 2021 року

### ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА

1. Тема кваліфікаційної роботи бакалавра: «Автоматизоване робоче місце адміністратора автомобільної парковки»

2. Завдання видано студенту Лавренюку Богдану Сергійовичу  
 (прізвище, ім'я, по батькові)

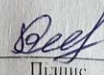
3. Керівник роботи ст. викладач кафедри КНІТ Скрипник Тетяна Казимирівна  
 (посада, прізвище, ім'я, по батькові)

4. Затверджено наказом університету від « 05 » лютого 2021 р. № 11

5. Зміст пояснювальної записки (перелік задач) та вихідні дані:

Метою роботи є розробка автоматизованого робочого місця адміністратора автомобільної парковки. При роботі з орендами парковочних місць слід забезпечити можливість реєстрації початку та завершення перебування автомобіля на паркомісці з вказанням запланованої та фактичної дати і часу. При роботі з наявним фондом парковочних місць слід враховувати тип паркомісць згідно їх призначення.

Виконавець: студент 4 курсу, група КН-17-1  
 Курс, група виконавця

  
 Підпис

Б.С. Лавренюк  
 Ініціали, прізвище

Керівник: старший викладач кафедри КНІТ  
 Науковий ступінь, посада

  
 Підпис

Т.К. Скрипник  
 Ініціали, прізвище

## Анотація

Тема кваліфікаційної роботи бакалавра: «Автоматизоване робоче місце адміністратора автомобільної парковки»

Виконавець кваліфікаційної роботи бакалавра: студент групи КН-17-1 Лавренюк Богдан Сергійович

Керівник кваліфікаційної роботи бакалавра: ст. викладач кафедри КНІТ Скрипник Тетяна Казимирівна

Кваліфікаційна робота бакалавра містить:

Пояснювальна записка				Кількість додатків
Сторінок	Рисунків	Таблиць	Джерел інформації	
56	31	12	38	4

Метою кваліфікаційної роботи бакалавра є розробка автоматизованого робочого місця адміністратора автомобільної парковки. Для розробки інформаційної системи на платформі ASP.NET було використано мову програмування C#, із використанням MVC шаблону та СКБД MS SQL Server.

Розроблена система призначена для адміністратора автомобільної парковки. Реалізоване автоматизоване робоче місце може використовуватися для підвищення ефективності виконання роботи адміністратора автомобільної парковки.

Напрямами практичного використання розробленої інформаційної системи визначено автоматизацію процесів орендування паркувальних місць та зменшення об'єму навантаження для адміністратора автомобільної парковки.

Ключові слова: адміністратор, парковка, інформаційна система, автоматизоване робоче місце, оренда.

Виконавець:

студент 4 курсу, група КН-17-1

Курс, група виконавця

  
Підпис

Б.С. Лавренюк

Ініціали, прізвище

## Зміст

Перелік скорочень .....	6
Вступ.....	7
Розділ 1	
Характеристика предметної області та постановка задачі .....	8
1.1 Аналіз предметної області .....	8
1.2 Аналіз існуючого програмного забезпечення предметної області .....	18
1.3 Аналіз сучасних засобів створення програмного забезпечення .....	23
1.4 Постановка задачі та вимоги до розробки інформаційної системи.....	25
Розділ 2	
Проектування інформаційної системи .....	26
2.1 Функціональна структура та бізнес-процеси системи .....	26
2.2 Інформаційна структура системи .....	28
2.3 Вибір засобів розробки інформаційної системи .....	33
2.3.1 Опис мови програмування .....	33
2.3.2 Вибір фреймворку.....	35
2.3.3 Вибір системи керування базами даних .....	36
Розділ 3	
Програмна реалізація інформаційної системи .....	39
3.1 Структура та функціональне призначення складових системи .....	39
3.2 Особливості реалізації складових системи .....	41
3.3 Тестування інформаційної системи .....	44
3.4 Інструкція користувача.....	50
3.5 Вимоги до розгортання інформаційної системи.....	55
Висновки .....	56
Перелік посилань.....	57
Додатки	

**Перелік скорочень**

<b>Скорочення, термін, позначення</b>	<b>Пояснення</b>
БД	База даних
АРМ	Автоматизоване робоче місце
ІТ	Інформаційні технології
КРБ	Кваліфікаційна робота бакалавра
СКБД	Система керування базами даних
CLR	Common Language Runtime
MS	Microsoft
MVC	Model-View-Controller

## Вступ

Початок використання людиною транспортних засобів датується 3500 роком до н.е. саме тоді було проведено одомашнення коней та розроблено перші механізми в яких їх використовували, а саме вози та колісниці [1]. В міру розвитку технологій відбувалося вдосконалення засобів переміщення людини. Сьогодні під транспортним засобом розуміється технічний пристрій призначений для переміщення людей та вантажів з одного місця в інше [2]. Незважаючи на велику кількість різноманітних видів транспорту (повітряний, водний, залізничний та інші) самим популярним є автомобільний транспорт [3].

Автомобільний транспорт забезпечує задоволення потреб населення та суспільного виробництва у перевезеннях пасажирів та вантажів за допомогою автомобільних транспортних засобів [4]. За його допомогою відбувається транспортування від 10 до 80% товарів. Основними перевагами автомобільного перевезення вантажів та пасажирів є:

- висока мобільність та маневреність;
- автономність роботи транспорту;
- висока швидкість;
- широка сфера застосування [5].

Значна популярність автомобільного транспорту призвела до збільшення кількості інфраструктурних будівель потрібних для його функціонування, одними з таких будівель є різноманітні парковки та паркінги. Використання людиною паркувального місця фактично означає його оренду, дату та час початку і закінчення котрої фіксує відповідальний працівник парковки.

Враховуючи кількість автомобільного транспорту, який кожного дня відвідує парковки та об'єм інформації який при цьому реєструється в системі необхідною стає автоматизації даного процесу. Найкращим рішенням для цього є реалізації веб-додатку призначено для адміністратора парковки.

## **Розділ 1**

### **Характеристика предметної області та постановка задачі**

#### **1.1 Аналіз предметної області**

Задоволення потреб населення та суспільного виробництва у перевезеннях пасажирів та вантажів забезпечується автомобільними транспортними засобами [6]. Основними сферами використання автомобільного транспорту є перевезення товарів торгівлі, будівництва, сільського господарства на невеликі відстані та забезпечення міжміських перевезень населення. На нього припадає перевезення до 80% всіх вантажів, які транспортуються сушею. Автомобільний транспорт також використовують для перевезення товарів з коротким терміном придатності та цінних речей, які потребуючих швидкої доставки. Сьогодні без автомобільного транспорту неможлива діяльність ні однієї галузі господарства [7].

Автомобільний транспорт є абсолютним лідером з кількості пасажирських перевезень, що відповідно призводить до збільшення кількості автомобілів. Для забезпечення ефективного функціонування автомобільного транспорту потрібна наявність відповідної інфраструктури, а саме: доріг, транспортних вузлів, ремонтних сервісів та паркувальних місць. Забезпечення потрібної кількості парковок та організації їх ефективної роботи є особливо актуальною в умовах постійного збільшення кількості автомобілів [8].

Парковка може представляти собою будівлю або спеціально обладнаний відкритий майданчик, який призначений для тимчасового чи постійного зберігання транспортних засобів, в основному автомобілів [9]. Однак окрім парковок існують також стоянки, хоча цих два терміни часто використовуються як тотожні насправді між ними є значні відмінності. Отже, парковка є місцем де можна залишити будь-яких транспортний засіб в неробочому стані на нетривалий час, при цьому плата за паркування не проводиться. Стоянка ж дозволяє залишити автовласнику свій транспорт на тривалий час і за певну плату перекласти відповідальність за його збереження на співробітника стоянки, до

стоянок також відносяться гаражі та паркінги. Також інколи поняття стоянка та парковка використовуються для характеризування платної території для паркування. Хоча автомобіліст платить диспетчеру за час протягом якого автомобіль перебуватиме на території, відповідальності за цілісність транспорту та речей які в ньому є диспетчер нести не буде, оскільки в даному випадку охороняється тільки відповідна територія., а не транспорт на ній. Основними відмінності між парковкою та стоянкою є:

- Парковка здійснюється безкоштовно на нетривалий проміжок часу.
- Протягом тривалої парковки транспорт не охороняється та ніхто не несе відповідальність за збереження автомобіля та речей які в ньому знаходяться.
- Стоянка це платна послуга, протягом надання якої автомобіль та майно що в ньому перебуває, знаходиться в цілковитій безпеці.
- Стоянка може проводитися тільки у відведених для цього місцях.
- Стоянка може проводитися протягом довго періоду часу;
- Стоянка може мати вигляд гаража або території під навісом, що дозволить вберегти автомобіль від атмосферних опадів [10].

Парковки та стоянки для автомобілів зазвичай розміщують у межах житлових комплексів та на території торгівельно-розважальних центрів, супермаркетів, готелів, вокзалів та аеропортів. Одним з прикладів парковок розміщених на території аеропортів є стоянка Skidata встановлена в Харківському міжнародному аеропорті. Дана система паркування максимально спрощує процес використання та оплати стоянки. Щоб орендувати місце на стоянці Skidata потрібно під'їхати до терміналу та натиснути на ньому відповідну кнопку, у результаті чого видається талон на якому зазначено час в'їзду на парковку (рисунок 1.1). Після отримання талону шлагбаум автоматично відкриється, талон потрібно зберігати до моменту виїзду з парковки [11]. На стоянці використовуються паркові термінали які виконують контроль в'їзду та виїзду автомобілів з її території, саме вони перевіряють наявність оплаченого місця в клієнта та вирішують чи впускати його на територію.



Рисунок 1.1 – Термінал стоянки Skidata

У великих містах набирає розмаху проектування і будівництво багаторівневих паркінгів, особливістю яких є наявність декількох рівнів на яких можна проводити паркування. Перша багаторівнева автостоянкою була відкрита в травні 1901 року компанією City & Suburban Electric Carriage Company в центральній частині Лондона. Стоянка була площею 19000 квадратних футів та могла розмістити 100 автомобілів на семи поверхах [12]. Багаторівневі стоянки в основному є самостійні будівлями призначеними виключно для паркування автомобілів. Проте існують і комбіновані варіанти побудови багаторівневих стоянок, такі споруди об'єднують житлові будинки і підземні паркінги.

Також існують конструкційні паркінги, які виготовлені із сталевих елементів, з'єднаних між собою. Особливостями конструкції із сталі є достатньо велика міцність та менша ціна, у порівнянні із іншими типами конструкцій, такими як бетонна та дерев'яна. Превагами сталевих конструкційні автомобільних стоянок є:

- дешевизна виготовлення та будівництва (збирання конструкції);
- легкість у обслуговуванні;
- швидкість монтажу (займає декілька днів);

- довговічність (гарантія на понад 50 років);
- транспортабельність;
- можливість легкого розширення конструкції [13].

Стельова плита автостоянки зі сталеву конструкцією, як правило, виготовляється з композитних матеріалів, таких як гофровані сталеві листи та бетон. Поверхня паркінгу на першому поверсі може залишатися оголеною або покритою епоксидною смолою або гудроном (рисунок 1.2).



Рисунок 1.2 – Автостоянка із сталеву конструкцією

В мегаполісах активно використовуються механізовані стоянки, які дозволяють ефективно використовувати невелику площу, відведену під стоянку автомобілів. Принцип роботи механізованої парковки достатньо простий: автомобіль під'їжджає до робочого майданчика, пасажери залишають салон, після чого спеціальний механізований пристрій відправляє авто на певне місце зберігання. Компактні парковки можуть застосовуватися як в новобудовах, будучи закладеними в план проектування, так і вбудовуватися в уже існуючі архітектурні об'єкти [14]. Механічні парковки можуть бути представлені наступними видами:

– баштові-системи вертикального завантаження автомобілів. Кожна з таких систем є багаторівною самонесучою конструкцією, яка складається з підйомника та маніпулятора. По боках підйомника знаходяться відсіки, в яких впорядковано розміщені транспортні засоби;

– роторні системи представляються напівавтоматичним обладнанням, що може забезпечити одночасну стоянку для 12 автомобілів. Розмір самої системи не перевищує розмірів парковки для двох автомобілів. Завантаження транспортного на платформу відбувається протягом 1,5-3 хвилин при цьому застосовується система вертикального підйому. Перевагою даного типу механізованої парковки є легкість монтування та демонтування (рисунок 1.3);

– парковка “пазл” – модель, сконструйована за системою матриці з порожньою коміркою. Це багаторівна рама, де позиціонування автомобіля виконується за рахунок пересування вгору і вниз піддонів верхнього рівня, а нижній рівень при цьому переміщається вправо/вліво для того, щоб звільнити потрібну комірку. Може вбудовуватися в будівлі або встановлюватися на відкритих майданчиках;

– стелажна – влаштована за типом автоматизованого складу. Це один з найпопулярніших варіантів. Система передбачає високу швидкість установки і подачі автомобіля (близько 2,5 хвилин). Стоянка складається з стелажа з відсіками для машин на піддонах одного або декількох підйомних пристроїв і візків переміщення авто в горизонтальному напрямку [15].

Також для оптимізації процесу паркування автомобілів використовуються пересувні платформи. Такі пристрої служать для того, щоб переміщати машини, що на них знаходяться. Вони можуть бути включені в систему паркінгу або використовуватися як самостійні механізми для стоянки автомобілів. Використання всіх типів механізованих парковок значно спрощує процес паркування автомобілів.

Користування механізованими стоянками повністю автоматизоване, однак на великих стоянках завжди працює один - двоє співробітників, які надають допомогу у разі виникнення проблем. До недоліків механізованих стоянок

можна віднести їх залежність від електропостачання. При вимкненні струму механізми стоянки перестануть працювати, що відповідно робить неможливою видачу транспорту водіям. Для вирішення даної проблеми на таких стоянках зазвичай використовують альтернативні джерела живлення (сонячні батареї)

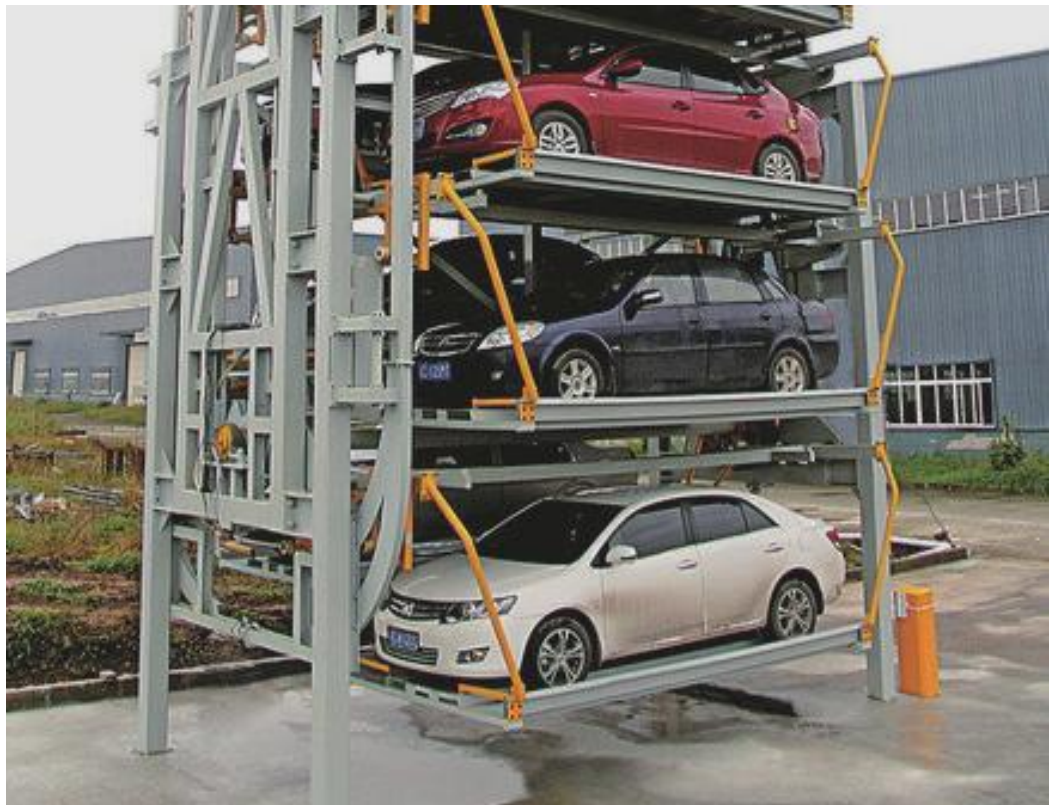


Рисунок 1.3 – Приклад роторної паркувальної системи

При в'їзді на парковку з автоматичною системою керування водій автомобіля має зупинитися перед системою, що розташована на в'їзді. В дану систему має бути попередньо внесена інформація про водія. Після зупинки йому потрібно натиснути на кнопку у результаті чого з'являється білет із штрих-кодом. Якщо система розпізнала водія має відбутися підняття шлагбауму після чого автомобіліст може зайняти вказане на білеті місце. Коли автомобіль заїде на територію паркінгу шлагбаум автоматично унеможливить в'їзд на неї, але якщо автомобіль не проїде з якоїсь причини його то на шлагбаумові встановленні відповідні сенсори, що не дозволять нанести пошкодження автомобілю .

При виїзді з парковки водій має показати білет із штрих-кодом, отриманий спочатку, касирові або відповідній установці. У результаті чого

проводиться сканування штрих-коду, далі у спеціальному вікні виводиться сума яку водієві потрібно оплатити. Після внесення потрібної кількості коштів відбудеться піднесення шлагбауму і водій зможе покинути місце стоянки (рисунок 1.4).

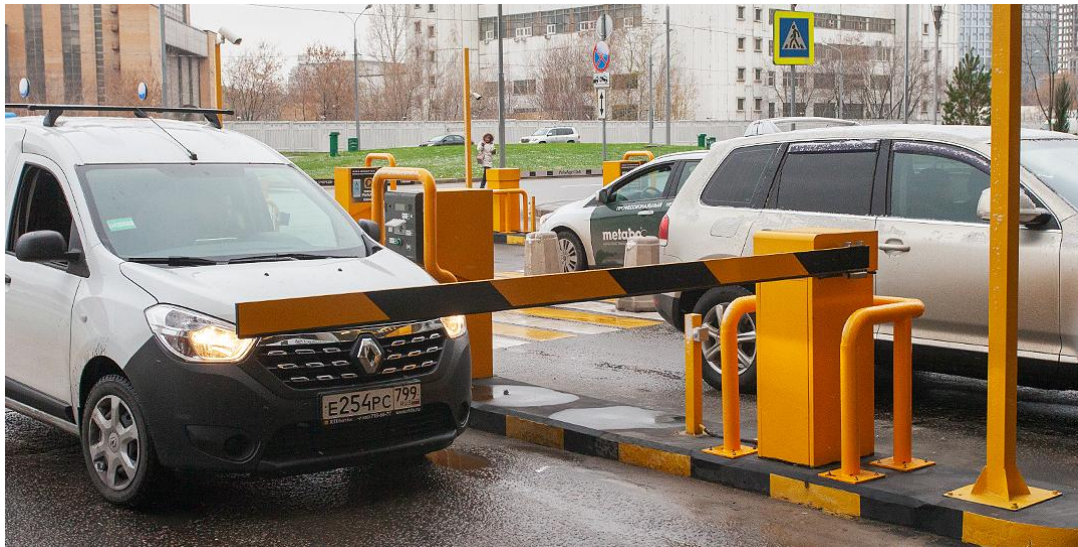


Рисунок 1.4 – Приклад системи автоматичної парковки

Головними функціями системи автоматичної парковки є ведення обліку в'їзду-виїзду транспорту та розрахунок вартості паркування. Процес взаємодії з клієнтом відбувається за допомогою автоматів оплати, видачі решти та друкування чека [16]. Також на автоматизованій парковці використовуються паркувальні стійки розташовані при в'їзді на парковку призначення яких проводити контроль в'їзду автомобілів (рисунок 1.5).

