

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет комп'ютерних наук і технологій

(повне найменування факультету)

Кафедра програмних засобів

(повне найменування кафедри)

Пояснювальна записка

до дипломного проєкту (роботи)

бакалавр

(ступінь вищої освіти)

на тему СТВОРЕННЯ ПРОГРАМНОЇ СИСТЕМИ ОПРАЦЮВАННЯ ДАНИХ
ДЛЯ АДМІНІСТРУВАННЯ ПОСЛУГ ПРОКАТУ ТРАНСПОРТНИХ ЗАСОБІВ
CREATING A DATA PROCESSING SYSTEM FOR
ADMINISTERING VEHICLE RENTAL SERVICES

Виконав(ла): студент(ка) 4 курсу, групи КНТ-212

Спеціальності 122 Комп'ютерні науки

(код і найменування спеціальності)

Освітня програма (спеціалізація)

Комп'ютерні науки

ДЕЙНЕГА В.О.

(ПРИЗВИЩЕ та ініціали)

Керівник СКРУПСЬКИЙ С.Ю.

(ПРИЗВИЩЕ та ініціали)

Рецензент КОЦУР М.І.

(ПРИЗВИЩЕ та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

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

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

ЗАТВЕРДЖУЮ

Завідувач кафедри ПЗ, д.т.н, проф.
Сергій СУББОТІН
“ ” 2026 року

З А В Д А Н Н Я

НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА(КИ)

ДЕЙНЕГИ Владислава Олеговича

(ПРИЗВИЩЕ, ім'я, по батькові)

1. Тема проєкту (роботи) Створення програмної системи опрацювання даних для адміністрування послуг прокату транспортних засобів. Creating a Data Processing System for Administering Vehicle Rental Services

керівник проєкту (роботи) к.т.н., доцент, СКРУПСЬКИЙ Степан Юрійович,
(науковий ступінь, вчене звання, ПРИЗВИЩЕ, ім'я, по батькові)

затверджені наказом закладу вищої освіти від “ 7 ” квітня 2026 року № 139

2. Строк подання студентом проєкту (роботи) 03 червня 2026 року

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

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

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

Слайди презентації

6. Консультанти розділів проєкту (роботи)

Розділ	ПРИЗВИЩЕ, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-4 Основна частина	СКРУПСЬКИЙ С.Ю., доцент		
Нормоконтроль	ДЕЙНЕГА Л.Ю., ст. викладач		

7. Дата видачі завдання “ 7 ” квітня 2026 року.

КАЛЕНДАРНИЙ ПЛАН

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

Студент(ка)

(підпис) Владислав ДЕЙНЕГА
(Імя ПРИЗВИЩЕ)

Керівник проєкту (роботи)

(підпис) Степан СКРУПСЬКИЙ
(Імя ПРИЗВИЩЕ)

РЕФЕРАТ

Пояснювальна записка до дипломної кваліфікаційної роботи бакалавра:
111 с., 4 табл., 32 рис., 3 дод., 19 джерел.

C#, SQL SERVER, АДМІНІСТРУВАННЯ, БАЗА ДАНИХ, МОВА ПРОГРАМУВАННЯ, СИСТЕМА ОПРАЦЮВАННЯ ДАНИХ.

Об'єкт дослідження – процеси обчислень, пов'язані з організацією та опрацюванням даних для адміністрування послуг прокату транспортних засобів.

Предмет дослідження – програмні системи опрацювання даних для адміністрування послуг прокату транспортних засобів.

Мета роботи – розробка програмної системи опрацювання даних для адміністрування послуг прокату транспортних засобів для скорочення часу та зменшення трудомісткості оброблення замовлень на прокат автомобілів.

Матеріали, методи та технічні засоби: мова програмування C#, середовище розробки Visual Studio, система управління базами даних Microsoft SQL Server.

Результати. Розроблено проєктні рішення для створення системи опрацювання даних для адміністрування послуг прокату транспортних засобів. Створено систему опрацювання даних для адміністрування послуг прокату транспортних засобів. Проведено дослідження розробленої системи опрацювання даних для адміністрування послуг прокату транспортних засобів.

Висновки. Мету роботи досягнуто. Розроблено систему опрацювання даних для адміністрування послуг прокату транспортних засобів за допомогою мови програмування C#, середовища розробки Visual Studio та системи управління базами даних Microsoft SQL Server.

Галузь використання – оренда транспортних засобів.

ABSTRACT

Explanatory note to the diploma qualifying work of the bachelor: 111 pages, 4 tables, 32 figures, 3 appendixes, 19 sources.

C#, SQL SERVER, ADMINISTRATION, DATABASE, PROGRAMMING LANGUAGE, DATA PROCESSING SYSTEM.

The object of research is computing processes related to the organization and processing of data for the administration of vehicle rental services.

The subject of the research is data processing systems for the administration of vehicle rental services.

The purpose of this work is to develop the data processing system for administering vehicle rental services to reduce the time and labor intensity of processing car rental orders.

Materials, methods and technical tools: C# programming language, Visual Studio development environment, Microsoft SQL Server database management system.

Results. Project solutions for the creation of data processing system for administering vehicle rental services has been designed. The data processing system for administering vehicle rental services has been developed. The study of the developed data processing system for administering vehicle rental services has been conducted.

Conclusions. The goal of the work has been achieved. The data processing system for administering vehicle rental services using the C# programming language, the Visual Studio development environment and Microsoft SQL Server database management system has been developed.

Scope of use – vehicle rental.

ЗМІСТ

	С.
Перелік скорочень та умовних позначок.....	8
Вступ.....	9
1 Аналіз предметної області.....	12
1.1 Системи опрацювання даних для адміністрування послуг прокату транспортних засобів.....	12
1.2 Аналіз систем опрацювання даних для адміністрування послуг прокату транспортних засобів.....	21
1.3 Висновки за розділом 1	35
2 Матеріали і методи.....	37
2.1 Вибір мови програмування.....	37
2.2 Вибір середовища розробки для створення системи опрацювання даних для адміністрування послуг прокату транспортних засобів.....	40
2.3 Вибір засобів зберігання та опрацювання даних.....	44
2.4 Висновки за розділом 2	46
3 Опис системи опрацювання даних.....	48
3.1 Структура системи опрацювання даних для адміністрування послуг прокату транспортних засобів.....	48
3.2 Функціонування системи опрацювання даних для адміністрування послуг прокату транспортних засобів.....	51
3.3 Розробка бази даних системи опрацювання даних для адміністрування послуг прокату транспортних засобів.....	54
3.4 Проектування інтерфейсу системи опрацювання даних для адміністрування послуг прокату транспортних засобів.....	57
3.5 Висновки за розділом 3	59
4 Експлуатація, тестування та експериментальне дослідження системи опрацювання даних.....	61
4.1 Призначення й умови застосування системи опрацювання даних.....	61

4.2	Характеристики системи опрацювання даних для адміністрування послуг прокату транспортних засобів.....	62
4.3	Інструкція по експлуатації програми.....	63
4.3.1	Звернення до програми.....	63
4.3.2	Вхідні й вихідні дані.....	64
4.3.3	Повідомлення.....	64
4.4	Виконання системи опрацювання даних для адміністрування послуг прокату транспортних засобів.....	64
4.5	Тестування системи опрацювання даних для адміністрування послуг прокату транспортних засобів.....	71
4.6	Висновки за розділом 4.....	72
	Висновки.....	73
	Перелік джерел посилання.....	77
	Додаток А Технічне завдання.....	79
	Додаток Б Фрагмент тексту програми.....	84
	Додаток В Слайди презентації.....	104

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК

API – application programming interface;

АП – адміністрування послуг;

БД – база даних;

ІС – інформаційна система;

СОД – система опрацювання даних.

ВСТУП

Актуальність створення системи опрацювання даних для адміністрування послуг прокату транспортних засобів зумовлюється зростанням ролі сервісної економіки та підвищенням вимог до якості, швидкості й прозорості обслуговування клієнтів у сфері мобільності. Сучасний ринок прокату характеризується значною динамічністю, високою конкуренцією та необхідністю оперативного реагування на зміни попиту, що робить традиційні, частково ручні підходи до обліку й управління малоефективними. У таких умовах виникає об'єктивна потреба в упорядкованому збиранні, зберіганні та аналізі великих обсягів інформації, пов'язаної з клієнтами, транспортними засобами, договорами, платежами та технічним станом парку [1], [2].

Використання спеціалізованої системи опрацювання даних дозволяє забезпечити цілісність і актуальність інформаційних ресурсів, зменшити ймовірність помилок, що виникають унаслідок людського фактора, а також підвищити рівень контролю за всіма етапами надання послуг прокату. Автоматизація процесів адміністрування сприяє скороченню часу обслуговування клієнтів, оптимізації використання транспортних засобів і більш раціональному плануванню їх завантаженості. У результаті створюються передумови для підвищення економічної ефективності діяльності підприємства та покращення його конкурентних позицій на ринку [1], [2].

Таким чином, актуальність створення системи опрацювання даних для адміністрування послуг прокату транспортних засобів визначається потребою підвищення якості управління, зменшення витрат, забезпечення надійності інформації та адаптації діяльності підприємств до сучасних умов цифрової трансформації [3]-[6].

Проте деякі системи опрацювання даних для адміністрування послуг прокату транспортних засобів характеризуються високою складністю

адаптації до реальних бізнес-процесів конкретного підприємства, оскільки універсальні програмні рішення не завжди враховують специфіку різних видів прокату, відмінності у моделях обслуговування клієнтів та особливості правового регулювання. Недоліком є і підвищені вимоги до інформаційної безпеки, оскільки системи опрацювання даних у сфері прокату працюють з персональною та фінансовою інформацією клієнтів. Недостатній рівень захисту або помилки в налаштуванні доступу можуть призвести до витоку даних, що створює як репутаційні, так і правові ризики. Забезпечення належного рівня безпеки потребує додаткових ресурсів, регулярних оновлень та постійного контролю, що не завжди є економічно виправданим для невеликих компаній. Крім того, впровадження таких систем часто супроводжується організаційними труднощами, пов'язаними зі зміною усталених процесів роботи. Перехід від традиційних методів обліку до автоматизованих рішень може викликати опір з боку персоналу, зниження продуктивності на початкових етапах та необхідність тривалого періоду адаптації. У сукупності ці фактори свідчать про те, що, попри очевидні переваги, програмні системи опрацювання даних для адміністрування послуг прокату транспортних засобів мають низку недоліків, які потребують комплексного врахування під час їх вибору, розроблення та експлуатації.

Тому актуальною є розробка системи опрацювання даних для адміністрування послуг прокату транспортних засобів. У дипломній кваліфікаційній роботі бакалавра розв'язується актуальне завдання розробки системи опрацювання даних для адміністрування послуг прокату транспортних засобів для скорочення часу та зменшення трудомісткості оброблення замовлень на прокат автомобілів.

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

- виконати аналіз предметної області та систем опрацювання даних для адміністрування послуг прокату транспортних засобів;

- здійснити проєктування системи опрацювання даних для адміністрування послуг прокату транспортних засобів;
- створити систему опрацювання даних для адміністрування послуг прокату транспортних засобів;
- дослідити розроблену систему опрацювання даних для адміністрування послуг прокату транспортних засобів.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Системи опрацювання даних для адміністрування послуг прокату транспортних засобів

Система опрацювання даних для адміністрування послуг прокату транспортних засобів являє собою комплексне програмне рішення, призначене для автоматизованого управління всіма інформаційними процесами, що супроводжують діяльність підприємств з надання автомобілів у тимчасове користування. Така програмна система надає інтегроване середовище, у межах якого здійснюється накопичення, збереження, оброблення та аналіз відомостей про транспортний парк, клієнтів, договори оренди, фінансові операції та супутні сервісні процедури. Її функціонування ґрунтується на використанні сучасних інформаційних технологій, що забезпечують централізований доступ до даних і підтримують прийняття управлінських рішень [1].

Завдяки таким програмним системам забезпечується впорядковане ведення обліку транспортних засобів, контроль їх технічного стану та завантаженості, реєстрація звернень клієнтів і фіксація умов оренди. Програмна система виступає посередником між користувачем і великим обсягом даних, перетворюючи розрізнені відомості на структуровану інформацію, придатну для оперативного використання [1]

Застосування таких програмних систем сприяє підвищенню прозорості бізнес-процесів. Автоматизоване формування документів, розрахунок вартості послуг, ведення історії взаємодії з клієнтами дають змогу зменшити вплив людського чинника та мінімізувати кількість помилок. Значною мірою полегшується контроль за рухом транспортних засобів, термінами їх повернення, плануванням технічного обслуговування. У результаті створюються умови для більш раціонального використання ресурсів підприємства [1], [2].

Характерною особливістю подібних програмних комплексів є орієнтація на роботу в умовах постійної зміни даних. Інформація про наявність автомобілів, замовлення, платежі та додаткові послуги оновлюється безперервно, тому система повинна забезпечувати високу швидкість та надійність зберігання. Важливим є також можливість інтеграції з зовнішніми сервісами, платіжними платформами, засобами електронної комунікації, що розширює функціональні межі програмного продукту [1], [2].

Подібні програмні рішення створюють передумови для підвищення якості обслуговування клієнтів. Оперативний доступ до повної інформації про замовлення дає змогу швидко реагувати на запити, пропонувати індивідуальні умови, контролювати виконання договірних зобов'язань. Водночас керівництво підприємства отримує інструменти аналітичного оцінювання діяльності, формування звітності та прогнозування попиту [1], [2].

Програмні системи опрацювання даних для адміністрування послуг прокату транспортних засобів є важливим елементом цифрової інфраструктури сучасного сервісного бізнесу. Такі системи об'єднують інформаційні потоки, забезпечують їх узгоджене оброблення та створюють основу для ефективного управління, розвитку конкурентоспроможності й підвищення організаційної результативності підприємства [1], [2].

У процесі характеристики програмних систем опрацювання даних для адміністрування послуг прокату транспортних засобів доцільно додатково зосереджувати увагу на їх ролі у трансформації традиційних підходів до організації сервісної діяльності. Такі програмні системи розглядаються не лише як інструмент обліку, а як засіб побудови нової моделі взаємодії між компанією та споживачем, у якій значна частина операцій переноситься у цифрове середовище. Відбувається переорієнтація з паперових процедур на електронні процеси, що змінює характер управління та створює передумови для більш гнучкої організації роботи [1], [2].

Важливим аспектом є те, що програмні комплекси формують основу для стандартизації бізнес-процедур у сфері прокату. За їх допомогою

уніфікуються підходи до реєстрації клієнтів, оформлення договорів, обліку платежів і контролю використання транспортних засобів. Це дозволяє забезпечити єдині правила роботи незалежно від кількості філій чи масштабів автопарку. Стандартизоване середовище сприяє узгодженості дій персоналу та полегшує навчання нових працівників [2], [3].

Окремої уваги заслуговує аналітичний потенціал подібних програмних систем. Накопичені дані стають джерелом для виявлення закономірностей у попиті на різні категорії транспортних засобів, для оцінювання ефективності тарифної політики та визначення напрямів розвитку сервісу. На основі цифрових показників формується можливість порівнювати результати за різні періоди, відстежувати динаміку використання ресурсів та обґрунтовувати управлінські рішення. Таким чином програмне забезпечення перетворюється на інструмент стратегічного планування [2], [3].

Не менш суттєвою є комунікаційна функція систем опрацювання даних. Вони забезпечують інформаційний зв'язок між підрозділами підприємства, об'єднують роботу менеджерів, операторів, бухгалтерських і технічних служб. Через єдину платформу передаються відомості про стан замовлень, платежі, інциденти з транспортними засобами, що зменшує розриви між окремими етапами обслуговування. Підвищується узгодженість дій та скорочується час реагування на проблемні ситуації [2], [3].

У контексті сучасних тенденцій слід підкреслити орієнтацію таких програмних систем на підтримку віддалених форматів взаємодії. Користувачі отримують можливість здійснювати значну частину операцій без особистого відвідування офісу, а персонал може працювати з даними незалежно від місця перебування. Це надає бізнесу додаткової стійкості та дозволяє адаптуватися до змін у поведінці споживачів [2], [3].

Також доцільно відзначати значення програмних систем для підвищення рівня безпеки діяльності прокатних компаній. Електронна фіксація дій користувачів, збереження історії змін і розмежування прав доступу створюють умови для контролю за інформаційними потоками.

Завдяки цьому зменшуються ризики втрати даних, несанкціонованого втручання або фінансових зловживань [2], [3].

Розробка програмних систем опрацювання даних для адміністрування послуг прокату транспортних засобів здійснюється як комплексний процес, що охоплює послідовні етапи аналізу, проектування, реалізації та впровадження програмного забезпечення. На початковому етапі відбувається вивчення вимог до системи, що включає визначення цілей бізнесу, функціональних потреб компанії, особливостей роботи з клієнтами та транспортними засобами, а також вимог до безпеки та збереження даних. Під час цього аналізу збираються відомості про специфіку оренди транспортних засобів, варіанти тарифів, правила обліку та контролю, що дозволяє створити точну концепцію системи та сформулювати завдання для подальшої розробки [3], [4].

На стадії проектування формується архітектура системи, визначаються ключові модулі та взаємозв'язки між ними, розробляються структурні схеми бази даних та логіка обробки даних. При цьому особлива увага приділяється оптимізації процесів бронювання, обліку клієнтів, обліку пробігу та технічного стану транспортних засобів. Використовуються сучасні засоби проектування, включаючи CASE-системи, UML-діаграми, інструменти для моделювання процесів та створення прототипів інтерфейсів користувача. Архітектура системи формується з урахуванням масштабованості, інтеграційних можливостей і підтримки різних платформ, включно з веб- та мобільними додатками [3], [4].

Реалізація програмної системи передбачає безпосереднє створення програмного коду з використанням мов програмування, які забезпечують надійність і ефективність обробки даних, таких як C#, Java або Python, а також застосування фреймворків для веб-розробки. Паралельно розробляються модулі взаємодії з базами даних, що можуть реалізовуватися на платформах SQL Server, MySQL або Oracle, забезпечуючи надійне збереження та швидкий доступ до інформації. Особлива увага приділяється реалізації функціоналу

обліку клієнтів, бронювань, тарифів та платежів, автоматизації розрахунків вартості оренди та формуванню звітності. Для перевірки коректності роботи системи застосовуються засоби тестування, включаючи тести, інтеграційні тести та сценарії навантажувального тестування [3], [4].

Наступний етап передбачає впровадження системи в робоче середовище та її інтеграцію з існуючими інформаційними системами підприємства. Проводяться роботи з міграції даних, налаштування прав доступу, навчання персоналу та оптимізації робочих процесів. Використовуються засоби адміністрування серверів, вебсерверів та баз даних, системи управління конфігураціями, а також платформи для віддаленого доступу та моніторингу. Після впровадження здійснюється контроль ефективності роботи системи, збираються відгуки користувачів, оцінюється продуктивність і зручність використання, а також визначаються напрямки для подальшого вдосконалення [3], [4].

На завершальному етапі розробки систем опрацювання даних для адміністрування послуг прокату транспортних засобів передбачено обслуговування та розвиток програмного забезпечення, що включає оновлення функціоналу, виправлення помилок, адаптацію до змін у бізнес-процесах і законодавчих вимогах. Використовуються інструменти для автоматичного резервного копіювання, моніторингу безпеки та контролю доступу. Також забезпечується можливість масштабування системи для роботи з більшою кількістю транспортних засобів, клієнтів та точок обслуговування, що дозволяє компаніям підтримувати високий рівень сервісу та ефективно реагувати на зміни в попиті. Таким чином процес розробки включає комплексну інтеграцію аналітики, проектування, кодування, впровадження та підтримки системи, забезпечуючи повну цифровізацію процесів оренди транспортних засобів [3], [4].

Функціональні можливості програмних систем опрацювання даних для адміністрування послуг прокату транспортних засобів для організації прокату автомобілів прийнято розглядати з позиції їх значущості для забезпечення

базової діяльності та для формування додаткових переваг у конкурентному середовищі. Перед вибором конкретної системи доцільно чітко розмежувати ті можливості, без яких ефективна робота неможлива, і ті, що сприяють підвищенню якості сервісу та розширенню бізнесу [4], [5].

До ключових функціональних компонентів систем опрацювання даних для адміністрування послуг прокату транспортних засобів відносять інструменти керування бронюванням і замовленнями, що передбачають наявність єдиного центру адміністрування для різних каналів оформлення оренди, включаючи онлайн заявки, особисті звернення та корпоративні договори. Важливим є забезпечення постійного контролю за доступністю транспортних засобів із використанням засобів визначення їхнього місцеперебування, а також можливість гнучкого налаштування отримання та повернення автомобілів у різних пунктах обслуговування. Система має підтримувати зміну умов замовлення, його перенесення чи анулювання відповідно до встановлених правил, а також взаємодіяти з вебресурсами, мобільними застосунками та сторонніми сервісами. Доцільним вважається автоматичне інформування клієнтів за допомогою електронних повідомлень або текстових сповіщень [4], [5].

