

КВАЛІФІКАЦІЙНА РОБОТА

бакалавр
Освітній рівень


Інформаційна система автосалону
Назва теми

КВРІСТ. 200197.01.09.09 ПЗ
Шифр

Галузь знань 12 «Інформаційні технології»
Шифр, назва

Спеціальність 126 «Інформаційні системи та технології»
Шифр, назва

Освітня програма «Інформаційні системи та технології»
Назва

Виконав: студент III курсу, група ICTc-20-1  Д.О. Мальченко
Підпис Ініціали, прізвище

Керівник  К.Ю. Бобровнікова
Підпис, дата Ініціали, прізвище

Нормоконтролер  С.М. Лисенко
Підпис, дата Ініціали, прізвище

До захисту допускаю:
Зав. кафедри комп'ютерної
інженерії та інформаційних
систем

 Т.О. Говоруценко
Підпис Ініціали, прізвище

« 1 » червня 2023 р.

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ

Освітній рівень БАКАЛАВР

Галузь знань 12 ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

Спеціальність 126 ІНФОРМАЦІЙНІ СИСТЕМИ ТА ТЕХНОЛОГІЇ

Освітня програма «ІНФОРМАЦІЙНІ СИСТЕМИ ТА ТЕХНОЛОГІЇ»

ЗАТВЕРДЖУЮ

Зав. кафедри Т.О.Говорущенко

«10» 01 2023 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА

Мальченку Денису Олександровичу

Прізвище, ім'я, по батькові студента

1. Тема проекту (роботи) Інформаційна система автосалону

Керівник проекту (роботи) Бобровнікова К.Ю., доцент кафедри КІС.

Прізвище, ім'я, по батькові, науковий ступінь, вчене звання

Затверджена наказом ректора університету від 01.03.2023 р. № 10

2. Строк подання студентом проекту (роботи) на кафедру 01.06.2023 р.

3. Вихідні дані до проекту (роботи) Завдання на дипломне проектування

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

Аналіз предметної області та проектування інформаційних систем

Вибір засобів реалізації та проектування інформаційної системи автосалону

Реалізація інформаційної системи автосалону





5. Перелік графічного матеріалу (із зазначенням обов'язкових креслень) _____

Схема бази даних

Схема структурна варіантів використання

Креслення вигляду екранних форм

6. Консультанти розділів дипломного проекту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	Лисенко С.М., професор кафедри КІС		
Антиплагиат	Нічепорук А.О., доцент кафедри КІС		

7. Дата видачі завдання «__» _____ 2023 р.

КАЛЕНДАРНИЙ ПЛАН

№з/п	Назва етапів (розділів) дипломного проекту (роботи)	Термін виконання етапів проекту (роботи)	Примітка
1	Вибір напрямку дослідження та узгодження тематики кваліфікаційної роботи з керівником	11.01.2023	виконано
2	Ознайомлення з предметною областю; формулювання мети та задач дослідження; визначення об'єкта та предмета дослідження	01.02.2023	виконано
3	Робота над розділом 1 – аналіз предметної галузі та проектування інформаційних систем	01.03.2023	виконано
4	Робота над розділом 2 – Вибір засобів реалізації та проектування інформаційної системи автосалону	01.04.2023	виконано
5	Робота над розділом 3 – Реалізація інформаційної системи автосалону	30.04.2023	виконано
6	Оформлення пояснювальної записки згідно вимог	10.05.2023	виконано
7	Попередній захист ВКР	26.05.2023	виконано
8	Захист ВКР на засіданні ЕК	Червень 2023 року	

Студент


Підпис

Мальченко Д.О.

Ініціали, прізвище

Керівник проекту (роботи)


Підпис

Бобровнікова К.Ю.

Ініціали, прізвище

АНОТАЦІЯ

Тема кваліфікаційної роботи: «Інформаційна система автосалону».

Автор роботи: *Мальченко Денис Олександрович*.

Керівник роботи: *Бобровнікова Кіра Юліївна*.

Пояснювальна записка: 55 с., 28 рис., 10 табл., 3 дод., 60 джерел.

Графічна частина: 7 презентаційних слайдів

ІНФОРМАЦІЙНА СИСТЕМА, БАЗА ДАНИХ, MVC, АВТОСАЛОН,
ENTITY FRAMEWORK

Мета кваліфікаційної роботи: покращення ефективності та конкурентоспроможності автосалону шляхом оптимізації його бізнес-процесів по роботі із даними шляхом розробки інформаційної системи.

В сучасному світі, де конкуренція в галузі автопродажу стає все більш жорсткою, автосалони шукають способи покращити свою ефективність, забезпечити високу якість обслуговування клієнтів і підвищити свою конкурентоспроможність. Одним з ключових інструментів для досягнення цих цілей є впровадження інформаційних систем в автосалонах.





Інформаційна система автосалону – це комплексне рішення, яке поєднує в собі технології, програмне забезпечення та процеси, що допомагають автосалону ефективно керувати своїми операціями, зберігати та обробляти дані, автоматизувати бізнес-процеси і поліпшити якість обслуговування та надання послуг клієнтам.

Підпис студента

Дата 20.05.2023

ЗМІСТ

СКРОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ.....	4
ВСТУП.....	5
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПРОЄКТУВАННЯ ІНФОРМАЦІЙНИХ СИСТЕМ.....	7
1.1 Аналіз предметної області автосалона	7
1.2 Клієнт-серверна архітектура у процесі проєктування інформаційних систем	8
1.3 Методологічні підходи до проєктування інформаційних систем.....	13
1.4 Постановка задачі.....	23
2 ВИБІР ЗАСОБІВ РЕАЛІЗАЦІЇ ТА ПРОЄКТУВАННЯ БАЗИ ДАНИХ ІНФОРМАЦІЙНОЇ СИСТЕМИ АВТОСАЛОНУ	25
2.1 Вибір засобів реалізації для проєктування інформаційної системи автосалону.....	25
2.1.1 Патерн проєктування Model View Controller (MVC).....	25
2.1.2 Технологія Entity Framework.....	28
2.1.3 Веб-сервер IIS Express	29
2.2 Аналіз вимог до інформаційної системи автосалону.....	30
2.3 Проєктування концептуальної моделі бази даних інформаційної системи..	31
Висновки до розділу 2	42
3 РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ АВТОСАЛОНУ	43
3.1 Модель вимог та модель проєктування інформаційної системи автосалону	43
3.2 Структура інформаційної системи автосалону	47
3.3 Реалізація інформаційної системи автосалону.....	51
Висновки за розділом 3	58
ВИСНОВКИ	59
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ	61

						КвРІСТ. 200197.01.09.09 ПЗ		
Зм.	Арк.	№докум.	Підпис	Дата	Інформаційна система автосалону	Літера	Аркуш	Аркушів
Виконав		Мальченко Д.О.						
Перевір.		Бобровнікова К.Ю.					2	62
Н.контр.		Лисенко С.М.				ХНУ, ІСТс-20-1		
Затвер.		Говорущенко Т.О.						

Додаток А Копія креслення «Екранні форми».....	62
Додаток Б Копія креслення «Схема бази даних».....	63
Додаток В Копія креслення «Схема структурна варіантів використання та схема інформаційних потоків».....	64

					КВРІСТ. 200197.01.09.09 ПЗ	Арк.
Зм.	Арк.	№докум.	Підпис	Дата		3

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

БД – База даних

ІС – Інформаційна система

ІТ – Інформаційні технології

СУБД – Система управління базами даних

КМБД – Концептуальна модель бази даних

КС – Комп'ютерна система

EF – Entity Framework

MVC – Model View Controller

					КВРІСТ. 200197.01.09.09 ПЗ	Арк.
						4
Зм.	Арк.	№докум.	Підпис	Дата		

ВСТУП

На сучасному розвитку суспільства одними з найважливіших напрямів є інформаційні технології. Багато систем у тій чи іншій мірі взаємопов'язані із функціями тривалого зберігання та обробки даних. З кожним роком обсяг інформації безперервно збільшується, змушуючи витратити на свою обробку та подання все більшу кількість фінансових та трудових витрат. У зв'язку із цим дедалі більше стають необхідними сучасні автоматизовані інформаційні системи, які надають користувачам доступ до інформації у зручному та структурованому вигляді. Не осторонь серед таких систем стоїть й інформаційна система автосалону.

В сучасному світі, де конкуренція в галузі автопродажу стає все більш жорсткою, автосалони шукають способи покращити свою ефективність, забезпечити високу якість обслуговування клієнтів і підвищити свою конкурентоспроможність. Одним з ключових інструментів для досягнення цих цілей є впровадження інформаційних систем в автосалонах.

Інформаційна система автосалону – це комплексне рішення, яке поєднує в собі технології, програмне забезпечення та процеси, що допомагають автосалону ефективно керувати своїми операціями, зберігати та обробляти дані, автоматизувати бізнес-процеси і поліпшити якість обслуговування та надання послуг клієнтам.

Інформаційна система автосалону включає такі компоненти, як база даних автомобілів з повною інформацією про їхні технічні характеристики, ціни, наявність тощо, база даних клієнтів з історією їхніх покупок і персоналізованою інформацією, система управління запасами, система обліку продажів та фінансового управління, система планування і контролю графіків тест-драйвів, система роботи із постачальниками, система аналізу даних і звітності, тощо.

Завдяки цим компонентам, інформаційна система автосалону надає можливість ефективно керувати запасами автомобілів, швидко знаходити

					КВРІСТ. 200197.01.09.09 ПЗ	Арк. 5
Зм.	Арк.	№докум.	Підпис	Дата		

відповідні моделі для клієнтів, забезпечувати персоналізоване обслуговування, автоматизувати процеси продажу та фінансового обліку, а також аналізувати дані для прийняття обґрунтованих і зважених рішень для надання якісних послуг клієнтам та отримання максимальної вигоди автосалону.

Отже, можна відзначити, що розробка інформаційної системи для автосалону є досить актуальною темою, реалізація якої дозволить покращити бізнес процеси у даній області.

Метою роботи є покращення ефективності та конкурентоспроможності автосалону шляхом оптимізації його бізнес-процесів по роботі із даними шляхом розробки інформаційної системи.

Об'єктом дослідження є процес обліку інформації про автомобілі.

Предметом дослідження є інформаційна система автосалону.

					КВРІСТ. 200197.01.09.09 ПЗ	Арк.
						6
Зм.	Арк.	№докум.	Підпис	Дата		

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПРОЄКТУВАННЯ ІНФОРМАЦІЙНИХ СИСТЕМ

1.1 Аналіз предметної області автосалона

Автомобільний дилер – торгова компанія, яка є посередником між виробником транспортних засобів та бажаючими придбати автомобіль фізичними та юридичними особами. Головним видом діяльності дилера є продаж автомобілів. Велика кількість автомобільних дилерів має певний перелік марок автомобілів, які вони реалізують. У покупця зазвичай є можливість або вибрати автомобіль із присутніх у наявності і придбати його відразу, або залишити заявку на модель, що цікавить, з потрібними для нього характеристиками.

Дилер проводить передпродажну підготовку автомобіля та гарантійне обслуговування – здійснення під час гарантійного терміну планових техоглядів (діагностика, гарантійний ремонт, заміна експлуатаційних матеріалів тощо). Автомобільний салон є юридичною особою, яка має фірмовий знак, зареєстрований товарний знак, розрахунковий та інші рахунки в установах банків у рублях та іноземній валюті. У складі автосалону є адміністрація, відділ кадрів, бухгалтерія, відділ маркетингу та менеджменту, магазин.