Оплата послуги паркування та поповнення безконтактних карт оплати на парковках проводиться за допомогою автоматичних кас. Такі термінали можуть розміщуватися як в приміщення та і на вулиці, при цьому вони не отримують ніяких пошкоджень від можливих опадів, оскільки обладнанні герметичним корпусом та мають антивандальний захист. Також такі каси можуть комплектуватися спеціальними навісами, які дозволять комфортніше проводити оплату користувачам в поганих погодних умовах [17]. В даних терміналах є можливість проводити безготівкову оплату паркування (рисунок 1.6).



Рисунок 1.5 – Паркувальна стійка



Рисунок 1.6 – Автоматизована каса парковки

Стоянки можна класифікувати за: термінами зберігання транспорту; розміщенням відносно житлової забудови; розміщенням відносно інших будівель; кількістю поверхів; способами міжповерхового переміщення; типом зберігання транспорту та типом огорожувальних конструкцій [18]. В результаті

проведеного аналізу виконана класифікація стоянок за рядом параметрів (рисунок 1.7).

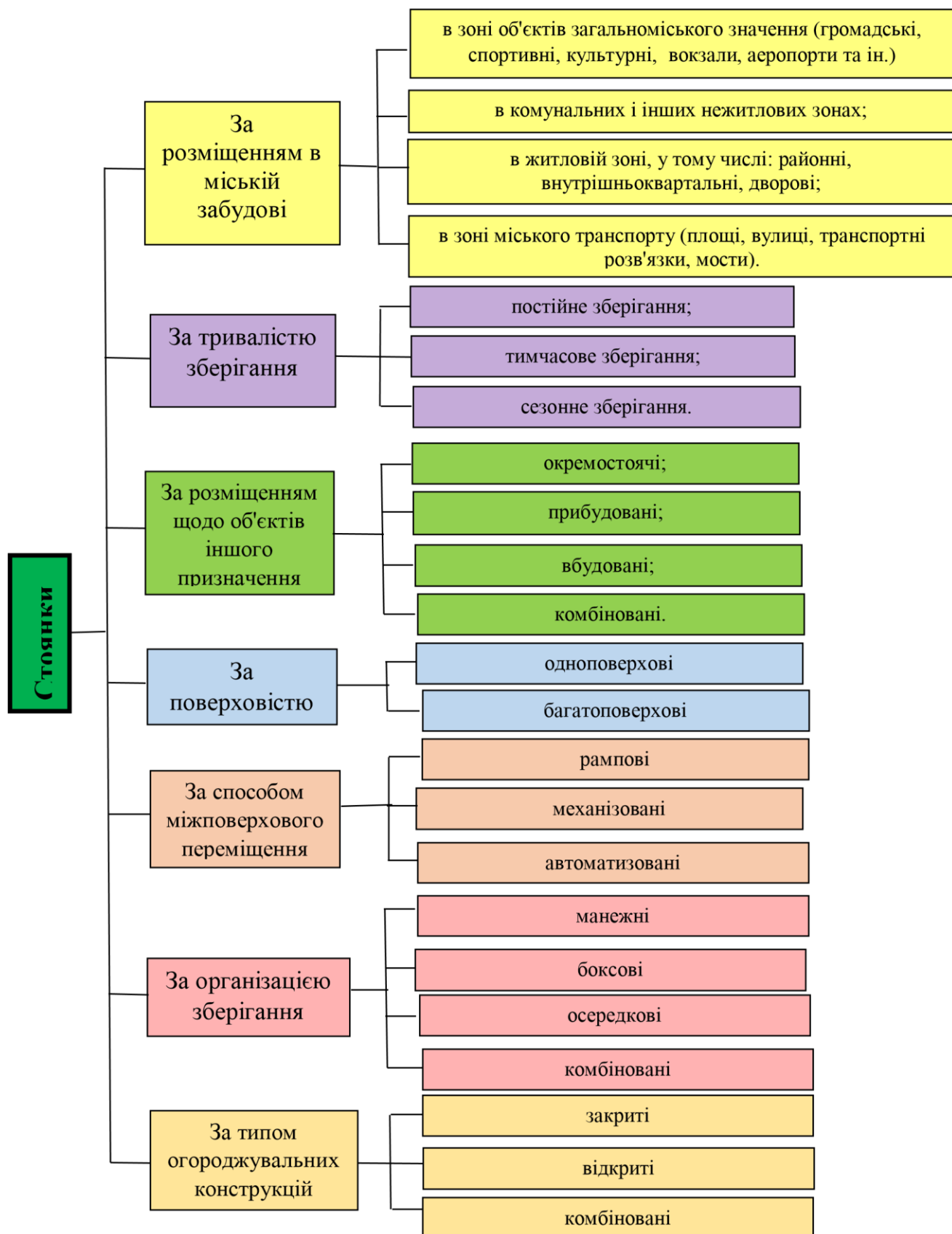


Рисунок 1.7 – Схема класифікації стоянок

При використанні паркувального місця клієнт фактично виконує його оренду терміни якої фіксуються робітником паркінгу (адміністратором). В процесі оформлення оренди паркувального місця також зазначаються дані клієнта, а саме: прізвище, ім'я, по батькові, номер телефону та прикріплюється фотографія, для достовірної ідентифікації клієнта. Окрім цього працівник вказує дані про автомобіль клієнта, а саме: модель, марку, колір, номерний знак, тип та прикріплює відповідне фото.

Враховуючи об'єм інформації якою повинний оперувати адміністратор парковки потрібно виконати автоматизацію процесу його роботи. Для чого доцільно реалізувати веб-додаток, який значно спростить процес орендування паркувальних місць та зменшить об'єм навантаження для адміністратора.

Отже, основними параметрами предметної області для задачі, що розглядається, є: працівник, паркувальне місце, клієнт, автомобіль та оренда (таблиця 1.1).

Таблиця 1.1. Параметри предметної області

<b>Параметр предметної області</b>	<b>Опис</b>
Працівник	Сутність «Працівник» є особою, що слідкує за процесом оренди паркувального місця. Важливими даними для неї є: прізвище, ім'я, по батькові, фото, логін та пароль, які потрібні для авторизації в системі
Паркувальне місце	Сутність «Паркувальне місце» є об'єктом, що здається в оренду клієнтові. Важливими параметрами для нього є: фото, опис, номер ряду, номер місця та сектор
Клієнт	Сутність «Клієнт» є особою, що орендує паркувальне місце. Важливими даними для неї є: прізвище, ім'я, по батькові, номер телефону та фото
Автомобіль	Сутність «Автомобіль» є об'єктом, для якого проводиться оренда. Важливими параметрами для нього є: фото, марка, модель, колір та номер авто
Оренда	Сутність «Оренда» фіксує зайняття паркувального місця автомобілем клієнта. Важливими параметрами для неї є: плановані дата та час початку і закінчення орендування та їхні фактичні відповідники.

## 1.2 Аналіз існуючого програмного забезпечення предметної області

Автомобільні парковки активно використовують різні програмні застосунки для пришвидшення процесу орендування паркувального місця та демонстрування списку вільних паркомісць. Реалізовані дані застосунки в основному для мобільних платформ.

Одним з прикладів реалізації такого застосунку є мобільний додаток «SMS парковка (949)» [19]. Після запуску програми перед користувачем відкривається карта з відміченими на ній паркінгами. У користувача є можливість вибрати паркінг та переглянути інформацію про час його роботи. Також додаток може автоматично побудувати маршрут від поточного місця користувача до вибраного паркінгу (рисунок 1.8)

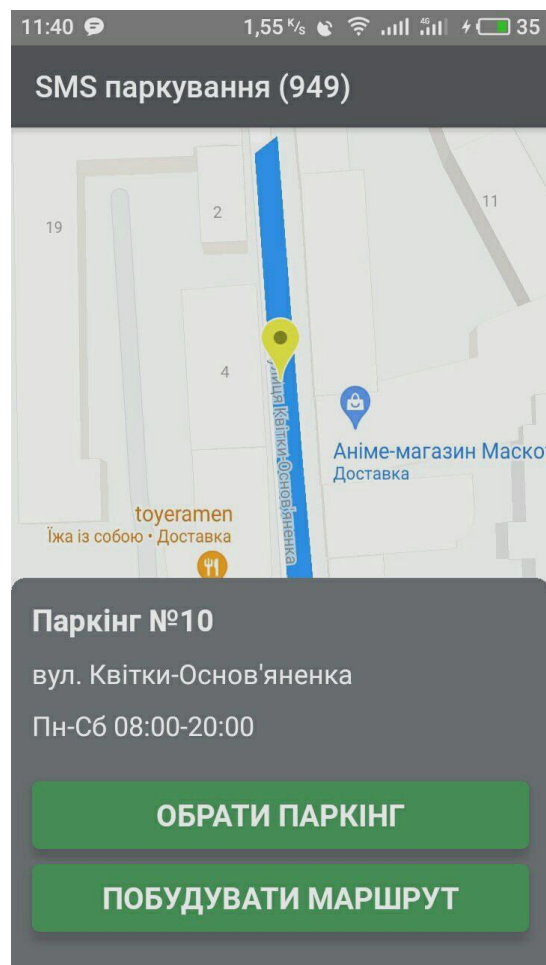


Рисунок 1.8 – Вибір паркінгу в програмі «SMS парковка (949)» [19]

Для того щоб виконати орендування паркувального місця користувачеві попередньо потрібно додати дані про автомобіль, а саме вказати марку, модель та його номер. В одного користувача може бути додано декілька автомобілів (рисунок 1.9).

При орендуванні паркувального місця також потрібно вказати бажану кількість годин паркування, від їхньої кількості залежатиме вартість парковки. Відповідно до вказаних даних проводиться оплата паркувального місця, для цього формується відповідне SMS-повідомлення яке необхідно відправити перед початком паркування. Успішне підтвердження оплати свідчитиме про початок паркувальної сесії. Якщо підтвердження не прийшло потрібно перевірити список активних сесії та у разі необхідності його відредагувати.

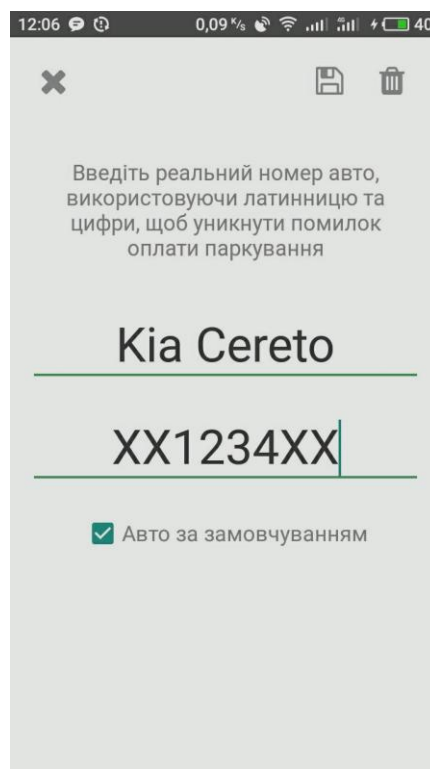


Рисунок 1.9 – Реєстрування автомобіля в додатку «SMS парковка (949)» [19]

У мобільному додатку «SMS парковка (949)» реалізовано перегляд наявних паркінгів, але не вказано кількість вільних паркувальних місць у даний момент часу, також не вказана інформація про саме паркомісце, що може призвести до непорозумінь під час паркування. Ще одним важливим недоліком є

відсутність можливості фіксування фактичного часу та дати перебування автомобіля на паркомісці.

Ще однією платформою, що дозволяє виконувати перегляд паркувальних місць є веб-сайт комунального підприємства «Харківпарксервіс» [20]. На даному веб-сайті можна переглянути список всіх паркових місць, що обслуговуються комунальним підприємством «Харківпарксервіс» (рисунок 1.10) та дізнатися деяку інформацію про них, а саме: адресу та графік роботи. Оплата оренди паркувального місця може виконуватися за допомогою «Паркувального талону» або «Електронним абонементом».

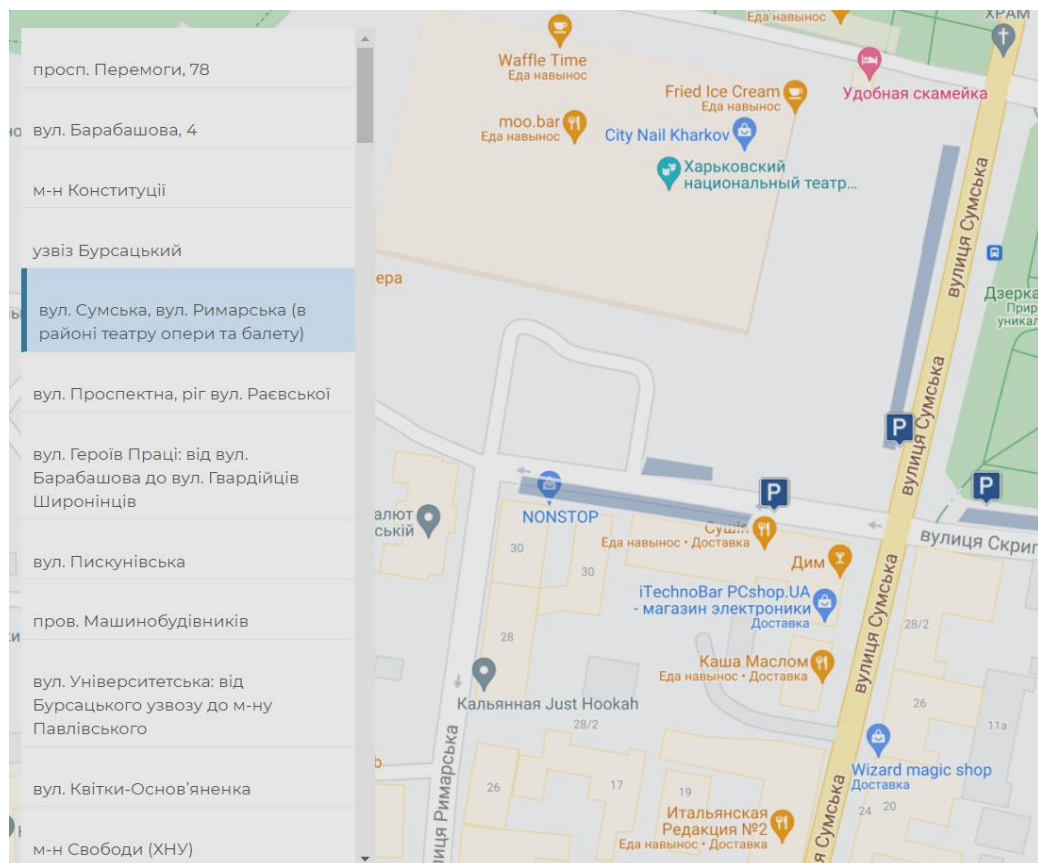


Рисунок 1.10 – Інформації про вибране паркове місце [20]

Хоч веб-сайт комунального підприємства «Харківпарксервіс» і демонструє деяку інформацію про стоянки її кількість дуже обмежена. Також на сайті відсутня можливість перегляду актуальної інформації про кількість зайнятих паркувальних місць. Ще одним упущенням даного веб-сайту є відсутність можливості автоматично побудувати маршрут до вибраної парковки.



користувач може оцінити якість паркінгу і залишити відповідний відгук про нього (рисунок 1.12).

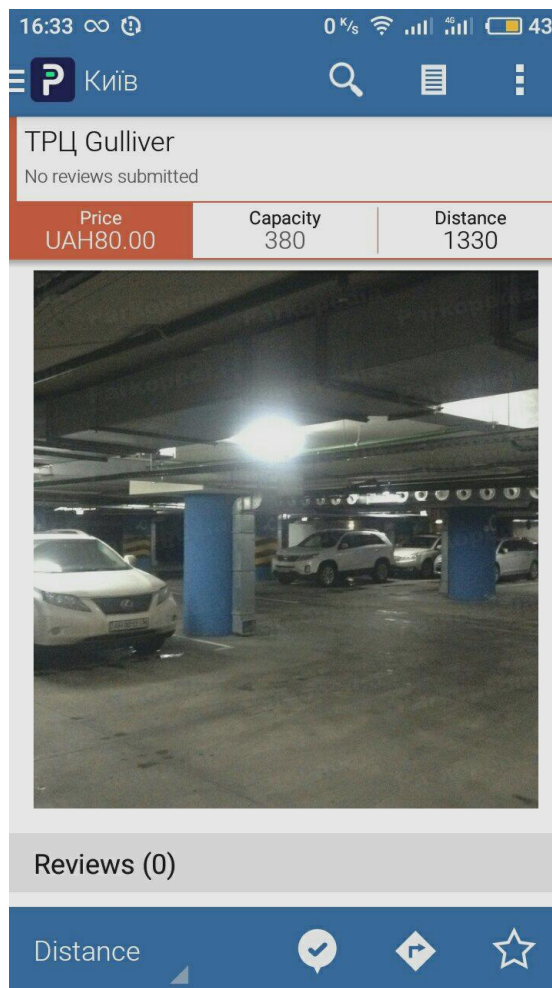


Рисунок 1.12 – Детальна інформація про паркомісце в додатку «Parkopedia Parking» [21]

Хоча додаток «Parkopedia Parking» володіє значним функціоналом для демонстрування інформації паркінгів в нього також існує ряд суттєвих недоліків, а саме: відсутність можливості вказання автомобілю користувача, відсутність можливості перегляду списку працівників.

Сьогодні можна знайти багато прикладів програм, в основному мобільних додатків, що дозволяють клієнту переглядати існуючі стоянки та орендувати на них паркувальні місця, але при цьому в них відсутній функціонал потрібний для роботи адміністратора парковки, а саме немає можливості перегляду списку всіх клієнтів та відсутні списки оренди паркувальних місць. Важливою є також

незначна кількість програм розроблених у вигляді веб-додатків, що створює на них значний попит. Отже, розробка автоматизованого робочого місця адміністратора автомобільної парковки реалізованого у вигляді веб-додатку є актуальною.

### **1.3 Аналіз сучасних засобів створення програмного забезпечення**

Розглядаючи предметну область було виявлено, що оптимальним способом реалізації всіх потреб адміністратора парковки є веб-додаток. На сьогоднішній день існує ряд найбільш перспективних платформ для створення веб-додатків, а саме: ASP.NET та PHP [22].

ASP.NET представляє собою технологію для створення веб-застосунків і веб-сервісів від компанії Майкрософт. Вона є складовою частиною платформи Microsoft.NET, яка використовується для написання серверних інтернет-додатків [23], його допомогою були розроблені такі відомі сайти як: MSN.com, Microsoft.com, Dell.com та багато інших.

До основних переваг ASP.NET можна віднести:

- ASP.NET має перевагу у швидкості в порівнянні з іншими технологіями, заснованими на скриптах (PHP та інші);
- розширюваний набір елементів управління і бібліотек класів дозволяє швидше розробляти застосунки;
- ASP.NET спирається на багатомовні можливості .NET, що дозволяє писати код сторінок на C#, VB, C/C++ та інших;
- поділ візуальної частини та бізнес-логіки;
- розширювана модель обробки запитів .