У межах управління автопарком необхідною є наявність повної інформаційної бази про кожний транспортний засіб із фіксацією технічних і реєстраційних даних, а також постійне оновлення його статусу. Значну роль відіграють засоби моніторингу переміщень, облік регламентних робіт, контроль витрат пального та аналіз інтенсивності експлуатації. Важливим напрямом є автоматизація документального супроводу, що передбачає формування електронних договорів із можливістю цифрового підпису, генерацію платіжних документів, збереження копій ідентифікаційних матеріалів клієнта та фіксацію стану автомобіля під час видачі й повернення [4], [5].

Фінансово-обліковий блок систем опрацювання даних для адміністрування послуг прокату транспортних засобів має забезпечувати

взаємодію з платіжними сервісами, підтримку розрахунків у різних валютах, автоматичне визначення податкових нарахувань і роботу з гарантійними внесками. Інтеграція з бухгалтерськими платформами сприяє узгодженості фінансових процесів. Окреме значення надається інструментам взаємодії з клієнтами, що включають централізоване зберігання даних, реалізацію програм заохочення, автоматичні нагадування та аналіз відгуків. Аналітичні можливості системи дозволяють оцінювати ефективність використання автопарку, структуру доходів, експлуатаційні витрати та формувати прогнози попиту [4], [5].

Для організації роботи персоналу у системах опрацювання даних для адміністрування послуг прокату транспортних засобів передбачається багаторівнева модель доступу з розмежуванням повноважень і веденням журналів дій. У разі розгалуженої мережі підприємства важливою є підтримка кількох підрозділів із можливістю централізованого управління ресурсами та консолідованого аналізу показників [4], [5].

Поряд із базовими можливостями систем опрацювання даних для адміністрування послуг прокату транспортних засобів все більшого значення набувають розширені технологічні інструменти. До них відносять інтелектуальне формування цін залежно від ринкових умов, застосування телематичних рішень для попереджувального обслуговування, використання мобільних застосунків для клієнтів і персоналу, автоматизоване визначення пошкоджень на основі аналізу зображень. Також важливими є інтеграційні механізми для взаємодії з туристичними платформами, підтримка багатомовності, обмін даними зі страховими сервісами та можливість роботи в умовах обмеженого доступу до мережі. Упровадження таких засобів сприяє перетворенню традиційного процесу оренди на цілісну цифрову екосистему, орієнтовану на масштабування та прийняття рішень на основі даних [4], [5].

Використання систем опрацювання даних для адміністрування послуг прокату транспортних засобів забезпечує істотне спрощення та впорядкування внутрішніх процесів підприємства. Запровадження автоматизованих

механізмів дозволяє відмовитися від значної частини рутинних дій, пов'язаних із перевіркою доступності автомобілів, оформленням документів та веденням фінансових розрахунків. Унаслідок цього зменшується вплив людського чинника, скорочується кількість помилок і підвищується загальна продуктивність роботи персоналу [4], [5].

Сучасні умови ринку формують підвищені вимоги до якості обслуговування клієнтів. Очікується можливість самостійного оформлення замовлення через мережу, оперативне отримання підтверджень та зрозуміла система формування вартості. Програмні платформи створюють умови для цілодобової взаємодії з користувачами, забезпечують стандартизоване інформування та сприяють формуванню довготривалих відносин із замовниками [4], [5].

Застосування засобів позиціонування та відповідних технологій надає можливість постійно контролювати стан і переміщення автопарку. Оператори отримують актуальні відомості про місцеперебування транспортних засобів, рівень витрат пального та можливі порушення режимів експлуатації. Такі інструменти підвищують прозорість діяльності та дозволяють своєчасно реагувати на нестандартні ситуації [4], [5].

Важливим результатом упровадження систем опрацювання даних для адміністрування послуг прокату транспортних засобів є оптимізація витрат на утримання техніки. Система нагадувань про регламентні роботи та планові огляди сприяє попередженню серйозних несправностей, зменшує простой та продовжує строк служби автомобілів. Завдяки цьому забезпечується стабільність надання послуг і раціональне використання ресурсів [4], [5].

Наявність розвинених аналітичних модулів формує основу для обґрунтованого управління бізнесом. Обробка накопичених даних дає змогу оцінювати рівень завантаження автопарку, рентабельність окремих напрямів, сезонні коливання попиту та інші показники, що впливають на прибутковість. Отримані відомості використовуються для коригування тарифної політики та планування подальшого розвитку [4], [5].

Цифрове ведення документації сприяє дотриманню вимог нормативного характеру, пов'язаних із оподаткуванням, страхуванням і безпекою перевезень. Централізоване зберігання інформації підвищує рівень її захищеності та спрощує процедури контролю. Водночас програмні рішення дозволяють масштабувати діяльність підприємства без суттєвого ускладнення управління, забезпечуючи єдине інформаційне поле для кількох підрозділів або філій [4], [5].

Процес вибору відповідної системи опрацювання даних для адміністрування послуг прокату транспортних засобів ґрунтується на всебічному аналізі особливостей конкретного бізнесу. Необхідно враховувати обсяг автопарку, заплановані темпи його розширення, характер надання послуг та потребу в додаткових технологіях. Значення має також форма розгортання платформи: використання хмарних сервісів забезпечує швидке впровадження та зменшення початкових витрат, тоді як локальні рішення надають більший контроль над даними [4], [5].

Порівняння доступних систем опрацювання даних для адміністрування послуг прокату транспортних засобів здійснюється з урахуванням функціональних можливостей, вартості володіння, рівня налаштування та перспектив подальшого розвитку [4], [5].

Визначено, що система опрацювання даних для адміністрування послуг прокату транспортних засобів являє собою комплексне програмне рішення, призначене для автоматизованого управління всіма інформаційними процесами, що супроводжують діяльність підприємств з надання автомобілів у тимчасове користування. Така програмна система надає інтегроване середовище, у межах якого здійснюється накопичення, збереження, оброблення та аналіз відомостей про транспортний парк, клієнтів, договори оренди, фінансові операції та супутні сервісні процедури. Її функціонування ґрунтується на використанні сучасних інформаційних технологій, що забезпечують централізований доступ до даних і підтримують прийняття управлінських рішень.

1.2 Аналіз систем опрацювання даних для адміністрування послуг прокату транспортних засобів

Проаналізуємо програмні системи опрацювання даних для адміністрування послуг прокату транспортних засобів [6]-[13].

Система опрацювання даних для адміністрування послуг прокату транспортних засобів AFG Rent A Car (рис. 1.1) являє собою комплексне програмне рішення, орієнтоване на підтримку повного циклу діяльності компанії з надання автомобілів в оренду. Передбачено можливість роботи з різними мовними та валютними параметрами, що забезпечує адаптацію до потреб користувачів з різних регіонів та спрощує взаємодію з клієнтами. У межах платформи реалізовано механізми централізованого зберігання інформації, опрацювання даних щодо транспортних засобів, клієнтів та договорів оренди, а також інструменти для формування внутрішньої звітності. Архітектура системи спрямована на підвищення ефективності управління бізнес-процесами та скорочення обсягів ручних операцій [6], [11].

The screenshot displays the 'Car Definition' window of the AFG Rent A Car system. It features a car image of a silver Porsche Cayon on the left. Below the image are various input fields for car details, including Code, Year, Plate, and Default Rent Price. A table on the right lists variables and their values, such as T. C. No., Color, and Category. The bottom of the window shows a navigation bar with icons and a status bar.

Variable	Value
T. C. No.	BR850124
Color	Gray
Sub color	Silver
Category	
Class	Cayon
Plate source / Place	
Plate category	
Shape	
Origin	
Empty weight	
G. V. weight	
Doors	4
Seats	4
Cylinders	8
Make	
Payload	
Insurance company	
Type of insurance	
Insurance number	
Insurance start dat	
Mortgage by	
Mortgage No.	

Рисунок 1.1 – Програмна система AFG Rent A Car [11]

Функціональні можливості програмної системи AFG Rent A Car охоплюють облік фінансових операцій, моніторинг стану автопарку, контроль за технічним обслуговуванням і страхуванням транспортних засобів, а також ведення аналітики діяльності підприємства. Передбачено використання різних механізмів роботи з базами даних, що дозволяє забезпечити стабільність функціонування за умов значних обсягів інформації. Система орієнтована на інтеграцію з віддаленими робочими місцями, завдяки чому співробітники можуть здійснювати спільну роботу в режимі онлайн та оперативно обмінюватися необхідними відомостями [6], [11].

Особливу увагу приділено питанням захисту інформації та розмежування доступу. Передбачено, що користувачі працюють лише з тими модулями та даними, які відповідають їхнім повноваженням, що мінімізує ризики несанкціонованого втручання. Реалізовано зручні засоби пошуку та фільтрації записів, які спрощують роботу уповноваженого персоналу з великими масивами відомостей. Завдяки цьому забезпечується цілісність даних і можливість швидкого отримання актуальної інформації для прийняття управлінських рішень [6], [11].

Програмна система AFG Rent A Car орієнтована як на невеликі компанії, так і на підприємства з розгалуженим автопарком. Передбачено можливість індивідуального ведення обліку кожного транспортного засобу без потреби у додатковому ліцензуванні окремих одиниць. Інтерфейс побудовано з урахуванням принципів простоти та зрозумілості, що дає змогу використовувати програму без глибоких технічних знань. Це сприяє швидкому впровадженню рішення в діяльність організації та зменшує витрати на навчання персоналу [6], [11].

У межах програмної системи AFG Rent A Car реалізовано інструменти для автоматизованого формування фінансових документів, розрахунку витрат і доходів, підготовки балансів та інших бухгалтерських звітів. Передбачено можливість фіксації витрат на паливо, обліку перевищення пробігу, контролю за штрафними нарахуваннями та простроченням повернення

автотранспортних засобів. Наявність таких функцій дозволяє здійснювати комплексний фінансовий контроль і підвищує прозорість господарської діяльності підприємства [6], [11].

Окремий напрям роботи системи AFG Rent A Car пов'язаний з підтримкою життєвого циклу транспортних засобів. Програма нагадує про необхідність проведення технічного обслуговування, закінчення строку дії реєстраційних документів і страховки, що сприяє своєчасному виконанню регламентних процедур. Також передбачено зберігання електронних копій документів і зображень автомобілів, що полегшує доступ до довідкової інформації та прискорює процес оформлення оренди [6], [11].

Для взаємодії з клієнтами реалізовано засоби автоматизованого інформування та формування персоналізованих документів. Система дозволяє налаштовувати шаблони договорів, квитанцій та інших матеріалів із використанням елементів корпоративного стилю. Це сприяє підвищенню якості обслуговування та формуванню єдиного інформаційного простору компанії [6], [11].

Таким чином, система опрацювання даних AFG Rent A Car розглядається як інтегроване середовище управління процесами прокату транспортних засобів, яке поєднує можливості обліку, контролю, аналітики та взаємодії з клієнтами. Її застосування дає змогу оптимізувати внутрішні процедури, зменшити кількість рутинних операцій та забезпечити більш ефективне використання ресурсів підприємства [6], [11].

Основними характеристиками програмного забезпечення AFG Rent A Car є такі [6], [11]:

- багатомовність та мультивалютність: програмна система AFG Rent A Car забезпечує роботу з різними мовами інтерфейсу та підтримує використання декількох валют, що дає змогу застосовувати систему в міжнародних компаніях та обслуговувати клієнтів з різних регіонів [6], [11];

- централізоване управління автопарком: у програмній системі AFG Rent A Car реалізовано можливість ведення єдиного реєстру транспортних

засобів із фіксацією їх технічних характеристик, статусу доступності та історії експлуатації [6], [11];

- автоматизація процесу бронювання: програмна система AFG Rent A Car дозволяє створювати, змінювати та відстежувати договори оренди з урахуванням індивідуальних умов для кожного клієнта [6], [11];

- інтегрований бухгалтерський облік: передбачено формування фінансових документів, облік доходів і витрат, підготовку звітів про прибутки та збитки без використання сторонніх програм [6], [11];

- контроль технічного обслуговування: програмна система AFG Rent A Car нагадує про необхідність проведення сервісних робіт, заміну витратних матеріалів та завершення строку дії страховки [6], [11];

- розмежування прав доступу: користувачам програмної системи AFG Rent A Car надаються різні рівні повноважень, що забезпечує захист даних та унеможливорює несанкціоновані зміни інформації [6], [11];

- підтримка віддаленої роботи: доступ до функцій програмної системи AFG Rent A Car можливий з різних робочих місць через мережу, що спрощує взаємодію між підрозділами компанії [6], [11];

- засоби аналітики та звітності: у програмній системі AFG Rent A Car передбачено побудову статистичних звітів, аналіз ефективності використання автомобілів та оцінювання фінансових показників [6], [11];

- налаштування шаблонів документів: програмна система AFG Rent A Car дозволяє адаптувати договори, квитанції та інші форми відповідно до корпоративного стилю підприємства [6], [11];

- облік додаткових нарахувань: у програмній системі AFG Rent A Car реалізовано розрахунок витрат на пальне, перевищення пробігу та штрафів за прострочення повернення автомобілів [6], [11];

- зберігання електронних матеріалів: у програмній системі AFG Rent A Car передбачено можливість прикріплення сканованих документів і зображень транспортних засобів до відповідних записів бази даних [6], [11];

- відсутність потреби у ліцензуванні кожного авто: програмна система AFG Rent A Car забезпечує роботу з необмеженою кількістю транспортних одиниць у межах однієї інсталяції [6], [11];

- зручність користувацького інтерфейсу: програмна система AFG Rent A Car орієнтований на користувачів без спеціальної технічної підготовки та не потребує тривалого навчання [6], [11];

- інтеграція з зовнішніми сервісами: передбачено отримання окремих даних з мережевих ресурсів без необхідності їх ручного введення [6], [11];

- підтримка різних типів договорів: у програмній системі AFG Rent A Car можливе оформлення оренди на різні періоди з урахуванням індивідуальних умов співпраці [6], [11].

Серед недоліків програмної системи AFG Rent A Car можна відзначити обмежену гнучкість при інтеграції з іншими сучасними інформаційними платформами, що ускладнює її використання в комплексних корпоративних середовищах. Певні функції орієнтовані на традиційні підходи до організації прокату і можуть не повною мірою відповідати новим бізнес-моделям та вимогам мобільних сервісів. Також спостерігається залежність від конкретних технологій баз даних, що потребує додаткових ресурсів для адміністрування та оновлення системи [6], [11].

Система опрацювання даних ScheduleBull (рис. 1.2) являє собою мережеве програмне середовище, орієнтоване на комплексне адміністрування процесів прокату транспортних засобів та суміжних сервісів. Рішення побудоване як універсальна платформа, що забезпечує взаємодію між різними учасниками ринку оренди, дозволяючи обмінюватися інформацією в єдиному інформаційному просторі. Програмний засіб ScheduleBull може використовуватися з різних пристроїв і не обмежується лише настільними комп'ютерами, що робить можливим доступ до функцій системи з будь-якого місця. Первинна концепція продукту була спрямована на автоматизацію діяльності компаній з прокату автомобілів, однак згодом функціональність була розширена і стала придатною для обслуговування інших напрямів,

пов'язаних з тимчасовим користуванням майном та організацією сервісних послуг [6], [7], [12].

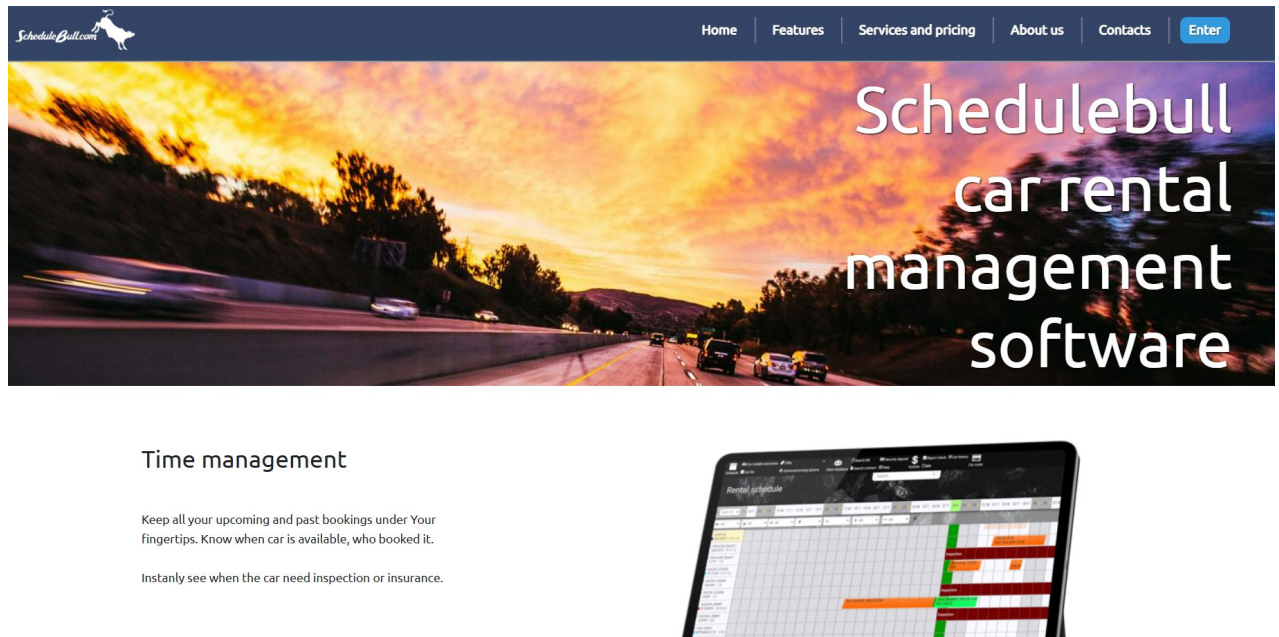


Рисунок 1.2 – Програмна система ScheduleBull [12]

У межах системи ScheduleBull реалізовано інструменти для ведення бази клієнтів, оформлення бронювань, планування робіт з технічного обслуговування та контролю стану автопарку. Значна увага приділена процесам формування рахунків та фінансових документів, що дає змогу здійснювати повний цикл обліку взаємовідносин із замовниками. Передбачено можливість генерування різноманітних звітів, які відображають поточний стан діяльності підприємства та сприяють прийняттю управлінських рішень. Архітектура програмного забезпечення є відкритою для розширення за допомогою інтерфейсу прикладного програмування, що дозволяє адаптувати функції відповідно до індивідуальних потреб організації [6], [7], [12].

Технологічна основа програмної системи Schedulebull спирається на сучасні вебстандарти, завдяки чому забезпечується оперативна синхронізація даних між робочими місцями та надійне збереження інформації. Механізми резервування даних спрямовані на мінімізацію ризиків втрати відомостей у разі технічних збоїв. Система здатна автоматично нагадувати користувачам

про важливі події, пов'язані з експлуатацією транспортних засобів та виконанням договірних зобов'язань. Окремі модулі орієнтовані на виявлення небажаних клієнтів, контроль штрафних нарахувань і гнучке управління тарифною політикою [6], [7], [12].

Програмне середовище ScheduleBull надає можливості інтеграції з корпоративними вебресурсами, що дозволяє організувати приймання замовлень безпосередньо через сайт компанії та забезпечити автоматичний обмін даними між зовнішніми сервісами і внутрішньою базою. Наявність вбудованої системи керування контентом спрощує підтримку інформаційних сторінок і дає змогу оперативно змінювати відомості про послуги та умови оренди. Таким чином формується єдиний інформаційний простір, у якому об'єднано процеси бронювання, фінансового обліку, управління автопарком і взаємодії з клієнтами [6], [7], [12].

Загалом програмну систему Schedulebull можна охарактеризувати як гнучку та масштабовану систему, що орієнтована на автоматизацію щоденної діяльності підприємств прокатної сфери. Вона поєднує засоби планування, обліку та комунікації, забезпечуючи узгоджену роботу персоналу й підвищення рівня обслуговування замовників. Використання єдиної платформи сприяє впорядкуванню інформаційних потоків, скороченню рутинних операцій та формуванню прозорого механізму управління послугами оренди транспортних засобів [6], [7], [12].