Генеральний директор здійснює загальне керівництво підприємством. У підпорядкуванні генерального директора може знаходитись заступник директора з економіки та фінансів та заступник директора з маркетингу та збуту. Основними підрозділами, що забезпечують діяльність автосалону є:

- Бухгалтерія здійснює прийом платежів від покупців, підрахунок та видача зарплати, ведення податкової звітності.
- Відділ маркетингу займається визначенням політики збуту з урахуванням існуючої динаміки ринку та ресурсів, координацією роботи всіх підрозділів.
- Відділ менеджменту займається оформленням замовлень, складанням звітів про замовлення та оплати цих замовлень.

					КВРІСТ. 200197.01.09.09 ПЗ	Арк. 7
Зм.	Арк.	№докум.	Підпис	Дата		

У спеціалізованому центрі при автосалоні клієнтам компанії надаються послуги з гарантійного та післягарантійного ремонту, діагностики автомобілів, а також кузовні та малярні роботи будь-якого ступеня складності.

1.2 Клієнт-серверна архітектура у процесі проектування інформаційних систем

Клієнт-серверна архітектура є популярним підходом до побудови інформаційних систем. Клієнт-сервер – обчислювальна чи мережева архітектура, у якій завдання чи мережне навантаження розподілені між постачальниками послуг, що називаються серверами, та замовниками послуг, званими клієнтами. Фактично клієнт та сервер - це програмне забезпечення. Зазвичай ці програми розташовані на різних обчислювальних машинах і взаємодіють між собою через обчислювальну мережу за допомогою мережевих протоколів, але можуть бути розташовані також і на одній машині. Програми-сервери очікують від клієнтських програм запити та надають їм свої ресурси у вигляді:

- Даних (наприклад, завантаження файлів за допомогою НТТР, FTP, BitTorrent, потокове мультимедіа або робота з базами даних).
- Сервісних функцій (наприклад, робота з електронною поштою, спілкування за допомогою систем миттєвого обміну повідомленнями або перегляд веб-сторінок у всесвітньому павутинні).

Оскільки одна програма-сервер може виконувати запити від багатьох програм-клієнтів, її розміщують на спеціально виділеній обчислювальній машині, налаштованій особливим чином, як правило, спільно з іншими програмами-серверами, тому продуктивність цієї машини повинна бути високою. Через особливу роль такої машини в мережі, специфіки її обладнання та програмного забезпечення, її також називають сервером, а машини, що виконують клієнтські програми, відповідно, клієнтами. Оскільки одна програма-сервер може виконувати запити від безлічі програм-клієнтів, її розміщують на спеціально виділеній обчислювальній машині, налаштованій особливим чином, як правило,

					КВРІСТ. 200197.01.09.09 ПЗ	Арк. 8
Зм.	Арк.	№докум.	Підпис	Дата		

спільно з іншими програмами-серверами, тому продуктивність цієї машини має бути високою. Через особливу роль такої машини в мережі, специфіки її обладнання та програмного забезпечення її також називають сервером, а машини, що виконують клієнтські програми, відповідно, клієнтами.

Архітектуру «клієнт-сервер» прийнято розділяти на три класи: одно-, дво- та трирівневу. Однак, не можна сказати, що у питанні про такий поділ у співтоваристві ІТ-фахівців існує повний консенсус. Багато хто називає однорівневу архітектуру дворівневою і навпаки, те саме можна сказати про співвідношення дво- та трирівневої архітектури. Архітектуру «клієнт-сервер» прийнято розділяти на три класи: одно-, дво- та трирівневу. Однак, не можна сказати, що у питанні про такий поділ у співтоваристві ІТ-фахівців існує повний консенсус. Багато хто називає однорівневу архітектуру дворівневою і навпаки, те саме можна сказати про співвідношення дво- і трирівневої архітектури.

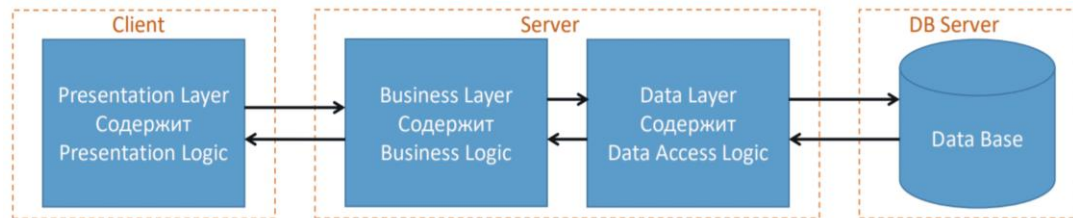
Однорівнева архітектура «клієнт-сервер» (1-Tier) – архітектура, де всі прикладні програми розосереджені по робочих станціях, які звертаються до спільного сервера баз даних або загального файлового сервера. Жодних прикладних програм сервер при цьому не виконує, тільки надає дані. Однорівнева архітектура «клієнт-сервер» (1-Tier) – є такою, де всі прикладні програми розосереджені по робочих станціях, які звертаються до спільного сервера баз даних або загального файлового сервера. Жодних прикладних програм сервер при цьому не виконує, лише надає дані.

В цілому, можна відзначити надійність такої архітектури, однак, їй складно керувати, оскільки в кожній робочій станції дані будуть присутні у різних варіантах. Тому виникає проблема їхньої синхронізації на окремих машинах. Загалом, як можна бачити з малюнка, в цій архітектурі проглядається ще один рівень – бази даних, що дає привід у багатьох випадках називати її дворівневою. будуть присутні у різних варіантах. Тому виникає проблема їхньої синхронізації на окремих машинах. Загалом, як можна бачити з рис. 1.1, у цій архітектурі проглядається ще один рівень – бази даних, що дає привід у багатьох випадках називати її дворівневою.

До дворівневої архітектури клієнт-сервер слід відносити таку, в якій прикладні програми зосереджені на сервері додатків (Application Server), наприклад, сервері CRM, а в робочих станціях знаходяться програми-клієнти, які надають для користувачів інтерфейс для роботи з додатками на загальному сервері. До дворівневої архітектури «клієнт-сервер» слід відносити таку, в якій прикладні програми зосереджені на сервері додатків (Application Server), наприклад, сервері CRM, а в робочих станціях знаходяться програми-клієнти, які надають користувачів інтерфейс для роботи з програмами на загальному сервері.

Така архітектура є найбільш логічною для архітектури «клієнт-сервер». У ній, проте, можна назвати два варіанта. Коли загальні дані зберігаються на сервері, а логіка їх обробки та бізнес-дані зберігаються на клієнтській машині, то така архітектура зветься "fat client thin server" (товстий клієнт, тонкий сервер). Коли дані, а й логіка їх обробки та бізнес-дані зберігаються на сервері, це називається "thin client fat server" (тонкий клієнт, товстий сервер). Така архітектура стала прообразом хмарних обчислень (Cloud Computing) Така архітектура є найбільш логічною для архітектури «клієнт-сервер». У ній, проте, можна назвати два варіанта. Коли загальні дані зберігаються на сервері, а логіка їх обробки та бізнес-дані зберігаються на клієнтській машині, то така архітектура зветься "fat client thin server" (товстий клієнт, тонкий сервер). Коли дані, а й логіка їх обробки та бізнес-дані зберігаються на сервері, це називається "thin client fat server" (тонкий клієнт, товстий сервер). Така архітектура стала прообразом хмарних обчислень (Cloud Computing).

У трирівневій архітектурі (рис. 1.1) сервер баз даних, файловий сервер та інші є окремим рівнем, результати роботи якого використовує сервер додатків. Логіка даних та бізнес-логіка знаходяться на сервері додатків. Всі звернення клієнтів до бази даних відбуваються через проміжне програмне забезпечення (middleware), яке знаходиться на сервері програм. Внаслідок цього, підвищується гнучкість роботи та продуктивність.



Рівень представлення
Статично або динамічно згенерований контент, що відображається за допомогою браузера (front-end)

Рівень логіки
Рівень підготовки даних для динамічно згенерованого контенту, рівень сервера додатків (application server).
Платформи JavaEE, ASP.NET PHP, тощо

Рівень даних
База даних, що включає в себе дані та систему керування ними, або ж готова RDBMS система, що надає доступ до даних та методи керування (back-end)

Рисунок 1.1 – Трьох рівнева архітектура «клієнт-сервер»

В окремий клас архітектури «клієнт-сервер» можна винести багаторівневу архітектуру, де кілька серверів додатків використовують результати роботи один одного, а також дані від різних серверів баз даних, файлових серверів та інших видів серверів.

По суті, попередній варіант, трирівнева архітектура – не більше ніж окремий випадок багаторівневої архітектури.

Перевагою багаторівневої архітектури є гнучкість надання послуг, які можуть бути комбінацією роботи різних додатків серверів різних рівнів та елементів цих додатків.

Очевидним недоліком є складність та багатокomпонентність такої архітектури.

Можна виділити наступні характеристики архітектури «клієнт-сервер»:

Асиметричність протоколів. Між клієнтами та сервером існують відносини «один до багатьох». Ініціатором діалогу із сервером зазвичай є клієнт.

Інкапсуляція послуг. Після отримання запиту на послугу від клієнта сервер вирішує, як повинна бути виконана дана послуга. Модифікація сервера може проводитися без впливу на роботу клієнтів, оскільки це не впливає на опублікований інтерфейс взаємодії між ними. Іншими словами, максимум, що при цьому може відчути користувач - незначна затримка відгуку сервера протягом невеликого часу апгрейду.

Цілісність. Програми та загальні дані для сервера управляються централізовано, що знижує вартість обслуговування та захищає цілісність даних. У той же час дані клієнтів залишаються персоніфікованими і незалежними.

Місцева прозорість. Сервер - це програмний процес, який може виконуватися на тій самій машині, що і клієнт, або на іншій машині, що підключена до мережі. Програмне забезпечення клієнт-сервер зазвичай приховує розташування сервера від клієнтів, перенаправляючи запит на послуги через мережу.

Обмін на основі повідомлень. Клієнти та сервер є нежорстко пов'язаними («loosely-coupled») процесами, які обмінюються повідомленнями: запитами на послуги та відповідями на них (рис. 1.2).

Модульний дизайн, здатний до розширення. Модульний дизайн програмної платформи «клієнт-сервер» надає їй стійкості до відмов, тобто відмова в якомусь модулі не викликає відмови всієї програми. У такій системі один або більше серверів можуть відмовити без зупинки всієї системи в цілому, до тих пір, поки послуги сервера, що відмовив, можуть бути надані з резервного сервера. Інша перевага модульності в тому, що програма «клієнт-сервер» може автоматично реагувати на підвищення або зниження навантаження на систему шляхом додавання або відключення послуг або серверів.

Незалежність від платформи. Ідеальна програма «клієнт-сервер» не залежить від платформ обладнання або операційної системи. Клієнти та сервери можуть розгортатися на різних апаратних платформах та різних операційних системах.

Масштабованість. Системи «клієнт-сервер» можуть масштабуватися як горизонтально (за кількістю серверів та клієнтів), так і вертикально (за продуктивністю та спектром послуг).

Розподіл функціоналу. Система «клієнт-сервер» – це співвідношення між процесами, що працюють на одній або різних машинах. Сервер – це процес надання послуг, клієнт – це споживач послуг.

					КВРІСТ. 200197.01.09.09 ПЗ	Арк.
						12
Зм.	Арк.	№докум.	Підпис	Дата		

Загальне використання ресурсів. Один сервер може надавати послуги безлічі клієнтів одночасно, і регулювати їх доступ до спільно використовуваних ресурсів.