Недоліками платформи ASP.NET можна вважати: необхідність встановлення додаткового програмного забезпечення (фреймворк, сервер, додаткові бібліотеки), більше споживання трафіку у порівнянні з PHP та необхідність додатково встановлювати IDE Mono для роботи на Mac чи Unix .

PHP представляє собою програмну платформу, що може працювати як в середовищі Unix так і в Windows. PHP – це система розробки скриптів яка включає в себе CGI – інтерфейс і набір функцій для доступу до бази даних. На PHP написанні такі сайти, як: Facebook, WordPress(та всі сайти які з його допомогою були розробленні), YouTube, Wikipedia та інші [24] .

Сайт написаний на PHP інтерпретується у HTML-код, а потім передається на сторону клієнта після закінчення його роботи автоматично відбувається очищення зайнятої ним пам'яті, для невеликих проектів це має значно впливає на їхню швидкодію. PHP краще всього підходить для писання невеликих додатків, оскільки виконувати тестування та модифікації коду в великих проектах достатньо важко. В основному PHP використовують для:

- Написання програмних кодів, виконання яких відбуватиметься на стороні сервера. Найбільш широко PHP використовується саме на серверній стороні роботи додатку. Для початку роботи PHP потрібний інтерпретатор коду у вигляді програми CGI або серверного модуля, сам веб-сервер та браузер.

- Написання скриптів в командній строчці. За допомогою PHP можна створити скрипт, що працюватиме незалежно від сервера та браузера. Скрипти які повинні виконуватися регулярно ( команда cron на Unix та Linux), ідеально реалізовувати на PHP.

- Написання віконних додатків, які працюють на клієнтській стороні. PHP є не найкращою мовою для створення подібних додатків, але така функція в нього передбачена, для створення такого роду додатків потрібно використовувати PHP-GTK [25].

Перевагами PHP можна вважати:

- крос платформне програмування;
- висока швидкість роботи і загальна швидкодія ресурсів;
- написання і редагування коду можна здійснювати у будь-якому текстовому редакторі;
- висока гнучкість, ємність та функціональність;

До недоліків платформи PHP можна віднести: відсутність типізації даних, інтерпретатор не повідомляє про помилки у коді, PHP-додатки працюють повільніше, не підтримується Unicode у версіях до 6.0 [26].

Під час розгляду даних платформ були виявленні позитивні і негативні сторони кожної із них. Беручи до уваги тип майбутнього додатку і особливості платформ, оптимальним буде використання технології ASP.NET.

#### **1.4 Постановка задачі та вимоги до розробки інформаційної системи**

Метою кваліфікаційної роботи бакалавра є розробка автоматизованого робочого місця адміністратора автомобільної парковки на платформі ASP.NET, що виконує наступні основні функції:

- робота з орендами паркувальних місць;
- робота з наявним фондом паркувальних місць;
- робота з даними клієнтів;
- робота з даними автомобілів;
- робота з даними працівників.

Автоматизоване робоче місце адміністратора автомобільної парковки слід реалізувати у вигляді веб-додатку.

При роботі з орендами парковочних місць слід забезпечити можливість реєстрації дати і часу початку й завершення як запланованого часу перебування автомобіля на паркомісці, так і фактичного. При роботі з наявним фондом парковочних місць слід враховувати тип паркомісця згідно їх призначення.

## Розділ 2

### Проектування інформаційної системи

#### 2.1 Функціональна структура та бізнес-процеси системи

Автоматизоване робоче місце адміністратора автомобільної парковки призначене для автоматизації процесів орендування паркувальних місць. Адміністратору автомобільної парковки потрібно реалізувати наступні функції: робота з орендою паркувальних місць; робота з наявним фондом парковочних місць; робота з даними клієнтів; робота з даними автомобілів; робота з даними працівників (рисунок 2.1).

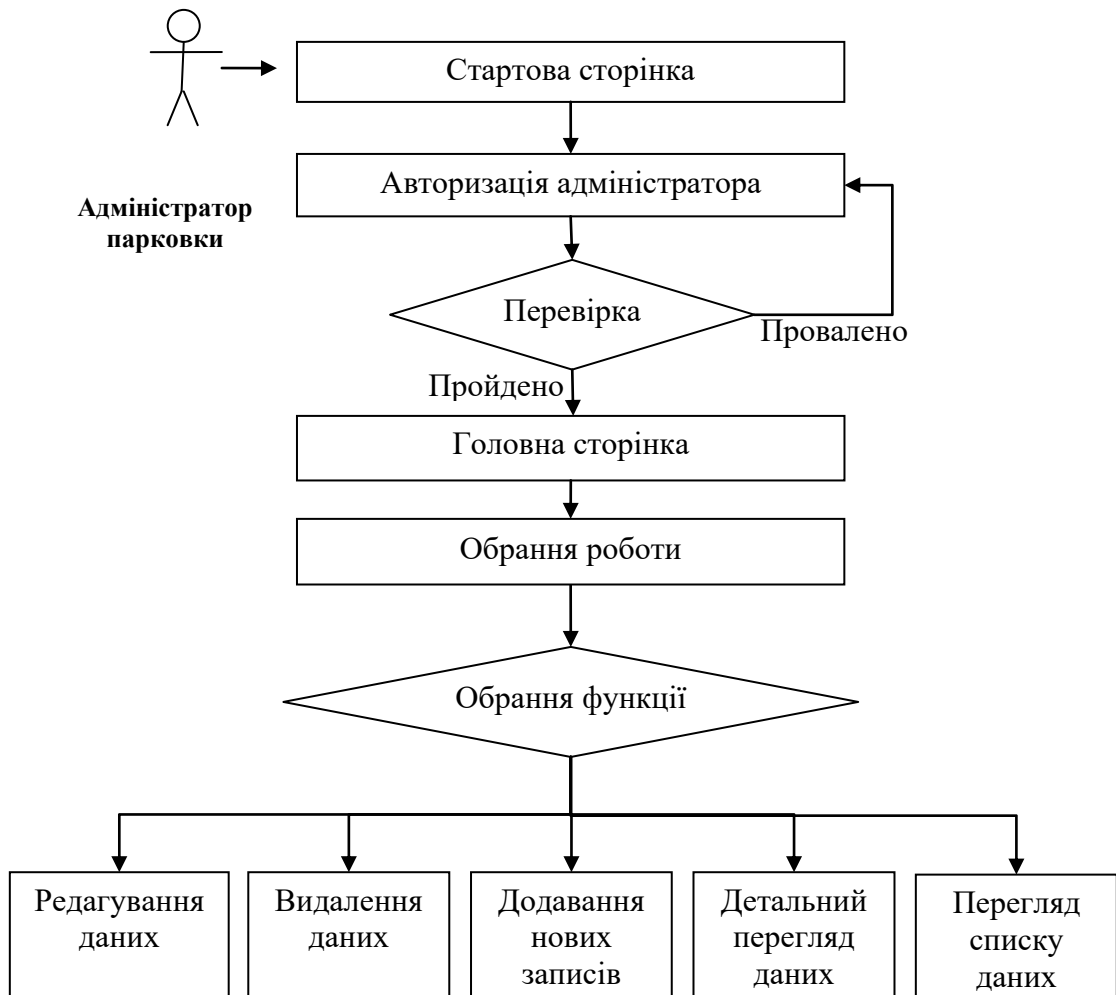


Рисунок 2.1 – Діаграма активності роботи адміністратора

*Бізнес-процес «Робота з орендою паркувальних місць».* Даний бізнес-процес надає адміністратору можливості роботи з даними оренди паркувальних місць. В процесі роботи з орендою паркомісця слід забезпечити можливість реєстрації початку та завершення перебування автомобіля на паркомісці з вказанням запланованої та фактичної дати і часу. Виконуючи роботу з орендою паркомісця обов'язково потрібно зазначити клієнта, автомобіль та паркувальне місце. Цей бізнес-процес призначений для автоматизації наступних функцій:

- перегляду списку оренд паркувальних місць;
- перегляду детальної інформацію про орендовані паркомісця;
- додавання оренди паркомісця;
- редагування вже існуючої інформації про оренду паркомісця;
- видалення інформації про оренду паркомісць.

*Бізнес-процес «Робота з наявним фондом паркомісць».* Цей бізнес-процес надає адміністратору можливість виконувати маніпуляції з паркомісцями. При додаванні паркомісця слід зазначити тип транспорту для якого воно розраховане та зазначити номер сектора, ряду та місця, для точної ідентифікації паркувального місця на паркінгу. Даний бізнес-процес призначений для автоматизації наступних функцій:

- перегляду списку паркувальних місць;
- перегляду детальної інформації про паркомісце;
- додавання паркомісця;
- редагування інформації про паркомісце.

*Бізнес-процес «Робота з даними клієнтів».* Даний бізнес-процес дозволяє адміністратору працювати з даними клієнтів. В ході реєстрації нового клієнта потрібно вказати його особисту інформацію. Цей бізнес-процес призначений для автоматизації наступних функцій:

- перегляду списку працівників;
- перегляду детальної інформації про працівника;
- додавання нових працівників;
- редагування даних існуючих працівників.

*Бізнес-процес «Робота з даними автомобілів».* Цей бізнес-процес надає адміністратору можливість працювати з даними автомобілів. В процесі роботи з даними автомобілів слід забезпечити можливість вказувати марку, модель, колір та номер автомобіля. Даний бізнес-процес призначений для автоматизації наступних функцій:

- перегляду списку автомобілів;
- перегляду детальної інформації про автомобіль;
- додавання нових автомобілів;
- редагування даних існуючих автомобілів.

*Бізнес-процес «Робота з даними працівників».* Даний бізнес-процес надає адміністратору можливість працювати з особистими даними працівників. В процесі реєстрації нового працівника потрібно вказати його особисту інформацію та дані для авторизації в системі. Цей бізнес-процес призначений для автоматизації наступних функцій:

- перегляду списку працівників;
- перегляду детальної інформації про працівника;
- додавання нових працівників;
- редагування даних існуючих працівників.

Виконання перерахованих вище бізнес-процесів забезпечує реалізацію функціональності для адміністратора автомобільної парковки. Автоматизація перерахованих бізнес-процесів приведе до підвищення ефективності роботи адміністратора.

## **2.2 Інформаційна структура системи**

Створення програми зазвичай розпочинається з проектування та розробки структури даних, для веб-додатку її роль виконує база даних. Правильно спроектована структура бази даних дозволить забезпечити цілісність даних та простоту роботи з ними.

У відповідності з поставленими завданнями для реалізації автоматизованого робочого місця адміністратора автомобільної парковки була спроектована та розроблена структура бази даних майбутнього веб-додатку. Розроблена архітектура дозволяє легко вносити в базу даних інформацію про оренду паркувальних місць, паркомісця, працівників, клієнтів та автомобілі. Схема створеної структури зображена на рисунку 2.2.

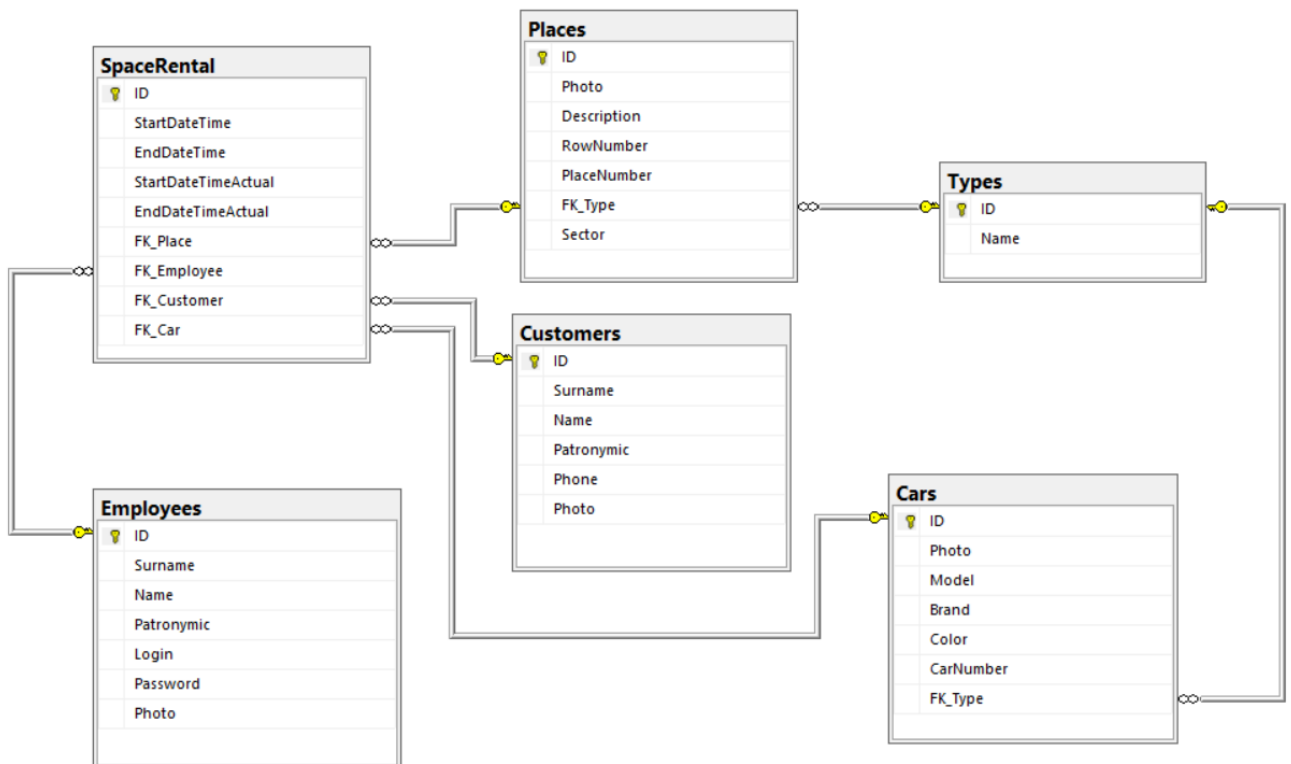


Рисунок 2.2 – Схема даних автомобільної парковки

Відповідно до структури спроектованої бази даних були відповідним чином розроблені її таблиці. В процесі створення таблиці заповнювалися початковим даними. Таблиця SpaceRental призначена для зберігання інформації про паркувальні місця (таблиця 2.1). Дана таблиця є головною в розробленій схемі, вона активно взаємодіє майже із всіма іншими таблицями.

Таблиця Employees призначена для зберігання інформації працівників паркінгу (таблиця 2.2).

Таблиця 2.1 – Атрибути таблиці SpaceRental

№ п/п	Назва атрибуту	Тип даних	Опис
1.	ID	int	Первинний ключ, числовий ідентифікатор для однозначного визначення запису таблиці
2.	StartDateTime	datetime	Плановані дата та час початку оренди
3.	EndDateTime	datetime	Плановані дата та час завершення оренди
4.	StartDateTimeActual	datetime	Фактичні дата та час початку оренди
5.	EndDateTimeActual	datetime	Фактичні дата та час завершення оренди «
6.	FK_Place	int	Вторинний ключ, посилання на запис таблиці «Places» призначений для співставлення з відповідним паркувальним місцем
7.	FK_Employee	int	Вторинний ключ, посилання на запис таблиці «Employees» призначений для співставлення з відповідним працівником
8.	FK_Customer	int	Вторинний ключ, посилання на запис таблиці «Customers» призначений для співставлення з відповідним клієнтом
9.	FK_Car	int	Вторинний ключ, посилання на запис таблиці «Cars» призначений для співставлення з відповідним автомобілем

Таблиця Customers призначена для зберігання інформації клієнті, які користуються паркінгом (таблиця 2.3).

Таблиця 2.2 – Атрибути таблиці Employees

№ п/п	Назва атрибуту	Тип даних	Опис
1.	ID	int	Первинний ключ, числовий ідентифікатор для однозначного визначення запису таблиці
2.	Surname	nvarchar(250)	Прізвище працівника
3.	Name	nvarchar(250)	Ім'я працівника
4.	Patronymic	nvarchar(250)	По батькові працівника
5.	Login	nvarchar(250)	Логін працівника
6.	Password	nvarchar(250)	Пароль працівника
7.	Photo	byte[]	Фото працівника

Таблиця 2.3 – Атрибути таблиці Customers

№ п/п	Назва атрибуту	Тип даних	Опис
1.	ID	int	Первинний ключ, числовий ідентифікатор для однозначного визначення запису таблиці
2.	Surname	nvarchar(250)	Прізвище клієнта
3.	Name	nvarchar(250)	Ім'я клієнта
4.	Patronymic	nvarchar(250)	По батькові клієнта
5.	Phone	nvarchar(250)	Номер телефону клієнта
6.	Photo	byte[]	Фото клієнта

Таблиця Types призначена для зберігання інформації про типи транспорту (таблиця 2.4).

Таблиця 2.4 – Атрибути таблиці Types

№ п/п	Назва атрибуту	Тип даних	Опис
1.	ID	int	Первинний ключ, числовий ідентифікатор для однозначного визначення запису таблиці
2.	Name	nvarchar(250)	Назва типу

Таблиця Cars призначена для зберігання інформації про автомобілі клієнтів паркінгу (таблиця 2.5).

Таблиця 2.5 – Атрибути таблиці Cars

№ п/п	Назва атрибуту	Тип даних	Опис
1.	ID	int	Первинний ключ, числовий ідентифікатор для однозначного визначення запису таблиці
2.	Photo	byte[]	Фото автомобіля
3.	Model	nvarchar(250)	Модель автомобіля
4.	Brand	nvarchar(250)	Марка автомобіля
5.	Color	nvarchar(250)	Колір автомобіля
6.	CarNumber	int	Номер автомобіля
7.	FK_Type	int	Вторинний ключ, посилання на запис таблиці «Types» призначений для співставлення з відповідним типом транспорту

Таблиця Places призначена для зберігання інформації про паркувальні місця (таблиця 2.6).

Таблиця 2.6 – Атрибути таблиці Places

№ п/п	Назва атрибуту	Тип даних	Опис
1.	ID	int	Первинний ключ, числовий ідентифікатор для однозначного визначення запису таблиці
2.	Photo	byte[]	Фото паркомісця
3.	Description	nvarchar(250)	Опис паркомісця
4.	RowNumber	int	Номер ряду паркомісця
5.	PlaceNumber	int	Номер місця в ряду
6.	FK_Type	int	Вторинний ключ, посилання на запис таблиці «Types» призначений для співставлення з відповідним типом транспорту
7.	Sector	nvarchar(250)	Сектор в якому розташоване паркомісце

Таким чином, розроблена база даних дозволить зберігати всю потрібну інформацію для правильної роботи автоматизованого робочого місця адміністратора автомобільної парковки.

## **2.3 Вибір засобів розробки інформаційної системи**

Для розробки веб-додатку автоматизованого робочого місця адміністратора автомобільної парковки необхідно обрати мову програмування, фреймворк та СКБД, за допомогою яких проводитиметься розробка проекту.

### **2.3.1 Опис мови програмування**

Для створення веб-додатку автоматизованого робочого місця адміністратора автомобільної парковки було обрано мову програмування C#. Сьогодні це одна з найпопулярніших мов програмування, за її допомогою пишуться різноманітні проекти: починаючи від незначних десктопних додатків до величезних веб-сайтів якими кожного дня користуються тисячі людей [27].

На сьогоднішній момент це є одна з найпотужніших та найпопулярніших мов програмування в галузі інформаційних технологій. На ній пишуться програми різного виду: від невеликих десктопних додатків до великих веб-порталів і веб-сервісів, які щодня використовують мільйони людей.

У порівнянні з іншими мовами програмування, C# – є відносно молодою мовою. Перша його версія вийшла разом з релізом Microsoft Visual Studio .NET 2002 року, а поточною версією мови є C# 9.0, яка вийшла 2020 року разом з релізом .NET Core 5. Це говорить про те що даний продукт постійно підтримується розробниками, та користується великим попитом.