Програмна система ScheduleBull має такі особливості [6], [7], [12]:

- веборієнтована платформа: програмна система ScheduleBull функціонує через браузер і не потребує встановлення складного клієнтського програмного забезпечення, що дає можливість працювати з різних пристроїв та локацій [6], [7], [12];

- управління автопарком: у програмній системі ScheduleBull реалізовано засоби обліку транспортних засобів, контролю їх технічного стану та планування сервісних робіт [6], [7], [12];

- ведення бази клієнтів: у програмній системі ScheduleBull передбачено централізоване збереження інформації про замовників, історію взаємодії та договірні відносини [6], [7], [12];
- підтримка бронювань: у програмній системі ScheduleBull забезпечено створення, редагування та супровід замовлень на оренду з урахуванням часових параметрів і доступності автомобілів [6], [7], [12];
- формування рахунків: програмна система ScheduleBull автоматизує підготовку фінансових документів та розрахунок вартості послуг відповідно до обраних тарифів [6], [7], [12];
- багатомовність: інтерфейс програмної системи ScheduleBull може використовуватися різними мовами, що спрощує роботу в міжнародному середовищі [6], [7], [12];
- інтеграційний API: у програмній системі ScheduleBull наявні механізми підключення зовнішніх сервісів і вебсайтів для обміну даними та розширення функцій [6], [7], [12];
- автоматичні нагадування: у програмній системі ScheduleBull реалізовано інформування користувачів про важливі події, пов'язані з договорами та технічним обслуговуванням [6], [7], [12];
- звітно-аналітичні засоби: у програмній системі ScheduleBull передбачено генерування різних типів звітів для контролю діяльності та прийняття управлінських рішень [6], [7], [12];
- розмежування доступу: система ScheduleBull підтримує розподіл прав між користувачами для забезпечення безпеки даних [6], [7], [12];
- синхронізація інформації: у програмній системі ScheduleBull використано сучасні вебтехнології для оперативного оновлення даних між робочими місцями [6], [7], [12];
- управління тарифами: у системі ScheduleBull надається можливість гнучкого налаштування цінових таблиць і умов оренди [6], [7], [12];
- робота з платежами: у програмній системі ScheduleBull підтримується облік надходжень і контроль фінансових зобов'язань клієнтів [6], [7], [12];

- виявлення ризикових клієнтів: у програмній системі ScheduleBull наявні інструменти для ідентифікації небажаних замовників та мінімізації можливих втрат [6], [7], [12];
- вбудована система керування контентом: у програмній системі ScheduleBull забезпечується можливість редагування інформаційних матеріалів без залучення сторонніх засобів [6], [7], [12];
- кросплатформеність: доступ до функцій можливий з персональних комп'ютерів і мобільних пристроїв [6], [7], [12];
- резервування даних: у програмній системі ScheduleBull передбачено механізми збереження інформації для захисту від технічних збоїв [6], [7], [12];
- підтримка різних сфер оренди: ScheduleBull може застосовуватися не лише для автомобілів, а й для інших об'єктів прокату [6], [7], [12];
- швидке оформлення договорів: процес підготовки документів оптимізовано завдяки шаблонам та автоматичному заповненню [6], [7], [12];
- масштабованість: архітектура ScheduleBull дозволяє використовувати систему як малими, так і великими підприємствами [6], [7], [12].

Серед недоліків програмної системи Schedulebull можна відзначити обмежену можливість глибокого налаштування окремих бізнес-процесів без використання додаткового програмування, залежність від стабільного підключення до мережі Інтернет, що ускладнює роботу в умовах нестійкого зв'язку, а також необхідність адаптації інтерфейсу під специфіку національного законодавства та локальних правил ведення обліку. Крім того, для повноцінного використання інтеграційних можливостей потрібні технічні фахівці, що може підвищувати витрати на впровадження та супровід, а універсальний характер платформи інколи зумовлює надмірну складність для невеликих компаній із простими схемами роботи [6], [7], [12].

Програмна система Flit2GO (рис. 1.3) являє собою комплексне рішення, спрямоване на цифрову трансформацію діяльності компаній, що надають послуги прокату та підписки на транспортні засоби [8], [9], [13].

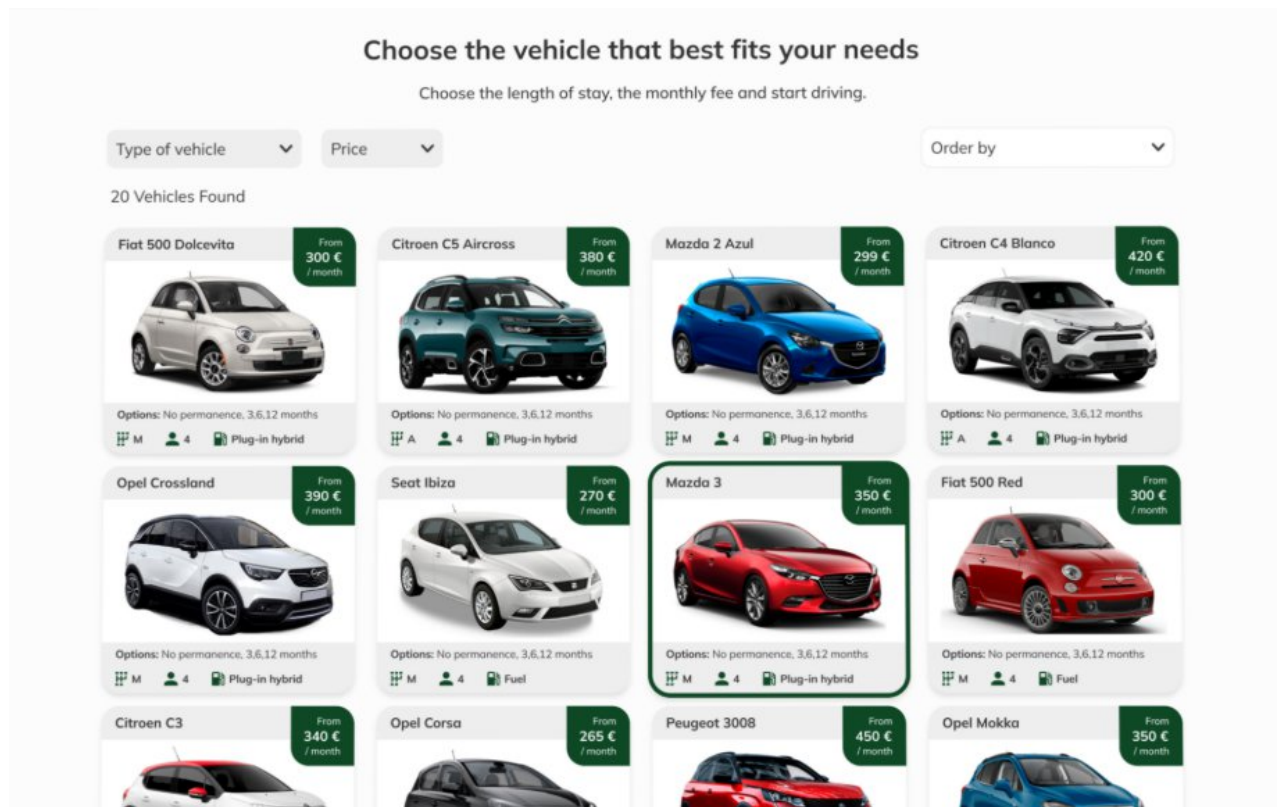


Рисунок 1.3 – Програмна система Flit2GO [13]

В основі програмної системи Flit2GO закладено підхід повної автоматизації взаємодії з клієнтом, завдяки чому всі етапи обслуговування можуть здійснюватися в електронному форматі без необхідності фізичної присутності в офісі. Програмна система Flit2GO орієнтована на переведення традиційних процедур оренди у цифровий простір, що дозволяє значно спростити процес оформлення замовлень, підготовки договорів та ведення фінансової документації. Значна увага приділяється формуванню єдиного інформаційного середовища, у якому поєднуються дані про клієнтів, транспортні засоби та поточні операції компанії [8], [9], [13].

У межах функціонування програмної системи Flit2GO забезпечується ведення електронних профілів користувачів, у яких зберігаються всі відомості, необхідні для подальшого обслуговування. Програмна система Flit2GO підтримує облік технічного стану автопарку, контроль проведення профілактичних робіт, планування прибирання та ревізій, а також інтеграцію з бухгалтерськими процесами підприємства. Завдяки використанню єдиного

центру управління стає можливим координувати як короткострокову оренду, так і більш гнучкі формати користування транспортом, зокрема лізингові або передплатні моделі. Такий підхід сприяє розширенню спектра послуг та підвищенню рівня їх доступності для різних категорій споживачів [8], [9], [13].

Важливою складовою програмної системи Flit2GO є підтримка веборієнтованого середовища бронювання, що дозволяє клієнтам самостійно обирати транспортні засоби та оформлювати замовлення в режимі реального часу. Усі необхідні дії, пов'язані з укладанням договору, підтвердженням особи та оплатою, виконуються за допомогою електронних інструментів. Програмна система Flit2GO передбачає використання цифрових підписів і формування електронних рахунків, що мінімізує обсяг паперового документообігу. Одночасно з цим забезпечується зручний інтерфейс для внутрішнього персоналу, який отримує можливість відстежувати стан замовлень, контролювати доступність техніки та оперативно реагувати на запити користувачів [8], [9], [13].

Архітектура програмної системи Flit2GO орієнтована на роботу з різними типами транспортних засобів, що робить його придатним для підприємств із неоднорідним автопарком. Передбачено механізми автоматичного супроводу процесу оренди від моменту реєстрації клієнта до завершення користування об'єктом прокату, включаючи фіксацію стану транспортного засобу за допомогою цифрових матеріалів. Така організація роботи сприяє підвищенню прозорості операцій та зменшенню ризику помилок. У результаті впровадження Flit2GO формується цілісна екосистема управління мобільністю, що поєднує інтереси клієнтів і компанії, забезпечуючи більш якісний та сучасний формат надання послуг [8], [9], [13].

Програмна система Flit2GO має такі особливості [8], [9], [13]:

- повна цифровізація процесів: оформлення оренди, підписання договорів та формування рахунків здійснюється в електронному середовищі Flit2GO без використання паперових носіїв, що прискорює обслуговування клієнтів та зменшує адміністративні витрати [8], [9], [13];

– автоматизоване управління автопарком: програмна система Flit2GO забезпечує контроль технічного стану транспортних засобів, планування профілактичних робіт та облік експлуатаційних подій протягом усього життєвого циклу автомобіля [8], [9], [13];

– інтегрований облік клієнтів: ведення цифрових профілів користувачів дозволяє зберігати історію взаємодії, документи та параметри попередніх замовлень для подальшого використання [8], [9], [13];

– підтримка різних моделей користування: програмна система Flit2GO орієнтоване не лише на короткострокову оренду, а й на гнучкі формати лізингу та передплати транспортних засобів [8], [9], [13];

– веборієнтоване бронювання: замовлення автомобілів виконується через онлайн інтерфейс Flit2GO у режимі реального часу з можливістю вибору параметрів оренди [8], [9], [13];

– електронна ідентифікація користувача: підтвердження особи та перевірка документів здійснюються за допомогою цифрових інструментів Flit2GO без відвідування офісу [8], [9], [13];

– автоматичний супровід оренди: від моменту реєстрації до завершення користування програмною системою Flit2GO фіксуються всі етапи, включаючи фотодокументування стану транспортного засобу [8], [9], [13];

– інтеграція з бухгалтерією: програмна система Flit2GO формує фінансові документи та забезпечує їх передачу до облікових підсистем підприємства [8], [9], [13];

– підтримка різних типів техніки: можливе використання програмної системи Flit2GO для автомобілів, мототранспорту та інших засобів мобільності в межах єдиного інформаційного простору [8], [9], [13];

– можливість брендованого додатка: інтерфейс програмної системи Flit2GO може адаптуватися під корпоративний стиль компанії, що підвищує впізнаваність сервісу серед клієнтів [8], [9], [13].

Серед недоліків програмної системи Flit2GO можна відзначити залежність ефективності роботи від стабільності Інтернет з'єднання та якості

інтеграції з зовнішніми сервісами, що в окремих випадках може ускладнювати використання платформи малими компаніями з обмеженими технічними ресурсами. Також розширений функціонал потребує певного періоду адаптації персоналу та налаштування бізнес-процесів відповідно до логіки програмного середовища [8], [9], [13].

Порівняльну характеристику програмних систем опрацювання даних для адміністрування послуг прокату транспортних засобів наведено у таблиці 1.1.

Таблиця 1.1 – Порівняння програмних систем опрацювання даних для адміністрування послуг прокату транспортних засобів

Критерій порівняння	AFG Rent A Car [11]	ScheduleBull [12]	Flit2GO [13]
Підтримка багатьох мов	+	+	+—
Мультивалютність	+	+—	+—
Інтеграція із зовнішніми API	+—	+	+
Розширені звіти та аналітика	+	+	+—
Підтримка кількох сфер оренди	+—	+	+
Простота освоєння для малих компаній	+—	+—	+
Гнучкі формати користування	—	+—	+

За результатами проведеного аналізу можна зробити висновок, що у наш час існує досить багато систем опрацювання даних для адміністрування послуг прокату транспортних засобів. Проте деякі системи опрацювання даних для адміністрування послуг прокату транспортних засобів характеризуються високою складністю адаптації до реальних бізнес-процесів конкретного підприємства, оскільки універсальні програмні рішення не завжди враховують специфіку різних видів прокату, відмінності у моделях

обслуговування клієнтів та особливості правового регулювання. Недоліком є і підвищені вимоги до інформаційної безпеки, оскільки системи опрацювання даних у сфері прокату працюють з персональною та фінансовою інформацією клієнтів. Недостатній рівень захисту або помилки в налаштуванні доступу можуть призвести до витоку даних, що створює як репутаційні, так і правові ризики. Забезпечення належного рівня безпеки потребує додаткових ресурсів, регулярних оновлень та постійного контролю, що не завжди є економічно виправданим для невеликих компаній. Крім того, впровадження таких систем часто супроводжується організаційними труднощами, пов'язаними зі зміною усталених процесів роботи. Тому актуальною є розробка системи опрацювання даних для адміністрування послуг прокату транспортних засобів.

При розробці системи опрацювання даних для адміністрування послуг прокату транспортних засобів необхідно забезпечити такі функціональні вимоги:

- підтримка можливості збереження у базі даних відомостей про клієнтів служби прокату транспортних засобів;
- можливість реєстрації та авторизації користувачів системи з розмежуванням прав доступу;
- підтримка можливості внесення та коригування інформації про моделі автомобілів і конкретні транспортні засоби автопарку;
- можливість фіксування операцій бронювання та фактичної оренди автомобілів із збереженням відповідних параметрів;
- можливість обліку місць розташування транспортних засобів;
- підтримка можливості формування та перегляду переліку доступних автомобілів за заданими умовами;
- можливість отримання інформаційних повідомлень про результати виконання операцій у системі.

1.3 Висновки за розділом 1

Визначено, що система опрацювання даних для адміністрування послуг прокату транспортних засобів являє собою комплексне програмне рішення, призначене для автоматизованого управління всіма інформаційними процесами, що супроводжують діяльність підприємств з надання автомобілів у тимчасове користування. Така програмна система надає інтегроване середовище, у межах якого здійснюється накопичення, збереження, оброблення та аналіз відомостей про транспортний парк, клієнтів, договори оренди, фінансові операції та супутні сервісні процедури. Її функціонування ґрунтується на використанні сучасних інформаційних технологій, що забезпечують централізований доступ до даних і підтримують прийняття управлінських рішень.

За результатами проведеного аналізу зроблено висновок, що у наш час існує досить багато систем опрацювання даних для адміністрування послуг прокату транспортних засобів. Проте деякі системи опрацювання даних для адміністрування послуг прокату транспортних засобів характеризуються високою складністю адаптації до реальних бізнес-процесів конкретного підприємства, оскільки універсальні програмні рішення не завжди враховують специфіку різних видів прокату, відмінності у моделях обслуговування клієнтів та особливості правового регулювання. Недоліком є і підвищені вимоги до інформаційної безпеки, оскільки системи опрацювання даних у сфері прокату працюють з персональною та фінансовою інформацією клієнтів. Недостатній рівень захисту або помилки в налаштуванні доступу можуть призвести до витоку даних, що створює як репутаційні, так і правові ризики. Забезпечення належного рівня безпеки потребує додаткових ресурсів, регулярних оновлень та постійного контролю, що не завжди є економічно виправданим для невеликих компаній. Крім того, впровадження таких систем часто супроводжується організаційними труднощами, пов'язаними зі зміною усталених процесів роботи. Тому актуальною є розробка системи

опрацювання даних для адміністрування послуг прокату транспортних засобів.

Сформульовано функціональні вимоги до системи опрацювання даних для адміністрування послуг прокату транспортних засобів.

2 МАТЕРІАЛИ І МЕТОДИ

2.1 Вибір мови програмування

При розробці системи опрацювання даних для адміністрування послуг прокату транспортних засобів було використано мову програмування C# [14], [15].

Мова C# являє собою універсальний інструмент для розроблення сучасних інформаційних систем, який поєднує строгість класичних мов із гнучкістю сучасних підходів до програмної інженерії. Мова програмування C# активно розвивається та підтримується, що забезпечує її актуальність упродовж тривалого часу і знижує ризики морального застарівання створеного програмного продукту. Регулярне оновлення мови супроводжується появою нових синтаксичних конструкцій і функціональних можливостей, які спрощують написання читабельного та підтримуваного коду без зниження його надійності [14], [15].

Важливим аспектом є орієнтація C# на безпеку виконання програм. Автоматичне керування пам'яттю, контроль доступу до ресурсів і механізми обробки винятків сприяють підвищенню стабільності програмних систем та зменшенню кількості критичних помилок під час експлуатації. Це має особливе значення для прикладних систем, що працюють у режимі постійного доступу та обслуговують велику кількість користувачів. Крім того, C# надає засоби для реалізації перевірок коректності даних і бізнес-правил безпосередньо на рівні програмного коду, що підвищує загальну надійність системи [14], [15].

Окремої уваги заслуговує екосистема розробки, сформована навколо мови C#. Потужні середовища програмування забезпечують засоби налагодження, тестування та профілювання, що істотно полегшує процес створення складних програмних продуктів. Інтеграція з системами контролю версій, інструментами безперервної інтеграції та розгортання сприяє впровадженню сучасних практик командної розробки та підвищує якість

кінцевого результату. Завдяки цьому мова C# часто використовується у проєктах, де важливими є керуваність процесу розроблення та прогнозованість результатів [14], [15].

Кросплатформність мови дає змогу створювати програмні рішення, що функціонують у різних операційних середовищах без суттєвих змін коду. Це розширює можливості подальшого використання та масштабування розроблених систем. У поєднанні з підтримкою модульності та повторного використання компонентів C# забезпечує основу для побудови гнучких і розширюваних архітектур, що відповідають вимогам сучасних інформаційних систем і сприяють ефективному розвитку програмних продуктів у довгостроковій перспективі [14], [15].

Доцільність вибору мови програмування C# для створення системи опрацювання даних для адміністрування послуг прокату транспортних засобів зумовлюється поєднанням її технологічних можливостей, зрілості екосистеми та орієнтації на розробку прикладних інформаційних систем корпоративного рівня. C# спроектована як мова для побудови надійних, масштабованих і структурованих програмних рішень, що є критично важливим для систем, які працюють з великими обсягами даних і забезпечують безперервну підтримку бізнес-процесів у режимі реального часу [14], [15].

Використання платформи .NET, у межах якої функціонує C#, дозволяє ефективно реалізувати багаторівневу архітектуру системи, чітко розмежовуючи рівні представлення, бізнес-логіки та доступу до даних. Такий підхід сприяє підвищенню керуваності програмного коду, спрощує його супровід і подальший розвиток, що особливо актуально для систем адміністрування прокату, які з часом розширюються та модифікуються відповідно до змін вимог користувачів і ринку. Наявність розвинених засобів роботи з базами даних, зокрема через технології об'єктно-реляційного відображення, забезпечує зручну та безпечну взаємодію з інформаційними ресурсами [14], [15].

Важливим аргументом на користь мови програмування C# є високий рівень типізації та вбудовані механізми контролю помилок, що знижує ймовірність появи критичних збоїв під час виконання програм. Для систем опрацювання даних це має принципове значення, оскільки помилки в обробці інформації можуть призводити до фінансових втрат або порушення обслуговування клієнтів. Крім того, мова підтримує сучасні парадигми програмування та надає інструменти для створення асинхронних і паралельних процесів, що підвищує продуктивність системи при одночасній роботі з великою кількістю запитів [14], [15].

Не менш важливою є і практична складова вибору C#. Широке поширення цієї мови та наявність великої кількості перевірених бібліотек і фреймворків скорочують час розроблення й зменшують витрати на впровадження системи. Підтримка різних типів застосунків, включаючи настільні, веб- та серверні рішення, дозволяє створювати універсальні системи адміністрування прокату транспортних засобів, які можуть адаптуватися до різних форматів використання. У сукупності зазначені чинники обґрунтовують доцільність вибору мови програмування C# як ефективного інструмента для розроблення сучасної системи опрацювання даних у сфері прокату транспортних засобів [14], [15].

Отже, для реалізації системи опрацювання даних для адміністрування послуг прокату транспортних засобів обрано мову програмування C#, яка дає змогу створювати програмні рішення, що функціонують у різних операційних середовищах без суттєвих змін коду, що розширює можливості подальшого використання та масштабування розроблених програмних систем. Крім того, у поєднанні з підтримкою модульності та повторного використання компонентів мова C# забезпечує основу для побудови гнучких і розширюваних архітектур, що відповідають вимогам сучасних інформаційних систем і сприяють ефективному розвитку програмних продуктів у довгостроковій перспективі.

Обґрунтування вибору мови програмування для розробки системи опрацювання даних для адміністрування послуг прокату транспортних засобів наведено у таблиці 2.1.