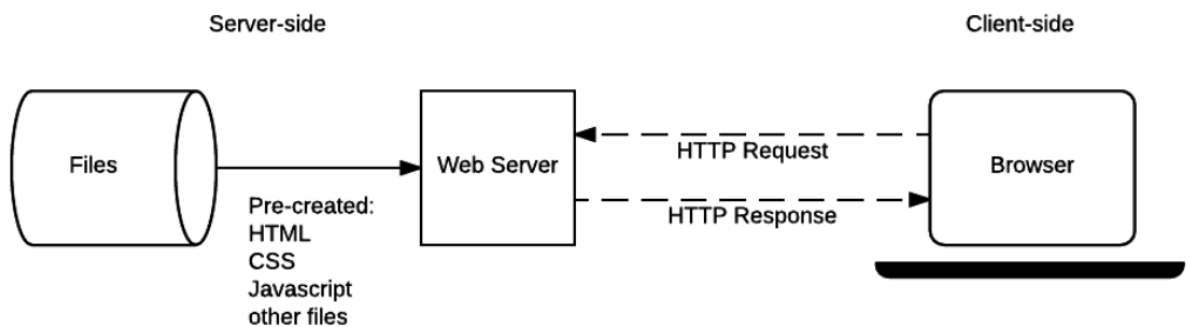


Рисунок 1.2 – Взаємодія між веб-сервером та браузером

1.3 Методологічні підходи до проектування інформаційних систем

У реальних умовах проектування – це пошук способу, який задовольняє вимоги функціональності системи засобами наявних технологій з урахуванням заданих обмежень.

До будь-якого проекту пред'являється ряд абсолютних вимог, наприклад, максимальний час розробки проекту, максимальні грошові вкладення в проект і т.д. Одна із складнощів проектування полягає в тому, що воно не є таким структурованим завданням, як аналіз вимог до проекту або реалізація того чи іншого проектного рішення.

Вважається, що складну систему неможливо описати у принципі. Це зокрема стосується систем управління підприємством. Одним з основних аргументів є зміна умов функціонування системи, наприклад, директивна зміна тих чи інших потоків інформації новим керівництвом. Ще один аргумент - обсяги технічного завдання, які для великого проекту можуть становити сотні сторінок, тоді як технічний проект може містити помилки.

Організацію проектування інформаційної системи прийнято розділяти на два типи:

Канонічне проектування відбиває особливості технології оригінального (індивідуального) процесу.

Типове проектування, для якого характерне типове проектне рішення (ТПР), тиражується та придатне до багаторазового використання.

Канонічне проектування, головним чином застосовується для локальних і відносно невеликих інформаційних систем із мінімальним використанням типових рішень. Адаптація проектних рішень відбувається лише за допомогою перепрограмування програмних модулів.

Організація канонічного проектування із використанням каскадної моделі життєвого циклу передбачає поділ процесу на наступні стадії та етапи:

- Передпроектна стадія. Проводиться передпроектний аналіз та складається технічне завдання. Тобто формуються вимоги до ІВ, розробляється її концепція, складається техніко-економічне обґрунтування та пишеться ТЗ.
- Проектна стадія передбачає складання ескізних та технічних проектів, розробку робочої документації.
- Післяпроектна стадія дає старт заходам щодо впровадження ІС, навчання персоналу, аналізу результатів випробування. Частиною цієї стадії стає супровід ІС та усунення виявлених недоліків.

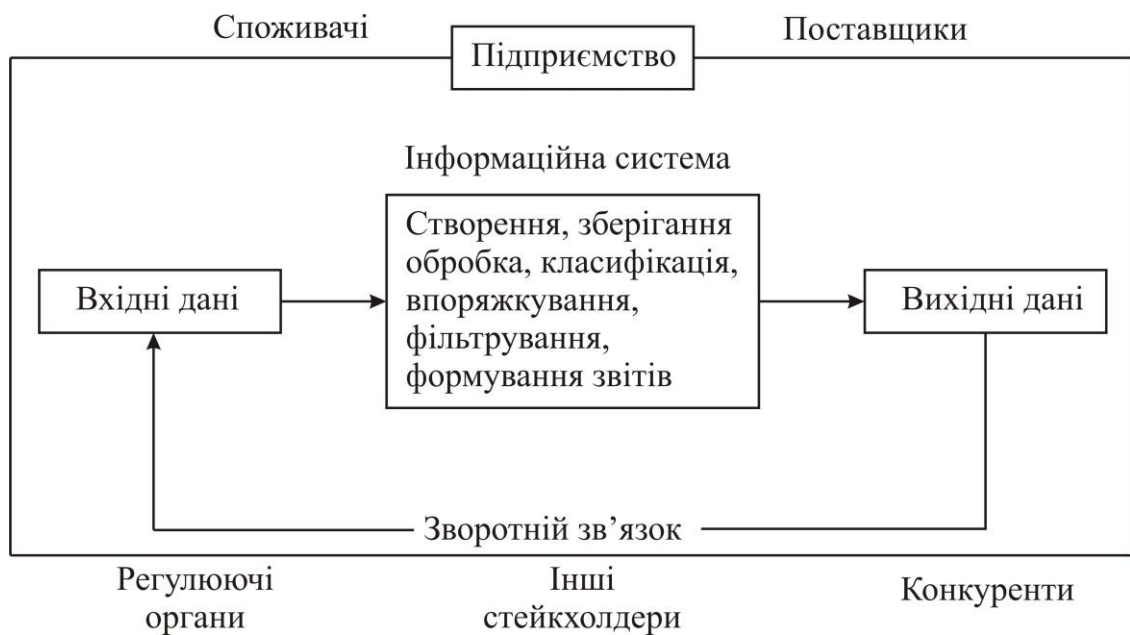


Рисунок 1.3 – Загальна схема інформаційної системи

Етапи, у разі потреби, можна укрупнювати чи деталізувати – поєднувати послідовні етапи, виключати «зайві», розпочинати виконання чергової стадії до завершення попередньої.

Метод типового проектування відрізняється можливістю декомпозиції проектованої ІС з поділом на компоненти, до яких входять програмні модулі, підсистеми, комплекси завдань та ін. Для реалізації компонентів можна скористатися типовими рішеннями, які вже існують на ринку, і налаштувати їх під потрібні конкретної організації. При цьому типове проектування передбачає обов'язкову наявність документації, що описує деталі типового проектного рішення і процедури налаштування.

Декомпозиція може мати кілька рівнів, що дозволяє виділити класи типового проектного рішення:

- Елементні – за окремим завданням (елементом).
- Підсистемні – за окремими підсистемами.
- Об'єктні – галузеві типові проектні рішення, містять весь набір підсистем.

Етап аналізу передбачає докладне дослідження бізнес-процесів (функцій, визначених на етапі вибору стратегії) та інформації, необхідної для їх виконання (сутностей, їх атрибутів та зв'язків). На цьому етапі створюється інформаційна модель, а на наступному етапі проектування – модель даних.

Вся інформація про систему, зібрана на етапі визначення стратегії, формалізується та уточнюється на етапі аналізу. Особливу увагу слід приділити повноті переданої інформації, аналізу інформації на предмет відсутності протиріч, а також пошуку інформації, що не використовується взагалі або дублюється. Як правило, замовник не відразу формує вимоги до системи загалом, а формулює вимоги до окремих її компонентів. Зверніть увагу на узгодженість цих компонентів.

Аналітики збирають та фіксують інформацію у двох взаємопов'язаних формах:

- Функції – інформація про події та процеси, що відбуваються у бізнесі.

– Сутності – інформація про речі, що мають значення для організації та про які щось відомо.

Двома класичними результатами аналізу є:

– Ієрархія функцій, яка розбиває процес обробки на складові (що робиться і з чого це складається).

– Модель «сутність-зв'язок» (Entry Relationship model, ER-модель), яка описує сутності, їх атрибути та зв'язки (відносини) між ними.

Ці результати є потрібними, але не достатніми. До достатніх результатів слід віднести діаграми потоків даних та діаграми життєвих циклів сутностей. Часто помилки аналізу виникають при спробі показати життєвий цикл сутності на діаграмі ER.

Розглянемо далі три найбільш застосовувані методології структурного аналізу:

– Діаграми «сутність-зв'язок» (Entity-Relationship Diagrams, ERD), які служать для формалізації інформації про сутності та їх відносини.

– Діаграми потоків даних (Data Flow Diagrams, DFD), що служать для формалізації представлення функцій системи.

– Діаграми переходів станів (State Transition Diagrams, STD), які відображають поведінку системи, що залежить від часу; Діаграми життєвих циклів сутностей відносяться саме до цього класу діаграм.

Діаграми потоків даних DFD показує зовнішні по відношенню до системи джерела та стоки (адресати) даних, ідентифікує логічні функції (процеси) та групи елементів даних, що пов'язують одну функцію з іншою (потоки), а також ідентифікує сховища (накопичувачі) даних, до яких здійснюється доступ. Структури потоків даних та визначення їх компонентів зберігаються та аналізуються у словнику даних. Кожна логічна функція (процес) може бути деталізована за допомогою нижнього рівня DFD; коли подальша деталізація перестає бути корисною, переходять до вираження логіки функції за допомогою специфікації процесу (міні-специфікації). Вміст кожного сховища зберігають у словнику даних, модель даних сховища розкривається за допомогою ER-діаграм.

Зокрема, у DFD не показуються процеси, які управляють власне потоком даних і не наводяться відмінності між допустимими та неприпустимими шляхами (рис. 1.4). DFD містять безліч корисної інформації, а також:

- Дозволяють представити систему з погляду даних.
- Ілюструють зовнішні механізми подачі даних, які вимагатимуть наявності спеціальних інтерфейсів.
- Дозволяють уявити як автоматизовані, і ручні процеси системи.
- Виконують орієнтоване дані секціонування всієї системи.