C# представляє собою об'єктно-орієнтовану мову програмування, значна частина функціональних можливостей якої перейнята від мов попередників Java та C++. Тому, C# реалізує три основні парадигми об'єктно-орієнтованих мов

програмування: інкапсуляцію, поліморфізм та успадкуванням. Розглянемо кожну з цих властивостей окремо:

– *Інкапсуляція* - це механізм програмування, який об'єднує разом код і дані, якими він маніпулює, виключаючи як втручання ззовні, так і неправильне використання даних. Коли код і дані зв'язуються разом подібним чином, створюється об'єкт. Іншими словами, об'єкт - це елемент, що підтримує інкапсуляція.

– *Поліморфізм* - це властивість, що дозволяє одному інтерфейсу отримувати доступ до загального класу дій. У більш загальному сенсі поняття поліморфізму нерідко виражається наступним чином: "один інтерфейс - безліч методів". Це означає, що для групи взаємопов'язаних дій можна розробити загальний інтерфейс. Це допомагає значно спростити програму.

*Успадкування* являє собою процес, в ході якого один об'єкт набуває властивості іншого об'єкта. Це дуже важливий процес, оскільки він забезпечує принцип ієрархічної класифікації. Якщо не користуватися ієрархіями, то для кожного об'єкта довелося б явно визначати всі його властивості. А якщо скористатися спадкуванням, то досить визначити лише ті властивості, які роблять об'єкт особливим в його класі. Він може також успадковувати загальні властивості свого батька. Отже, завдяки механізму успадкування один об'єкт стає окремим екземпляром більш загального класу [28].

Об'єктно-орієнтований підхід надає можливість створення великих розширюваних додатків. Програми написані на C# виконуються в віртуальній системі виконання .NET, яка в свою чергу викликає загальне середовище виконання (CLR) та набір бібліотеки класів [29]. Набори різноманітних бібліотек та шаблонів дозволяють використовувати вже готові програмні рішення, що значно пришвидшує процес розробки .

Отже, використання об'єктно-орієнтованого підходу та вбудованих бібліотек дозволить створити веб-додаток з потужним функціоналом, який використовуватиметься для автоматизації робочого місця адміністратора автомобільної парковки.

### 2.3.2 Вибір фреймворку

В процесі розробки автоматизованого робочого місця адміністратора автомобільної парковки, був проведений вибір фреймворку, вході якого було розглянуто ASP.NET MVC, ASP.NET Core та ASP.NET WebForms.

ASP.NET MVC є фреймворком для створення веб-додатків, що реалізує шаблон Model-View-Controller. Цей шаблон дозволяє легко відділити бізнес логіку від програмної реалізації, для цього проект поділяється на компоненти, кожен з яких потрапляє в одну з трьох категорій:

- модель (model) – містить логіку, яка пов'язана з даними;
- представлення (view) – інтерфейс додатку, який відображає дані;
- контролер (controller) – обробка всієї бізнес-логіки і вхідних запитів [30].

Основними перевагами ASP.NET MVC є:

- можливість розширення ієрархії класів;
- можливість реалізації бізнес-процесів будь-якої складності;
- швидкість роботи програми;
- тестування [31].

ASP.NET Core є вільно розповсюджуваним кросплатформним фреймворком призначеним для створення веб-додатків з відкритим вихідним кодом [32].

Основними перевагами ASP.NET Core є:

- кросплатформність;
- відкритий вихідний код.

ASP.NET WebForms – це платформа для створення веб-додатків підтримуваних технологією Microsoft ASP.NET, що була включена в початковий випуск .NET Framework 1.0 як перша модель програмування [33]. Додатки на Web Forms можуть бути написані на будь-якій мові програмування, яка підтримує мову CLR (Common Language Runtime), прикладом є C# або Visual

Basic. Основними блоками сторінок веб-форм є елементи управління, які представляють собою повторно використовувані компоненти, що відповідають за рендеринг HTML-розмітки і реагування на події.

Превагами ASP.NET WebForms є:

- можливість швидкої розробки додатків;
- підтримка дизайнера клієнтської частини в Visual Studio;
- значна кількість бібліотек контролерів, які підтримуються WebForms;
- низький рівень входу для початківців.

Недоліками ASP.NET WebForms є:

- складність відділення бізнес логіки від програмної реалізації;
- складність під час проведення unit-тестування;
- занадто великий розмір сторінок, через ViewState [34].

Враховуючи переваги та недоліки розглянутих шаблонів для розробки автоматизованого робочого місця адміністратора автомобільної парковки, було обрано фреймворк ASP.NET MVC.

### **2.3.3 Вибір системи керування базами даних**

При розробці автоматизованого робочого місця адміністратора автомобільної парковки був проведений вибір СКБД, у якому проводитиметься розробка бази даних.

MySQL – система керування реляційними базами даних. Наразі ця система належить компанії Oracle, але вона була створена компанією «ТсХ» для підвищення швидкодії обробки великих баз даних. Вона була створена як альтернатива комерційним системам. MySQL використовує мову запитів SQL. Вона прекрасно підходить як для середніх, так і для великих проектів.

MySQL працює з багатьма фреймворками та на багатьох платформах, має якісну документацію, високі стандарти безпеки, велику кількість розробників, які використовують дану СКБД. Крім цього завдяки популярності даної СКБД, є висока ймовірність, що вона буде і надалі дороблятися її розробниками [35].

Превагами MySQL є:

- масштабованість, може підтримувати роботу над великих БД;
- кросплатформеність;
- зв'язність;
- швидкість роботи;
- зручність експлуатації;
- відкритий код.

Недоліками MySQL є:

- недостатня надійність;
- достатньо низька швидкість розробки [36].

SQL Server – це система керування базами даних, яка розроблена корпорацією Microsoft. Для створення запитів використовується мова SQL, яка є основною стандартною мовою програмування ANSI / ISO [37]. Крім того SQL Server також використовує розширення цієї мови під назвою Transact-SQL, що має набагато більший функціонал, та надає додаткові можливості по оголошенню змінних, обробки виключень, та використання збережених процедур.

До переваг MS SQL Server можна віднести:

- ефективність, інструменти для керування доступом, призначені для захисту конфіденційної інформації користувачів;
- надійність та безпека, SQL Server містить алгоритми шифрування даних;
- можливість взаємодії з хмарою Windows Azure.

До недоліків MS SQL Server можна віднести:

- Можливість роботи тільки на операційній системі Windows;
- Достатньо висока ціна повної версії програми [38].

Враховуючи особливості поставлених завдань для розробки АРМ адміністратора автомобільної парковки було вирішено обрати СКБД MS SQL Server

Результатом виконання кваліфікаційної роботи бакалавра має бути автоматизована інформаційна система адміністратора автомобільної парковки, призначений для автоматизації процесу орендування паркувальних місць, збереження та отримання інформації з бази даних. Для написання програмного продукту було обрано платформу розробки ASP.NET, мову програмування C#, із використанням MVC шаблону та СКБД MS SQL Server. Застосування такої системи допоможе автоматизувати роботу адміністратора автомобільної парковки.

## Розділ 3

### Програмна реалізація інформаційної системи

#### 3.1 Структура та функціональне призначення складових системи

На основі розробленої структури веб-додатку було створено діаграму класів, яка буде використовуватися при створенні проекту. Розроблена структура зображена на рисунку 3.1

Оскільки для реалізації веб-додатку було вирішено використовувати фреймворк ASP.NET MVC то для основних таблиць з бази даних були реалізовані відповідні моделі, контролери та набори представлень: Index, Create, Delete, Details та Edit. Представлення Index відображає списки даних відповідної моделі. За допомогою представлення Create виконується додавання нових записів до бази даних. Використовуючи представлення Delete можна видалити непотрібні записи з таблиці. Представлення Details відображає детальну інформацію про вибраний запис. За допомогою представлення Edit можна виконати редагування запису.

Класи Cars, Customers, Employees, Places, SpaceRental та Types є моделями відповідних таблиць з бази даних. Для кожного з цих класів розроблені контролери, які забезпечують зв'язок між моделями та представленнями.

Також в проекті присутній HomeController, що виконує виклик стартової сторінки (представлення Index) на якій виводиться загальна інформація про паркінг. Крім того в контролері є представлення «About» яке відображає інформацію про програму та її розробника.

Клас CustomRoleProvider використовується для розділення ролей користувачів. У даному класі реалізується інтерфейс RoleProvider та розширюються його методи GetRolesForUser() та IsUserInRole(). За допомогою методу GetRolesForUser() можна виконувати додавання нових ролей для користувачів. У методі IsUserInRole() реалізована логіка роботи з відповідною роллю.

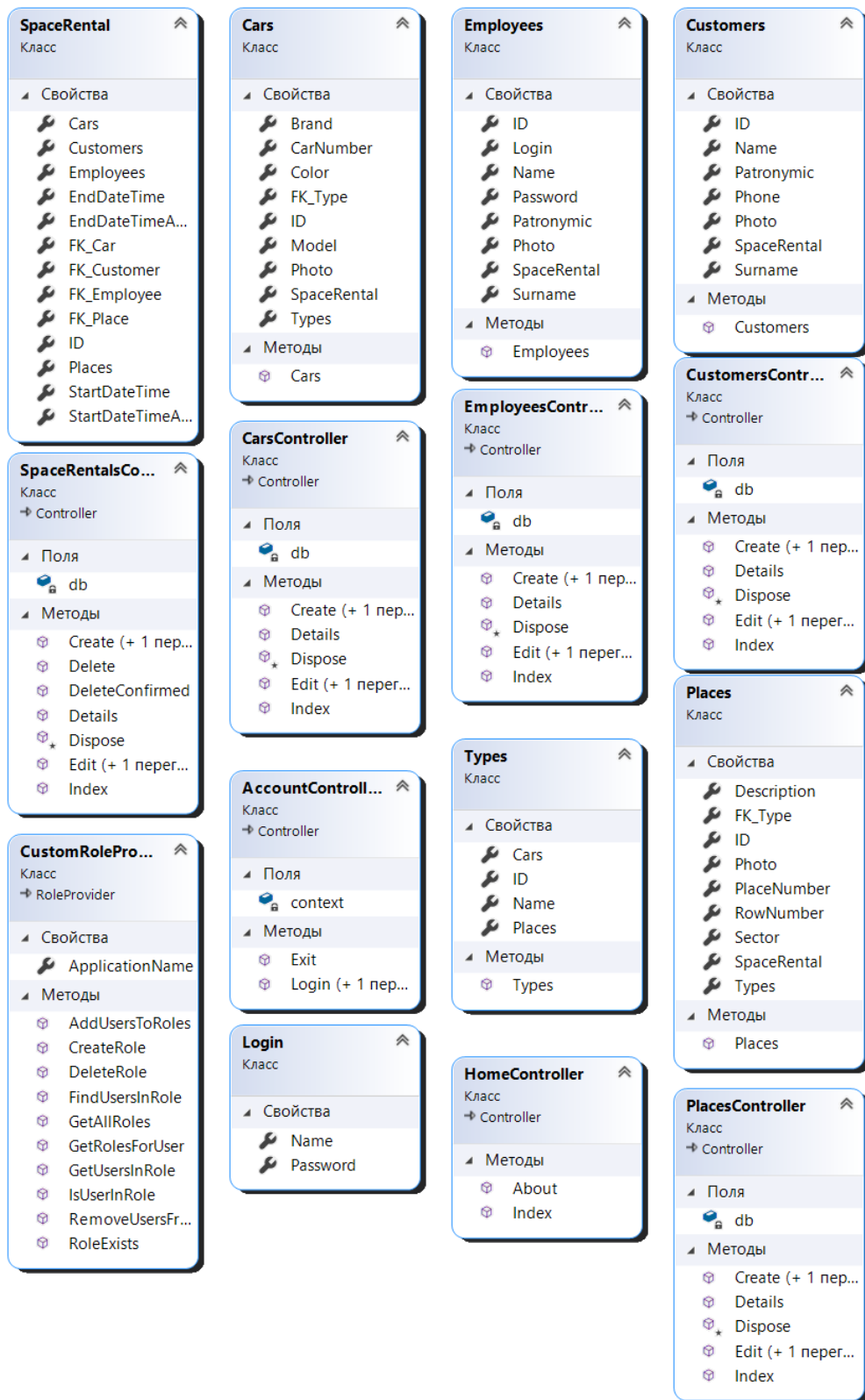


Рисунок 3.1 – Діаграма класів веб-додатку

Клас Login використовуються як модель для входу в систему, з ним взаємодіє контролер AccountController. У контролері реалізовані методи Login() та Exit(). У методі Login() виконується перевірка введених логіну та паролю, у

разі успішного її проходження відбувається перехід на головну сторінку веб-додатку. Метод Exit() виконує вихід адміністратора з веб-додатку.

Розроблена структура веб-додатку дозволяє повністю реалізувати основні функції автоматизованого робочого місця адміністратора автомобільної парковки, а також надає можливість подальшого розширення його функціоналу.

### 3.2 Особливості реалізації складових системи

Основними функціями адміністратора є введення нових даних, редагування і видалення вже існуючих. Тому для кожної таблиці з якою має працювати адміністратор були реалізовані відповідні модулі, а саме: Create, Edit та Delete.

Модуль «Create» призначений для додавання нових записів в відповідні таблиці. Після запуску сторінки перед адміністратором буде відображена форма у якій потрібно заповнити поля бажаними даними (рисунк 3.2).

The screenshot shows a web application interface for adding transport. The header is dark blue with navigation links: Паркування, Паркомісця, Працівники, Клієнти, Автомобілі, Про програму, and a 'Вихід' (Exit) button. The main title is 'Додати транспорт'. The form contains the following fields:

- Картинка: Выберете файл 3.jpg
- Модель: T1
- Марка: Volkswagen
- Колір: Червоний
- Номер транспорту: 4523
- Тип: Автомобіль (dropdown menu)

Below the form are two buttons: 'Додати' (Add) in green and 'На головну' (Home) in blue. At the bottom, there is a footer: 'Автоматизоване робоче місце адміністратора автомобільної парковки' and 'Лавренко КН-17-1 2021'.

Рисунок 3.2 – Приклад заповнення представлення «Create»

Коли всі поля форми заповнення коректними даними можна виконати додавання запису натиснувши на кнопку «Додати». Запит на додавання нового запису обробляється наступним програмним кодом:

```
public ActionResult Create([Bind(Include =
"ID,Photo,Model,Brand,Color,CarNumber,FK_Type")] Cars cars, HttpPostedFileBase
uploadImage)
{
    if (ModelState.IsValid && uploadImage != null)
    {
        byte[] imageData = null;
        using (var binaryReader = new BinaryReader(uploadImage.InputStream))
        {
            imageData = binaryReader.ReadBytes(uploadImage.ContentLength);
        }
        cars.Photo = imageData;
        db.Cars.Add(cars);
        db.SaveChanges();
        return RedirectToAction("Index", "Cars");
    }
    else
    {
        ViewBag.Message = "Будь ласка заповніть всі поля";
        ViewBag.FK_Type = new SelectList(db.Types, "ID", "Name", cars.FK_Type);
        return View(cars);
    }
}
```

Модуль «Edit» призначений для редагування існуючих даних. Після того як адміністратор вибере запис в якому потрібно змінити дані перед ним буде відкрита відповідна форма з полями для редагування (рисунок 3.3).

The screenshot shows a web application interface for editing car information. At the top, there is a dark blue navigation bar with icons and text links: 'Паркування', 'Паркомісія', 'Працівники', 'Клієнти', 'Автомобілі', and 'Про програму'. A 'Вихід' button is in the top right. Below the navigation bar is the title 'Редагування'. The main content area contains a form with the following elements:

- Картинка:** A photo of a light green and white Volkswagen van. Below it is a file upload control with the text 'Выберите файл' and 'файл не выбран'.
- Модель:** A text input field containing 'T1'.
- Марка:** A text input field containing 'Volkswagen'.
- Колір:** A text input field containing 'Червоний'.
- Номер транспорту:** A text input field containing '4523'.
- Тип:** A dropdown menu with 'Автомобіль' selected.
- Buttons:** A green 'Зберегти' button and a blue 'На головну' button.

Рисунок 3.3 – Форма редагування інформації про автомобіль

Для збереження внесених змін потрібно натиснути на кнопку «Зберегти». Запит на редагування вибраного запису обробляється наступним програмним кодом:

```
[HttpPost]
[ValidateAntiForgeryToken]
public ActionResult Edit([Bind(Include =
"ID,Photo,Model,Brand,Color,CarNumber,FK_Type")] Cars cars, HttpPostedFileBase
uploadImage)
{
    if (ModelState.IsValid)
    {
        if (uploadImage != null)
        {
            byte[] imageData = null;
            using (var binaryReader = new BinaryReader(uploadImage.InputStream))
            {
                imageData = binaryReader.ReadBytes(uploadImage.ContentLength);
            }
            cars.Photo = imageData;
        }
        db.Entry(cars).State = EntityState.Modified;
        db.SaveChanges();
        return RedirectToAction("Index");
    }
    ViewBag.FK_Type = new SelectList(db.Types, "ID", "Name", cars.FK_Type);
    return View(cars);
}
```

Модуль «Delete» використовується для видалення не актуальних записів з бази даних. Коли адміністратор вибрав запис який необхідно видалити потрібно додатково підтвердити свій вибір, дана функція реалізована для уникнення випадкової втрати даних через неуважність (рисунок 3.4).

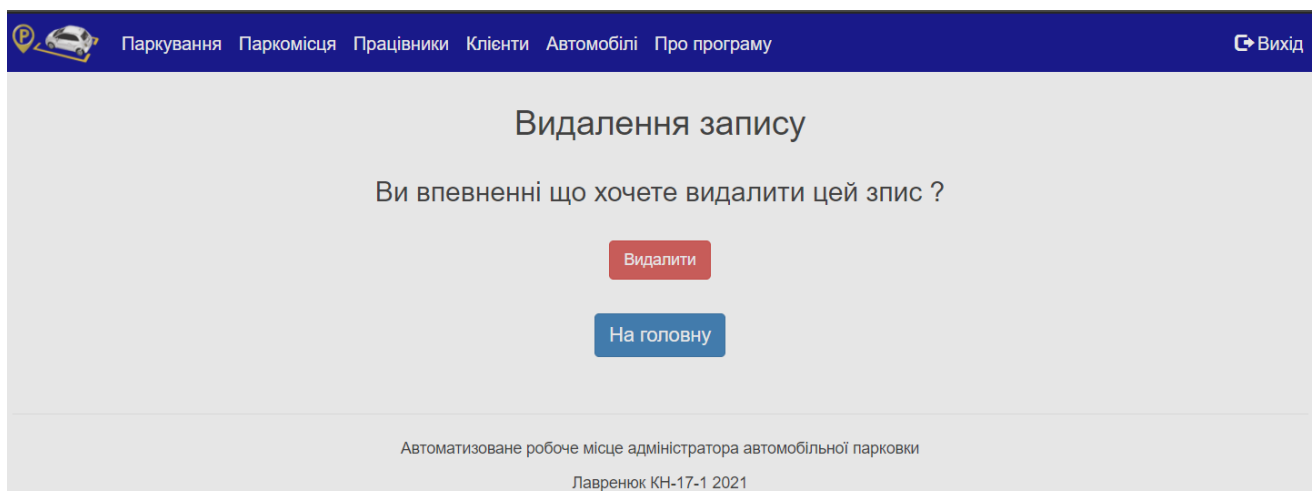


Рисунок 3.4 – Підтвердження видалення запису

Для підтвердження видалення запису потрібно натиснути на кнопку «Видалити». Запит на видалення вибраного запису обробляється наступним програмним кодом:

```
[HttpPost, ActionName("Delete")]
[ValidateAntiForgeryToken]
public ActionResult DeleteConfirmed(int id)
{
    SpaceRental spaceRental = db.SpaceRental.Find(id);
    db.SpaceRental.Remove(spaceRental);
    db.SaveChanges();
    return RedirectToAction("Index");
}
```

Реалізовані модулі надають адміністратору функціональні можливості для автоматизованого керування даними автомобільної парковки.