Таблиця 2.1 – Обґрунтування вибору мови програмування для розробки системи опрацювання даних для адміністрування послуг прокату транспортних засобів

Критерій порівняння мов програмування	Мова програмування		
	C#	Java	PHP
Масштабованість системи	+	+	–
Асинхронність та паралельне програмування	+	+-	–
Легкість супроводу та модифікації коду	+	+-	+-
Безпека виконання та керування пам'яттю	+	+	–
Контроль помилок	+	+	–

2.2 Вибір середовища розробки для створення системи опрацювання даних для адміністрування послуг прокату транспортних засобів

В якості середовища розробки для створення системи опрацювання даних для адміністрування послуг прокату транспортних засобів було обрано середовище Visual Studio [16], [17].

Середовище розробки Visual Studio орієнтовано на підтримку різних типів програмних продуктів, що робить його ефективним не лише для корпоративних систем, але й для веб-, мобільних та хмарних додатків. Visual Studio забезпечує роботу з кількома мовами програмування одночасно,

включаючи C#, C++, F#, Python та інші, що дозволяє реалізовувати багатомовні проєкти та легко інтегрувати сторонні компоненти без значних змін у кодовій базі [16], [17].

Важливою характеристикою середовища розробки Visual Studio є потужна інтеграція з системами керування базами даних та хмарними сервісами, що дозволяє швидко налаштовувати доступ до SQL Server, MySQL, PostgreSQL або Azure, а також розгортати додатки у хмарному середовищі. Це особливо актуально для систем адміністрування прокату транспортних засобів, де потрібно обробляти великі обсяги даних і забезпечувати доступ у реальному часі [16], [17].

Середовище Visual Studio підтримує модульне розширення через плагіни та додаткові інструменти, що дозволяє адаптувати середовище під конкретні потреби проєкту та команди розробників. Можна інтегрувати інструменти для автоматичного тестування, аналізу коду на вразливості, документування або DevOps-процеси, що підвищує якість продукту та спрощує його супровід [16], [17].

Не менш важливим є зручний інтерфейс та можливість персоналізації середовища, що сприяє підвищенню продуктивності розробників і зменшенню часу на виконання рутинних завдань. Засоби візуального проєктування інтерфейсів і генерації коду для користувацьких форм значно спрощують створення функціональних і сучасних інтерфейсів, необхідних для роботи з клієнтами та внутрішніми користувачами системи [16], [17].

Крім того, Visual Studio надає розширені можливості для налагодження багатопотокових та розподілених програм, включаючи інтегроване профілювання продуктивності, трасування виконання та аналіз споживання ресурсів. Це дозволяє забезпечити стабільну роботу системи при високих навантаженнях і гарантувати швидкий доступ до даних без простоїв [16], [17].

У комплексі ці характеристики роблять Visual Studio потужним, надійним та гнучким середовищем для створення сучасних інформаційних

систем, здатних ефективно адмініструвати послуги прокату транспортних засобів та масштабуватися відповідно до потреб бізнесу [16], [17].

Доцільність вибору середовища розробки Visual Studio для створення системи опрацювання даних для адміністрування послуг прокату транспортних засобів зумовлюється його широкими функціональними можливостями, інтеграцією з мовою програмування C# та підтримкою сучасних підходів до розробки корпоративних інформаційних систем. Visual Studio забезпечує повноцінне середовище, в якому поєднуються редактор коду, засоби налагодження, тестування та управління проектами, що дозволяє розробникам виконувати всі етапи створення програмного продукту в одному інтерфейсі без необхідності використання додаткових інструментів [16], [17].

Важливою перевагою є висока інтеграція з платформою .NET, що робить можливим ефективне використання всіх сучасних технологій для роботи з базами даних, асинхронними процесами, багаторівневою архітектурою та вебсервісами. Інструменти автоматичного рефакторингу, підсвічування синтаксису, автодоповнення коду та аналізу помилок дозволяють підвищити продуктивність розробки, зменшити кількість помилок і спростити супровід системи в майбутньому [16], [17].

Середовище також пропонує потужні засоби для налагодження та профілювання продуктивності, що особливо важливо для систем адміністрування прокату транспортних засобів, де велика кількість одночасних запитів і обробка даних у реальному часі є критичною. Крім того, Visual Studio підтримує інтеграцію з системами контролю версій, інструментами безперервної інтеграції та розгортання, що спрощує командну розробку та забезпечує впорядковане управління змінами в коді [16], [17].

Ще одним важливим аспектом є наявність великої кількості готових шаблонів, бібліотек і розширень, які прискорюють процес створення функціональних модулів, таких як облік транспортних засобів, бронювання, керування договорами та платежами. Це дозволяє скоротити терміни розробки

та зменшити витрати, одночасно забезпечуючи високу якість і надійність програмного продукту [16], [17].

Таким чином, використання Visual Studio створює комплексне, ефективне та стабільне середовище для розробки системи опрацювання даних для адміністрування послуг прокату транспортних засобів, що значно підвищує продуктивність розробників, спрощує супровід та гарантує стабільну роботу кінцевої системи [16], [17].

Обґрунтування вибору середовища розробки для створення системи опрацювання даних для адміністрування послуг прокату транспортних засобів наведено у таблиці 2.2.

Таблиця 2.2 – Обґрунтування вибору середовища розробки для створення системи опрацювання даних для адміністрування послуг прокату транспортних засобів

Критерій порівняння середовищ розробки	Середовища розробки		
	Visual Studio	Rider	MonoDevelop
Зручність налагодження ПЗ	+	+-	+-
Робота з багаторівневою архітектурою	+	+-	+-
Профілювання продуктивності	+	+-	-
Зручність інструментів для тестування	+	+-	-
Інтеграція з системами контролю версій	+	+	+-

Таким чином, для створення системи опрацювання даних для адміністрування послуг прокату транспортних засобів обрано середовище розробки Visual Studio, що обумовлено його широкими функціональними

можливостями, інтеграцією з мовою програмування C# та підтримкою сучасних підходів до розробки корпоративних інформаційних систем. Visual Studio забезпечує повноцінне середовище, в якому поєднуються редактор коду, засоби налагодження, тестування та управління проєктами, що дозволяє розробникам виконувати всі етапи створення програмного продукту в одному інтерфейсі без необхідності використання додаткових інструментів.

2.3 Вибір засобів зберігання та опрацювання даних

Для зберігання та опрацювання даних було обрано систему управління базами даних SQL Server [18], [19].

Доцільність вибору системи керування базами даних SQL Server для зберігання та обробки даних системи опрацювання послуг прокату транспортних засобів обґрунтовується її високою продуктивністю, надійністю та розширеними можливостями для роботи з великими обсягами структурованої інформації. SQL Server забезпечує централізоване зберігання даних, що дозволяє гарантувати їх цілісність, уніфікованість і актуальність, а також створює умови для ефективного контролю доступу до інформації та її захисту від несанкціонованих змін [18], [19].

Система управління базами даних SQL Server пропонує потужні інструменти для організації реляційних структур, що спрощує моделювання даних про транспортні засоби, клієнтів, бронювання, договори та фінансові операції. Завдяки підтримці транзакцій та механізмів відновлення після збоїв SQL Server забезпечує надійність обробки даних навіть у випадках аварій або високого навантаження. Це особливо важливо для систем адміністрування прокату, де своєчасність і точність інформації безпосередньо впливають на якість обслуговування клієнтів та ефективність використання транспортного парку [18], [19].

Крім того, система управління базами даних SQL Server підтримує інтеграцію з мовою програмування C# і середовищем розробки Visual Studio,

що дозволяє реалізовувати ефективний доступ до бази даних, виконувати складні запити, зберігати процедури та тригери для автоматизації бізнес-логіки. Можливості аналітики та побудови звітів на базі SQL Server дають змогу отримувати оперативну статистику про завантаженість транспортних засобів, попит на різні послуги та фінансові показники, що сприяє прийняттю обґрунтованих управлінських рішень [18], [19].

Не менш важливим є масштабований характер SQL Server, який дозволяє збільшувати обсяги зберігання та продуктивність бази даних без значних змін у структурі системи. Підтримка реплікації, кластеризації та резервного копіювання гарантує безперервність роботи системи та захист від втрати даних [18], [19].

Обґрунтування вибору засобів зберігання та опрацювання даних наведено у таблиці 2.3.

Таблиця 2.3 – Обґрунтування вибору засобів зберігання та опрацювання даних

Критерій порівняння систем управління базами даних	Системи управління базами даних		
	SQL Server	MySQL	PostgreSQL
Надійність та стійкість до збоїв	+	+-	+
Інтеграція з C# та Visual Studio	+	+-	+-
Безпека та контроль доступу	+	+-	+
Легкість адміністрування та моніторингу	+	+-	+-
Підтримка складних запитів і процедур	+	+-	+

Таким чином, для зберігання та опрацювання даних обрано систему управління базами даних SQL Server завдяки поєднанню високої надійності, продуктивності та розширених можливостей обробки даних, які є критично важливими для систем адміністрування послуг прокату транспортних засобів. SQL Server забезпечує ефективну роботу з великими обсягами структурованої інформації, підтримує транзакційність і цілісність даних, а також тісно інтегрується з мовою програмування C# і середовищем розробки Visual Studio, що спрощує реалізацію бізнес-логіки та супровід системи. Масштабованість, розвинені механізми безпеки та зручні засоби адміністрування роблять SQL Server оптимальним вибором для створення стабільного, керованого й перспективного програмного продукту.

2.4 Висновки за розділом 2

Для реалізації системи опрацювання даних для адміністрування послуг прокату транспортних засобів обрано мову програмування C#, яка дає змогу створювати програмні рішення, що функціонують у різних операційних середовищах без суттєвих змін коду, що розширює можливості подальшого використання та масштабування розроблених програмних систем. Крім того, у поєднанні з підтримкою модульності та повторного використання компонентів мова C# забезпечує основу для побудови гнучких і розширюваних архітектур, що відповідають вимогам сучасних інформаційних систем і сприяють ефективному розвитку програмних продуктів у довгостроковій перспективі.

Для створення системи опрацювання даних для адміністрування послуг прокату транспортних засобів обрано середовище розробки Visual Studio, що обумовлено його широкими функціональними можливостями, інтеграцією з мовою програмування C# та підтримкою сучасних підходів до розробки корпоративних інформаційних систем. Visual Studio забезпечує повноцінне середовище, в якому поєднуються редактор коду, засоби налагодження, тестування та управління проєктами, що дозволяє розробникам виконувати всі

етапи створення програмного продукту в одному інтерфейсі без необхідності використання додаткових інструментів.

Для зберігання та опрацювання даних обрано систему управління базами даних SQL Server завдяки поєднанню високої надійності, продуктивності та розширених можливостей обробки даних, які є критично важливими для систем адміністрування послуг прокату транспортних засобів. SQL Server забезпечує ефективну роботу з великими обсягами структурованої інформації, підтримує транзакційність і цілісність даних, а також тісно інтегрується з мовою програмування C# і середовищем розробки Visual Studio, що спрощує реалізацію бізнес-логіки та супровід системи. Масштабованість, розвинені механізми безпеки та зручні засоби адміністрування роблять SQL Server оптимальним вибором для створення стабільного, керованого й перспективного програмного продукту.

3 ОПИС СИСТЕМИ ОПРАЦЮВАННЯ ДАНИХ

3.1 Структура системи опрацювання даних для адміністрування послуг прокату транспортних засобів

Структуру системи опрацювання даних для адміністрування послуг прокату транспортних засобів наведено на рис. 3.1.

Структура системи опрацювання даних для адміністрування послуг прокату транспортних засобів сформована з урахуванням необхідності чіткого розмежування логічних рівнів програмного забезпечення та забезпечення можливості його подальшого розвитку. У процесі проєктування було прийнято рішення відокремити механізми роботи з інформаційними ресурсами від компонентів, що відповідають за взаємодію з користувачем, оскільки такий підхід сприяє підвищенню гнучкості архітектури та спрощує внесення змін без суттєвого впливу на інші частини програми. Основу системи опрацювання даних для адміністрування послуг прокату транспортних засобів становить набір програмних модулів, орієнтованих на обробку даних предметної області, які реалізують операції з відомостями про транспортні засоби, клієнтів, користувачів, процеси оренди, тарифи та місця видачі й повернення автомобілів.

Для забезпечення взаємодії з базою даних передбачено спеціалізований компонент, що виконує функції підключення до сховища, формування запитів, отримання результатів та перетворення їх у вигляд об'єктів програмної моделі. Окремі програмні сутності відповідають за представлення моделей автомобілів, характеристик конкретних транспортних засобів, даних про клієнтів і співробітників, а також за фіксацію операцій бронювання та розрахунку вартості оренди відповідно до встановлених тарифів.

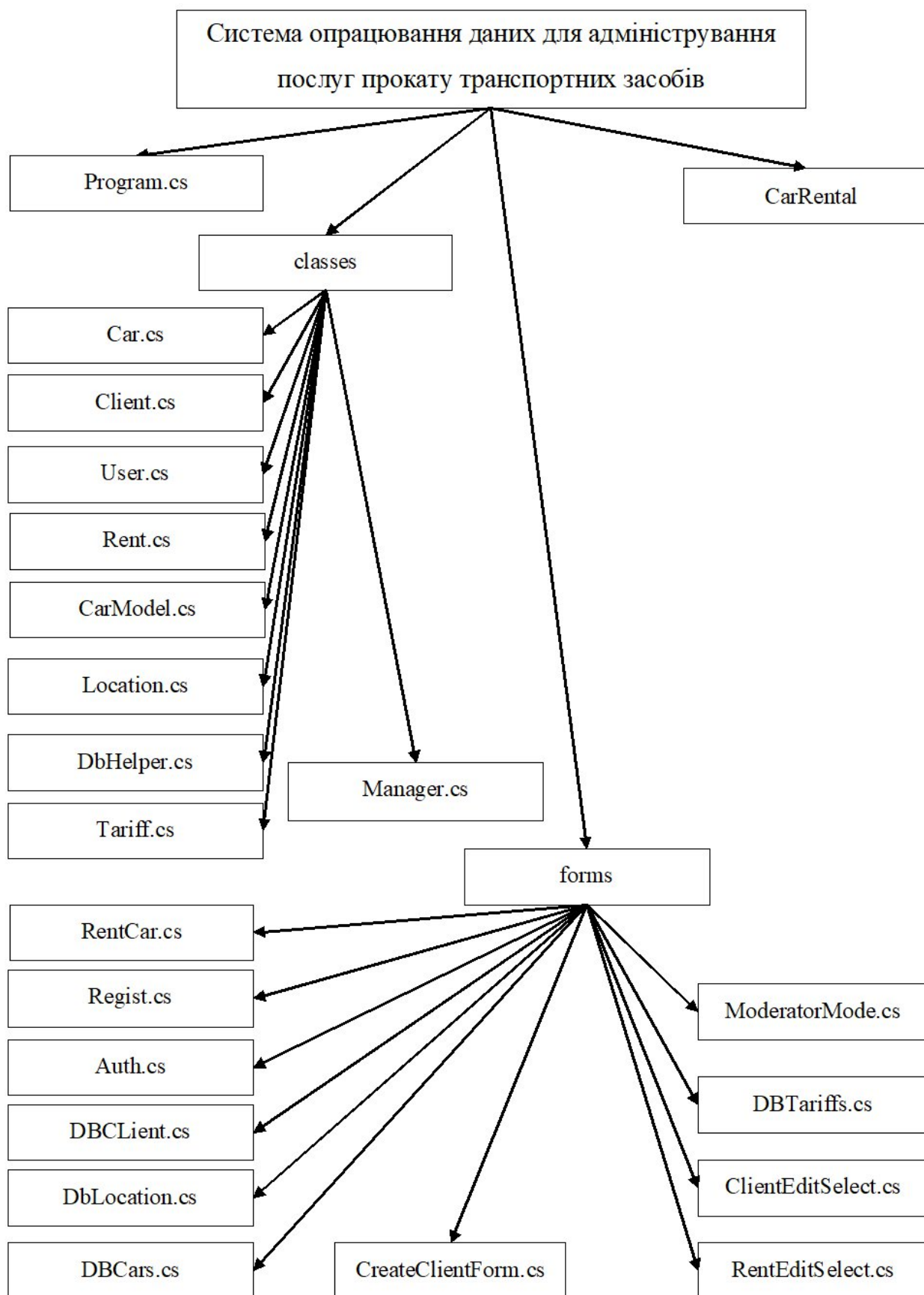


Рисунок 3.1 – Структура системи опрацювання даних для адміністрування послуг прокату транспортних засобів

У складі проекту передбачено наявність файлу бази даних, що забезпечує фізичне зберігання всієї інформації, та головного програмного модуля, який визначає точку запуску застосунку і координує взаємодію між його складовими. Значну частину структури займає підсистема користувацького інтерфейсу, призначена для створення візуальних форм, відображення даних та обробки дій користувачів, що дозволяє забезпечити зручний і зрозумілий спосіб роботи з функціональними можливостями системи.

У сукупності така організація програмних компонентів формує цілісну архітектуру, у якій логіка опрацювання даних, механізми доступу до бази та інтерфейсні елементи функціонують узгоджено, що створює передумови для стабільної експлуатації системи та її подальшої модернізації відповідно до потреб адміністрування послуг прокату транспортних засобів.

Запропонована структура програми свідчить про орієнтацію на чітке розмежування функціональних обов'язків між окремими компонентами системи, що сприяє підвищенню впорядкованості та керованості програмного коду. Відокремлення рівня роботи з даними від рівня користувацького інтерфейсу дозволяє забезпечити незалежність логіки опрацювання інформації від способів її подання, що створює можливості для подальшої модернізації та адаптації програмного продукту до нових вимог без необхідності кардинальної перебудови всієї системи.

Сформована архітектура забезпечує узгоджену взаємодію між модулями доступу до бази даних, об'єктами предметної області та інтерфейсними елементами, завдяки чому досягається цілісність процесів адміністрування послуг прокату транспортних засобів. Такий підхід підвищує надійність функціонування програми, спрощує її супровід і розширення, а також створює основу для ефективного використання системи в умовах реальної експлуатації.

3.2 Функціонування системи опрацювання даних для адміністрування послуг прокату транспортних засобів

Функціонування системи опрацювання даних для адміністрування послуг прокату транспортних засобів подамо за допомогою схеми, зображеної на рис. 3.2.

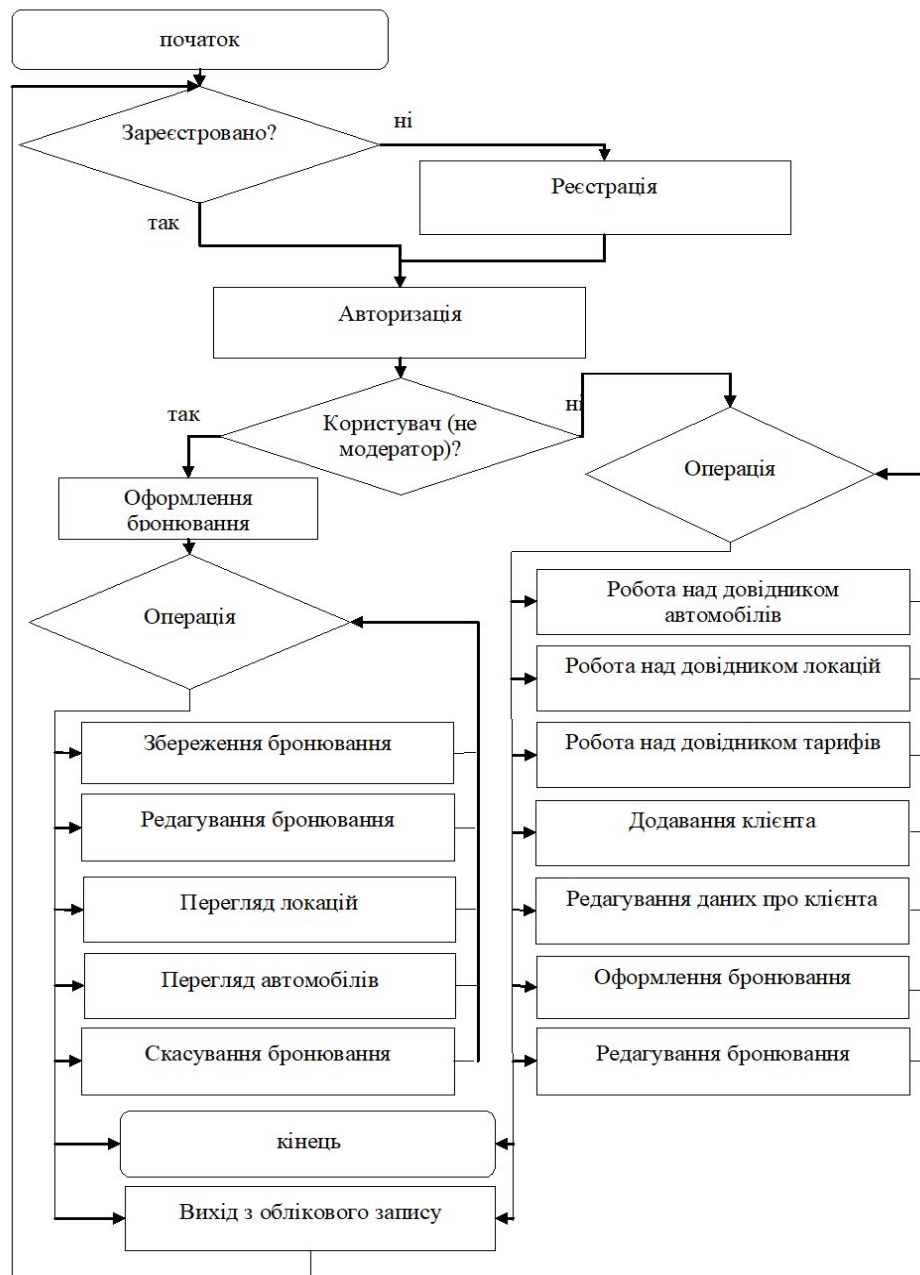


Рисунок 3.2 – Схема функціонування системи опрацювання даних для адміністрування послуг прокату транспортних засобів

Функціонування системи опрацювання даних для адміністрування послуг прокату транспортних засобів ґрунтується на постійній взаємодії користувача з елементами графічного інтерфейсу та програмними модулями обробки інформації. Усі дії в межах програми здійснюються через візуальні вікна та діалогові форми, за допомогою яких забезпечується доступ до необхідних функцій і сервісів. Кожен елемент інтерфейсу пов'язаний із відповідними процедурами обробки даних, що реалізують звернення до бази даних, отримання актуальної інформації та її подальше відображення у зручному для користувача вигляді.