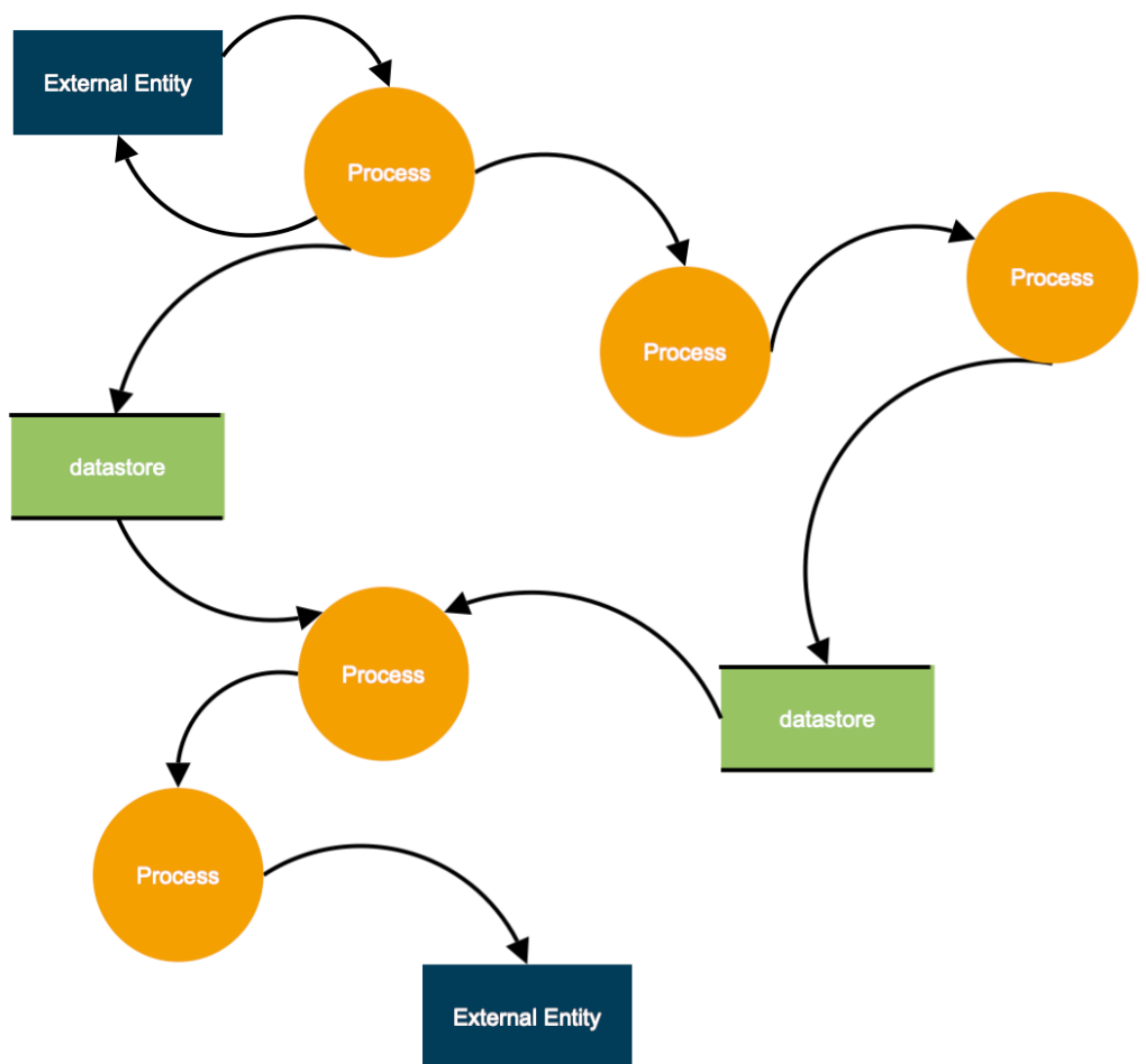


Рисунок 1.4 – Схематичне подання DFD діаграми

дозволяють описати їхні властивості. Зв'язки визначають взаємодію між сутностями і можуть бути один до одного, один до багатьох або багато до багатьох (1.4).

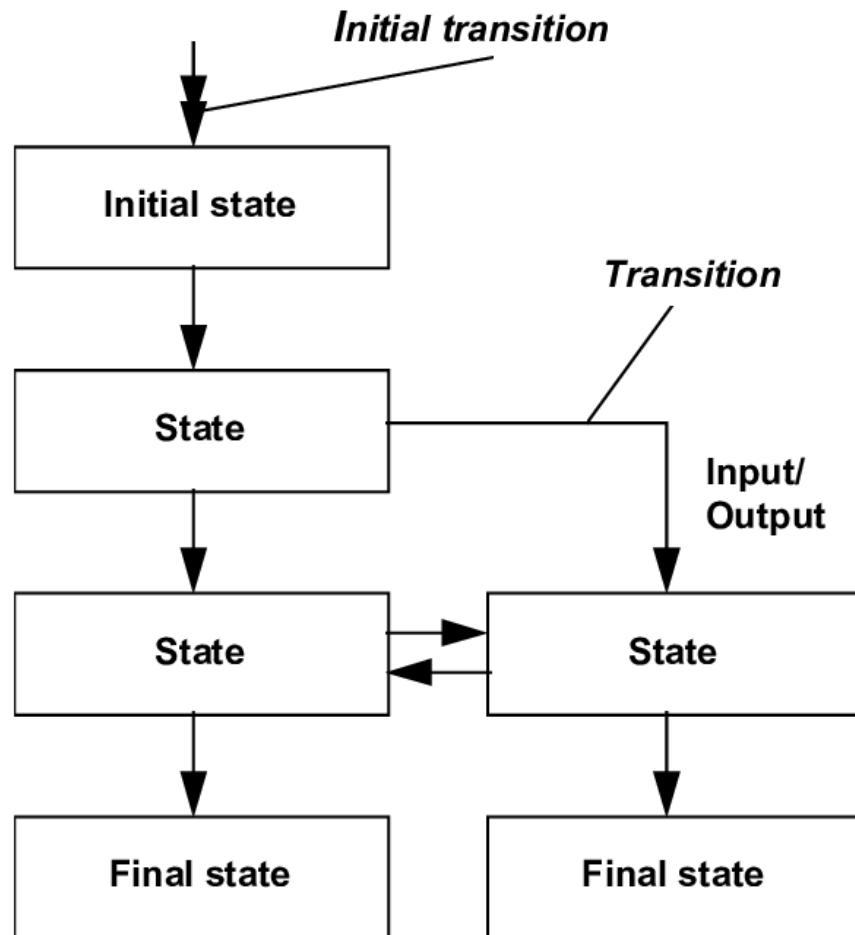


Рисунок 1.5 – Схематичне подання STD діаграми

Основні елементи ER-діаграми:

1. Сутність (Entity): Відображається у вигляді прямокутника з назвою сутності всередині. Наприклад, сутність "Книга" може мати атрибути, такі як "Назва", "Автор" і "Рік видання".

2. Атрибут (Attribute): Відображається як овал всередині сутності і описує властивості цієї сутності. Наприклад, атрибут "Назва" може мати тип "Рядок", атрибут "Рік видання" може мати тип "Ціле число".

3. Зв'язок (Relationship): Відображається як лінія, що з'єднує дві сутності. Зв'язки можуть мати напрямок (один до одного, один до багатьох, багато до

багатьох) та можуть мати атрибути. Наприклад, зв'язок "Автор написав Книгу" між сутностями "Автор" і "Книга" може мати атрибут "Дата написання".

4. Ключ (Key): Позначається підкресленням або підкресленням преривчастої лінії. Ідентифікує унікальний ідентифікатор для сутності. Наприклад, у сутності "Книга" ключем може бути атрибут "ISBN", який є унікальним для кожної книги.

ER-діаграми можуть бути розширені для включення додаткових елементів, таких як спадковість (inheritance), агрегація (aggregation) та асоціація (association), що дозволяють подібніше описати взаємодію між сутностями.

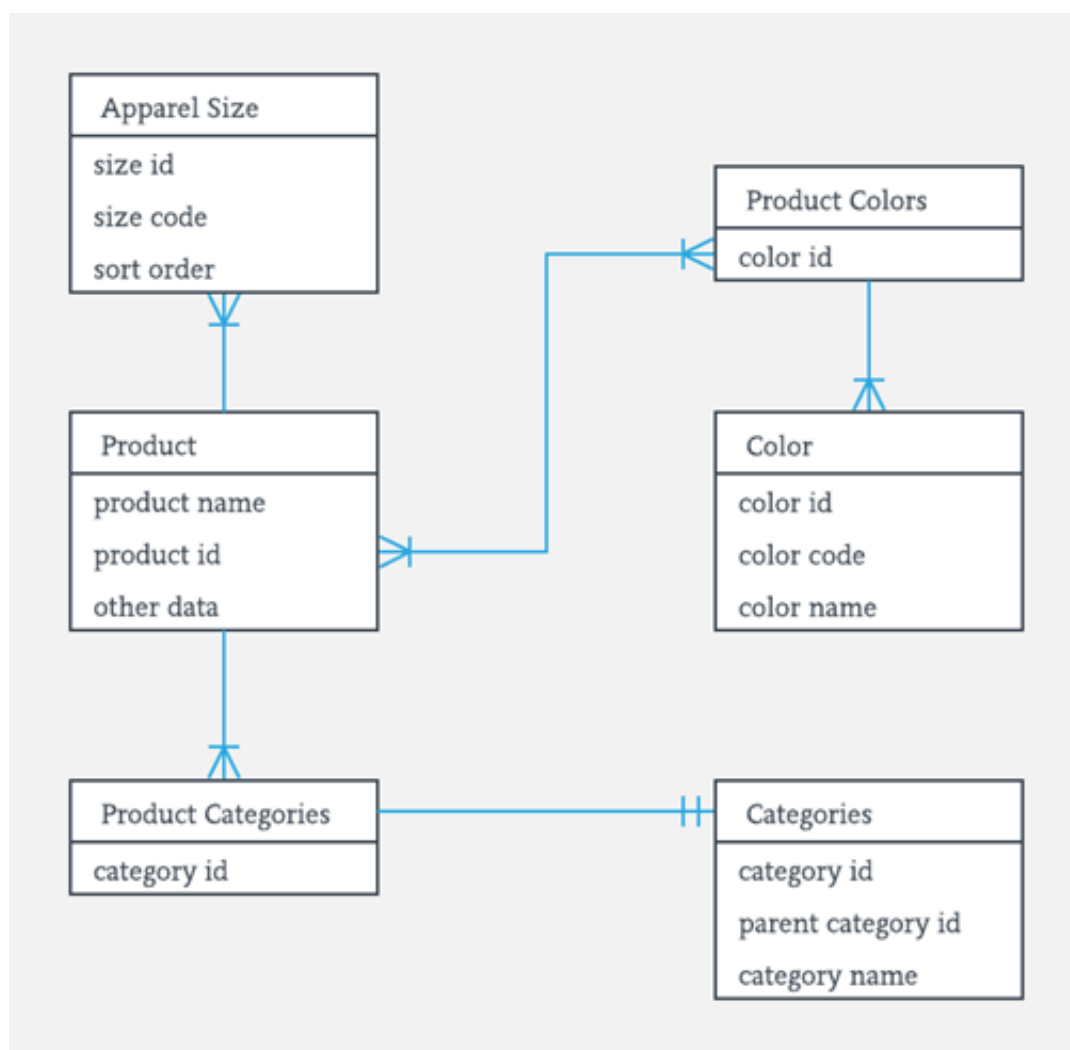


Рисунок 1.6 – Приклад ER діаграми

вже набагато більше, і вони докладніші. Отримані дані, і навіть характеристики, передані групами тестування, можуть показати, що вибір СУБД на етапі визначення стратегії був неправильним і обрана СУБД неспроможна задовольняти тим чи іншим вимогам інформаційної системи. Такі дані можуть бути отримані щодо вибору апаратної платформи і операційної системи. Отримання подібних результатів ініціює зміну даних, отриманих на етапі визначення стратегії, наприклад, перераховується кошторис витрат на проект.

Вибір засобів розробки також уточнюється на етапі аналізу. Через те, що етап аналізу дає повніше уявлення про інформаційну систему, ніж було на етапі визначення стратегії, план робіт то, можливо скоригований. Якщо обраний на попередньому етапі засіб розробки не дозволяє виконати ту чи іншу частину робіт у заданий термін, то приймається рішення про зміну термінів (зазвичай це збільшення терміну розробки) або про зміну засобу розробки. Здійснюючи вибір тих чи інших засобів, слід враховувати наявність висококваліфікованого персоналу, який володіє вибраними засобами розробки, а також адміністраторів обраної СУБД. Ці рекомендації також уточнюватимуть дані етапу вибору стратегії (сукупність умов, за яких передбачається експлуатувати майбутню систему).

Уточнюються також обмеження, ризики, критичні чинники. Якщо будь-які вимоги не можуть бути задоволені в інформаційній системі, реалізованій з використанням СУБД та програмних засобів, обраних на етапі визначення стратегії, то це також ініціює уточнення та зміну одержуваних даних (зрештою кошторису витрат та планів робіт, а можливо, і зміна вимог замовника до системи, наприклад, їх ослаблення). Докладніше описуються ті можливості, які будуть реалізовані у системі.

Специфічні особливості процесу проектування дають змогу виділяти методології, побудовані на різних принципах. Серед основних сучасних методологій проектування інформаційних систем можна виділити наступні:

SADT. Методологія функціонального моделювання робіт, яка заснована на структурному аналізі та графічному поданні організації як системи функцій. Тут

						КВРІСТ. 200197.01.09.09 ПЗ	Арк. 22
Зм.	Арк.	№докум.	Підпис	Дата			

виділяється функціональна, інформаційна та динамічна моделі. В даний час методологія відома як нотація (стандарт) IDEF0. Аналізований процес графічно представляється як чотирикутника, де зверху зображуються регламентуючі і управляючі впливу, знизу – об'єкти управління, зліва – вхідні дані, а справа – вихідні.