### 3.3 Тестування інформаційної системи

Для перевірки коректності роботи автоматизованого робочого місця адміністратора автомобільної парковки проводиться тестування можливостей роботи з даними. Тестування складається з п'яти тест-кейсів, які перевіряють функції додавання, редагування, видалення, перегляду детальної інформації про запис в таблиці SpaceRentals та можливість авторизації адміністратора.

За допомогою першого тест-кейсу перевіряється можливість додавання нового запису для орендування паркомісця (таблиця 3.1). Результатом роботи тест-кейсу має бути додавання нового запису в таблицю SpaceRentals.

У результаті проходження тест-кейсу було успішно додано новий запис з вказаними вхідними даними. Зображення доданого запису продемонстровано на рисунку 3.5.

У другому тест-кейсі тестується можливість редагування даних вибраного запису, результатом успішного проходження якого є зміна даних в вибраних полях (таблиця 3.2).

Таблиця 3.1 – Тест-кейс TS0001

<b>Тест-кейс ID:</b> TS0001	<b>Пріоритет:</b> 1	<b>Створено:</b> 15.02.2021, Лавренюк Б.С.
<b>Назва:</b> Перевірка додавання нового запису для орендування паркомісця <b>Вхідні дані:</b> <b>Дата та час початку:</b> 20.06.2020 8:00:00 <b>Дата та час закінчення:</b> 21.06.2020 8:00:00 <b>Дата та час початку фактичні:</b> 20.06.2020 8:01:00 <b>Дата та час закінчення фактичні:</b> 21.06.2020 7:59:00 <b>Паркомісце:</b> А1 <b>Працівник:</b> Райчук <b>Клієнт:</b> Лайтер <b>Транспорт:</b> Volkswagen		
<b>Кроки</b>	<b>Очікуваний результат</b>	
<ol style="list-style-type: none"> <li>1. Авторизуватися в системі</li> <li>2. Перейти на сторінку «Паркування»</li> <li>3. Натиснути на кнопку «Орендувати паркомісце»</li> <li>4. Заповнити поля вхідними даними</li> <li>5. Натиснути кнопку «Додати»</li> <li>6. Перевірити додавання нового запису в таблицю</li> </ol>	Додавання нового запису для орендування паркомісця	
<b>Результат виконання тест-кейсу:</b> пройдено успішно		

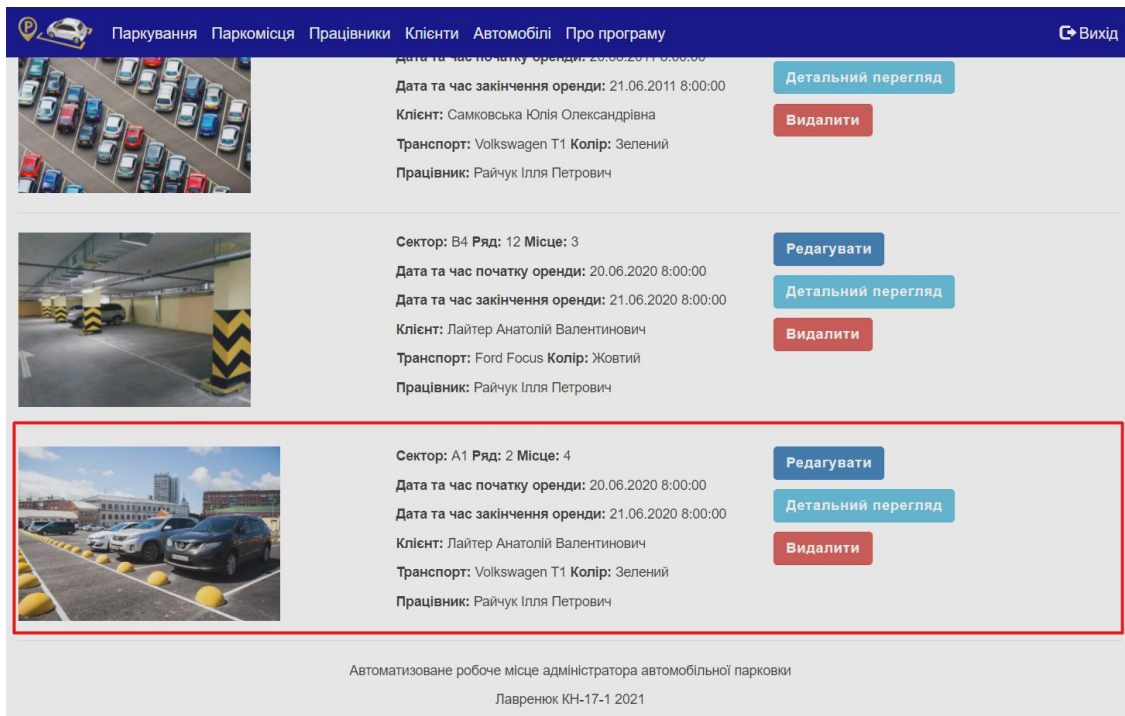


Рисунок 3.5 – Результат успішного проходження тест-кейсу

Таблиця 3.2 – Тест-кейс TS0002

<b>Тест-кейс ID:</b> TS0002	<b>Пріоритет:</b> 1	<b>Створено:</b> 15.02.2021, Лавренюк Б.С.
<b>Назва:</b> Тестування можливості редагування даних вибраного запису <b>Вхідні дані:</b> Останній доданий запис <b>Клієнт:</b> Зернюк <b>Транспорт:</b> Land Cruiser		
Кроки	Очікуваний результат	
<ol style="list-style-type: none"> <li>1. Авторизуватися в системі</li> <li>2. Перейти на сторінку «Паркування»</li> <li>3. Натиснути на кнопку «Редагувати»</li> <li>4. Зробити зміни відповідні вхідним даним</li> <li>5. Натиснути «Зберегти»</li> <li>6. Переглянути отриманні зміни</li> </ol>	Зміна значень полів «Клієнт» та «Транспорт»	
<b>Результат виконання тест-кейсу:</b> пройдено успішно		

У результаті проходження тест-кейсу було успішно змінено зазначенні поля. Результати редагування даних продемонстровано на рисунку 3.6.

The screenshot shows a web application interface for parking management. At the top, there is a navigation bar with a parking icon and links: Паркування, Паркомістя, Працівники, Клієнти, Автомобілі, Про програму, and a 'Вихід' button. Below the navigation bar, there are three rows of parking records. Each row consists of a photo of a parking lot, a list of details, and three action buttons: 'Редагувати' (Edit), 'Детальний перегляд' (Detailed view), and 'Видалити' (Delete). The third record is highlighted with red boxes around the 'Клієнт' and 'Транспорт' fields.

Image	Details	Actions
	Дата та час початку оренди: 20.06.2011 8:00:00 Дата та час закінчення оренди: 21.06.2011 8:00:00 Клієнт: Самковська Юлія Олександрівна Транспорт: Volkswagen T1 Колір: Зелений Працівник: Райчук Ілля Петрович	<a href="#">Детальний перегляд</a> <a href="#">Видалити</a>
	Сектор: B4 Ряд: 12 Місце: 3 Дата та час початку оренди: 20.06.2020 8:00:00 Дата та час закінчення оренди: 21.06.2020 8:00:00 Клієнт: Лайтер Анатолій Валентинович Транспорт: Ford Focus Колір: Жовтий Працівник: Райчук Ілля Петрович	<a href="#">Редагувати</a> <a href="#">Детальний перегляд</a> <a href="#">Видалити</a>
	Сектор: A1 Ряд: 2 Місце: 4 Дата та час початку оренди: 20.06.2020 8:00:00 Дата та час закінчення оренди: 21.06.2020 8:00:00 <b>Клієнт: Зернюк Іван Іванович</b> <b>Транспорт: Toyota Land Cruiser Колір: Сірий</b> Працівник: Райчук Ілля Петрович	<a href="#">Редагувати</a> <a href="#">Детальний перегляд</a> <a href="#">Видалити</a>

Рисунок 3.6 – Результати редагування даних

Третій тест-кейс призначений для перевірки отримання детальної інформації про вибраний запис (таблиця 3.3)

Таблиця 3.3 – Тест-кейс TS0003

<b>Тест-кейс ID:</b> TS0003	<b>Пріоритет:</b> 1	<b>Створено:</b> 15.02.2021, Лавренюк Б.С.
<b>Назва:</b> Перевірка отримання детальної інформації про вибраний запис		
<b>Вхідні дані:</b> Останній доданий запис		
Кроки	Очікуваний результат	
<ol style="list-style-type: none"> <li>1. Авторизуватися в системі</li> <li>2. Перейти на сторінку «Паркування»</li> <li>3. Натиснути на кнопку «Детальний перегляд»</li> <li>4. Переглянути детальну інформацію про вибраний запис</li> </ol>	Перехід на сторінку з детальною інформацією про вибраний запис.	
<b>Результат виконання тест-кейсу:</b> пройдено успішно		

У результаті проходження тест-кейсу була успішно переглянута детальна інформація про вибраний запис. Результати детального перегляду інформацію про запис продемонстровано на рисунку 3.7.

**Паркомісце:**  
 Сектор: A1 Ряд: 2 Місце: 4  
 Дата та час початку оренди: 20.06.2020 8:00:00  
 Дата та час закінчення оренди: 21.06.2020 8:00:00  
 Дата та час початку фактичний: 20.06.2020 8:01:00  
 Дата та час закінчення фактичний: 21.06.2020 7:59:00  
 Опис: Паркомісце розташоване в центрі міста. Зручне розташування

**Автомобіль:**  
 Марка: Toyota Модель: Land Cruiser  
 Копір: Сирій  
 Номер: 5656

**Працівник:** Райчук Ілля Петрович

**Клієнт:** Зернюк Іван Іванович

Редагувати На головну

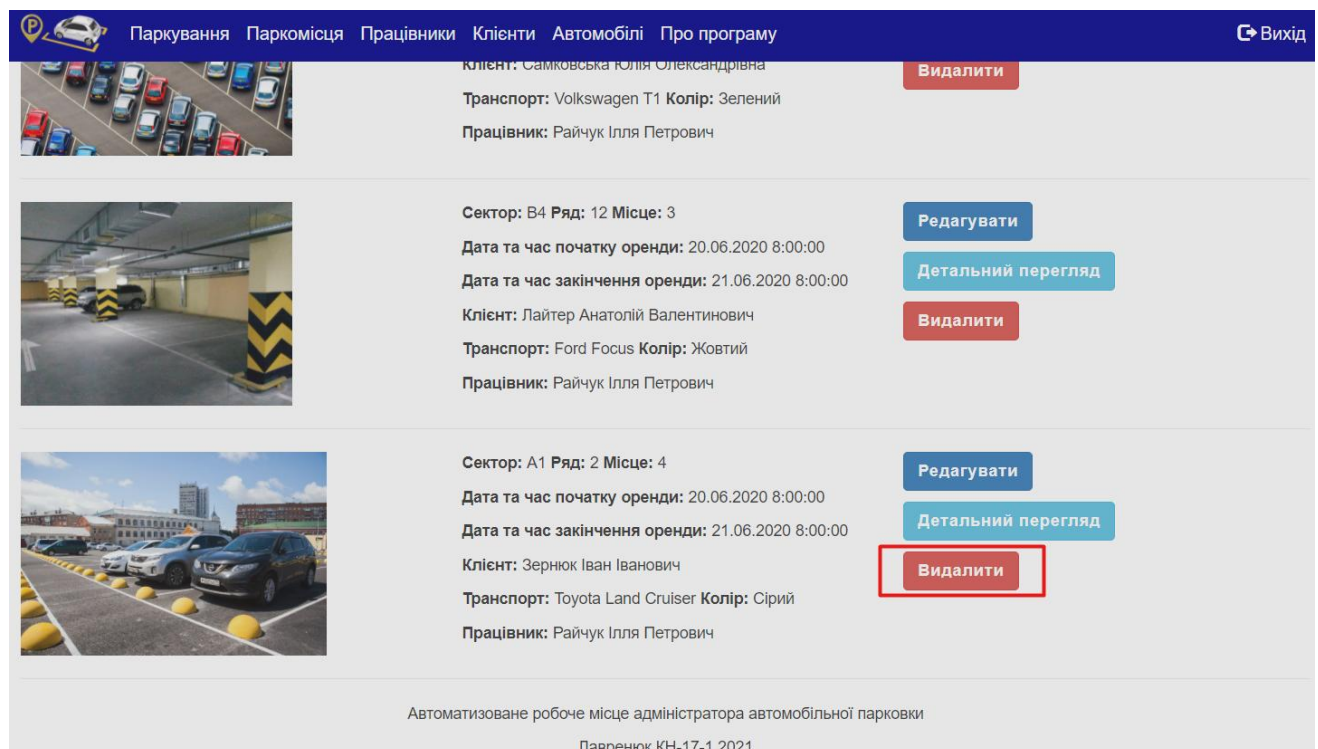
Рисунок 3.7 – Детальний перегляд даних вибраного запису

Четвертий тест-кейс перевіряє можливість видалення запису з таблиці SpaceRentals (таблиця 3.4)

Таблиця 3.4 – Тест-кейс TS0004

<b>Тест-кейс ID:</b> TS0004	<b>Пріоритет:</b> 2	<b>Створено:</b> 15.02.2021,Лавренюк Б.С.
<b>Назва:</b> Тестування видалення запису з таблиці		
<b>Вхідні дані:</b> Останній доданий запис		
<b>Кроки</b>	<b>Очікуваний результат</b>	
<ol style="list-style-type: none"> <li>1. Авторизуватися в системі</li> <li>2. Перейти на сторінку «Паркування»</li> <li>3. Вибрати потрібний запис</li> <li>4. Натиснути на кнопку «Видалити»</li> <li>5. Підтвердити свій вибір</li> </ol>	Видалення запису з бази даних	
<b>Результат виконання тест-кейсу:</b> пройдено успішно		

Результатом проходження тест-кейсу було успішне видалення запису з таблиці. Процес видалення запису зображено на рисунку 3.8.



The screenshot displays a web application interface for car rental management. The interface is divided into a header, a main content area, and a footer. The header contains navigation links: Паркування, Паркомісця, Працівники, Клієнти, Автомобілі, Про програму, and a 'Вихід' (Exit) button. The main content area lists three car rental records. Each record includes a thumbnail image of a car, a 'Видалити' (Delete) button, and a 'Детальний перегляд' (Detailed view) button. The 'Видалити' button for the third record is highlighted with a red box. The footer contains the text: 'Автоматизоване робоче місце адміністратора автомобільної парковки' and 'Лавренюк КН-17-1 2021'.

Клієнт	Транспорт	Колір	Працівник	Дія
Самковська Юлія Олександрівна	Volkswagen T1	Зелений	Райчук Ілля Петрович	Видалити
Лайтер Анатолій Валентинович	Ford Focus	Жовтий	Райчук Ілля Петрович	Видалити
Зернюк Іван Іванович	Toyota Land Cruiser	Сірий	Райчук Ілля Петрович	Видалити

Рисунок 3.8 – Демонстрація видалення запису

За допомогою п'ятого тест-кейсу виконується перевірка можливості авторизації адміністратора в системі (таблиця 3.5). У результаті успішного проходження тест-кейсу перед адміністратором має відкрити головна сторінка веб-додатку, а саме сторінка «Паркування транспорту»

Таблиця 3.5 – Тест-кейс TS0005

<b>Тест-кейс ID:</b> TS0005	<b>Пріоритет:</b> 1	<b>Створено:</b> 15.02.2021, Лавренюк Б.С.
<b>Назва:</b> Перевірка можливості авторизації адміністратора		
<b>Вхідні дані:</b>		
<i>Логін: admin</i>		
<i>Пароль: admin</i>		
<b>Кроки</b>	<b>Очікуваний результат</b>	
<ol style="list-style-type: none"> <li>1. Перейти на початкову сторінку сайту</li> <li>2. Перейти на сторінку «Авторизації»</li> <li>3. Ввести вхідні дані</li> <li>4. Натиснути кнопку «Вхід»</li> </ol>	Успішне проходження авторизації та перехід на «Паркування транспорту»	
<b>Результат виконання тест-кейсу:</b> пройдено успішно		

У результаті проходження тест-кейсу було успішно проведено авторизацію адміністратора в системі. Зображення головної сторінки показано на рисунку 3.5.

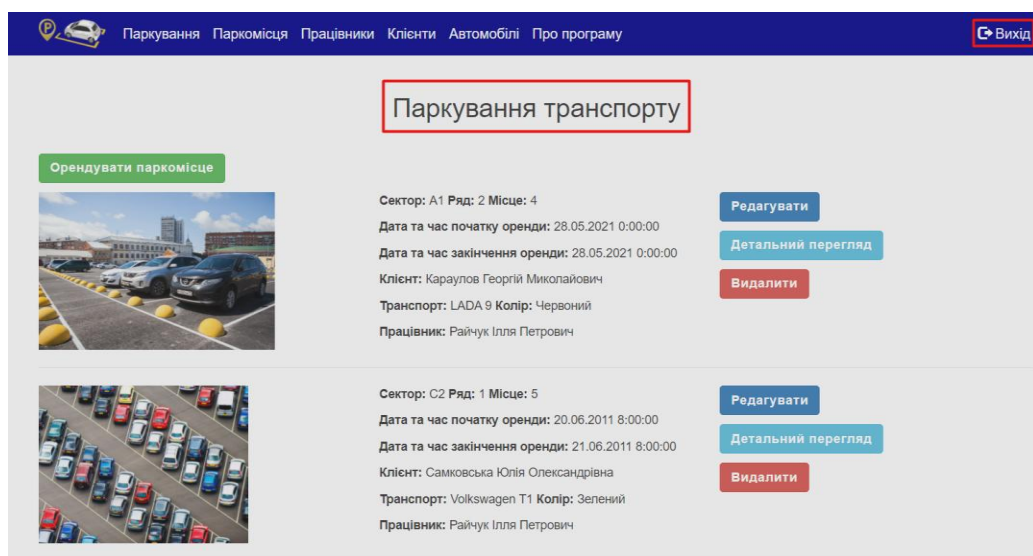


Рисунок 3.9 – Результат успішного проходження тест-кейсу

Протестувавши функції додавання, редагування, видалення та перегляду детальної інформації про запис некоректної поведінки веб-додатку виявлено не було, всі розроблені тест-кейси пройдені успішно.

### 3.4 Інструкція користувача

Після переходу на сайт перед користувачем відкриється стартова сторінка на якій виведена загальна інформація про паркування транспорту (рисунок 3.10).