Початковим етапом роботи системи є перевірка облікових даних особи, яка намагається отримати доступ до функціональних можливостей програмного продукту. За відсутності зареєстрованого профілю передбачено можливість створення нового облікового запису з подальшим внесенням необхідних персональних відомостей. Після підтвердження автентичності користувача система автоматично визначає його роль і відповідний набір доступних операцій, що дозволяє розмежувати повноваження та забезпечити контроль над виконанням дій.

У подальшому робота програми відбувається відповідно до статусу користувача. Для осіб, відповідальних за безпосереднє обслуговування клієнтів, надаються засоби оформлення нових замовлень на оренду транспортних засобів, коригування вже внесених відомостей, перегляду інформації про наявні автомобілі та місця їх видачі. Передбачено можливість скасування помилково створених записів, а також завершення сеансу роботи або виходу з особистого облікового запису. Усі ці дії супроводжуються автоматичним оновленням даних у сховищі та контролем їхньої узгодженості.

Користувачі з розширеними правами доступу мають змогу здійснювати адміністрування довідкової інформації, редагувати відомості про транспортні засоби, тарифи, локації та клієнтів, а також виконувати повний цикл роботи з бронюваннями. Система забезпечує перевірку коректності введених даних, запобігає дублюванню записів і підтримує логічну цілісність взаємопов'язаних

об'єктів. Усі операції ініціюються діями користувача, а програмне забезпечення реагує на них шляхом виконання відповідних алгоритмів обробки та оновлення інформаційних ресурсів.

Діаграму прецедентів системи опрацювання даних для адміністрування послуг прокату транспортних засобів подано на рис. 3.3.

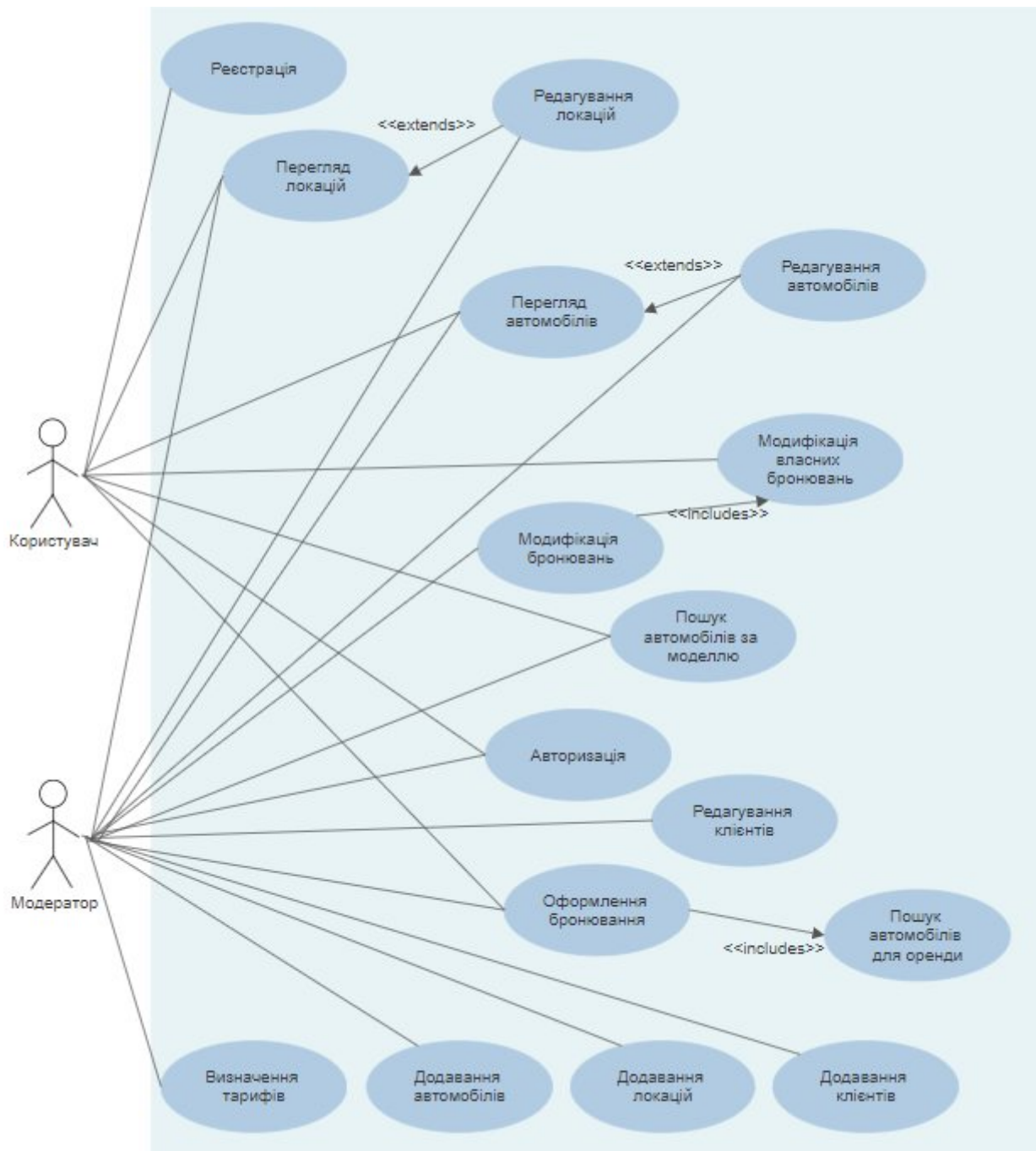


Рисунок 3.3 – Діаграма прецедентів системи опрацювання даних для адміністрування послуг прокату транспортних засобів

Таким чином, функціонування програми характеризується чіткою орієнтацією на подієву модель взаємодії, у якій користувач виступає ініціатором процесів, а система забезпечує їх коректне та послідовне виконання. Реалізований механізм роботи дозволяє підтримувати повний цикл адміністрування послуг прокату транспортних засобів, забезпечуючи актуальність даних, зручність користування та надійність обробки інформації.

3.3 Розробка бази даних системи опрацювання даних для адміністрування послуг прокату транспортних засобів

База даних системи опрацювання даних для адміністрування послуг прокату транспортних засобів призначена для зберігання даних, необхідних для роботи програмної системи.

Схему бази даних системи опрацювання даних для адміністрування послуг прокату транспортних засобів наведено на рис. 3.4.

Схема бази даних системи опрацювання даних для адміністрування послуг прокату транспортних засобів сформована як основа інформаційного забезпечення всього програмного комплексу, оскільки саме на рівні сховища даних відбувається накопичення та підтримка відомостей, необхідних для реалізації функцій бронювання, обліку автомобілів і взаємодії з клієнтами. Проектування структури бази даних було здійснено з урахуванням особливостей предметної області та орієнтації на використання можливостей реляційної системи керування базами даних Microsoft SQL Server, що дозволяє забезпечити цілісність, узгодженість і надійність збереження інформації. Модель даних побудована на основі виділення ключових об'єктів діяльності служби прокату та відображення між ними логічних взаємозв'язків, які відтворюють реальні процеси оформлення оренди та експлуатації транспортних засобів.

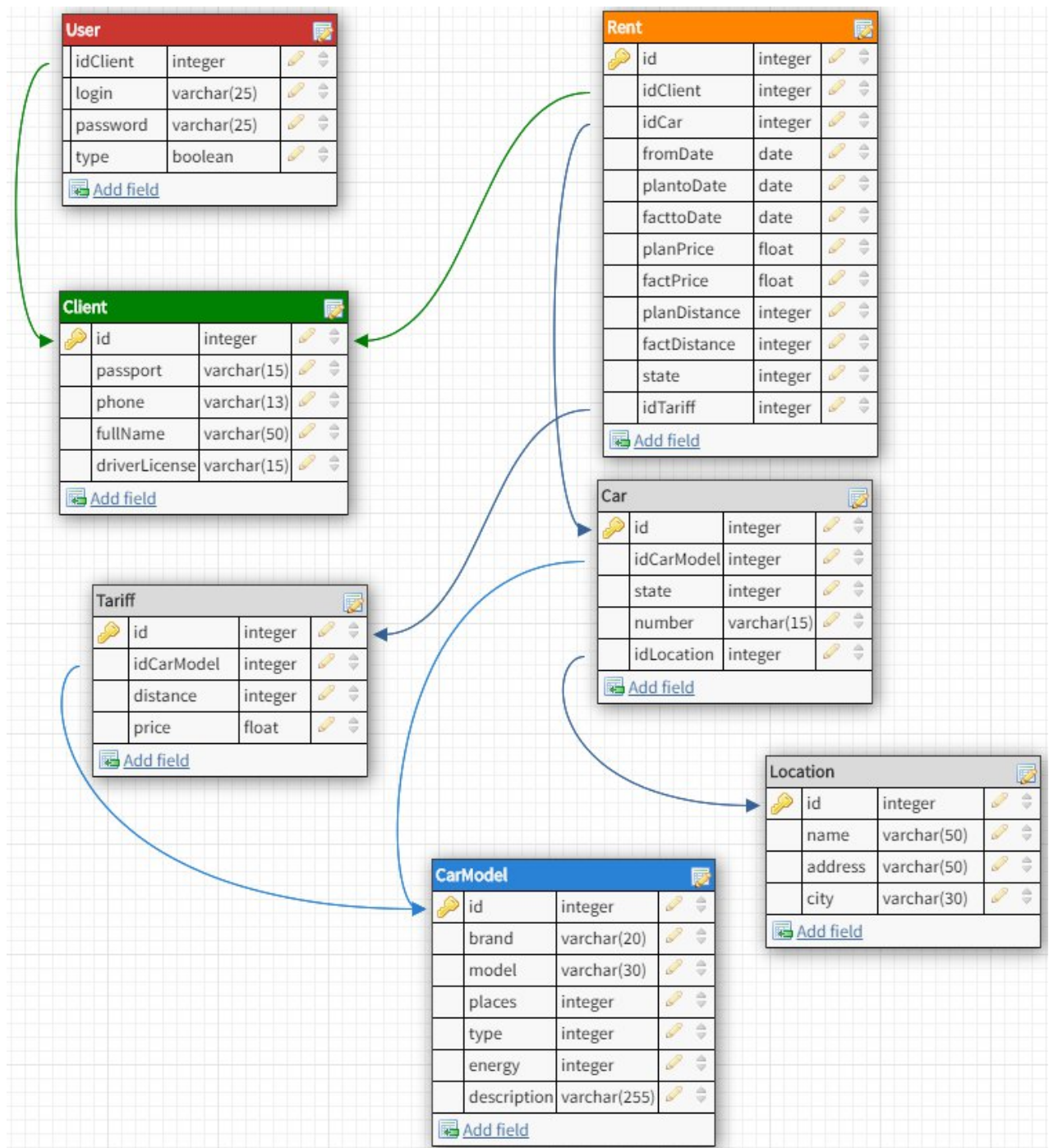


Рисунок 3.4 – Схема бази даних системи опрацювання даних для адміністрування послуг прокату транспортних засобів

У структурі бази даних передбачено зберігання відомостей про облікові записи користувачів системи, дані про клієнтів, характеристики моделей автомобілів, інформацію про конкретні транспортні засоби, записи про здійснені або заплановані орендні операції, відомості про діючі тарифи та про місця видачі й повернення автомобілів. Розмежування інформації про

користувачів та клієнтів реалізовано шляхом виділення окремих логічних сутностей, що дозволяє розділити дані, необхідні для автентифікації в системі, та відомості, які використовуються під час оформлення договорів оренди. Такий підхід забезпечує гнучкість у роботі з різними категоріями осіб, у тому числі з тими, що не мають власного облікового запису.

Між основними елементами схеми встановлено відношення, які відображають правила функціонування сервісу прокату. Кожен клієнт може багаторазово користуватися послугами оренди, тому записи про орендні операції пов'язуються з відповідною особою через механізм зовнішніх ключів. Аналогічним чином кожний автомобіль може брати участь у значній кількості бронювань, проте окрема операція оренди стосується лише одного транспортного засобу, що забезпечує однозначність обліку. Тарифи, за якими розраховується вартість послуг, також пов'язуються з орендними записами, оскільки для кожної операції має застосовуватися лише один чинний тариф, сформований з урахуванням моделі автомобіля та умов використання.

Інформація про конкретні автомобілі підпорядковується даним про їхні моделі, що дає можливість групувати транспортні засоби за спільними характеристиками та застосовувати до них відповідні правила тарифікації. Місця розташування автомобілів виступають окремим елементом схеми, з яким пов'язуються відомості про кожну одиницю автопарку, що дозволяє контролювати процес видачі та повернення транспортних засобів у різних локаціях. Для реалізації зазначених взаємозв'язків у таблицях бази даних передбачено використання зовнішніх ключів, за допомогою яких забезпечується узгодженість між даними різних об'єктів та дотримання обмежень цілісності.

Розроблена структура таблиць створює можливості для виконання всіх необхідних операцій з інформацією, включаючи додавання нових записів, їх редагування, пошук та формування аналітичних вибірок. Завдяки чіткому визначенню атрибутів і зв'язків між ними забезпечується коректне

відображення бізнес-процесів служби прокату у вигляді реляційної моделі, придатної для подальшого розширення та оптимізації.

Отже, спроектована база даних становить логічно впорядковане інформаційне ядро програми, яке відтворює основні об'єкти та відношення предметної області й забезпечує надійне зберігання та обробку даних, необхідних для повноцінного функціонування системи адміністрування послуг прокату транспортних засобів.

3.4 Проєктування інтерфейсу системи опрацювання даних для адміністрування послуг прокату транспортних засобів

Проєктування інтерфейсу взаємодії користувача з системою опрацювання даних для адміністрування послуг прокату транспортних засобів виконано з урахуванням вимог технічного завдання. При розробленні системи опрацювання даних для адміністрування послуг прокату транспортних засобів враховані функціональні вимоги до програми:

- підтримка можливості збереження у базі даних відомостей про клієнтів служби прокату транспортних засобів;
- можливість реєстрації та авторизації користувачів системи з розмежуванням прав доступу;
- підтримка можливості внесення та коригування інформації про моделі автомобілів і конкретні транспортні засоби автопарку;
- можливість фіксування операцій бронювання та фактичної оренди автомобілів із збереженням відповідних параметрів;
- можливість обліку місць розташування транспортних засобів;
- підтримка можливості формування та перегляду переліку доступних автомобілів за заданими умовами;
- можливість отримання інформаційних повідомлень про результати виконання операцій у системі.

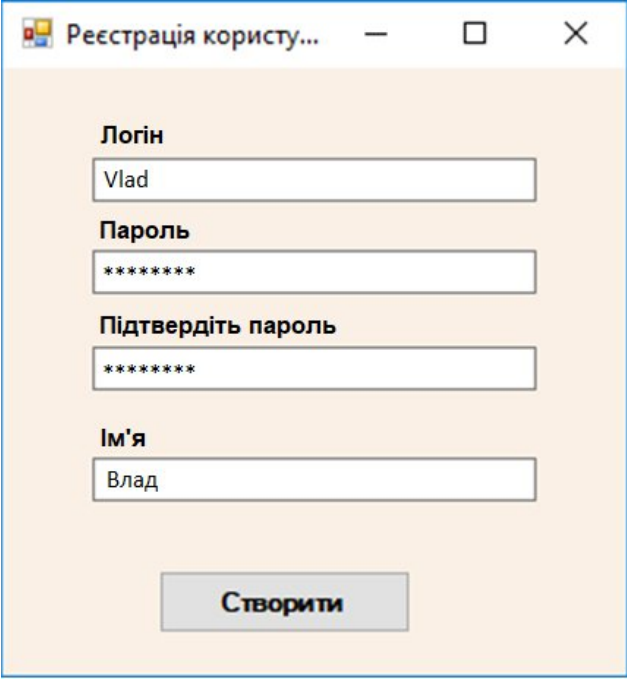
Інтерфейс головної форми системи опрацювання даних для адміністрування послуг прокату транспортних засобів наведено на рис. 3.5.

Рисунок 3.5 – Інтерфейс головної форми системи опрацювання даних для адміністрування послуг прокату транспортних засобів

Форму роботи з системою у режимі адміністратора наведено на рис. 3.6.

Рисунок 3.6 – Форма роботи з системою у режимі адміністратора

Інтерфейс форми реєстрації наведено на рис. 3.7.



The image shows a screenshot of a web application window titled "Реєстрація користу...". The window has a light orange background and contains four input fields and a button. The first field is labeled "Логін" (Login) and contains the text "Vlad". The second field is labeled "Пароль" (Password) and contains seven asterisks "*****". The third field is labeled "Підтвердіть пароль" (Confirm password) and also contains seven asterisks "*****". The fourth field is labeled "Ім'я" (Name) and contains the text "Влад". Below the input fields is a gray button labeled "Створити" (Create).

Рисунок 3.7 – Інтерфейс форми реєстрації

3.5 Висновки за розділом 3

Розроблено систему опрацювання даних для адміністрування послуг прокату транспортних засобів.

Запропоновано структуру системи опрацювання даних для адміністрування послуг прокату транспортних засобів, яка сформована з урахуванням необхідності чіткого розмежування логічних рівнів програмного забезпечення та забезпечення можливості його подальшого розвитку. У процесі проєктування було прийнято рішення відокремити механізми роботи з інформаційними ресурсами від компонентів, що відповідають за взаємодію з користувачем, оскільки такий підхід сприяє підвищенню гнучкості архітектури та спрощує внесення змін без суттєвого впливу на інші частини програми. У сукупності така організація програмних компонентів формує цілісну архітектуру, у якій логіка опрацювання даних, механізми доступу до бази та інтерфейсні елементи функціонують узгоджено, що створює

передумови для стабільної експлуатації системи та її подальшої модернізації відповідно до потреб адміністрування послуг прокату транспортних засобів.

Описано функціонування системи опрацювання даних для адміністрування послуг прокату транспортних засобів. Функціонування програми характеризується чіткою орієнтацією на подієву модель взаємодії, у якій користувач виступає ініціатором процесів, а система забезпечує їх коректне та послідовне виконання. Реалізований механізм роботи дозволяє підтримувати повний цикл адміністрування послуг прокату транспортних засобів, забезпечуючи актуальність даних, зручність користування та надійність обробки інформації.

Розроблено базу даних системи опрацювання даних для адміністрування послуг прокату транспортних засобів. У структурі бази даних передбачено зберігання відомостей про облікові записи користувачів системи, дані про клієнтів, характеристики моделей автомобілів, інформацію про конкретні транспортні засоби, записи про здійснені або заплановані орендні операції, відомості про діючі тарифи та про місця видачі й повернення автомобілів. Розмежування інформації про користувачів та клієнтів реалізовано шляхом виділення окремих логічних сутностей, що дозволяє розділити дані, необхідні для автентифікації в системі, та відомості, які використовуються під час оформлення договорів оренди. Такий підхід забезпечує гнучкість у роботі з різними категоріями осіб, у тому числі з тими, що не мають власного облікового запису.

Виконано проектування інтерфейсу взаємодії користувача з системою опрацювання даних для адміністрування послуг прокату транспортних засобів.

4 ЕКСПЛУАТАЦІЯ, ТЕСТУВАННЯ ТА ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ СИСТЕМИ ОПРАЦЮВАННЯ ДАНИХ

4.1 Призначення й умови застосування системи опрацювання даних

Система опрацювання даних призначена для підтримки процесу адміністрування послуг прокату транспортних засобів. Система опрацювання даних написана на мові C#, яка дає змогу створювати програмні рішення, що функціонують у різних операційних середовищах без суттєвих змін коду, що розширює можливості подальшого використання та масштабування розроблених програмних систем. В якості середовища розробки для створення системи опрацювання даних для адміністрування послуг прокату транспортних засобів обрано Visual Studio, що обумовлено його широкими функціональними можливостями, інтеграцією з мовою програмування C# та підтримкою сучасних підходів до розробки корпоративних інформаційних систем.