RAD. Методологія швидкої розробки програм. У RAD швидка розробка додатків можлива рахунок застосування компонентно-орієнтованого конструювання. Методологія застосовується на проектах з обмеженим бюджетом, нечіткими вимогами до ІВ, за стислих термінів реалізації. До неї вдаються, якщо інтерфейс користувача можна продемонструвати в прототипі, а проект розділити на функціональні елементи.

RUP. У методології RUP реалізуються ітераційний та нарощуваний (інкрементний) підходи. Побудова системи відбувається з урахуванням архітектури інформаційної системи, а планування і проектне управління – з урахуванням функціональних вимог до ІВ. Розробка загальної інформаційної системи відбувається ітераціями, як комплекс окремих невеликих проектів зі своїми планами та завданнями. Для ітераційного циклу характерна періодичний зворотний зв'язок та адаптація до ядра ІВ.

1.4 Постановка задачі

Розробка інформаційної системи для автосалону є актуальною та корисною задачею в сучасному бізнес-середовищі. Зокрема розробка інформаційної системи автосалону дозволить вирішити наступні завдання:

– Ефективне керування даними: Автосалони мають великий обсяг даних, включаючи інформацію про моделі автомобілів, клієнтів, продажі, запаси тощо. Інформаційна система дозволить ефективно збирати, зберігати і керувати цими даними, забезпечуючи легкий доступ до них, швидкість і точність обробки.

– Автоматизація бізнес-процесів: Інформаційна система може автоматизувати багато рутинних завдань, таких як управління запасами, облік

продажів, розрахунок цін, створення договорів тощо. Це полегшить роботу персоналу автосалону, зменшує ймовірність помилок і прискорює виконання процесів.

– Краще обслуговування клієнтів: Завдяки інформаційній системі, автосалон може зберігати повну історію клієнтів, їхні вподобання, попередні покупки, графіки тест-драйвів тощо. Це дозволяє персоналу надавати персоналізоване обслуговування, швидко знаходити відповідні автомобілі для клієнтів і підвищувати загальне задоволення клієнтів.

– Аналіз та прогнозування: Інформаційна система надасть можливість аналізувати дані автосалону, отримувати звіти, статистику і ділову аналітику.

Для вирішення поставленого завдання необхідно виконання наступних етапів:

1. проаналізувати підходи до створення інформаційних систем, а також відомі технології та засоби, що використовуються для проєктування інформаційних систем;
2. здійснити визначення вимог до інформаційної системи автосалону;
3. провести аналіз предметної області та розробити концептуальну модель бази даних;
4. виконати проєктування інформаційної системи автосалону;
5. реалізувати інформаційну систему автосалону;
6. здійснити аналіз отриманих результатів.

2 ВИБІР ЗАСОБІВ РЕАЛІЗАЦІЇ ТА ПРОЄКТУВАННЯ БАЗИ ДАНИХ ІНФОРМАЦІЙНОЇ СИСТЕМИ АВТОСАЛОНУ

2.1 Вибір засобів реалізації для проєктування інформаційної системи автосалону

Під час проєктування інформаційної системи автосалону було проведено розроблення всіх складових компонентів системи, які відповідають за роботу зі створенням, додаванням, оновленням і видаленням інформації. Для цього було використано такі програмні технології та засоби:

- Середовище розробки Visual Studio 2022, яке надає потужні інструменти для створення та відлагодження програмного забезпечення.

- Технологію Entity Framework, яка забезпечує об'єктно-орієнтований доступ до бази даних і спрощує роботу з ними.

- Технологію MVC (Model-View-Controller), яка визначає підхід до побудови веб-додатків, розділяючи логіку додатку на моделі, представлення та контролери.

- IIS Express, який є вбудованим веб-сервером для розробки та відлагодження веб-додатків на локальному комп'ютері.

2.1.1 Патерн проєктування Model View Controller (MVC)

В якості патерну проєктування в процесі розробки інформаційної системи автосалону було використано патерн MVC. Архітектура MVC (Model, View, Controller) – це шаблон проєктування, який використовується для створення програм. Цей архітектурний шаблон в основному використовувався для веб-додатків.

Загалом відомо, що патерн MVC має три компоненти: модель, подання та контролер. Проте можна виділити також ще один компонент, який не менш

важливий для того, щоб вся архітектура працювала – маршрутизатор. Таким чином патерн MVC складають компоненти:

- Представлення – місце взаємодії із користувачем.
- Маршрутизатор – обробляє запити, які потрібно виконати.
- Контролер – обробляє запити і надсилає відповідь для представлення.
- Модель – проміжне програмне забезпечення, яке обробляє операції з базою даних.

Представлення (View) представляє спосіб представлення даних у програмі. Представлення створюються на основі даних, зібраних із моделі. Запитуючи інформацію із моделі, користувач отримує вихідну презентацію. Окрім представлення даних із таблиць, представлення також відображає дані з інших джерел. Усі компоненти інтерфейсу користувача, такі як текстові поля, випадаючі меню тощо, відобразатимуться у будь-якому поданні клієнта.

Контролери – це ті компоненти програми, які обробляють взаємодію користувача. Введення користувача інтерпретується контролером, змушуючи модель і представлення змінюватися на основі інформації, яку він отримує. Встановлюючи зв'язок із пов'язаним представленням контролера, користувач може змінювати вигляд представлення (наприклад, прокручувати документ) і оновлювати стан пов'язаної моделі (наприклад, зберігати документ).

В загальному, компонент моделі зберігає дані та логіку. Наприклад, об'єкт Controller буде отримувати інформацію про клієнта з бази даних. Дані передаються між компонентами контролера або між елементами бізнес-логіки. Він маніпулює даними та надсилає їх назад до бази даних або використовується для відтворення тієї самої інформації.

Окрім того, даний компонент патерну MVC, відповідає на запити від представлень і має інструкції від контролера, які дозволяють йому самостійно оновлюватися. Це також найнижчий рівень шаблону, який відповідає за збереження даних.

На рис. 2.1 наведено схему взаємозв'язків між складовими компонентами в архітектурі MVC.

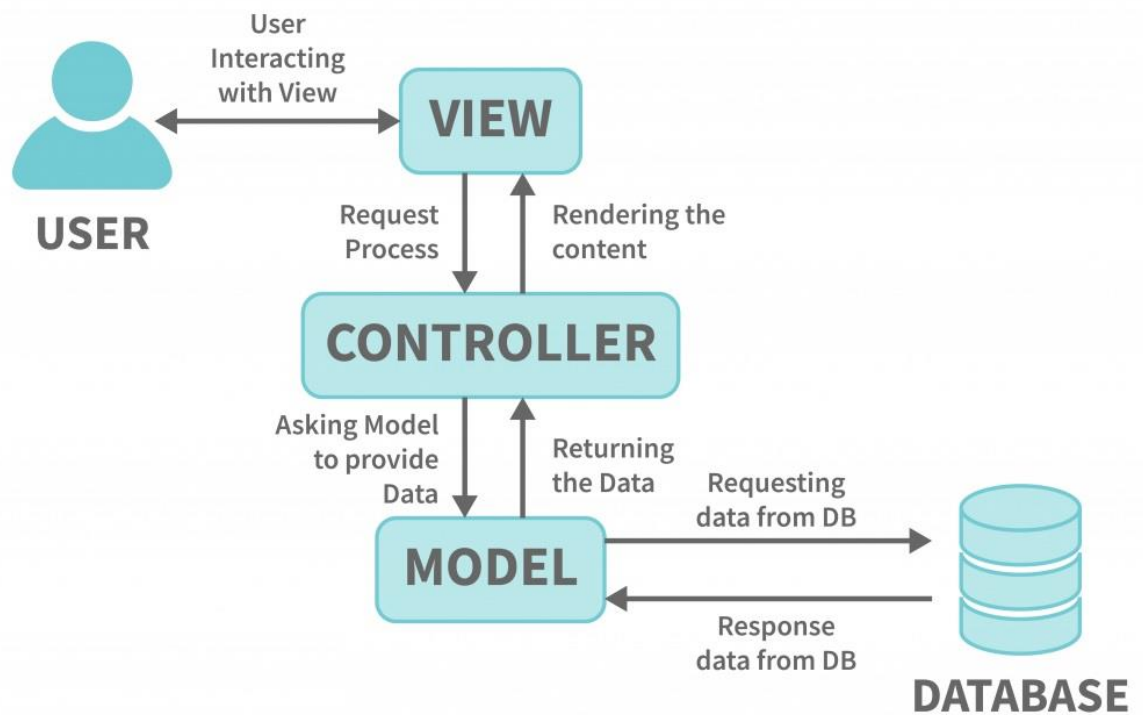


Рисунок 2.1 – Архітектура Model, View, Controller

Високорівневий алгоритм роботи відповідно до архітектури Model, View, Controller можна подати наступними кроками (вважатимемо, що користувач здійснив запит до сторінки із адресою <https://www.interviewbit.com/online-python-compiler>):

1. Спочатку запит спершу надходить до маршрутизатора кореня програми (<https://www.interviewbit.com>). Тоді кореневий маршрутизатор розбирає URL-адресу, за допомогою якої контролер повинен передати запит.
2. Тепер контролер із назвою (`online-python-compiler`) отримує запит. А потім цей контролер передає запит відповідній дії.
3. Далі Action визначає тип запиту та пов'язане з ним ім'я, обробляє запит і надсилає відповідний результат до View.
4. Представлення обробляє запит, у випадку, якщо існує вимога взаємодії з базою даних, тоді конкретна дія взаємодіє з моделлю, яка запитує дані з бази даних. І повертається до дії. Потім дія надсилає відповідні результати до View.

2.1.2 Технологія Entity Framework

Робота із базою даних здійснювалась за допомогою фреймворку Entity Framework. Entity Framework – це платформа ORM (Object Relational Mapping) із відкритим кодом для додатків .NET, які підтримує Microsoft. Це дозволяє розробникам працювати з даними за допомогою об'єктів класів, не зосереджуючись на таблицях і стовпцях бази даних, де зберігаються дані. За допомогою Entity Framework розробники можуть працювати на вищому рівні абстракції, коли вони мають справу з даними. За допомогою Entity Framework є можливість створювати та підтримувати орієнтовані на дані програми з меншим кодом порівняно з традиційними програмами. Entity Framework – це Object-Relational Mapper, який дозволяє розробникам .NET працювати з базою даних за допомогою об'єктів .NET. Це усуває необхідність доступу до коду даних, які розробники повинні написати. На рис. 2.2 зображено місце фреймворку Entity Framework у структурі розробки програмного забезпечення.