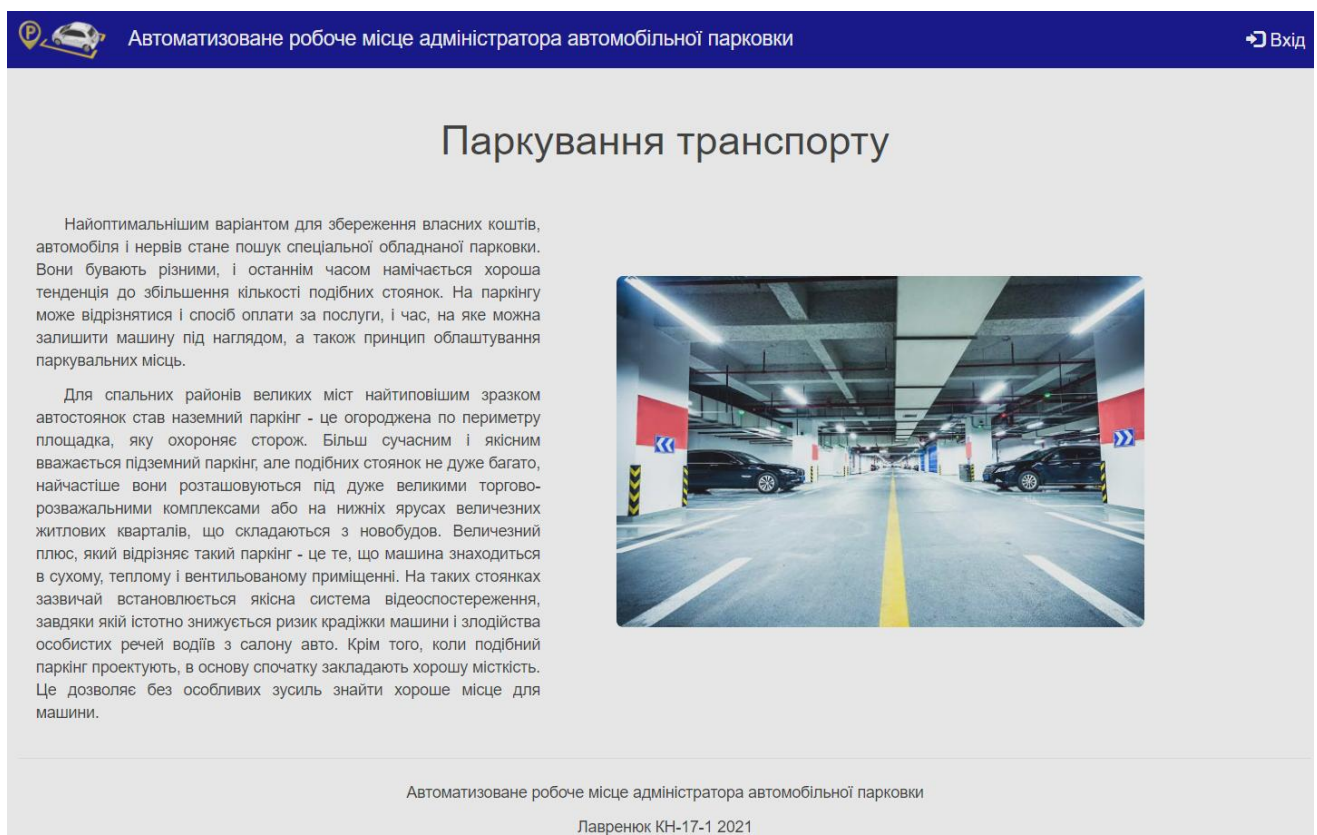


Рисунок 3.10 – Стартова сторінка веб-додатку

В верхньому правому куті екрану розміщена кнопка «Вхід» після її натискання відкривається сторінка авторизації користувача. На завантаженій сторінці потрібно вказати логін та пароль адміністратора. Для проходження авторизації тестовими даними в поля потрібно ввести слово: admin (рисунок 3.11).

Рисунок 3.11 – Сторінка авторизації

Після успішного проходження авторизації на сайті перед адміністратором відкриється сторінка «Паркування транспорту». На завантаженій сторінці можна переглядати інформацію про оренду паркових місць (рисунок 3.12).

Зображення	Сектор: A1 Ряд: 2 Місце: 4	Редагувати
	Дата та час початку оренди: 28.05.2021 0:00:00	Детальний перегляд
	Дата та час закінчення оренди: 28.05.2021 0:00:00	Видалити
	Клієнт: Караулов Георгій Миколайович	
	Транспорт: LADA 9 Колір: Червоний	
	Працівник: Райчук Ілля Петрович	
Зображення	Сектор: C2 Ряд: 1 Місце: 5	Редагувати
	Дата та час початку оренди: 20.06.2011 8:00:00	Детальний перегляд
	Дата та час закінчення оренди: 21.06.2011 8:00:00	Видалити
	Клієнт: Самковська Юлія Олександрівна	
	Транспорт: Volkswagen T1 Колір: Зелений	
	Працівник: Райчук Ілля Петрович	

Рисунок 3.12 – Стрінка «Паркування транспорту»

Також на сторінці «Паркування транспорту» можна орендувати паркомісце, для цього потрібно натиснути на кнопку з назвою «Орендувати місце» в лівому верхньому куті екрану. На сторінці «Оренда паркомісця» вводяться дані необхідні для орендування паркомісця. Після заповнення всіх

полів потрібно натиснути на кнопку «Додати», у результаті чого відбудеться створення нового запису з вказаними параметрами. При натисканні на кнопку «На головну» відбудеться перехід на сторінку «Паркування транспорту» (рисунок 3.13).

Паркування Паркомісця Працівники Клієнти Автомобілі Про програму Вихід

### Оренда паркомісця

Дата та час початку: 20.06.2011 8:00:00

Дата та час закінчення: 21.06.2011 8:00:00

Дата та час початку фактичні: 20.06.2011 8:00:00

Дата та час закінчення фактичні: 21.06.2011 8:00:00

Паркомісця: B4

Працівник: Райчук

Клієнт: Самковська

Транспорт: Toyota

[Додати](#)

[На головну](#)

Автоматизоване робоче місце адміністратора автомобільної парковки  
Лавренко КН-17-1 2021

Рисунок 3.13 – Сторінка «Оренда паркомісця»

Крім того з сторінки «Паркування транспорту» можна перейти на сторінку редагування запису, для цього потрібно натиснути кнопку «Редагувати» виділену темно синім кольором. На сторінці «Редагування» можна змінювати значення полів вибраного запису. Для збереження внесених змін потрібно натиснути на кнопку «Зберегти» (рисунок 3.14). Якщо адміністратор передумав вносити зміни до запису то з сторінки «Редагування» можна повернутися до переліку орендованих пакових місць, для цього потрібно натиснути кнопку «На головну».

Редагувати

Дата та час початку 28.05.2021 0:00:00

Дата та час закінчення 28.05.2021 0:00:00

Дата та час початку фактичні 28.05.2021 0:00:00

Дата та час закінчення фактичні 28.05.2021 0:00:00

Паркомісце A1

Працівник Райчук

Клієнт Караулов

Транспорт 9

Зберегти

На головну

Автоматизоване робоче місце адміністратора автомобільної парковки  
Лавренко КН-17-1 2021

Рисунок 3.14 – Стрінка «Редагування»

Також на сторінці «Паркування транспорту» можна переглянути детальну інформацію про оренду паркового місця, для цього потрібно натиснути на кнопку «Детальний перегляд». На завантаженій сторінці відображається детальна інформація про паркомісце, працівника який виконував додавання запису, клієнта та його автомобіль (рисунок 3.15). Якщо було виявлено, що дані заповненні невірно їх одразу можна відредагувати натиснувши на кнопку «Редагувати». Також з сторінки детального перегляду інформації можна повернутися на головну сторінку сайту натиснувши кнопку «На головну» або на логотип в лівому верхньому куті екрану.

Крім того на сторінці «Паркування транспорту» є можливість видалити непотрібний запис, для цього потрібно натиснути на кнопку «Видалити». Після завантаження сторінки потрібно підтвердити бажання видалити запис натиснувши кнопку «Видалити», якщо адміністратор передумав видалити запис йому потрібно натиснути на кнопку «На головну», або на логотип в верхній частині екрану.

**Детальний перегляд інформації про зайняте паркомісце**

**Паркомісце:**  
 Сектор: В4 Ряд: 12 Місце: 3  
 Дата та час початку оренди: 20.06.2020 8:00:00  
 Дата та час закінчення оренди: 21.06.2020 8:00:00  
 Дата та час початку фактичний: 20.06.2020 8:00:00  
 Дата та час закінчення фактичний: 21.06.2020 8:00:00  
 Опис: Підземний паркінг. Захистить автомобіль від опадів

**Працівник:** Райчук Ілля Петрович

**Автомобіль:**  
 Марка: Ford Модель: Focus  
 Колір: Жовтий  
 Номер: 3412

**Клієнт:** Лайтер Анатолій Валентинович

Редагувати На головну

Рисунок 3.15 – Сторінка з детальною інформацією про запис

На сайті також можна переглянути інформацію про паркомісця, працівників, клієнтів та автомобілі виконавши перехід за відповідним посиланням в верхній частині екрану (рисунок 3.16).

Для кожної з зазначених категорій реалізовані функції додавання, редагування та перегляду детальної інформації про запис.

Паркування Паркомісця Працівники Клієнти Автомобілі Про програму

Рисунок 3.16 – Навігаційна меню сайту

При переході на сторінку «Про програму» перед користувачем буде відображена інформація про розробника та технології які використовувалися при створенні веб-додатку (рисунок 3.17).

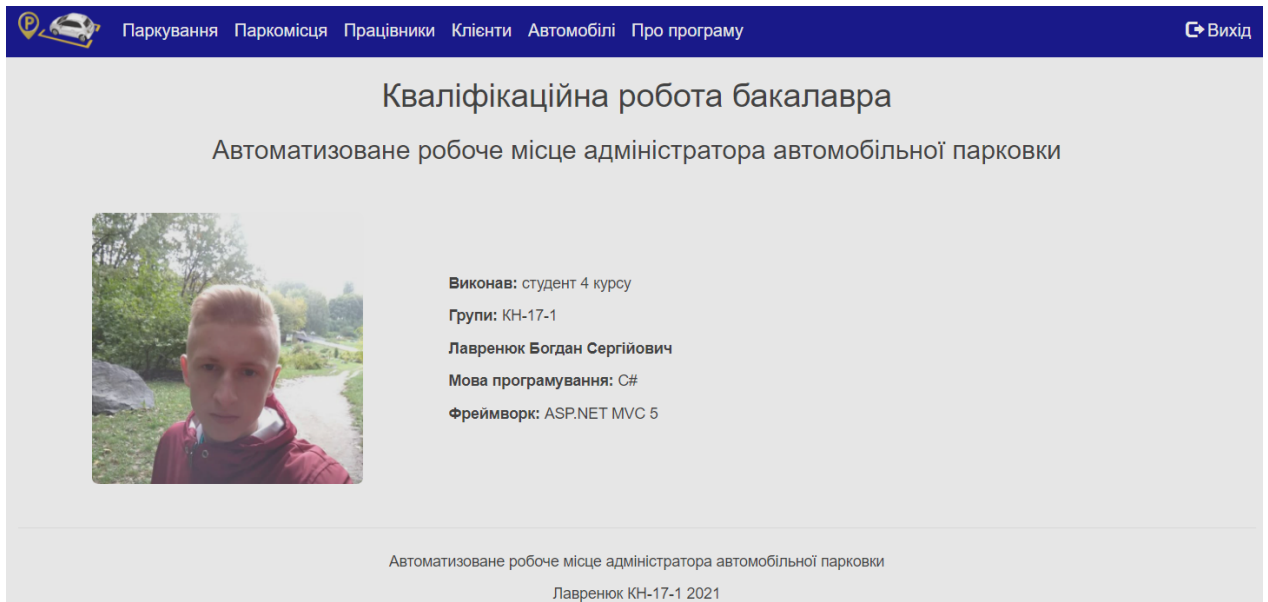


Рисунок 3.17 – Сторінка «Про програму»

Аналізуючи всі особливості розробленого веб-додатку можна зробити висновок, що навігація в даному проекті є інтуїтивно зрозумілою, це зменшує рівень знань необхідних користувачеві перед початком роботи з проектом.

### 3.5 Вимоги до розгортання інформаційної системи

Вимоги до браузера:

- підтримка CSS 3;
- підтримка HTML 5;

Вимоги до провайдера:

- використання серверу IIS;
- підтримка фреймворку ASP.NET MVC 5.

Вимоги до СКБД: MS SQL SERVER 2017.

Мінімальні вимоги до користувача:

- підключення до Інтернету;
- використання пристроїв з великим розширенням екрану (не підтримується мобільна версія).

## Висновки

В результаті виконання кваліфікаційної роботи бакалавра було створено автоматизоване робоче місце адміністратора автомобільної парковки. Розробка проводилася на платформі ASP.NET з використанням фреймворку ASP.NET MVC. Вимоги поставленні до роботи виконанні у повному обсязі, а саме розроблено автоматизоване робоче місце адміністратора автомобільної парковки у вигляді веб-додатку з реалізацією наступних функцій:

- робота з орендами паркувальних місць;
- робота з наявним фондом паркувальних місць (демонстрування переліків паркувальних місць);
- робота з даними клієнтів;
- робота з даними автомобілів;
- робота з даними працівників.

Розроблене автоматизоване робоче місце може використовуватися для підвищення ефективності виконання роботи адміністратора автомобільної парковки.

## Перелік посилань

1. Хронологія розвитку транспортних засобів. URL: <https://cutt.ly/Tng1rmT>
2. Транспортний засіб URL: <https://cutt.ly/3nhoGNL>
3. Види транспорту та їх властивості. URL: [https://www.lobanov-logist.ru/library/all\\_articles/54408/](https://www.lobanov-logist.ru/library/all_articles/54408/)
4. Автомобільний транспорт. URL: <https://cutt.ly/znhaApl>
5. Автомобільний транспорт переваги та недоліки. URL: <https://studfile.net/preview/7293774/page:2/>
6. Загальні положення організації та діяльності автомобільного транспорту. URL: <https://zakon.rada.gov.ua/laws/show/2344-14#Text>
7. Автомобільний транспорт. URL: <http://www.transportall.ru/auto/>
8. Кількість авто на 1000 жителів. URL: <https://www.the-village.com.ua/village/city/city-news/309073-kilkist-avto-v-kievi-perevischila-400-na-1000-zhiteliv>
9. Автостоянка. URL: <https://cutt.ly/Anh0bzY>
10. Що таке паркінг - визначення, особливості та види. URL: <https://zubilovaz.ru/uk/chto-yavlyaetsya-parkovkoi-opredelenie-chto-takoe-parking---opredelenie/>
11. Стоянка. URL: <https://hrk.aero/stoyanka/>
12. Багатоповерхова автостоянка – Multistorey car park. URL: [https://uk.jejakjabar.com/wiki/multistorey\\_car\\_park](https://uk.jejakjabar.com/wiki/multistorey_car_park)
13. Автоматичні паркувальні системи. URL: <https://viramax.ua/uk/automated-parking/>
14. Види механічних парковок і їх особливості. URL: <https://www.igb-parkings.com/about/>
15. Види механічних парковок і їх особливості. URL: <https://ukr.thehouseofchronic.com/4299734-automatic-multi-level-parking-types-of-parking>

16. Парковки та паркінги. Як вирішити проблему парковки у великих містах. URL: <http://smart-parking.com.ua/prensa/101115/center.shtml>
17. Механізована автомобільна стоянка. URL: <https://cutt.ly/hnh6OmO>
18. Класифікація автомобільних стоянок. URL: <https://cutt.ly/Anv7zqL>
19. SMS паркування Харків (949). URL: <https://play.google.com/store/apps/details?id=com.skukunin.rnsm parking>
20. Комунальне підприємство «Харківпарксервіс». URL: <https://parking.kh.city/parking-list/>
21. Parkopedia Parking. URL: <https://play.google.com/store/apps/details?id=com.parkopedia>
22. Засоби створення WEB-сайтів URL: [https://pidruchniki.com/1970070547797/informatika/zasobi\\_stvorennya\\_web-saytiv](https://pidruchniki.com/1970070547797/informatika/zasobi_stvorennya_web-saytiv)
23. ASP.NET. URL: <https://uk.wikipedia.org/wiki/ASP.NET>
24. Що вибрати: ASP.NET чи PHP. URL: [https://skillbox.ru/media/code/chto\\_vybrat\\_asp\\_net\\_ili\\_php/](https://skillbox.ru/media/code/chto_vybrat_asp_net_ili_php/)
25. Що таке PHP. URL: <http://www.lib.mdpu.org.ua/e-book/web/lec1.htm>
26. PHP VS ASP.NET: Як правильно вибрати? URL: <https://www.pixelcrayons.com/blog/web/php-vs-asp-net-how-to-choose-the-right-one/>
27. С# – Переваги та недоліки. URL: <https://shwanoff.ru/plus-minus-c-sharp/>
28. С#. URL: [https://ru.wikipedia.org/wiki/C\\_Sharp](https://ru.wikipedia.org/wiki/C_Sharp)
29. Короткий огляд мови С# URL: <https://docs.microsoft.com/ru-ru/dotnet/csharp/tour-of-csharp/>
30. Особливості ASP.NET MVC URL: <https://metanit.com/sharp/mvc5/1.1.php>
31. Переваги ASP.NET MVC URL: [https://professorweb.ru/my/ASP\\_NET/mvc/level1/1\\_2.php](https://professorweb.ru/my/ASP_NET/mvc/level1/1_2.php)
32. ASP.NET Core URL: [https://ru.wikipedia.org/wiki/ASP.NET\\_Core](https://ru.wikipedia.org/wiki/ASP.NET_Core)
33. ASP.NET WebForms: можливості використання. URL: <https://dotnet.microsoft.com/apps/aspnet/web-forms>

34. ASP.NET MVC vs. WebForms. URL: <https://habr.com/ru/post/47249/>
35. Що таке MySQL. URL: <https://mchost.ru/articles/chto-takoe-mysql/>
36. Основні переваги СКБД MySQL. URL: <https://studfile.net/preview/5607354/page:3/>
37. SQL Server. URL: [https://uk.wikipedia.org/wiki/Microsoft\\_SQL\\_Server](https://uk.wikipedia.org/wiki/Microsoft_SQL_Server)
38. Реляційні СКБД – порівняння MySQL та SQL-сервер. URL: <https://www.hostinger.com.ua/rukovodstva/reljacionnye-subd-sravnenie-mysql-i-sql-server/>

# ДОДАТКИ

## Додаток А

### Програмні коди

#### AccountController.cs:

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;
using System.Web.Security;
using Parking.Models;

namespace Parking.Controllers
{
    public class AccountController : Controller
    {
        ParkingEntities4 context = new ParkingEntities4();
        public ActionResult Login()
        {
            return View();
        }

        [HttpPost]
        [ValidateAntiForgeryToken]
        public ActionResult Login(Login model)
        {
            if (ModelState.IsValid)
            {
                Employees employees = null;
                using (ParkingEntities4 db = new ParkingEntities4())
                {
                    employees = db.Employees.FirstOrDefault(u => u.Login == model.Name &&
u.Password == model.Password);
                }
                if ( employees != null)
                {
                    FormsAuthentication.SetAuthCookie(model.Name, true);
                    if (employees != null )
                        return RedirectToAction("Index", "SpaceRentals");
                }
                else
                {
                    ModelState.AddModelError("", "Користувача з таким логіном та паролем
немає");
                }
            }
            return View(model);
        }

        public ActionResult Exit()
        {
            FormsAuthentication.SignOut();
            return RedirectToAction("Index", "Home");
        }
    }
}

```

#### CarsController.cs