Система опрацювання даних для адміністрування послуг прокату транспортних засобів забезпечує виконання таких функцій:

- підтримка можливості збереження у базі даних відомостей про клієнтів служби прокату транспортних засобів;
- можливість реєстрації та авторизації користувачів системи з розмежуванням прав доступу;
- підтримка можливості внесення та коригування інформації про моделі автомобілів і конкретні транспортні засоби автопарку;
- можливість фіксування операцій бронювання та фактичної оренди автомобілів із збереженням відповідних параметрів;
- можливість обліку місць розташування транспортних засобів;
- підтримка можливості формування та перегляду переліку доступних автомобілів за заданими умовами;

– можливість отримання інформаційних повідомлень про результати виконання операцій у системі.

Для роботи системи опрацювання даних для адміністрування послуг прокату транспортних засобів потрібні такі технічні та програмні ресурси: комп'ютер, оснащений процесором з тактовою частотою не нижче 2,0 ГГц, що дозволяє виконувати операції обробки запитів до бази даних та роботу графічного інтерфейсу без помітних затримок. Обсяг оперативної пам'яті має становити не менше 4 ГБ. Для розміщення самої програми, файлів бази даних та допоміжних компонентів необхідно щонайменше 10 ГБ вільного простору на жорсткому диску або твердотільному накопичувачі.

У програмному середовищі обов'язковою є наявність операційної системи сімейства Windows актуальної версії з підтримкою платформи .NET. Для зберігання й обробки інформації повинна бути встановлена система керування базами даних Microsoft SQL Server або її сумісна редакція. Для можливості супроводу й подальшого розвитку програми доцільною є наявність середовища розробки Visual Studio.

Також рекомендованою є наявність стандартних засобів введення та виведення інформації, зокрема монітора для коректного відображення форм інтерфейсу, клавіатури та маніпулятора типу миша.

Функціональні характеристики розробленої системи опрацювання даних для адміністрування послуг прокату транспортних засобів наведено у технічному завданні (додаток А). Фрагмент тексту програмного коду системи опрацювання даних для адміністрування послуг прокату транспортних засобів наведено у додатку Б.

4.2 Характеристики системи опрацювання даних для адміністрування послуг прокату транспортних засобів

Система опрацювання даних для адміністрування послуг прокату транспортних засобів характеризується комплексною спрямованістю на

автоматизацію основних управлінських та облікових процесів підприємства, що надає послуги оренди. Програмний продукт орієнтовано на забезпечення централізованого зберігання відомостей про транспортні одиниці, клієнтів, укладені договори та здійснені операції, завдяки чому формується єдине інформаційне середовище для роботи персоналу. Передбачено можливість оперативного внесення, редагування та перегляду даних.

Функціональні можливості системи дозволяють здійснювати облік доступності транспортних засобів, контролювати строки їх використання та фіксувати фінансові розрахунки між підприємством і замовниками. Інтерфейс користувача побудовано з урахуванням принципів зручності та логічної послідовності дій, що спрощує виконання типових операцій без необхідності залучення складних інструкцій. Реалізовано механізми пошуку та фільтрації відомостей за різними ознаками, завдяки чому забезпечується швидке отримання необхідних даних для прийняття управлінських рішень.

Програмне рішення підтримує розмежування прав доступу, що дає змогу організувати роботу різних категорій користувачів відповідно до їхніх службових повноважень. Передбачено формування звітної інформації у вигляді структурованих документів, придатних для подальшого аналізу та використання у внутрішній діяльності установи. Архітектура системи побудована з урахуванням можливості подальшого розширення функціональності та інтеграції з іншими програмними засобами, що створює передумови для її довготривалої експлуатації та адаптації до змін у процесах надання послуг прокату.

4.3 Інструкція по експлуатації програми

4.3.1 Звернення до програми

Для запуску системи опрацювання даних для адміністрування послуг прокату транспортних засобів необхідно запустити виконуваний файл.

4.3.2 Вхідні й вихідні дані

Вхідними даними до системи опрацювання даних для адміністрування послуг прокату транспортних засобів є інформація про клієнтів, параметри транспортних засобів, умови та терміни оренди, дані щодо тарифів і місць отримання або повернення автомобілів.

Вихідними даними системи опрацювання даних для адміністрування послуг прокату транспортних засобів є інформація, що надається користувачеві у відповідь на виконані запити та операції, зокрема відомості про доступні автомобілі, перелік оформлених бронювань, дані про клієнтів і фінансові нарахування.

4.3.3 Повідомлення

Користувач системи опрацювання даних для адміністрування послуг прокату транспортних засобів може отримати такі повідомлення: сповіщення про успішну авторизацію або відмову у доступі, попередження про некоректність введених даних, інформування про відсутність доступних транспортних засобів за заданими параметрами, підтвердження виконання операцій збереження чи редагування.

4.4 Виконання системи опрацювання даних для адміністрування послуг прокату транспортних засобів

Після запуску системи опрацювання даних для адміністрування послуг прокату транспортних засобів користувачу відображається форма авторизації (рис. 4.1), де необхідно ввести облікові дані, що були створені раніше, та підтвердити їх коректність за допомогою відповідного елемента керування.

Авторизація

Логін

Пароль

Підтвердити

Реєстрація

Рисунок 4.1 – Форма авторизації у системі опрацювання даних для адміністрування послуг прокату транспортних засобів

У разі відсутності облікових даних користувач має можливість ініціювати процедуру створення нового запису, що здійснюється через окреме вікно реєстрації (рис. 4.2).

Реєстрація користу...

Логін

Vlad

Пароль

Підтвердіть пароль

Ім'я

Влад

Створити

Рисунок 4.2 – Форма реєстрації у системі опрацювання даних для адміністрування послуг прокату транспортних засобів

У межах системи передбачено існування декількох категорій користувачів, які відрізняються обсягом доступних повноважень. Звичайні користувачі можуть самостійно проходити процедуру реєстрації, тоді як облікові записи з розширеними правами створюються виключно адміністратором шляхом безпосереднього внесення даних до сховища інформації. Для отримання такого доступу необхідно звернутися до відповідальної особи, що здійснює адміністрування системи.

Після успішного проходження процедури ідентифікації користувачеві відкривається основне робоче вікно (рис. 4.3), призначене для оформлення замовлення на оренду транспортного засобу.

The screenshot shows a web application window titled "Бронювання автомобіля для оренди". The interface is divided into several sections:

- Вийти**: A button at the top center.
- Період часу**: A section containing two input fields: "Дата початку" (01.05.2026) and "Кількість днів" (1).
- Вибрати автомобіль**: A section with a "Пошук за моделлю" subsection containing "Виробник" and "Модель" input fields, a "Знайти" button, and a "Локація" dropdown menu with an "Обрати" button.
- Обрати модель**: A section with a "Локація" dropdown menu, "Виробник" and "Модель" dropdown menus, and an "Обрати" button.
- Бронювання**: A sidebar section with a "Номер" dropdown menu, "Змінити", and "Скасувати" buttons.
- Перегляд**: A sidebar section with "Автомобілі" and "Локації" buttons.
- Bottom Bar**: Contains "Дистанція (км)" input field, "Розрахована вартість" (0.00), and a "Забронювати" button.

Рисунок 4.3 – Форма бронювання автотранспортних засобів

Інтерфейс оформлення оренди побудовано таким чином, щоб користувач міг поетапно визначити всі необхідні параметри майбутньої послуги. Спочатку встановлюються часові межі використання автомобіля, для

чого задається початкова дата та тривалість експлуатації. Далі здійснюється вибір конкретного транспортного засобу, що може виконуватися за допомогою різних механізмів пошуку та відбору відповідно до моделі або інших характеристик. Така організація процесу дає змогу адаптувати процедуру замовлення до індивідуальних потреб клієнта.

Після визначення основних параметрів користувач вводить відомості про плановану відстань поїздки, на основі чого система автоматично формує попередній розрахунок вартості послуги (рис. 4.4).

Бронювання автомобіля для оренди

Вийти

Період часу

Дата початку: 01.05.2026

Кількість днів: 5

Вибрати автомобіль

Пошук за моделлю

Виробник: Ford

Моделль: Fiesta

Знайти

Покація: Запоріжжя

Обрати

Обрати модель

Покація:

Виробник:

Моделль:

Обрати

Бронювання

Номер:

Змінити

Скасувати

Перегляд

Автомобілі

Локації

Дистанція (км): 500

Розрахована вартість: 5500.00

Забронювати

Рисунок 4.4 – Обчислення вартості оренди автотransпортного засобу

Обчислення здійснюється шляхом визначення середнього пробігу за одиницю часу та підбору тарифу, що відповідає обраному автомобілю і заданим умовам експлуатації. Отримане значення множиться на тривалість оренди, у результаті чого формується підсумкова сума до сплати. У разі зміни будь-якого з параметрів розрахунок виконується повторно без додаткових дій

з боку користувача. Після підтвердження замовлення інформація стає доступною для подальшого опрацювання уповноваженим працівником, який за необхідності може внести корективи до визначеної вартості, і такі зміни надалі відображатимуться під час перегляду відповідного запису.

У разі входу до системи з обліковими даними, що відповідають ролі менеджера, користувачеві надається доступ до спеціалізованого інтерфейсу керування (рис. 4.5), призначеного для виконання адміністративних і організаційних операцій. Відкрите робоче вікно об'єднує інструменти для роботи з основними інформаційними об'єктами та процесами, які забезпечують повноцінне функціонування служби прокату. Структура інтерфейсу побудована таким чином, щоб розмежувати дії, пов'язані з підтримкою довідкових відомостей, обробленням замовлень на оренду та веденням інформації про клієнтів.

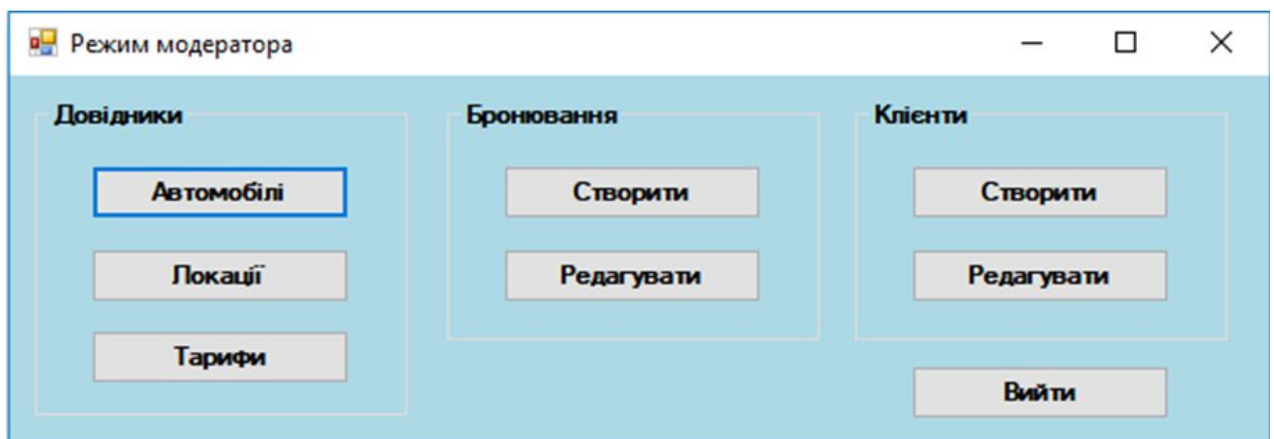
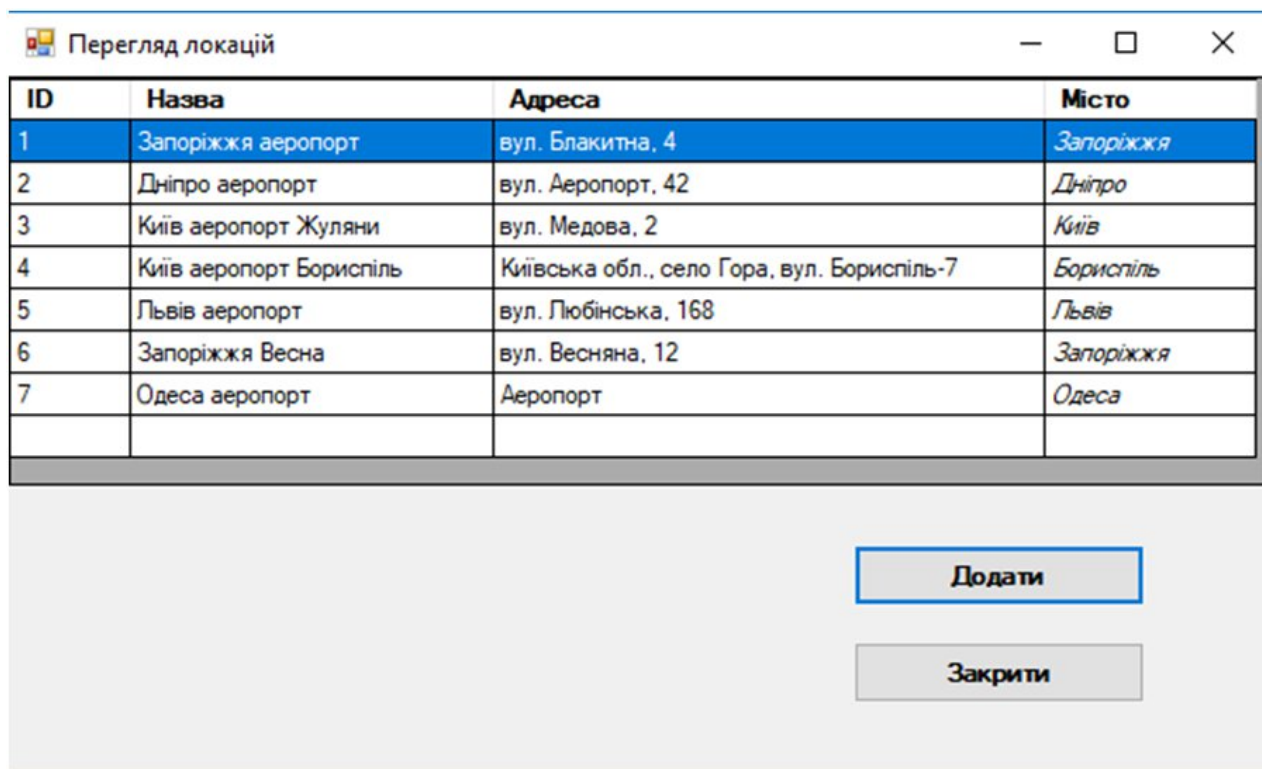


Рисунок 4.5 – Робота з програмною системою у режимі адміністратора

У межах розділу, призначеного для роботи з базовими даними, передбачено доступ до відомостей про транспортні засоби та місця їх розташування. За допомогою відповідних елементів керування менеджер може переглядати наявні записи, вносити зміни до характеристик автомобілів, уточнювати інформацію про локації або коригувати умови нарахування плати за користування (рис. 4.6).



ID	Назва	Адреса	Місто
1	Запоріжжя аеропорт	вул. Блакитна, 4	Запоріжжя
2	Дніпро аеропорт	вул. Аеропорт, 42	Дніпро
3	Київ аеропорт Жуляни	вул. Медова, 2	Київ
4	Київ аеропорт Бориспіль	Київська обл., село Гора, вул. Бориспіль-7	Бориспіль
5	Львів аеропорт	вул. Любінська, 168	Львів
6	Запоріжжя Весна	вул. Весняна, 12	Запоріжжя
7	Одеса аеропорт	Аеропорт	Одеса

Додати

Закрити

Рисунок 4.6 – Редагування списку локацій

Окремий блок інтерфейсу орієнтований на роботу з орендними операціями та дозволяє створювати нові записи або вносити зміни до вже існуючих. Процедура формування замовлення для менеджера реалізована у вигляді вікна (рис. 4.7), подібного до того, що використовується звичайним користувачем, однак доповненого розширеними функціями контролю. Уповноважений працівник має можливість визначати поточний етап виконання замовлення, фіксувати його підтвердження, скасування або завершення, а також у разі необхідності вказувати фактичну суму оплати. Система підтримує декілька логічних станів оренди, що відображають шлях замовлення від моменту створення до остаточного закриття, включаючи етапи перевірки та відстеження повернення автомобіля.

Редагування бронювання автомобіля для оренди

Клієнт

Прізвище та ім'я **Знайти** ID

Період часу

Дата початку Кількість днів

Вибрати автомобіль

Пошук за моделлю

Виробник Модель **Знайти**

Локація **Обрати**

Обрати модель

Локація

Виробник Модель **Обрати**

Дистанція (км) Розрахована вартість

Фактична вартість

Номер: 24 Статус

заповнено
заповнено
підтверджено
скасовано
в оренді
відслідковується
завершено

Зберегти

Рисунок 4.7 – Робота адміністратора з замовленням автомобіля

Робота з інформацією про клієнтів здійснюється через окремі засоби інтерфейсу, які дають змогу як редагувати наявні відомості, так і створювати нові записи (рис. 4.8). Під час вибору режиму редагування відкривається

форма для пошуку та подальшого коригування даних конкретної особи. У випадку додавання нового клієнта менеджеру надається можливість внести необхідні відомості без попередньої наявності облікового запису в системі. Такий підхід є доцільним у ситуаціях, коли замовлення оформлюється безпосередньо в пункті обслуговування, а клієнт не здійснював самостійної реєстрації. Реалізований механізм забезпечує гнучкість у роботі персоналу та дозволяє охопити різні сценарії взаємодії з відвідувачами служби прокату.

Рисунок 4.8 – Додавання інформації про клієнта

4.5 Тестування системи опрацювання даних для адміністрування послуг прокату транспортних засобів

Проведено тестування системи опрацювання даних для адміністрування послуг прокату транспортних засобів.

Підтверджено, що системи опрацювання даних для адміністрування послуг прокату транспортних засобів працює коректно і стабільно, виконуючи всі необхідні функціональні вимоги та успішно справляючись із основною задачею підтримки адміністрування послуг прокату транспортних засобів.

Розроблена система забезпечує організацію та опрацювання даних, пов'язаних із підтримкою процесів адміністрування послуг прокату транспортних засобів.

4.6 Висновки за розділом 4

Описано систему опрацювання даних для адміністрування послуг прокату транспортних засобів.

Проведено тестування створеної системи опрацювання даних для адміністрування послуг прокату транспортних засобів. Результати тестування показали, що система забезпечує організацію та опрацювання даних, пов'язаних із підтримкою процесів адміністрування послуг прокату транспортних засобів.

ВИСНОВКИ

В ході виконання дипломної кваліфікаційної роботи бакалавра було проаналізовано та досліджено процеси обчислень, пов'язані з організацією та опрацюванням даних для адміністрування послуг прокату транспортних засобів.

Визначено, що система опрацювання даних для адміністрування послуг прокату транспортних засобів являє собою комплексне програмне рішення, призначене для автоматизованого управління всіма інформаційними процесами, що супроводжують діяльність підприємств з надання автомобілів у тимчасове користування. Така програмна система надає інтегроване середовище, у межах якого здійснюється накопичення, збереження, оброблення та аналіз відомостей про транспортний парк, клієнтів, договори оренди, фінансові операції та супутні сервісні процедури. Її функціонування ґрунтується на використанні сучасних інформаційних технологій, що забезпечують централізований доступ до даних і підтримують прийняття управлінських рішень.

За результатами проведеного аналізу зроблено висновок, що у наш час існує досить багато систем опрацювання даних для адміністрування послуг прокату транспортних засобів. Проте деякі системи опрацювання даних для адміністрування послуг прокату транспортних засобів характеризуються високою складністю адаптації до реальних бізнес-процесів конкретного підприємства, оскільки універсальні програмні рішення не завжди враховують специфіку різних видів прокату, відмінності у моделях обслуговування клієнтів та особливості правового регулювання. Недоліком є і підвищені вимоги до інформаційної безпеки, оскільки системи опрацювання даних у сфері прокату працюють з персональною та фінансовою інформацією клієнтів. Недостатній рівень захисту або помилки в налаштуванні доступу можуть призвести до витоку даних, що створює як репутаційні, так і правові ризики. Забезпечення належного рівня безпеки потребує додаткових ресурсів,

регулярних оновлень та постійного контролю, що не завжди є економічно виправданим для невеликих компаній. Крім того, впровадження таких систем часто супроводжується організаційними труднощами, пов'язаними зі зміною усталених процесів роботи. Тому актуальною є розробка системи опрацювання даних для адміністрування послуг прокату транспортних засобів.

Сформульовано функціональні вимоги до системи опрацювання даних для адміністрування послуг прокату транспортних засобів.