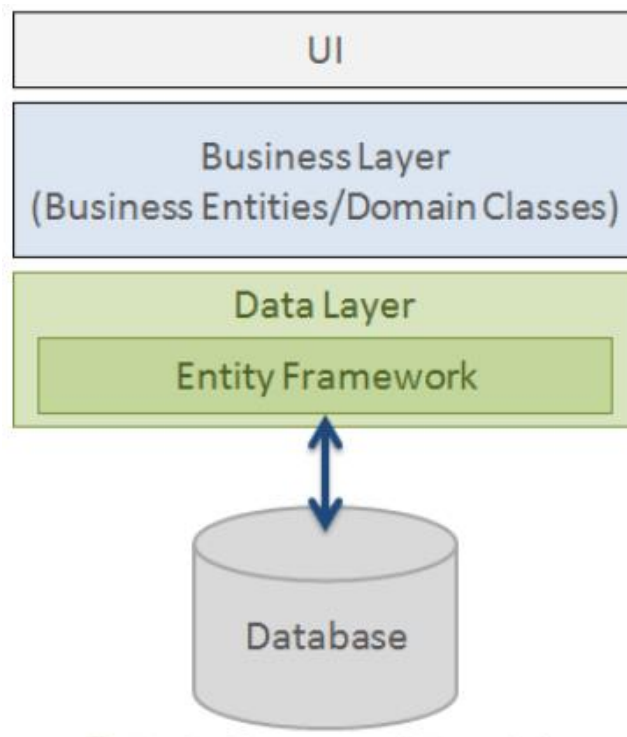


Рисунок 2.2 – місце фреймворку Entity Framework у структурі розробки програмного забезпечення

Entity Framework є ORM. Метою ORM є підвищення продуктивності розробника шляхом зменшення надлишкових завдань, які використовуються в додатку.

Entity Framework може генерувати необхідні команди бази даних для читання або запису даних у базі даних і виконувати їх. У випадку надсилання запитів, є можливість виражати запити щодо домену об'єкта за допомогою LINQ to Entities. Entity Framework виконає відповідний запит до бази даних, а потім матеріалізує результати в екземпляри об'єктів домену для роботи з програмою. Слід відзначити, що існують також й інші ORM, зокрема NHibernate і LLBL Gen Pro. Оскільки більшість ORM зазвичай відображають тип домену безпосередньо в схемі бази даних, то при реалізації даної інформаційної системи автосалону було використано саме Entity Framework.

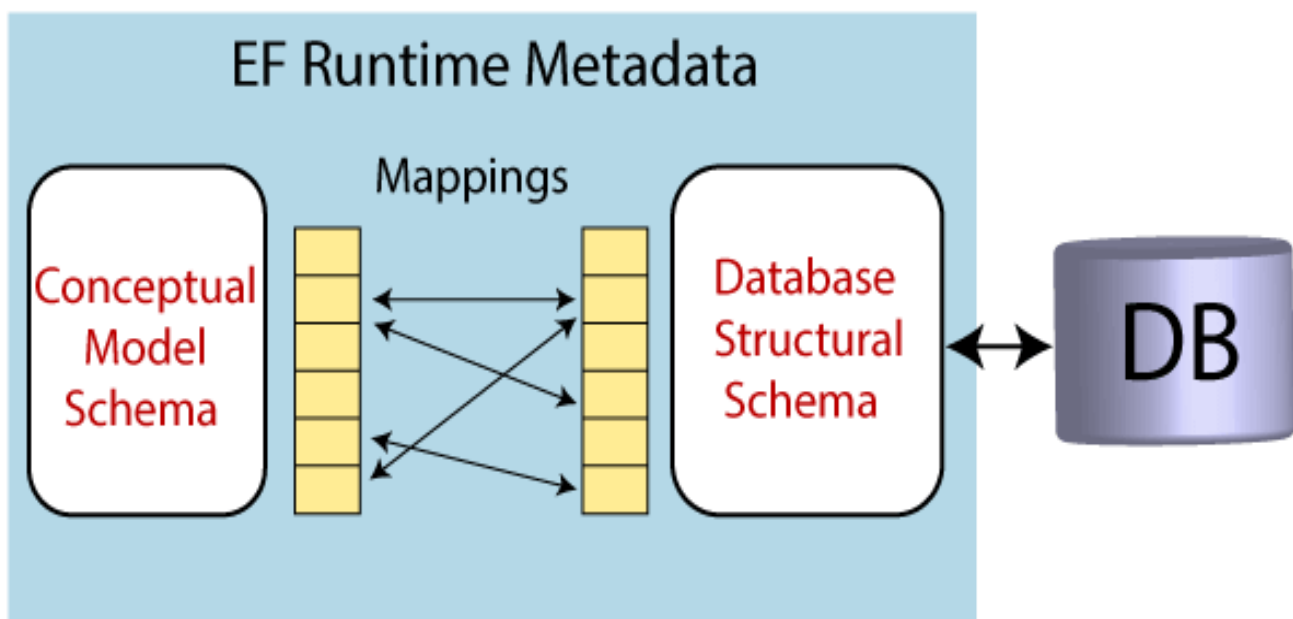


Рисунок 2.3 – Процес відображення даних у Entity Framework.

2.1.3 Веб-сервер IIS Express

IIS Express (Internet Information Services Express) є безкоштовною версією веб-сервера IIS від Microsoft, яка може бути використана для локальної розробки і тестування веб-додатків. Основними перевагами IIS Express є:

– Локальна розробка: IIS Express дозволяє вам розробляти і тестувати веб-додатки на вашому локальному комп'ютері перед розгортанням на реальному веб-сервері. Він надає схожу функціональність з повноцінним IIS, але має менші системні вимоги.

– Легкість використання: IIS Express має простий інтерфейс та легко налаштовується. Він може автоматично запускатися під час відкриття проекту в Visual Studio і підтримує багато розробницьких функцій, таких як перезавантаження веб-сайту після змін коду.

– Підтримка різних протоколів і платформ: IIS Express підтримує HTTP, HTTPS, FTP, і він може працювати з різними версіями .NET Framework. Він також підтримує платформу ASP.NET для розробки веб-додатків.

Беручи до уваги наведені переваги різних програмних засобів та технологій, було вирішено обрати IIS Express в якості веб-сервера, а також використовувати Entity Framework та паттерн MVC для розробки інформаційної веб-системи автосалону.

Таким чином, після розгляду переваг та недоліків різних програмних засобів і технологій, прийнято рішення використовувати IIS Express як веб-сервер, а також фреймворк Entity Framework і паттерн MVC для розробки інформаційної веб-системи автосалону.

2.2 Аналіз вимог до інформаційної системи автосалону

Один з важливих етапів розробки інформаційної системи полягає у визначенні вимог, яким повинна відповідати ця система. Вирізняють функційні та нефункційні вимоги. Функціональні вимоги є вимогами, що стосуються функціонування системи або ПЗ. Вони описують, які дії система або програмне забезпечення повинні виконувати, які функції вони мають здійснювати і які результати повинні бути отримані.

узагальненого розуміння даних, їх структури та взаємозв'язків, незалежно від реалізації в конкретній системі управління базами даних. Концептуальна модель бази даних дозволяє встановити загальний фреймворк для подальшого проектування фізичної моделі бази даних, що враховуватиме специфічні вимоги технології та забезпечить ефективне зберігання і обробку даних.

Головна мета проектування концептуальної моделі бази даних (МБД) полягає у тому, щоб представити предметну область таким чином, щоб й технічні, й нетехнічні користувачі могли легко зрозуміти. Концептуальна модель є загальною та не залежить від конкретної технології бази даних, що дозволяє її легко змінювати за потребою.

Відповідно до поставленого завдання та для реалізації покладених функцій інформаційної системи по збереженню та обробці даних, основою даної інформаційної системи автосалону є база даних. Створення та робота із базою даних здійснюватиметься із використанням об'єктно-орієнтовану технологію доступу до даних Entity Framework. В якості підходу було використано Code First, що визначає спочатку моделі сутностей у кодї, а далі створює базу даних по цих моделях.

Провівши аналіз та дослідження предметної галузі та проаналізувавши вимоги до проектованої інформаційної системи автосалону було виділено такі сутності у базі даних: автомобіль (Car), клієнт (Client), тип кузова автомобіля (CarBody), комплектація (Configuration), двигун (Engine), спосіб перемикання передач (Transmission), менеджер із продажу (Manager), договір купівлі (Order), спосіб оплати (PaymentMethod), посада менеджера (Position), постачальник (Supplier),

Виконаємо проектування необхідних моделей у вигляді класів та подамо їх в вигляді таблиць (таблиці 2.1 – 2.8).

Таблиця 2.1 – Поля таблиці Car та їх тип

Поле	Тип	Опис
CarID	int	Ідентифікатор автомобіля (primary key)
SupplierID	int	Ідентифікатор поставщика (foreign key), для зв'язку із сутністю Supplier
CarBodyID	int	Ідентифікатор типу кузова (foreign key), для зв'язку із сутністю CarBody
EngineID	int	Ідентифікатор типу двигуна (foreign key), для зв'язку із сутністю Engine
TransmissionID	int	Ідентифікатор типу перемикачання передач (foreign key), для зв'язку із сутністю Transmission
ConfigurationID	int	Ідентифікатор конфігурації (foreign key), для зв'язку із сутністю Configuration
Mark	nvarchar(60)	Марка автомобіля
Model	nvarchar(60)	Модель автомобіля
AvailableCount	int	Доступна кількість
Color	nvarchar(40)	Колір
Vin	nvarchar(20)	Vin код
Price	int	Ціна
ManufacturingDate	Datetime	Дата виготовлення

```

public class Car
{
    public int CarID { get; set; }
    public int SupplierID { get; set; }
    public int CarBodyID { get; set; }
    public int EngineID { get; set; }
    public int TransmissionID { get; set; }
    public int ConfigurationID { get; set; }
}

```

```

public string Mark { get; set; }
public string Model { get; set; }
public int AvailableCount { get; set; }
public string Color { get; set; }
[StringLength(20)]
public string Vin { get; set; }
[DataType(DataType.Date)]
public DateTime ManufacturingDate { get; set; }
public virtual ICollection<Order> Orders { get; set; }
}

```

Поле CarID у таблиці Car є стовпцем первинного ключа у таблиці бази даних, що відповідає даному класу. Поле Orders виступає полем навігації. В полі навігації знаходяться інші сутності, які пов'язані з даною сутністю. Тобто у таблиці Car поле Orders міститиме всі сутності Order, які є пов'язаними із даною сутністю Car. Далі визначимо у базі даних набір сутностей, які розширюють сутність Car, тобто мають із сутністю Car відношення один до багатьох (рис. 2.4).

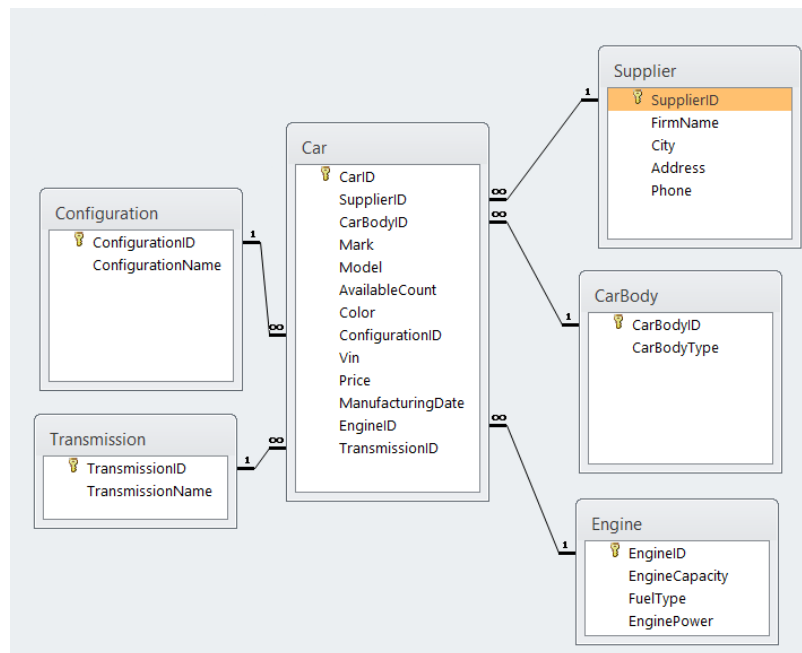


Рисунок 2.4 – Розширення сутності Car сутностями Engine, Transmission, Configuration, CarBody, Supplier

Цими сутностями виступаються Engine (двигун), Transmission (тип перемикання передач), Configuration (комплектація), CarBody (тип кузова), а також Supplier (постачальник).