```

using System;

```

```

using System.Collections.Generic;
using System.Data;
using System.Data.Entity;
using System.IO;
using System.Linq;
using System.Net;
using System.Web;
using System.Web.Mvc;
using Parking.Models;

namespace Parking.Controllers
{
    public class CarsController : Controller
    {
        private ParkingEntities4 db = new ParkingEntities4();

        public ActionResult Index()
        {
            var cars = db.Cars.Include(c => c.Types);
            return View(cars.ToList());
        }

        public ActionResult Details(int? id)
        {
            if (id == null)
            {
                return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
            }
            Cars cars = db.Cars.Find(id);
            if (cars == null)
            {
                return HttpNotFound();
            }
            return View(cars);
        }

        public ActionResult Create()
        {
            ViewBag.FK_Type = new SelectList(db.Types, "ID", "Name");
            return View();
        }

        [HttpPost]
        [ValidateAntiForgeryToken]
        public ActionResult Create([Bind(Include =
"ID,Photo,Model,Brand,Color,CarNumber,FK_Type")] Cars cars, HttpPostedFileBase uploadImage)
        {
            if (ModelState.IsValid && uploadImage != null)
            {
                byte[] imageData = null;
                using (var binaryReader = new BinaryReader(uploadImage.InputStream))
                {
                    imageData = binaryReader.ReadBytes(uploadImage.ContentLength);
                }
                cars.Photo = imageData;
                db.Cars.Add(cars);
                db.SaveChanges();
                return RedirectToAction("Index", "Cars");
            }
            else
                ViewBag.Message = "Будь ласка заповніть всі поля";

            ViewBag.FK_Type = new SelectList(db.Types, "ID", "Name", cars.FK_Type);
            return View(cars);
        }

        public ActionResult Edit(int? id)
        {
            if (id == null)

```

```

    {
        return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
    }
    Cars cars = db.Cars.Find(id);
    if (cars == null)
    {
        return HttpNotFound();
    }
    ViewBag.FK_Type = new SelectList(db.Types, "ID", "Name", cars.FK_Type);
    return View(cars);
}
[HttpPost]
[ValidateAntiForgeryToken]
public ActionResult Edit([Bind(Include =
"ID,Photo,Model,Brand,Color,CarNumber,FK_Type")] Cars cars, HttpPostedFileBase uploadImage)
{
    if (ModelState.IsValid)
    {
        if (uploadImage != null)
        {
            byte[] imageData = null;
            using (var binaryReader = new BinaryReader(uploadImage.InputStream))
            {
                imageData = binaryReader.ReadBytes(uploadImage.ContentLength);
            }
            cars.Photo = imageData;
        }
        db.Entry(cars).State = EntityState.Modified;
        db.SaveChanges();
        return RedirectToAction("Index");
    }
    ViewBag.FK_Type = new SelectList(db.Types, "ID", "Name", cars.FK_Type);
    return View(cars);
}

protected override void Dispose(bool disposing)
{
    if (disposing)
    {
        db.Dispose();
    }
    base.Dispose(disposing);
}
}
}

```

### SpaceRentalsController.cs

```

using System;
using System.Collections.Generic;
using System.Data;
using System.Data.Entity;
using System.Linq;
using System.Net;
using System.Web;
using System.Web.Mvc;
using Parking.Models;

namespace Parking.Controllers
{
    public class SpaceRentalsController : Controller
    {
        private ParkingEntities4 db = new ParkingEntities4();
        public ActionResult Index()
        {

```

```

        var spaceRental = db.SpaceRental.Include(s => s.Cars).Include(s =>
s.Customers).Include(s => s.Employees).Include(s => s.Places);
        return View(spaceRental.ToList());
    }
    public ActionResult Details(int? id)
    {
        if (id == null)
        {
            return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
        }
        SpaceRental spaceRental = db.SpaceRental.Find(id);
        if (spaceRental == null)
        {
            return HttpNotFound();
        }
        return View(spaceRental);
    }
    public ActionResult Create()
    {
        ViewBag.FK_Car = new SelectList(db.Cars, "ID", "Brand");
        ViewBag.FK_Customer = new SelectList(db.Customers, "ID", "Surname");
        ViewBag.FK_Employee = new SelectList(db.Employees, "ID", "Surname");
        ViewBag.FK_Place = new SelectList(db.Places, "ID", "Sector");
        return View();
    }
    [HttpPost]
    [ValidateAntiForgeryToken]
    public ActionResult Create([Bind(Include =
"ID,StartDateTime,EndDateTime,StartDateTimeActual,EndDateTimeActual,FK_Place,FK_Employee,FK_C
ustomer,FK_Car")] SpaceRental spaceRental)
    {
        if (ModelState.IsValid)
        {
            db.SpaceRental.Add(spaceRental);
            db.SaveChanges();
            return RedirectToAction("Index");
        }

        ViewBag.FK_Car = new SelectList(db.Cars, "ID", "Model", spaceRental.FK_Car);
        ViewBag.FK_Customer = new SelectList(db.Customers, "ID", "Surname",
spaceRental.FK_Customer);
        ViewBag.FK_Employee = new SelectList(db.Employees, "ID", "Surname",
spaceRental.FK_Employee);
        ViewBag.FK_Place = new SelectList(db.Places, "ID", "Description",
spaceRental.FK_Place);
        return View(spaceRental);
    }

    public ActionResult Edit(int? id)
    {
        if (id == null)
        {
            return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
        }
        SpaceRental spaceRental = db.SpaceRental.Find(id);
        if (spaceRental == null)
        {
            return HttpNotFound();
        }
        ViewBag.FK_Car = new SelectList(db.Cars, "ID", "Model", spaceRental.FK_Car);
        ViewBag.FK_Customer = new SelectList(db.Customers, "ID", "Surname",
spaceRental.FK_Customer);
        ViewBag.FK_Employee = new SelectList(db.Employees, "ID", "Surname",
spaceRental.FK_Employee);
        ViewBag.FK_Place = new SelectList(db.Places, "ID", "Sector",
spaceRental.FK_Place);
    }

```

```

        return View(spaceRental);
    }
    [HttpPost]
    [ValidateAntiForgeryToken]
    public ActionResult Edit([Bind(Include =
"ID,StartDateTime,EndDateTime,StartDateTimeActual,EndDateTimeActual,FK_Place,FK_Employee,FK_C
ustomer,FK_Car")] SpaceRental spaceRental)
    {
        if (ModelState.IsValid)
        {
            db.Entry(spaceRental).State = EntityState.Modified;
            db.SaveChanges();
            return RedirectToAction("Index");
        }
        ViewBag.FK_Car = new SelectList(db.Cars, "ID", "Model", spaceRental.FK_Car);
        ViewBag.FK_Customer = new SelectList(db.Customers, "ID", "Surname",
spaceRental.FK_Customer);
        ViewBag.FK_Employee = new SelectList(db.Employees, "ID", "Surname",
spaceRental.FK_Employee);
        ViewBag.FK_Place = new SelectList(db.Places, "ID", "Sector",
spaceRental.FK_Place);
        return View(spaceRental);
    }

    public ActionResult Delete(int? id)
    {
        if (id == null)
        {
            return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
        }
        SpaceRental spaceRental = db.SpaceRental.Find(id);
        if (spaceRental == null)
        {
            return HttpNotFound();
        }
        return View(spaceRental);
    }
    [HttpPost, ActionName("Delete")]
    [ValidateAntiForgeryToken]
    public ActionResult DeleteConfirmed(int id)
    {
        SpaceRental spaceRental = db.SpaceRental.Find(id);
        db.SpaceRental.Remove(spaceRental);
        db.SaveChanges();
        return RedirectToAction("Index");
    }

    protected override void Dispose(bool disposing)
    {
        if (disposing)
        {
            db.Dispose();
        }
        base.Dispose(disposing);
    }
}
}

```

### CustomRoleProvider.cs

```

using Parking.Models;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Security;

```

```

namespace Parking.Providers
{
    public class CustomRoleProvider : RoleProvider
    {
        public override string ApplicationName { get => throw new NotImplementedException();
set => throw new NotImplementedException(); }

        public override void AddUsersToRoles(string[] usernames, string[] roleNames)
        {
            throw new NotImplementedException();
        }

        public override void CreateRole(string roleName)
        {
            throw new NotImplementedException();
        }

        public override bool DeleteRole(string roleName, bool throwOnPopulatedRole)
        {
            throw new NotImplementedException();
        }

        public override string[] FindUsersInRole(string roleName, string usernameToMatch)
        {
            throw new NotImplementedException();
        }

        public override string[] GetAllRoles()
        {
            throw new NotImplementedException();
        }

        public override string[] GetRolesForUser(string username)
        {
            string[] roles = new string[] { };
            using (ParkingEntities4 db = new ParkingEntities4())
            {
                Employees employees = db.Employees.FirstOrDefault(u => u.Login == username);

                if (employees != null )
                {
                    roles = new string[] { "Admin" };
                }

                return roles;
            }
        }

        public override string[] GetUsersInRole(string roleName)
        {
            throw new NotImplementedException();
        }

        public override bool IsUserInRole(string username, string roleName)
        {
            bool outputResult = false;

            using (ParkingEntities4 db = new ParkingEntities4())
            {
                Получаем пользователя
                Club_members user = db.Club_members.FirstOrDefault(u => u.Email == username);

                if (user != null)
                {

```

```

        Role role = db.Role.Find(user.FK_Role);
        if (role != null && role.Name == roleName)
            outputResult = true;
    }
    return outputResult;
}
}

public override void RemoveUsersFromRoles(string[] usernames, string[] roleNames)
{
    throw new NotImplementedException();
}

public override bool RoleExists(string roleName)
{
    throw new NotImplementedException();
}
}
}

```

### Login.cs

```

using System;
using System.Collections.Generic;
using System.ComponentModel.DataAnnotations;
using System.Linq;
using System.Web;

namespace Parking.Models
{
    public class Login
    {
        [Required]
        public string Name { get; set; }

        [Required]
        [DataType(DataType.Password)]
        public string Password { get; set; }
    }
}

```

### Create.cshtml

```

@model Parking.Models.Cars

@{
    ViewBag.Title = "Додати";
}

<h2>Додати транспорт</h2>

@using (Html.BeginForm("Create", "Cars", FormMethod.Post, new { enctype = "multipart/form-data" }))
{
    @Html.AntiForgeryToken()

    <div class="form-horizontal">

        <hr />
        <h5 style="color:red">@ViewBag.Message</h5>
        <div class="form-group">
            @Html.Label("Картинка", htmlAttributes: new { @class = "control-label col-md-2" })
            <div class="row control-label col-md-2">
                <div class="col-md-4"> <input type="file" name="uploadImage" /></div>

```

```

        </div>

    </div>

    <div class="form-group">
        @Html.LabelFor(model => model.Model, "Модель", htmlAttributes: new { @class =
"control-label col-md-2" })
        <div class="col-md-10">
            @Html.EditorFor(model => model.Model, new { htmlAttributes = new { @class =
"form-control" } })
            @Html.ValidationMessageFor(model => model.Model, "", new { @class = "text-danger"
})
        </div>
    </div>

    <div class="form-group">
        @Html.LabelFor(model => model.Brand, "Марка", htmlAttributes: new { @class =
"control-label col-md-2" })
        <div class="col-md-10">
            @Html.EditorFor(model => model.Brand, new { htmlAttributes = new { @class =
"form-control" } })
            @Html.ValidationMessageFor(model => model.Brand, "", new { @class = "text-danger"
})
        </div>
    </div>

    <div class="form-group">
        @Html.LabelFor(model => model.Color, "Колір", htmlAttributes: new { @class =
"control-label col-md-2" })
        <div class="col-md-10">
            @Html.EditorFor(model => model.Color, new { htmlAttributes = new { @class =
"form-control" } })
            @Html.ValidationMessageFor(model => model.Color, "", new { @class = "text-danger"
})
        </div>
    </div>

    <div class="form-group">
        @Html.LabelFor(model => model.CarNumber, "Номер транспорту", htmlAttributes: new {
@class = "control-label col-md-2" })
        <div class="col-md-10">
            @Html.EditorFor(model => model.CarNumber, new { htmlAttributes = new { @class =
"form-control" } })
            @Html.ValidationMessageFor(model => model.CarNumber, "", new { @class = "text-
danger" })
        </div>
    </div>

    <div class="form-group">
        @Html.LabelFor(model => model.FK_Type, "Тип", htmlAttributes: new { @class =
"control-label col-md-2" })
        <div class="col-md-10">
            @Html.DropDownList("FK_Type", null, htmlAttributes: new { @class = "form-control"
})
            @Html.ValidationMessageFor(model => model.FK_Type, "", new { @class = "text-
danger" })
        </div>
    </div>

    <div class="form-group">
        <div class="col-md-offset-2 col-md-10">
            <input type="submit" value="Додати" class="btn btn-success" />

        </div>
    </div>
</div class="form-group">

```

```

        <div class="col-md-offset-2 col-md-10">
            @Html.ActionLink(" На голову", "Index", "Cars", new { ID = 0 }, new { @class =
"btn btn-primary", @style = " font-size: 11pt;" })
        </div>
    </div>
</div>
<hr />
}

```

### PlacesController.cs

```

using System;
using System.Collections.Generic;
using System.Data;
using System.Data.Entity;
using System.IO;
using System.Linq;
using System.Net;
using System.Web;
using System.Web.Mvc;
using Parking.Models;

namespace Parking.Controllers
{
    public class PlacesController : Controller
    {
        private ParkingEntities4 db = new ParkingEntities4();
        public ActionResult Index()
        {
            var places = db.Places.Include(p => p.Types);
            return View(places.ToList());
        }
        public ActionResult Details(int? id)
        {
            if (id == null)
            {
                return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
            }
            Places places = db.Places.Find(id);
            if (places == null)
            {
                return HttpNotFound();
            }
            return View(places);
        }
        public ActionResult Create()
        {
            ViewBag.FK_Type = new SelectList(db.Types, "ID", "Name");
            return View();
        }
        [HttpPost]
        [ValidateAntiForgeryToken]
        public ActionResult Create([Bind(Include =
"ID,Photo,Description,RowNumber,PlaceNumber,FK_Type,Sector")] Places places,
HttpPostedFileBase uploadImage)
        {

            if (ModelState.IsValid && uploadImage != null)
            {
                byte[] imageData = null;
                using (var binaryReader = new BinaryReader(uploadImage.InputStream))
                {

```

```

        imageData = binaryReader.ReadBytes(uploadImage.ContentLength);
    }
    places.Photo = imageData;

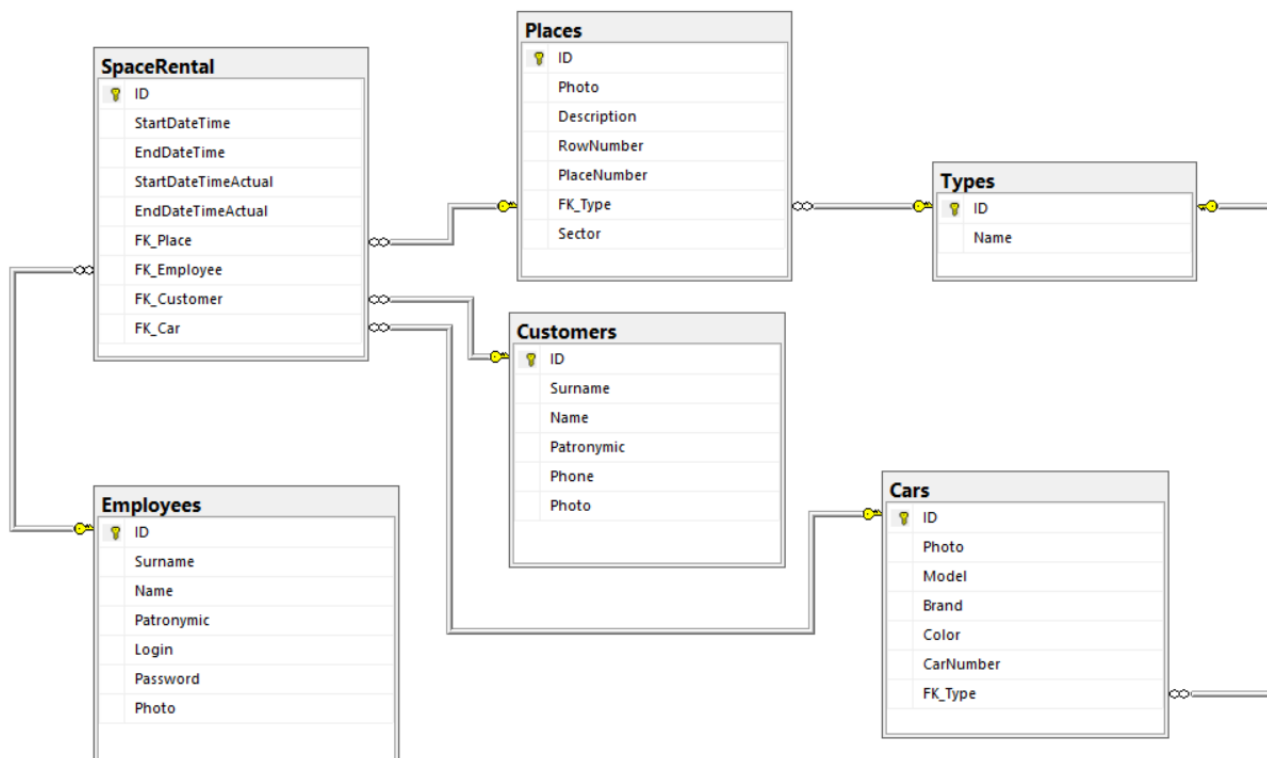
    db.Places.Add(places);
    db.SaveChanges();
    return RedirectToAction("Index", "Places");
}
else
    ViewBag.Message = "Будь ласка заповніть всі поля";
ViewBag.FK_Type = new SelectList(db.Types, "ID", "Name", places.FK_Type);
return View(places);
}
public ActionResult Edit(int? id)
{
    if (id == null)
    {
        return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
    }
    Places places = db.Places.Find(id);
    if (places == null)
    {
        return HttpNotFound();
    }
    ViewBag.FK_Type = new SelectList(db.Types, "ID", "Name", places.FK_Type);
    return View(places);
}

[HttpPost]
[ValidateAntiForgeryToken]
public ActionResult Edit([Bind(Include =
"ID,Photo,Description,RowNumber,PlaceNumber,FK_Type,Sector")] Places places)
{
    if (ModelState.IsValid)
    {
        db.Entry(places).State = EntityState.Modified;
        db.SaveChanges();
        return RedirectToAction("Index");
    }
    ViewBag.FK_Type = new SelectList(db.Types, "ID", "Name", places.FK_Type);
    return View(places);
}

protected override void Dispose(bool disposing)
{
    if (disposing)
    {
        db.Dispose();
    }
    base.Dispose(disposing);
}
}
}
}