Для реалізації системи опрацювання даних для адміністрування послуг прокату транспортних засобів обрано мову програмування C#, яка дає змогу створювати програмні рішення, що функціонують у різних операційних середовищах без суттєвих змін коду, що розширює можливості подальшого використання та масштабування розроблених програмних систем. Крім того, у поєднанні з підтримкою модульності та повторного використання компонентів мова C# забезпечує основу для побудови гнучких і розширюваних архітектур, що відповідають вимогам сучасних інформаційних систем і сприяють ефективному розвитку програмних продуктів у довгостроковій перспективі.

Для створення системи опрацювання даних для адміністрування послуг прокату транспортних засобів обрано середовище розробки Visual Studio, що обумовлено його широкими функціональними можливостями, інтеграцією з мовою програмування C# та підтримкою сучасних підходів до розробки корпоративних інформаційних систем. Visual Studio забезпечує повноцінне середовище, в якому поєднуються редактор коду, засоби налагодження, тестування та управління проєктами, що дозволяє розробникам виконувати всі етапи створення програмного продукту в одному інтерфейсі без необхідності використання додаткових інструментів.

Для зберігання та опрацювання даних обрано систему управління базами даних SQL Server завдяки поєднанню високої надійності, продуктивності та розширених можливостей обробки даних, які є критично важливими для систем адміністрування послуг прокату транспортних засобів.

SQL Server забезпечує ефективну роботу з великими обсягами структурованої інформації, підтримує транзакційність і цілісність даних, а також тісно інтегрується з мовою програмування C# і середовищем розробки Visual Studio, що спрощує реалізацію бізнес-логіки та супровід системи. Масштабованість, розвинені механізми безпеки та зручні засоби адміністрування роблять SQL Server оптимальним вибором для створення стабільного, керованого й перспективного програмного продукту.

Розроблено систему опрацювання даних для адміністрування послуг прокату транспортних засобів.

Запропоновано структуру системи опрацювання даних для адміністрування послуг прокату транспортних засобів, яка сформована з урахуванням необхідності чіткого розмежування логічних рівнів програмного забезпечення та забезпечення можливості його подальшого розвитку. У процесі проєктування було прийнято рішення відокремити механізми роботи з інформаційними ресурсами від компонентів, що відповідають за взаємодію з користувачем, оскільки такий підхід сприяє підвищенню гнучкості архітектури та спрощує внесення змін без суттєвого впливу на інші частини програми. У сукупності така організація програмних компонентів формує цілісну архітектуру, у якій логіка опрацювання даних, механізми доступу до бази та інтерфейсні елементи функціонують узгоджено, що створює передумови для стабільної експлуатації системи та її подальшої модернізації відповідно до потреб адміністрування послуг прокату транспортних засобів.

Описано функціонування системи опрацювання даних для адміністрування послуг прокату транспортних засобів. Функціонування програми характеризується чіткою орієнтацією на подієву модель взаємодії, у якій користувач виступає ініціатором процесів, а система забезпечує їх коректне та послідовне виконання. Реалізований механізм роботи дозволяє підтримувати повний цикл адміністрування послуг прокату транспортних засобів, забезпечуючи актуальність даних, зручність користування та надійність обробки інформації

Розроблено базу даних системи опрацювання даних для адміністрування послуг прокату транспортних засобів. У структурі бази даних передбачено зберігання відомостей про облікові записи користувачів системи, дані про клієнтів, характеристики моделей автомобілів, інформацію про конкретні транспортні засоби, записи про здійснені або заплановані орендні операції, відомості про діючі тарифи та про місця видачі й повернення автомобілів. Розмежування інформації про користувачів та клієнтів реалізовано шляхом виділення окремих логічних сутностей, що дозволяє розділити дані, необхідні для автентифікації в системі, та відомості, які використовуються під час оформлення договорів оренди. Такий підхід забезпечує гнучкість у роботі з різними категоріями осіб, у тому числі з тими, що не мають власного облікового запису.

Виконано проєктування інтерфейсу взаємодії користувача з системою опрацювання даних для адміністрування послуг прокату транспортних засобів.

Проведено тестування створеної системи опрацювання даних для адміністрування послуг прокату транспортних засобів. Результати тестування показали, що система забезпечує організацію та опрацювання даних, пов'язаних із підтримкою процесів адміністрування послуг прокату транспортних засобів.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Car Rental Management Software: Top Key Features, Business Benefits, and Top Solutions. URL: <https://www.mobilityinfotech.com/blog/car-rental-management-software-guide> (date of access: 30.04.2026).
2. What Is Car Rental Management Software and How Does It Work. URL: <https://traveltekpro.com/what-is-car-rental-management-software-and-how-does-it-work> (date of access: 30.04.2026).
3. Car Rental Software Features. URL: <https://www.carrentalsolutions.com/features.html> (date of access: 30.04.2026).
4. Car Rental Software. URL: <https://www.travelopro.com/car-rental-management-software.php> (date of access: 30.04.2026).
5. Car Rental Software. URL: https://www.reddit.com/user/shrugir/comments/1exkyvg/car_rental_software (date of access: 30.04.2026).
6. Best Car Rental Software Solutions: Key Features to Look For. URL: <https://financesonline.com/car-rental/> (date of access: 30.04.2026).
7. The Best Free and Open Source Car Rental Software Solutions. URL: <https://www.goodfirms.co/car-rental-software/blog/best-free-open-source-car-rental-software-solutions> (date of access: 30.04.2026).
8. Best Car Rental Software. URL: <https://www.capterra.com/car-rental-software/> (date of access: 30.04.2026).
9. Car Rental Software. URL: <https://www.trustradius.com/categories/car-rental> (date of access: 30.04.2026).
10. Best Car Rental Software. URL: <https://www.g2.com/categories/car-rental> (date of access: 30.04.2026).
11. AFG Rent A Car. URL: <https://www.afgsoftware.com/english/rentacar/> (date of access: 30.04.2026).
12. ScheduleBull. URL: <https://schedulebull.com/en/car-rental-software> (date of access: 30.04.2026).

13. Flit2GO Digitize your car rental company. URL: <https://flit2go.com/en/descubre-flit2go/> (date of access: 30.04.2026).
14. C# Tutorial. URL: <https://www.w3schools.com/cs/index.php> (date of access: 30.04.2026).
15. C# Win Forms. URL: https://www.dotnetperls.com/category_c-winforms (date of access: 30.04.2026).
16. Visual Studio product family documentation. URL: <https://learn.microsoft.com/uk-ua/visualstudio/?view=visualstudio> (date of access: 30.04.2026).
17. Visual Studio. URL: <https://www.reddit.com/r/VisualStudio/> (date of access: 30.04.2026).
18. SQL Server. URL: <https://www.sqlservercentral.com/articles/sql-server-2022-build-list> (date of access: 30.04.2026).
19. SQL Server technical documentation. URL: <https://learn.microsoft.com/uk-ua/sql/sql-server/?view=sql-server-ver17> (date of access: 30.04.2026).

ДОДАТОК А
Технічне завдання

Вступ

Програмна система опрацювання даних може використовуватися для адміністрування послуг прокату транспортних засобів.

A.1 Підстава для розробки

Підставою для розробки є завдання на дипломну кваліфікаційну роботу на тему «Створення програмної системи опрацювання даних для адміністрування послуг прокату транспортних засобів», затверджене наказом Національного університету «Запорізька політехніка» № 139 від 7 квітня 2026 р.

A.2 Призначення розробки

Система опрацювання даних призначена для забезпечення адміністрування послуг прокату транспортних засобів.

A.3 Основні вимоги до програми, що розробляється

A.3.1 Вимоги до функціональних характеристик

Система опрацювання даних для адміністрування послуг прокату транспортних засобів забезпечує виконання таких функцій:

- підтримка можливості збереження у базі даних відомостей про клієнтів служби прокату транспортних засобів;
- можливість реєстрації та авторизації користувачів системи з розмежуванням прав доступу;
- підтримка можливості внесення та коригування інформації про моделі автомобілів і конкретні транспортні засоби автопарку;

- можливість фіксування операцій бронювання та фактичної оренди автомобілів із збереженням відповідних параметрів;
- можливість обліку місць розташування транспортних засобів;
- підтримка можливості формування та перегляду переліку доступних автомобілів за заданими умовами;
- можливість отримання інформаційних повідомлень про результати виконання операцій у системі.

A.3.2 Вимоги до інтерфейсу програми

Інтерфейс системи опрацювання даних для адміністрування послуг прокату транспортних засобів повинен бути зручним для користувачів. Повинна бути забезпечена можливість роботи в візуальному режимі.

A.3.3 Вимоги до надійності

Система опрацювання даних для адміністрування послуг прокату транспортних засобів повинна забезпечити надійне функціонування.

A.3.4 Умови експлуатації

Для експлуатації системи опрацювання даних для адміністрування послуг прокату транспортних засобів необхідна наявність персонального комп'ютера.

A.3.5 Вимоги до складу та параметрів технічний засобів

Для роботи системи опрацювання даних для адміністрування послуг прокату транспортних засобів потрібні такі технічні та програмні ресурси:

комп'ютер, оснащений процесором з тактовою частотою не нижче 2,0 ГГц, що дозволяє виконувати операції обробки запитів до бази даних та роботу графічного інтерфейсу без помітних затримок. Обсяг оперативної пам'яті має становити не менше 4 ГБ. Для розміщення самої програми, файлів бази даних та допоміжних компонентів необхідно щонайменше 10 ГБ вільного простору на жорсткому диску або твердотільному накопичувачі.

У програмному середовищі обов'язковою є наявність операційної системи сімейства Windows актуальної версії з підтримкою платформи .NET. Для зберігання й обробки інформації повинна бути встановлена система керування базами даних Microsoft SQL Server або її сумісна редакція, що забезпечує виконання SQL-запитів, підтримку транзакцій та механізми резервного копіювання. Для можливості супроводу й подальшого розвитку програми доцільною є наявність середовища розробки Visual Studio з необхідними компонентами .NET, а також засобами доступу до SQL Server.

Також рекомендованою є наявність стандартних засобів введення та виведення інформації, зокрема монітора для коректного відображення форм інтерфейсу, клавіатури та маніпулятора типу миша.

A.3.6 Вимоги до маркування і пакування

Система опрацювання даних для адміністрування послуг прокату транспортних засобів може бути записана на будь-якому носії інформації.

На пакуванні повинна бути назва – «Система опрацювання даних для адміністрування послуг прокату транспортних засобів».

A.4 Вхідні дані до роботи

Вхідними даними до системи опрацювання даних для адміністрування послуг прокату транспортних засобів є інформація про клієнтів, параметри

транспортних засобів, умови та терміни оренди, дані щодо тарифів і місць отримання або повернення автомобілів.

Вихідними даними системи опрацювання даних для адміністрування послуг прокату транспортних засобів є інформація, що надається користувачеві у відповідь на виконані запити та операції, зокрема відомості про доступні автомобілі, перелік оформлених бронювань, дані про клієнтів і фінансові нарахування.

ДОДАТОК Б

Фрагмент тексту програми

```

namespace PrjNms
{
    public class Prgm
    {
        public static void Main(string[] args)
        {
            var bld = WebApplication.CreateBuilder(args);
            bld.Services.AddSingleton<ClntSrv>();
            bld.Services.AddSingleton<CrSrv>();
            bld.Services.AddSingleton<BrnSrv>();
            bld.Services.AddSingleton<TrffSrv>();
            bld.Services.AddSingleton<LctnSrv>();
            var app = bld.Build();

            app.MapGet("/clnt/lst", (ClntSrv srv) =>
            {
                return srv.GtLst();
            });

            app.MapPost("/clnt/add", (ClntSrv srv, Clnt m) =>
            {
                srv.Add(m);
                return Results.Ok();
            });

            app.MapPost("/clnt/upd", (ClntSrv srv, Clnt m) =>
            {
                srv.Upd(m);
                return Results.Ok();
            });

            app.MapGet("/cr/lst", (CrSrv srv) =>
            {
                return srv.GtLst();
            });

            app.MapPost("/cr/add", (CrSrv srv, Cr m) =>
            {
                srv.Add(m);
                return Results.Ok();
            });

            app.MapPost("/cr/upd", (CrSrv srv, Cr m) =>
            {
                srv.Upd(m);
                return Results.Ok();
            });
        }
    }
}

```

```

    app.MapGet("/brn/lst", (BrnSrv srv) =>
    {
        return srv.GtLst();
    });

    app.MapPost("/brn/add", (BrnSrv srv, Brn m) =>
    {
        srv.Add(m);
        return Results.Ok();
    });

    app.MapPost("/brn/upd", (BrnSrv srv, Brn m) =>
    {
        srv.Upd(m);
        return Results.Ok();
    });

    app.Run();
}

public class Clnt
{
    public string nm { get; set; }
    public string phn { get; set; }
    public string ml { get; set; }
    public string dsc { get; set; }
}

public class Cr
{
    public string cd { get; set; }
    public string mdl { get; set; }
    public string stt { get; set; }
    public string lct { get; set; }
}

public class Brn
{
    public string clnt { get; set; }
    public string cr { get; set; }
    public string trff { get; set; }
    public string lct { get; set; }
}

public class Trff
{

```

```

        public string mdl { get; set; }
        public string prc { get; set; }
        public string dsc { get; set; }
    }

    public class Lctn
    {
        public string nm { get; set; }
        public string dsc { get; set; }
    }

    public class ClntSrv
    {
        private List<Clnt> lst = new List<Clnt>();

        public List<Clnt> GtLst()
        {
            return lst;
        }

        public void Add(Clnt m)
        {
            lst.Add(m);
        }

        public void Upd(Clnt m)
        {
            for (int i = 0; i < lst.Count; i++)
            {
                if (lst[i].nm == m.nm)
                {
                    lst[i] = m;
                }
            }
        }

        public void Rmv(string nm)
        {
            var tmp = new List<Clnt>();
            for (int i = 0; i < lst.Count; i++)
            {
                if (lst[i].nm != nm)
                {
                    tmp.Add(lst[i]);
                }
            }
            lst = tmp;
        }
    }

```

```

public Clnt Fnd(string nm)
{
    for (int i = 0; i < lst.Count; i++)
    {
        if (lst[i].nm == nm)
        {
            return lst[i];
        }
    }
    return null;
}

public void Clr()
{
    lst = new List<Clnt>();
}

public int Cnt()
{
    int c = 0;
    for (int i = 0; i < lst.Count; i++)
    {
        c++;
    }
    return c;
}

public bool Chk(string nm)
{
    for (int i = 0; i < lst.Count; i++)
    {
        if (lst[i].nm == nm)
        {
            return true;
        }
    }
    return false;
}
}

public class CrSrv
{
    private List<Cr> lst = new List<Cr>();

    public List<Cr> GtLst()
    {
        return lst;
    }
}

```

```

    }

    public void Add(Cr m)
    {
        lst.Add(m);
    }

    public void Upd(Cr m)
    {
        for (int i = 0; i < lst.Count; i++)
        {
            if (lst[i].cd == m.cd)
            {
                lst[i] = m;
            }
        }
    }

    public void Rmv(string cd)
    {
        var tmp = new List<Cr>();
        for (int i = 0; i < lst.Count; i++)
        {
            if (lst[i].cd != cd)
            {
                tmp.Add(lst[i]);
            }
        }
        lst = tmp;
    }

    public Cr Fnd(string cd)
    {
        for (int i = 0; i < lst.Count; i++)
        {
            if (lst[i].cd == cd)
            {
                return lst[i];
            }
        }
        return null;
    }

    public int Cnt()
    {
        int c = 0;
        for (int i = 0; i < lst.Count; i++)
        {

```

```

        c++;
    }
    return c;
}

public bool Chk(string cd)
{
    for (int i = 0; i < lst.Count; i++)
    {
        if (lst[i].cd == cd)
        {
            return true;
        }
    }
    return false;
}

public void Clr()
{
    lst = new List<Cr>();
}

}

public class BrnSrv
{
    private List<Brn> lst = new List<Brn>();

    public List<Brn> GtLst()
    {
        return lst;
    }

    public void Add(Brn m)
    {
        lst.Add(m);
    }

    public void Upd(Brn m)
    {
        for (int i = 0; i < lst.Count; i++)
        {
            if (lst[i].clnt == m.clnt && lst[i].cr == m.cr)
            {
                lst[i] = m;
            }
        }
    }
}

```

```

public void Rmv(string cl, string cr)
{
    var tmp = new List<Brn>();
    for (int i = 0; i < lst.Count; i++)
    {
        if (!(lst[i].clnt == cl && lst[i].cr == cr))
        {
            tmp.Add(lst[i]);
        }
    }
    lst = tmp;
}

```

```

public Brn Fnd(string cl, string cr)
{
    for (int i = 0; i < lst.Count; i++)
    {
        if (lst[i].clnt == cl && lst[i].cr == cr)
        {
            return lst[i];
        }
    }
    return null;
}

```

```

public int Cnt()
{
    int c = 0;
    for (int i = 0; i < lst.Count; i++)
    {
        c++;
    }
    return c;
}

```

```

public bool Chk(string cl, string cr)
{
    for (int i = 0; i < lst.Count; i++)
    {
        if (lst[i].clnt == cl && lst[i].cr == cr)
        {
            return true;
        }
    }
    return false;
}

```

```

public void Clr()

```

```

        {
            lst = new List<Brn>();
        }
    }

public class TrffSrv
{
    private List<Trff> lst = new List<Trff>();

    public List<Trff> GtLst()
    {
        return lst;
    }

    public void Add(Trff m)
    {
        lst.Add(m);
    }

    public void Upd(Trff m)
    {
        for (int i = 0; i < lst.Count; i++)
        {
            if (lst[i].mdl == m.mdl)
            {
                lst[i] = m;
            }
        }
    }

    public Trff End(string mdl)
    {
        for (int i = 0; i < lst.Count; i++)
        {
            if (lst[i].mdl == mdl)
            {
                return lst[i];
            }
        }
        return null;
    }

    public void Rmv(string mdl)
    {
        var tmp = new List<Trff>();
        for (int i = 0; i < lst.Count; i++)
        {
            if (lst[i].mdl != mdl)

```

```

        {
            tmp.Add(lst[i]);
        }
    }
    lst = tmp;
}

public int Cnt()
{
    int c = 0;
    for (int i = 0; i < lst.Count; i++)
    {
        c++;
    }
    return c;
}

public void Clr()
{
    lst = new List<Trff>();
}

}

public class LctnSrv
{
    private List<Lctn> lst = new List<Lctn>();

    public List<Lctn> GtLst()
    {
        return lst;
    }

    public void Add(Lctn m)
    {
        lst.Add(m);
    }

    public void Upd(Lctn m)
    {
        for (int i = 0; i < lst.Count; i++)
        {
            if (lst[i].nm == m.nm)
            {
                lst[i] = m;
            }
        }
    }
}

```

```

public Lctn Fnd(string nm)
{
    for (int i = 0; i < lst.Count; i++)
    {
        if (lst[i].nm == nm)
        {
            return lst[i];
        }
    }
    return null;
}

public void Rmv(string nm)
{
    var tmp = new List<Lctn>();
    for (int i = 0; i < lst.Count; i++)
    {
        if (lst[i].nm != nm)
        {
            tmp.Add(lst[i]);
        }
    }
    lst = tmp;
}

public int Cnt()
{
    int c = 0;
    for (int i = 0; i < lst.Count; i++)
    {
        c++;
    }
    return c;
}

public void Clr()
{
    lst = new List<Lctn>();
}

}

public class RptMdl
{
    public string ttl { get; set; }
    public string bdy { get; set; }
}

public class RptSrv

```

```

{
    private List<RptMdl> lst = new List<RptMdl>();

    public void Add(RptMdl m)
    {
        lst.Add(m);
    }

    public List<RptMdl> GtLst()
    {
        return lst;
    }

    public void Clr()
    {
        lst = new List<RptMdl>();
    }

    public int Cnt()
    {
        int c = 0;
        for (int i = 0; i < lst.Count; i++)
        {
            c++;
        }
        return c;
    }

    public RptMdl Fnd(string ttl)
    {
        for (int i = 0; i < lst.Count; i++)
        {
            if (lst[i].ttl == ttl)
            {
                return lst[i];
            }
        }
        return null;
    }
}

public class MthExt
{
    public string Cmb(string a, string b)
    {
        string r = "";
        r = a + b;
        return r;
    }
}

```

```

}

public string Rpt(string s, int n)
{
    string r = "";
    for (int i = 0; i < n; i++)
    {
        r = r + s;
    }
    return r;
}

public bool Emp(string s)
{
    if (s == null)
    {
        return true;
    }
    if (s == "")
    {
        return true;
    }
    return false;
}

public int Lng(string s)
{
    int c = 0;
    for (int i = 0; i < s.Length; i++)
    {
        c++;
    }
    return c;
}

public string Sub(string s, int p, int l)
{
    string r = "";
    int c = 0;
    for (int i = p; i < s.Length; i++)
    {
        if (c < l)
        {
            r = r + s[i];
        }
        c++;
    }
    return r;
}