У сутності Engine виділимо поля EngineID (первинний ключ), EngineCapacity, FuelType та EnginePower (таблиця 2.2). Також у даній сутності введено поле навігації Cars, що містить колекцію всіх автомобілів, у яких встановлено даний вид трансмісії. Слід відзначити, що це поле буде присутнім у Transmission, Configuration, CarBody, Supplier, і виконуватиме аналогічні функції.

Таблиця 2.2 – Поля таблиці Engine

Поле	Тип	Опис
EngineID	int	Ідентифікатор двигуна (primary key)
EngineCapacity	int	Об'єм двигуна
FuelType	nvarchar(50)	Тип палива
EnginePower	int	Потужність двигуна

```
public class Engine
{
    public int EngineID { get; set; }
    public int EngineCapacity { get; set; }
    public int EnginePower { get; set; }
    [Display(Name = "Fuel Type")]
    public string FuelType { get; set; }
    public virtual ICollection<Car> Cars { get; set; }
}
```

Таблиця 2.3 – Поля таблиці Transmission

Поле	Тип	Опис
TransmissionID	int	Ідентифікатор трансмісії (primary key)
TransmissionName	nvarchar(40)	Назва


```

public class CarBody
{
    public int ConfigurationID { get; set; }
    [Display(Name = " Configuratio Name")]
    [StringLength(50)]
    public string CarBodyType { get; set; }
    public virtual ICollection<Car> Cars { get; set; }
}

```

Таблиця 2.6 – Поля таблиці Supplier

Поле	Тип	Опис
SupplierID	int	Ідентифікатор поставщика (primary key)
FirmName	nvarchar(50)	Назва поставщика
City	nvarchar(50)	Місто
Address	nvarchar(50)	Адреса
Phone	nvarchar(50)	Телефоний номер

```

public class Supplier
{
    public int SupplierID { get; set; }
    [Display(Name = " Firm Name ")]
    [StringLength(50)]
    public string FirmName { get; set; }
    [StringLength(50)]
    public string City { get; set; }
    [StringLength(50)]
    public string Address { get; set; }
    [StringLength(50)]
    public string Phone { get; set; }
    public virtual ICollection<Car> Cars { get; set; }
}

```

Виділимо сутність Order, яка складатиметься із таких полів: OrderID, OrerDate, CarID, ManagerID, ClientID, PaymentMethodID, Status (таблиця 2.7). У даній сутності поля OrderID, OrerDate, CarID, ManagerID, ClientID, PaymentMethodID виступатимуть як зовнішні ключі. Також введено поля навігації Car, Manager, Client, PaymentMethod для зв'язку із відповідними таблицями.

Таблиця 2.7 – Поля таблиці Order

Поле	Тип	Опис
OrderID	int	Ідентифікатор замовлення (primary key)
OrerDate	Datetime	Дата складання замовлення
CarID	int	Ідентифікатор автомобіля (foreign key)
ManagerID	int	Ідентифікатор манаджера із продажу (foreign key)
ClientID	int	Ідентифікатор клієнта (foreign key)
PaymentMethodID	int	Ідентифікатор способу оплати (foreign key)
Status	bool	Статус замовлення

```
public class Order
{
    public int OrderID { get; set; }
    public DateTime OrerDate { get; set; }
    public int CarID { get; set; }
    public int PaymentMethodID { get; set; }
    public int ManagerID { get; set; }
    public int ClientID { get; set; }
    public bool Status { get; set; }
    public virtual Car Car { get; set; }
    public virtual Manager Manager { get; set; }
    public virtual Client Client { get; set; }
    public virtual PaymentMethod PaymentMethod { get; set; }
}
```

}

Виділимо також сутність Client, яка зв'язана із сутністю Order відношенням один до багатьох (рис. 2.3). Окрім полів ClientSurname, ClientName, Phone, ClientPlaceResidence виділимо також навігаційне поле Orders, що містить всі замовлення, складені клієнтом (таблиця 2.8).

Таблиця 2.8 – Поля таблиці Client

Поле	Тип	Опис
ClientID	int	Ідентифікатор клієнта (primary key)
ClientSurname	nvarchar(50)	Прізвище клієнта
ClientName	nvarchar(50)	Ім'я клієнта
Phone	nvarchar(50)	Номер телефону клієнта
ClientPlaceResidence	nvarchar(50)	Місце проживання клієнта

```
public class Client
{
    public int ClientID { get; set; }
    [StringLength(50)]
    public string ClientSurname { get; set; }
    [StringLength(50)]
    public string ClientName { get; set; }
    [StringLength(50)]
    public string Phone { get; set; }
    [StringLength(50)]
    public string ClientPlaceResidence { get; set; }
    public virtual ICollection<Order> Orders { get; set; }
}
```

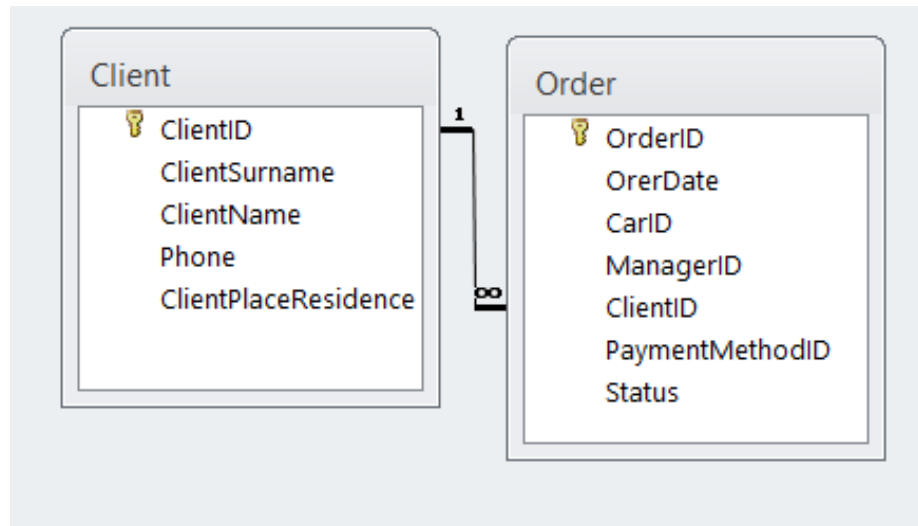


Рисунок 2.5 – Зв’язок сутностей Client та Order

Також виділимо сутність, яка описує менеджера із продажів – Manager. Поле Orders є колекцією всіх замовлень, пов’язаних із даним менеджером із продажів.

Таблиця 2.9 – Поля таблиці Manager

Поле	Тип	Опис
ManagerID	int	Ідентифікатор клієнта (primary key)
ManagerName	nvarchar(50)	Прізвище клієнта
ManagerSurname	nvarchar(50)	Ім’я клієнта
Salary	int	Номер телефону клієнта
Experience	Datetime	Місце проживання клієнта
PositionID	int	Ідентифікатор посади (foreign key)

```

public class Client
{
    public int ManagerID { get; set; }
    [StringLength(50)]
    public string ManagerName { get; set; }
    [StringLength(50)]
    public string ManagerSurname { get; set; }
}
  
```

```

public int Salary { get; set; }
[DataType(DataType.Date)]
[Display(Name = " Experience ")]
public DateTime Experience { get; set; }
public virtual ICollection<Order> Orders { get; set; }
}

```

Таблиця 2.10 – Поля таблиці Position

Поле	Тип	Опис
PositionID	int	Ідентифікатор посади (primary key)
PositionName	nvarchar(50)	Назва посади

В результаті спроектована концептуальна схема бази даних для інформаційної системи автосалону буде представлена схемою на рис. 2.6. Створена концептуальна модель бази даних допоможе чітко визначити сутності, їх атрибути, а також взаємозв'язки між ними у вибраній сфері. Це спростить розуміння способу збереження та використання даних в інформаційній системі. Крім того, проєктована модель бази даних наглядно демонструє ключові компоненти бази даних та їх залежності, що дозволить виявити помилки та протиріччя в області застосування.

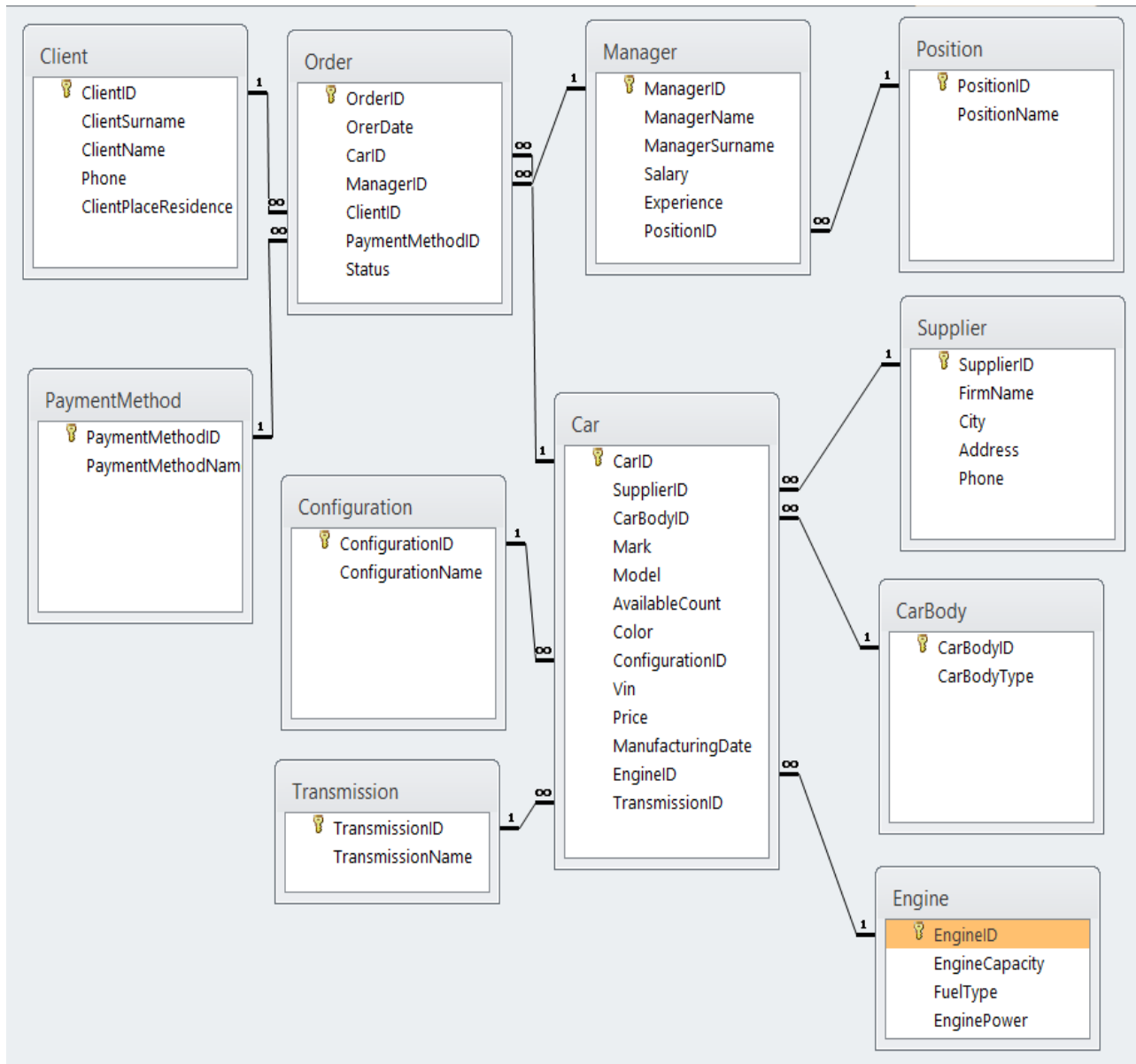


Рисунок 2.6 – Концептуальна схема бази даних інформаційної системи автосалону

Висновки до розділу 2

В результаті було проведено огляд та аналіз засобів та технологій, що були використанні для проєктування інформаційної системи автосалону. Таким чином було обрано такі засоби та технології як патерн проєктування MVC, ORM фреймворк Entity Framework, веб сервер IIS Express та середовище розробки програмного забезпечення Visual Studio 2022. Проведено дослідження предметної галузі та виділено сутності у базі даних інформаційної системи автосалону. Спроєктовано концептуальну модель бази даних для інформаційної системи автосалону.