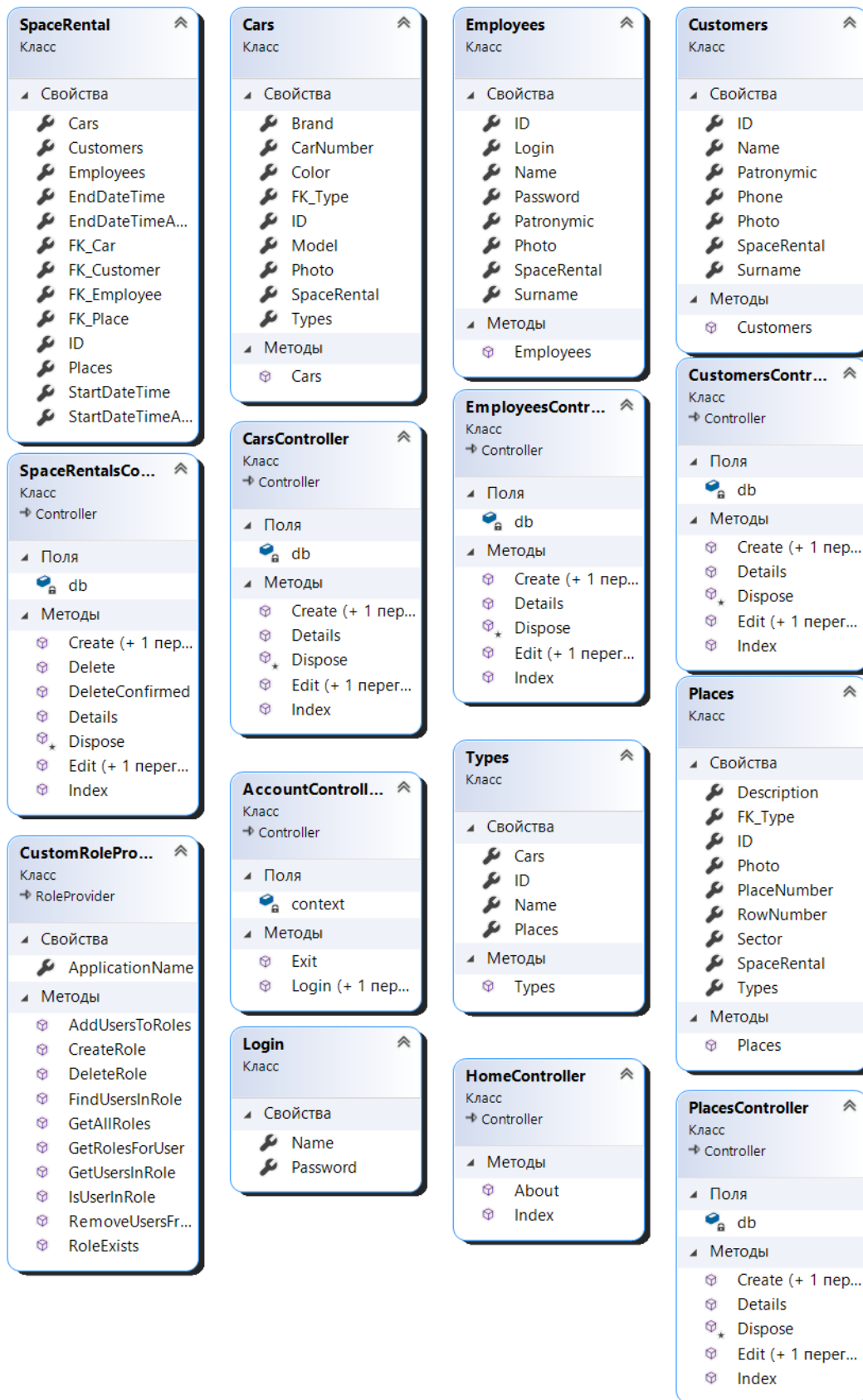
```

## Додаток Б

Структура бази даних автоматизованого робочого місця адміністратора  
автомобільної парковки

## Додаток В

## Розгорнута структура класів автоматизованого робочого місця адміністратора автомобільної парковки



## Додаток Г

### Презентаційний матеріал

# КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

## АВТОМАТИЗОВАНЕ РОБОЧЕ МІСЦЕ АДМІНІСТРАТОРА АВТОМОБІЛЬНОЇ ПАРКОВКИ

Виконав: студент 4 курсу групи КН-17-1 Лавренюк Б.С.

Керівник: старший викладач кафедри КНІТ Скрипник Т.К.

## Актуальність

Автомобільний транспорт є абсолютним лідером з кількості пасажирських перевезень, що відповідно призводить до збільшення кількості автомобілів. Для забезпечення ефективного функціонування автомобільного транспорту потрібна наявність відповідної інфраструктури, а саме: доріг, транспортних вузлів, ремонтних сервісів та паркувальних місць. Забезпечення потрібної кількості парковок та організації їх ефективної роботи є особливо актуальною в умовах постійного збільшення кількості автомобілів

При використанні паркувального місця клієнт фактично виконує його оренду терміни якої фіксуються робітником паркінгу (адміністратором). В процесі оформлення оренди паркувального місця також зазначаються дані клієнта, а саме: прізвище, ім'я, по батькові, номер телефону та прикріплюється фотографія, для достовірної ідентифікації клієнта. Окрім цього працівник вказує дані про автомобіль клієнта, а саме: модель, марку, колір, номерний знак, тип та прикріплює відповідне фото.

Враховуючи об'єм інформації якою повинний оперувати адміністратор парковки потрібно виконати автоматизацію процесу його роботи. Для чого доцільно реалізувати веб-додаток, який значно спростить процес орендування паркувальних місць та зменшить об'єм навантаження для адміністратора.

## Завдання

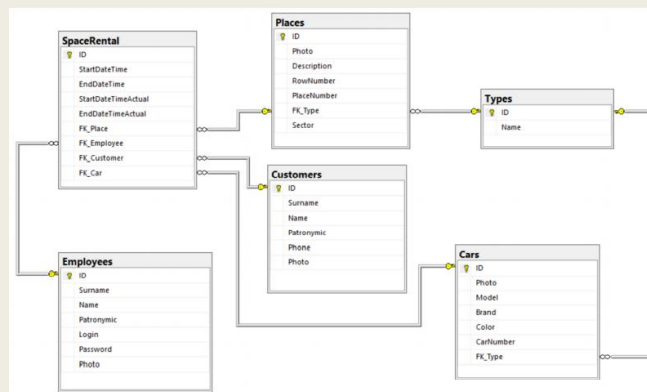
*Метою роботи є розробка автоматизованого робочого місця адміністратора автомобільної парковки.*

**З наступними функція:**

- робота з орендами паркувальних місць;
- робота з наявним фондом паркувальних місць;
- робота з даними клієнтів;
- робота з даними автомобілів;
- робота з даними працівників.

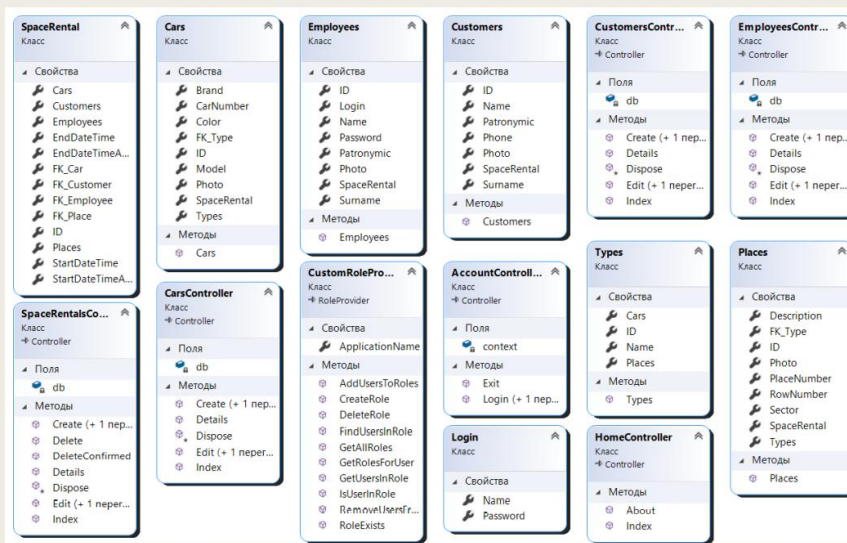
## Схема бази даних

У відповідності з поставленими завданнями для реалізації автоматизованого робочого місця адміністратора автомобільної парковки була спроектована та розроблена структура бази даних веб-додатку.



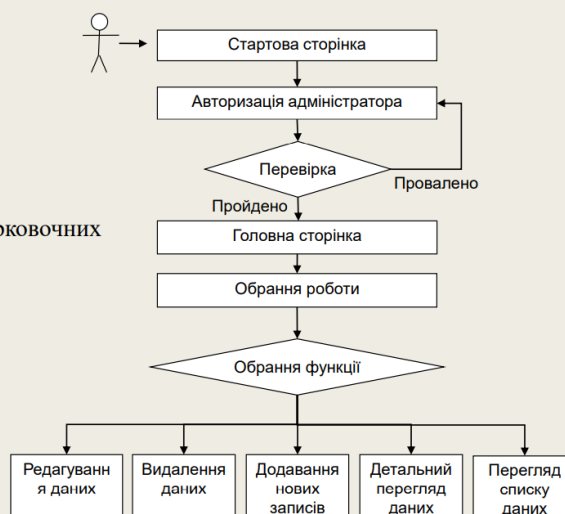
## Діаграма класів

На основі розробленої структури веб-додатку було створено діаграму класів, яка буде використовуватися при створенні проекту.



## Діаграма активності роботи адміністратора

Адміністратору автомобільної парковки потрібно реалізувати наступні функції: робота з орендою паркувальних місць; робота з наявним фондом парковочних місць; робота з даними клієнтів; робота з даними автомобілів; робота з даними працівників



## Головна сторінка веб-додатку

Автоматизоване робоче місце адміністратора автомобільної парковки Вхід

### Паркування транспорту

Найотимальнішим варіантом для збереження власних коштів, автомобіля і неріві стане пошук спеціальної обладнаної парковки. Вони бувають різними, і останнім часом намічається хороша тенденція до збільшення кількості подібних стоянок. На паркінгу може відрізнятися і спосіб оплати за послуги, і час, на яке можна залишити машину під наглядом, а також принцип облаштування паркувальних місць.

Для спальних районів великих міст найідеальнішим зразком автостоянок став наземний паркінг – це огорожена по периметру площа, яку охороняє сторож. Більш сучасним і якісним вважається підземний паркінг, але подібних стоянок не дуже багато, найчастіше вони розташовуються під дуже великими торговельно-офісними комплексами або на нижніх ярусах величезних житлових кварталів, що осядаються з новобудов. Величезний плюс, який відрізняє такий паркінг – це те, що машина знаходиться в сухому, теплому і вентильованому приміщенні. На таких стоянках зазвичай встановлюється якісна система відеоспостереження, завдяки якій істотно зникає ризик крадіжки машини і згодістю особистих речей водіїв з салону авто. Крім того, коли подібний паркінг проєктують, в основу спочатку закладають хорошу місцевість. Це дозволяє без особливих зусиль знайти хороше місце для машини.



Автоматизоване робоче місце адміністратора автомобільної парковки  
Лавренко КН-17-1 2021

## Перегляд списків клієнтів та їхнього транспорту

Паркування Парковки Працівники Клієнти Автомобілі Про програму Вхід

### Клієнти

[Додати клієнта](#)

Клієнт: Караулов Георгій Михайлович  
Номер телефону: 0362341231

[Редагувати](#)  
[Детальний перегляд](#)

---

Клієнт: Пайтер Анастасій Валентинович  
Номер телефону: 0500124870

Паркування Парковки Працівники Клієнти Автомобілі Про програму Вхід

### Транспорт

[Додати транспорт](#)

Марка: LADA Модель: 9  
Номер: 5637  
Колір: Червоний

[Редагувати](#)  
[Детальний перегляд](#)

---

Марка: Ford Модель: Focus  
Номер: 9412  
Колір: Жовтий

[Редагувати](#)  
[Детальний перегляд](#)

## Перегляд детальної інформації про оренду паркувального місця

**Детальний перегляд інформації про зайняте паркомісце**

**Паркомісце:**

Сектор: 04 Ряд: 12 Місце: 3  
 Дата та час початку оренди: 20.06.2020 8:00:00  
 Дата та час закінчення оренди: 21.06.2020 8:00:00  
 Дата та час початку фактичної: 20.06.2020 8:00:00  
 Дата та час закінчення фактичної: 21.06.2020 8:00:00  
 Опис: Підземний паркінг. Захистити автомобіль від оглядів

**Автомобіль:**

Марка: Ford Модель: Focus  
 Колір: Жовтий  
 Номер: 3412

**Працівник:** Райчук Ілля Петрович

**Клієнт:** Лайтер Анатолій Валентинович

Buttons: Редагувати, На головну

## Висновки

В результаті виконання кваліфікаційної роботи бакалавра було створено автоматизоване робоче місце адміністратора автомобільної парковки. Розробка проводилася на платформі ASP.NET з використанням фреймворку ASP.NET MVC. Вимоги поставленні до роботи виконанні у повному обсязі, а саме розроблено автоматизоване робоче місце адміністратора автомобільної парковки у вигляді веб-додатку.

Розроблене автоматизоване робоче місце може використовуватися для підвищення ефективності виконання роботи адміністратора автомобільної парковки.

# ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

## ГОЛОВІ ЕКЗАМЕНАЦІЙНОЇ КОМІСІЇ

Направляється студент Лавренюк Б. С. на захист дипломного проекту (роботи)  
(прізвище, ініціали)  
 за спеціальністю 122 - Комп'ютерні науки

На тему: Автоматизоване робоче місце адміністратора автомобільної парковки

Дипломний проект (робота), рецензія і довідка про перевірку на плагіат додаються.

Декан факультету



**САВЕНКО О.С.**

(прізвище та ініціали)

### ДОВІДКА УСПІШНОСТІ

Лавренюк Б. С. за період навчання на факультеті програмування та комп'ютерних і телекомунікаційних систем з 2017 по 2021 роки повністю виконав навчальний план спеціальності з такими розподілом оцінок за:  
 національною шкалою: відмінно 0,00 %, добре 0,00 %, задовільно 100,00.  
 шкалою ЄКТС: А 1,82 %, В 0,00 %, С 0,00 %, D 7,27 %, Е 90,91 %.

Методист факультету

*[Handwritten signature]*

(підпис)

(прізвище та ініціали)

### ВИСНОВОК КЕРІВНИКА ДИПЛОМНОГО ПРОЕКТУ (РОБОТИ) ТА ОБГРУНТУВАННЯ ОЦІНКИ

Студент Лавришук Ф.С. виконав кваліфікаційну роботу бакалавра на тему "Автоматизоване робоче місце адміністратора автомобільної парковки" у відповідності до завдання в повному обсязі.

Оцінка дипломного проекту (роботи) добре

Керівник дипломного проекту (роботи)

*[Handwritten signature]*

(підпис)

Сарипшик Ф.К.

(прізвище та ініціали)

" 08 " червня 2021 р.

### ВИСНОВОК КАФЕДРИ ПРО ДИПЛОМНИЙ ПРОЕКТ (РОБОТУ)

Дипломний проект (роботу) розглянуто. Студент Лавренюк Б. С. допускається до захисту цього

Завідувач кафедри

ХНУІТ

(назва)

*[Handwritten signature]*

(підпис, прізвище, ініціали)

" 09 " червня 2021 р.

08.06.2021

result\_6286528320802477475.html

Tue Jun 08 12:41:42 EEST 2021, Петровський Сергій Степанович, Хмельницький національний університет, ХНУ

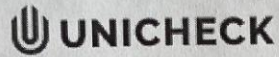
**Anti-Plagiarism v-15.257****Максимальное совпадение с одним документом 2.0%**

Словари проверки: en\_US, ru\_RU, ua\_UA. Ошибок в документах: 13%

ID: 92689 Название: Автоматизоване робоче місце адміністратора автомобільної парковки Добавлено в БД: 2021-06-08 Авторы: Б.С. Лавренюк Руководители: Т.К. Скрипник Консультанты: Опоненты:	Документ		Суммарное совпадение по Базе Данных	
	Символы	Лексемы	Символы	Лексемы
	51348	487	2305 (4%)	32 (7%)

## Источник плагиата

ID	Описание	Наличие плагиата в документе	
		Символы	Лексемы



Ім'я користувача:  
Кафедра КН

Дата перевірки:  
08.06.2021 13:33:47 EEST

Дата звіту:  
10.06.2021 18:06:38 EEST

ID перевірки:  
1008228639

Тип перевірки:  
Doc vs Internet + Library

ID користувача:  
100005671

Назва документа: 2021 КРБ Лавренюк 20210607 2 фінал Lite

Кількість сторінок: 54 Кількість слів: 7715 Кількість символів: 59475 Розмір файлу: 9.66 MB ID файлу: 1008302248

Виявлено модифікації тексту (можуть впливати на відсоток схожості)

9.42%  
Схожість

Найбільша схожість: 2.8% з джерелом з Бібліотеки (ID файлу: 1005675070)

5.16% Джерела з Інтернету 192

Сторінка 56

5.56% Джерела з Бібліотеки 40

Сторінка 56

0% Цитат

Вилучення цитат вимкнено

Вилучення списку бібліографічних посилань вимкнено

0%  
Вилучень

Немає вилучених джерел

Модифікації

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Підозріле форматування 19 сторінок

РІШЕННЯ ЕКСПЕРНОЇ КОМІСІЇ  
КАФЕДРИ КОМП'ЮТЕРНИХ НАУК ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ  
ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

Підтверджуємо ознайомлення з результатом звіту подібності щодо роботи, генерованого системою виявлення текстових збігів/ідентичності/схожості:

Назва: Автоматизоване робоче місце адміністратора автомобільної парковки

Автор: Лавренюк Богдан Сергійович

Спеціальність: 122 – Компютерні науки

Освітня програма: освітньо-професійна

Науковий керівник: Скрипник Тетяна Казимирівна, ст.викладач

Після аналізу звіту подібності зроблено такий висновок:

№	Висновок	Позначка про відповідність
1	Запозичення, виявлені в роботі, є законними і не є плагіатом. Робота приймається до захисту.	відповідає
2	Виявлені запозичення не є плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи. Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої та електронної версії роботи	
3	Виявлені запозичення не є плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнені. Робота може бути допущена до захисту (наступного року) після того як буде відкоригована та допрацьована і успішно пройде повторну перевірку на академічний плагіат.	
4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	

Підтвердження:

Запозичення, виявлені в роботі, є законними і не є плагіатом, оскільки:

- 1) запозичення розміщені в розділах аналізу існуючих аналогів та прототипів, які не описують безпосередньо авторське дослідження і не стосуються результатів роботи;
- 2) усі запозичення фрагментарні;
- 3) до запозичень входять фрагменти програмного коду, що на мають авторства і містять поширені конструкції;
- 4) серед запозичень знаходяться загальновідомі терміни, скорочення та визначення.

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ідентичності/схожості, складає 9,42 і адресується до першоджерел, що, з урахуванням наведених обґрунтувань, відповідає характеру наукового дослідження і свідчить на користь кваліфікаційної роботи.

Керівник роботи

\_\_\_\_\_

Т.К.Скрипник

Гарант ОП

\_\_\_\_\_

О. В. Мазурець

Завідувач кафедри КІСП

\_\_\_\_\_

О. В. Бармак

Хмельницький національний університет  
Кафедра комп'ютерних наук та інформаційних технологій

## РЕЦЕНЗІЯ

### на кваліфікаційну роботу

студента групи КН-17-1 Лавренюка Богдана Сергійовича

за темою Автоматизоване робоче місце адміністратора автомобільної парковки

Актуальність і значення теми: напрямами використання розробленої системи визначено автоматизацію орендування паркувальних місць та зменшення навантаження для адміністратора автомобільної парковки

1. Оцінка запропонованих моделей, підходів, алгоритмів, інформаційної складової та засобів розробки: при роботі з орендами парковочних місць слід забезпечити можливість реєстрації початку та завершення перебування атомобіл на паркомісці.
2. Оцінка розробленої інформаційної системи, її практична цінність та економічна доцільність: розроблена система призначена для адміністратора автомобільної парковки. Реалізоване автоматизоване робоче місце може використовуватись для підвищення ефективності виконання роботи адмістратора автомобільної парковки.
3. Загальний висновок та оцінка: вимоги поставленої задачі виконані в повному обсязі, автоматизоване робоче місце адміністратора автомобільної парковки ,

Робота заслугоує на оцінку « задовільно »

Рецензент к. ер.-м. н. доц. Значенко Н.О. Друж 08.06.2021р