```

```

        }
    }

public List<Cr> FltCrByStt(string stt)
{
    List<Cr> rsl = new List<Cr>();
    for (int i = 0; i < lst.Count; i++)
    {
        if (lst[i].stt == stt)
        {
            rsl.Add(lst[i]);
        }
    }
    return rsl;
}

public List<Brn> FltBrnByClnt(string cl)
{
    List<Brn> rsl = new List<Brn>();
    for (int i = 0; i < lst.Count; i++)
    {
        if (lst[i].clnt == cl)
        {
            rsl.Add(lst[i]);
        }
    }
    return rsl;
}

public bool ChngCrStt(string cd, string stt)
{
    for (int i = 0; i < lst.Count; i++)
    {
        if (lst[i].cd == cd)
        {
            lst[i].stt = stt;
            return true;
        }
    }
    return false;
}

public int ClcBrnCntByCr(string cd)
{
    int c = 0;
    for (int i = 0; i < lst.Count; i++)
    {
        if (lst[i].cr == cd)
        {

```

```

        c++;
    }
}
return c;
}

public List<Clnt> SrchClntByPhn(string ph)
{
    List<Clnt> rsl = new List<Clnt>();
    for (int i = 0; i < lst.Count; i++)
    {
        if (lst[i].phn == ph)
        {
            rsl.Add(lst[i]);
        }
    }
    return rsl;
}

public bool ChngTrffForBrn(string cl, string cr, string tr)
{
    for (int i = 0; i < lst.Count; i++)
    {
        if (lst[i].clnt == cl && lst[i].cr == cr)
        {
            lst[i].trff = tr;
            return true;
        }
    }
    return false;
}

public List<Cr> FltCrByLct(string lc)
{
    List<Cr> rsl = new List<Cr>();
    for (int i = 0; i < lst.Count; i++)
    {
        if (lst[i].lct == lc)
        {
            rsl.Add(lst[i]);
        }
    }
    return rsl;
}

public bool MvCrToLct(string cd, string lc)
{
    for (int i = 0; i < lst.Count; i++)
    {

```

```

        if (lst[i].cd == cd)
        {
            lst[i].lct = lc;
            return true;
        }
    }
    return false;
}

public string GtClntMl(string nm)
{
    for (int i = 0; i < lst.Count; i++)
    {
        if (lst[i].nm == nm)
        {
            return lst[i].ml;
        }
    }
    return "";
}

public bool UpdClntPhn(string nm, string ph)
{
    for (int i = 0; i < lst.Count; i++)
    {
        if (lst[i].nm == nm)
        {
            lst[i].phn = ph;
            return true;
        }
    }
    return false;
}

public List<Brn> FltBrnByLct(string lc)
{
    List<Brn> rsl = new List<Brn>();
    for (int i = 0; i < lst.Count; i++)
    {
        if (lst[i].lct == lc)
        {
            rsl.Add(lst[i]);
        }
    }
    return rsl;
}

public int ClcClntCnt()
{

```

```

        int c = 0;
        for (int i = 0; i < lst.Count; i++)
        {
            c = c + 1;
        }
        return c;
    }

    public bool RstCrStt(string cd)
    {
        for (int i = 0; i < lst.Count; i++)
        {
            if (lst[i].cd == cd)
            {
                lst[i].stt = "fr";
                return true;
            }
        }
        return false;
    }

    public List<Lctn> SrchLctByNm(string nm)
    {
        List<Lctn> rsl = new List<Lctn>();
        for (int i = 0; i < lst.Count; i++)
        {
            if (lst[i].nm == nm)
            {
                rsl.Add(lst[i]);
            }
        }
        return rsl;
    }

    public bool UpdLctDsc(string nm, string ds)
    {
        for (int i = 0; i < lst.Count; i++)
        {
            if (lst[i].nm == nm)
            {
                lst[i].dsc = ds;
                return true;
            }
        }
        return false;
    }

    public string BldBrnTxt(Brn b)
    {

```

```

        string t = "";
        t = t + b.clnt;
        t = t + "_";
        t = t + b.cr;
        t = t + "_";
        t = t + b.trff;
        return t;
    }

    public bool ChkCrFr(string cd)
    {
        for (int i = 0; i < lst.Count; i++)
        {
            if (lst[i].cd == cd && lst[i].stt == "fr")
            {
                return true;
            }
        }
        return false;
    }

    public int ClcCrCntByMdl(string mdl)
    {
        int c = 0;
        for (int i = 0; i < lst.Count; i++)
        {
            if (lst[i].mdl == mdl)
            {
                c++;
            }
        }
        return c;
    }

    public void MrgClntLst(List<Clnt> nw)
    {
        for (int i = 0; i < nw.Count; i++)
        {
            lst.Add(nw[i]);
        }
    }

    public void MrgCrLst(List<Cr> nw)
    {
        for (int i = 0; i < nw.Count; i++)
        {
            lst.Add(nw[i]);
        }
    }

```

```

public bool ChkLctEx(string nm)
{
    for (int i = 0; i < lst.Count; i++)
    {
        if (lst[i].nm == nm)
        {
            return true;
        }
    }
    return false;
}

```

```

public string GtCrLct(string cd)
{
    for (int i = 0; i < lst.Count; i++)
    {
        if (lst[i].cd == cd)
        {
            return lst[i].lct;
        }
    }
    return "";
}

```

```

public bool ClrBrnByClnt(string cl)
{
    bool f = false;
    List<Brn> tmp = new List<Brn>();
    for (int i = 0; i < lst.Count; i++)
    {
        if (lst[i].clnt != cl)
        {
            tmp.Add(lst[i]);
        }
        else
        {
            f = true;
        }
    }
    lst = tmp;
    return f;
}

```

```

public bool ClrCrByLct(string lc)
{
    bool f = false;
    List<Cr> tmp = new List<Cr>();
    for (int i = 0; i < lst.Count; i++)

```

```
{
    if (lst[i].lct != lc)
    {
        tmp.Add(lst[i]);
    }
    else
    {
        f = true;
    }
}
lst = tmp;
return f;
}
```

ДОДАТОК В
Слайди презентації

**Міністерство освіти і науки України
Національний університет «Запорізька політехніка»
Кафедра програмних засобів**

Створення програмної системи опрацювання даних для адміністрування послуг прокату транспортних засобів

Виконав: Владислав ДЕЙНЕГА
ст. групи КНТ-212

Керівник: Степан СКРУПСЬКИЙ
к.т.н., доцент

Рисунок В.1 – Слайд 1

Об'єкт, предметта мета роботи

Об'єкт дослідження – процеси обчислень, пов'язані з організацією та опрацюванням даних для адміністрування послуг прокату транспортних засобів.

Предмет дослідження – програмні системи опрацювання даних для адміністрування послуг прокату транспортних засобів.

Метою роботи є розробка програмної системи опрацювання даних для адміністрування послуг прокату транспортних засобів для скорочення часу та зменшення трудомісткості оброблення замовлень на прокат автомобілів.

Рисунок В.2 – Слайд 2

Завдання роботи

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

- виконати аналіз предметної області та систем опрацювання даних для адміністрування послуг прокату транспортних засобів;
- здійснити проектування системи опрацювання даних для адміністрування послуг прокату транспортних засобів;
- створити систему опрацювання даних для адміністрування послуг прокату транспортних засобів;
- дослідити розроблену систему опрацювання даних для адміністрування послуг прокату транспортних засобів.

3

Рисунок В.3 – Слайд 3

Порівняння існуючих аналогів

Критерій порівняння	AFG Rent A Car	ScheduleBull	Flit2GO
Підтримка багатьох мов	+	+	+-
Мультивалютність	+	+-	+-
Інтеграція із зовнішніми API	+-	+	+
Розширені звіти та аналітика	+	+	+-
Підтримка кількох сфер оренди	+-	+	+
Простота освоєння для малих компаній	+-	+-	+
Гнучкі формати користування	-	+-	+

4

Рисунок В.4 – Слайд 4

Постановка завдання

При розробці системи опрацювання даних для адміністрування послуг прокату транспортних засобів необхідно забезпечити такі функціональні вимоги:

- підтримка можливості збереження у базі даних відомостей про клієнтів служби прокату транспортних засобів;
- можливість реєстрації та авторизації користувачів системи з розмежуванням прав доступу;
- підтримка можливості внесення та коригування інформації про моделі автомобілів і конкретні транспортні засоби автопарку;
- можливість фіксування операцій бронювання та фактичної оренди автомобілів із збереженням відповідних параметрів;
- можливість обліку місць розташування транспортних засобів;
- підтримка можливості формування та перегляду переліку доступних автомобілів за заданими умовами;
- можливість отримання інформаційних повідомлень про результати виконання операцій у системі.

5

Рисунок В.5 – Слайд 5

Порівняння мов програмування

Критерій порівняння мов програмування	Мова програмування		
	C#	Java	PHP
Масштабованість системи	+	+	–
Асинхронність та паралельне програмування	+	+–	–
Легкість супроводу та модифікації коду	+	+–	+–
Безпека виконання та керування пам'яттю	+	+	–
Контроль помилок	+	+	–

6

Рисунок В.6 – Слайд 6

Порівняння середовищ розробки

Критерій порівняння середовищ розробки	Середовища розробки		
	Visual Studio	Rider	MonoDevelop
Зручність налагодження ПЗ	+	+-	+-
Робота з багаторівневою архітектурою	+	+-	+-
Профілювання продуктивності	+	+-	-
Зручність інструментів для тестування	+	+-	-
Інтеграція з системами контролю версій	+	+	+-

7

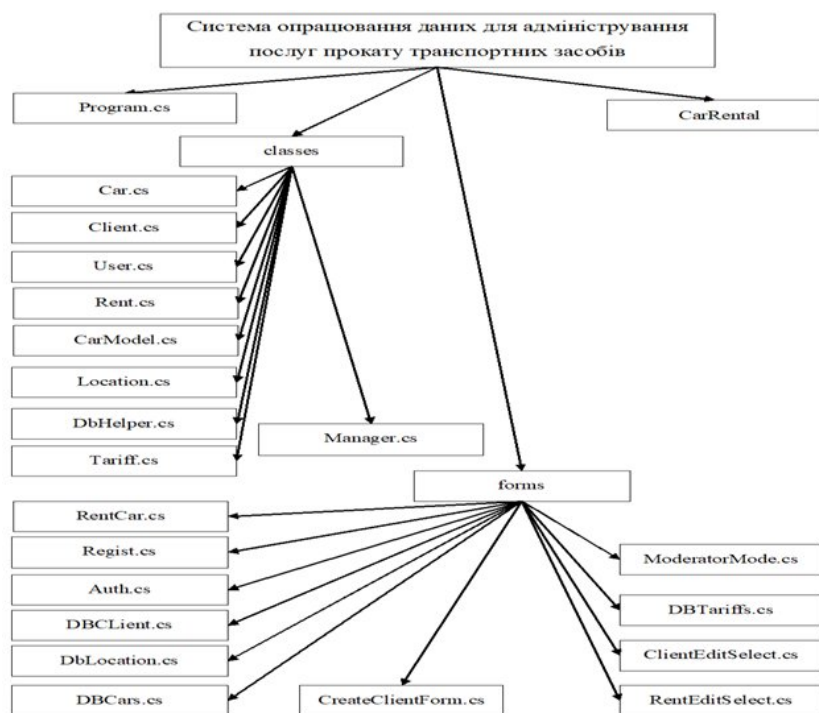
Рисунок В.7 – Слайд 7

Обґрунтування вибору засобів зберігання та опрацювання даних

Критерій порівняння систем управління базами даних	Системи управління базами даних		
	SQL Server	MySQL	PostgreSQL
Надійність та стійкість до збоїв	+	+-	+
Інтеграція з C# та Visual Studio	+	+-	+-
Безпека та контроль доступу	+	+-	+
Легкість адміністрування та моніторингу	+	+-	+-
Підтримка складних запитів і процедур	+	+-	+

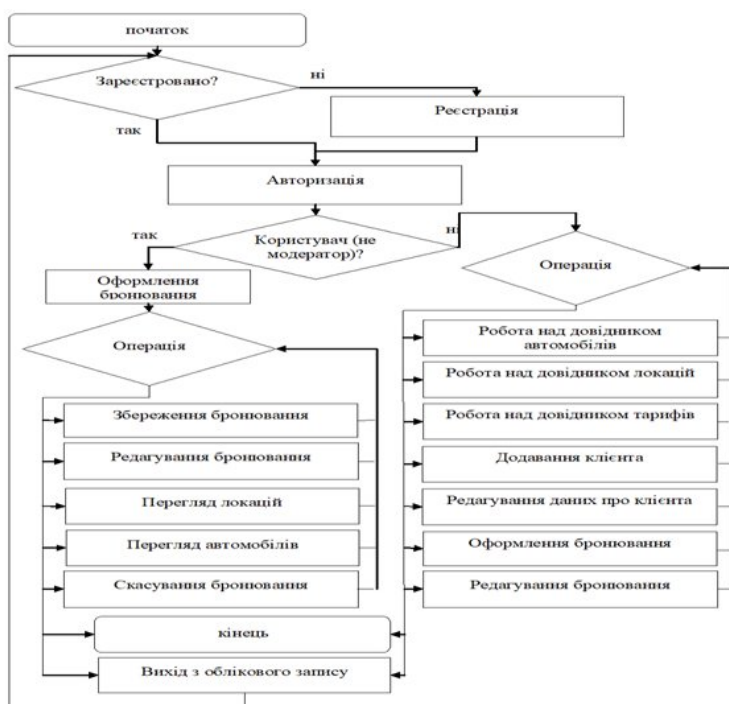
8

Рисунок В.8 – Слайд 8



Структура програми

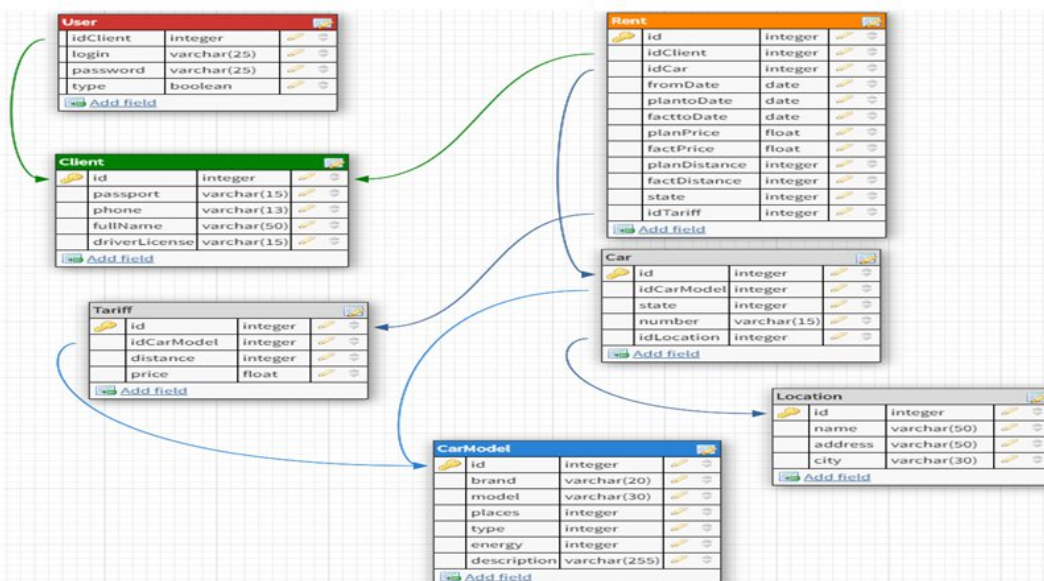
Рисунок В.9 – Слайд 9



Функціонування програми

Рисунок В.10 – Слайд 10

Схема бази даних



11

Рисунок В.11 – Слайд 11

Робота з програмою

The screenshot displays the 'Бронювання автомобіля для оренди' (Car rental booking) application. The main window includes the following sections:

- Період часу** (Time period): Fields for 'Дата початку' (Start date) set to 01.05.2026 and 'Кількість днів' (Number of days) set to 1.
- Вибрати автомобіль** (Select car): A search section with 'Виробник' (Manufacturer) and 'Модель' (Model) input fields, a 'Знайти' (Find) button, and a 'Локація' (Location) dropdown menu with an 'Обрати' (Select) button.
- Обрати модель** (Select model): A section with 'Локація' (Location) and 'Виробник' (Manufacturer) dropdown menus, a 'Модель' (Model) dropdown menu, and an 'Обрати' (Select) button.
- Бронювання** (Booking): A 'Номер' (Number) dropdown menu, 'Закінчити' (Finish), and 'Скасувати' (Cancel) buttons.
- Перегляд** (View): 'Автомобілі' (Cars) and 'Локації' (Locations) buttons.
- Summary**: 'Дистанція (км)' (Distance in km) field, 'Розрахована вартість' (Calculated cost) set to 0.00, and a 'Забронювати' (Book) button.

Two side panels are visible:

- Авторизація** (Authorization): Fields for 'Логін' (Login) and 'Пароль' (Password), with 'Підтвердити' (Confirm) and 'Реєстрація' (Registration) buttons.
- Реєстрація користувача** (User registration): Fields for 'Логін' (Login) with value 'Vlad', 'Пароль' (Password), 'Підтвердіть пароль' (Confirm password), and 'Ім'я' (Name) with value 'Влад', with a 'Створити' (Create) button.

12

Рисунок В.12 – Слайд 12

Робота адміністратора з програмою

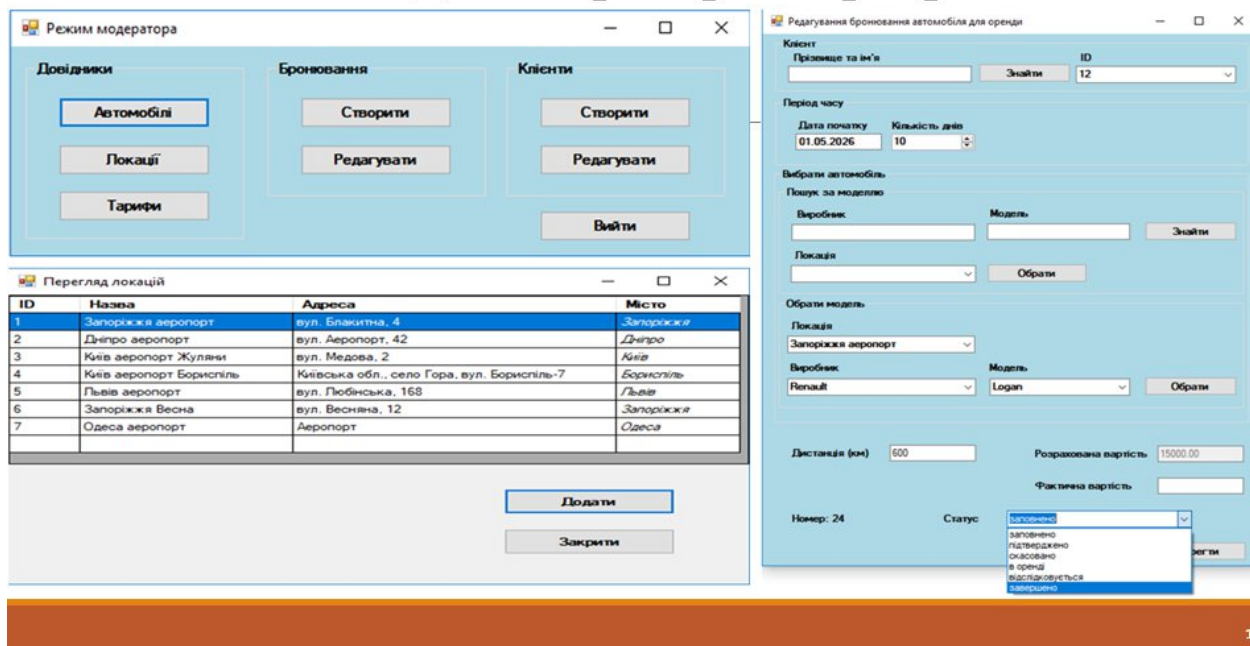


Рисунок В.13 – Слайд 13

Висновки

В ході виконання дипломної кваліфікаційної роботи бакалавра було проаналізовано та досліджено процеси обчислень, пов'язані з організацією та опрацюванням даних для адміністрування послуг прокату транспортних засобів.

Для реалізації системи опрацювання даних для адміністрування послуг прокату транспортних засобів обрано мову програмування C#, середовище розробки Visual Studio та систему управління базами даних SQL Server.

Запропоновано структуру системи опрацювання даних для адміністрування послуг прокату транспортних засобів, яка сформована з урахуванням необхідності чіткого розмежування логічних рівнів програмного забезпечення та забезпечення можливості його подальшого розвитку. Описано функціонування системи опрацювання даних для адміністрування послуг прокату транспортних засобів. Розроблено базу даних системи опрацювання даних для адміністрування послуг прокату транспортних засобів.

В результаті виконання завдання було отримано систему, що відповідає всім вимогам визначеним на початку розробки та закріпленим в технічному завданні. Розроблена система є зручною для використання та виконує всі необхідні функції.

Усі завдання дипломної кваліфікаційної роботи бакалавра повністю виконано.

Рисунок В.14 – Слайд 14