Зм.	Арк.	№докум.	Підпис	Дата

3 РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ АВТОСАЛОНУ

3.1 Модель вимог та модель проєктування інформаційної системи автосалону

В моделі вимог визначаються завдання, функціональні та нефункціональні вимоги, які встановлюються до системи. Для проєктованої інформаційної системи автосалону використаємо діаграму прецедентів (Use case diagram) у рамках UML.

Діаграма прецедентів показує прецеденти (варіанти використання) та акторів, а також взаємозв'язки між ними. Ця діаграма використовується для моделювання динамічних аспектів системи. Вона дозволяє зобразити контекст системи, підсистеми, класи та вимоги які пред'являються до поведінки зазначених елементів.

У проєктованій системі беруть участь менеджер із продажу автомобілів, клієнт, директор та працівник бухгалтерії. Менеджер із продажів реалізує функції створення нового замовлення на купівлю автомобіля (також видалення та редагування), додавання у систему нового клієнта (також видалення та редагування); додавання у систему нового поставщика (також видалення та редагування), а також консультування клієнта.

Клієнту надаються функції по перегляду доступних для продажу автомобілів та участь у створенні замовлення.

Директор автосалону виконує функції консультування та перегляду складених договорів про продаж автомобілів.

Працівник бухгалтерії відповідальний за перегляд складених договорів про продаж автомобілів. Ця інформація буде використана для нарахування заробітної плати менеджерам із продажу.

Подемо множину дії менеджера із продажу автомобілів, клієнта, директора та працівника бухгалтерії у вигляді UML діаграма варіантів використання, яка подана на рис. 3.1.

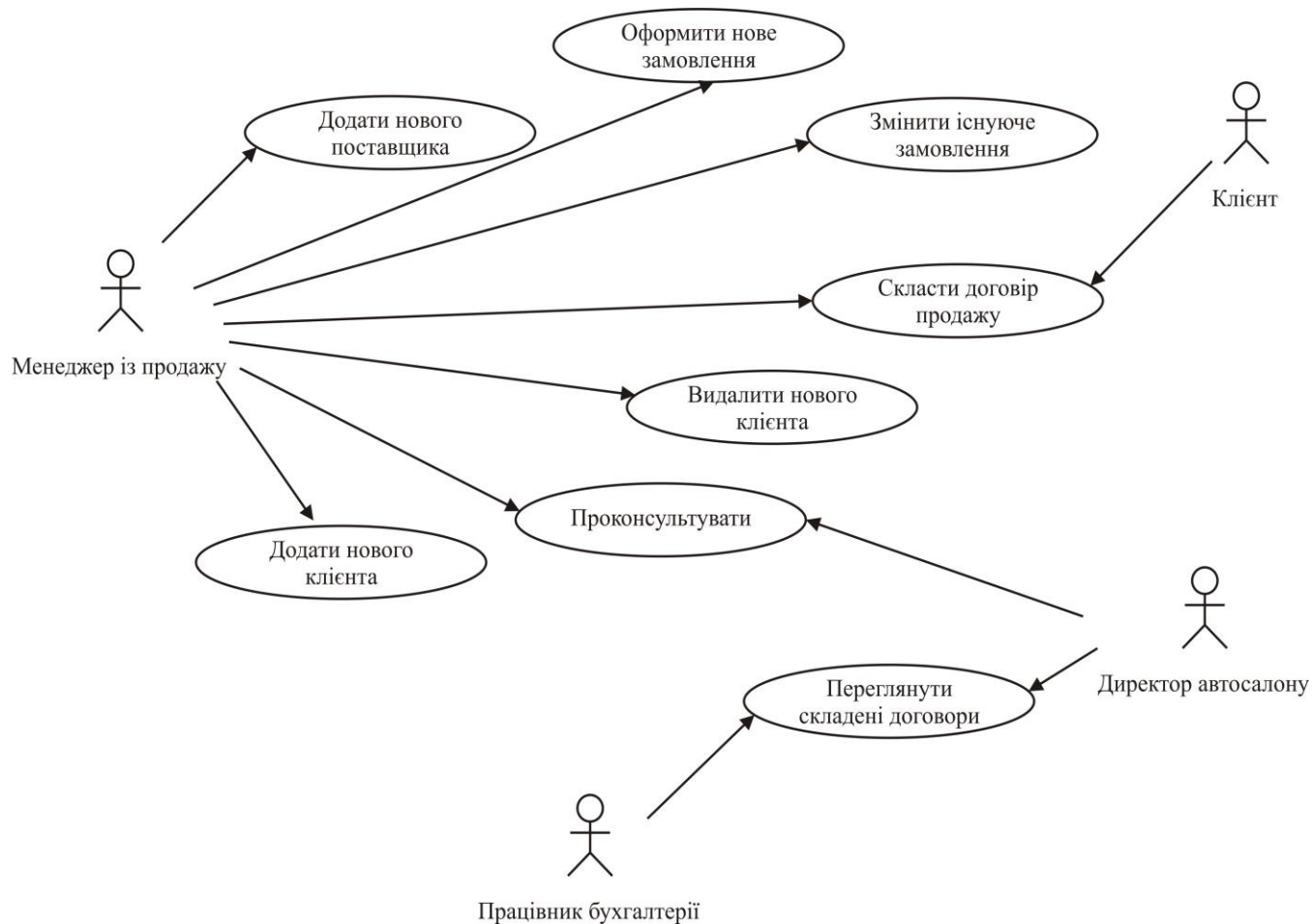


Рисунок 3.1 – UML діаграма варіантів використання

Модель проектування інформаційної системи відображає взаємодію процесів всередині системи, шляхи передачі даних, потоки даних та стан об'єктів. Для опису цієї моделі можна використати діаграми взаємодії, такі як діаграма послідовностей та діаграма кооперації.

Діаграми взаємодій відображають зв'язки між об'єктами та відносини між ними, включаючи повідомлення, якими об'єкти обмінюються. Діаграма послідовностей зосереджується на часовому порядку повідомлень, показуючи послідовність їх відправки та отримання. Діаграма кооперації, зі свого боку, акцентує увагу на структурній організації об'єктів, які взаємодіють, та на їх взаємодії через повідомлення.

Таким чином, за допомогою діаграм взаємодій можна якісно описати способи взаємодії та обміну даними між об'єктами системи, звертаючи увагу як на послідовність повідомлень, так і на структурні аспекти цієї взаємодії.

Зм.	Арк.	№докум.	Підпис	Дата

На рис. 3.2 подано діаграму послідовності, що відображає введення нового договору в систему обробки договорів інформаційної системи автосалону. Вона відповідає успішному варіанту перебігу подій. Існують п'ять об'єктів та один об'єкт – дійова особа (продавець).

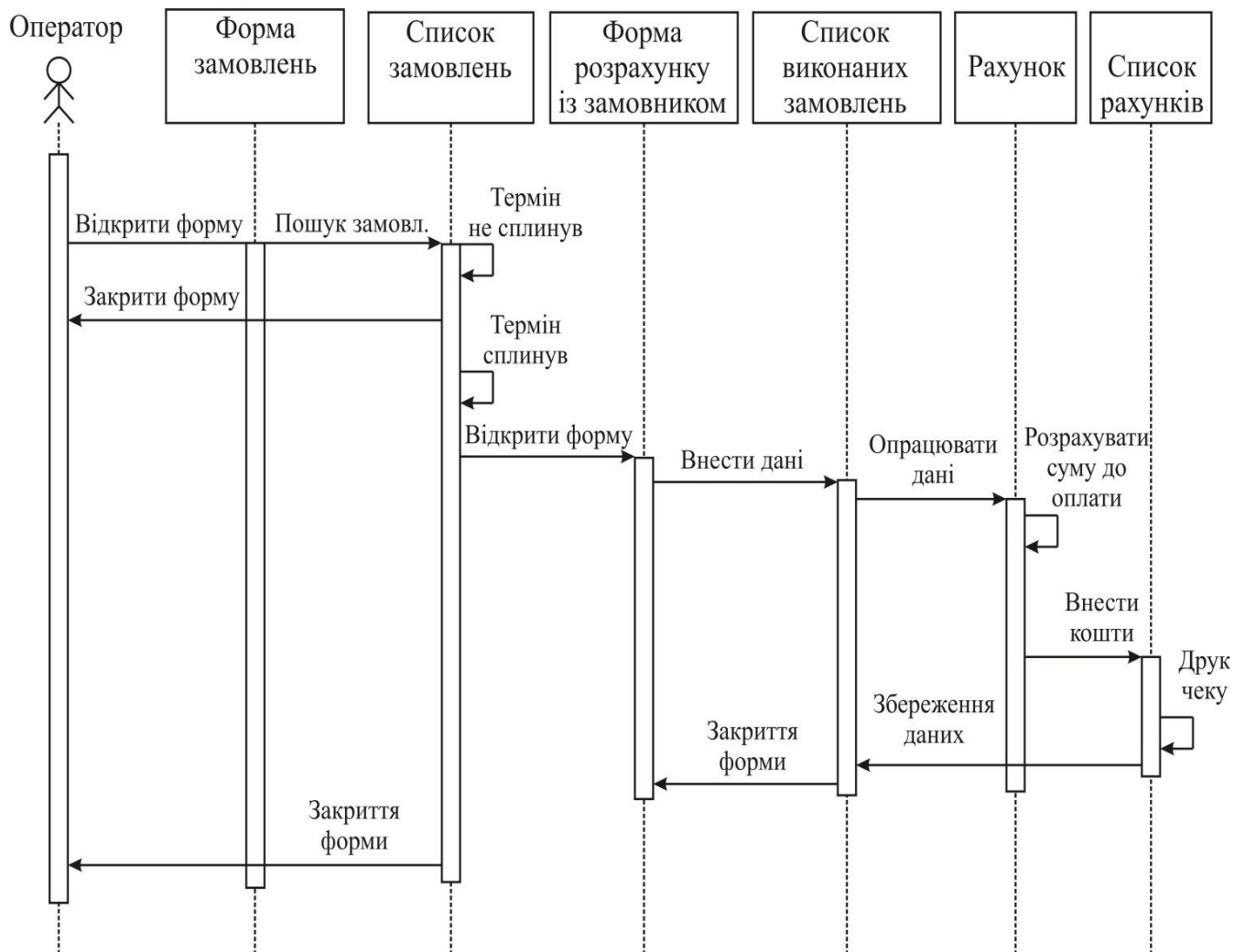


Рисунок 3.2 – UML діаграма послідовності виконання

Послідовність дій під час виконання цієї під-задачі, після входу в систему можна подати наступним чином:

- Створення нового договору.
- Відкриття форми.
- Введення інформації про замовника (ПІБ, № замовлення) та марка модель.

– Збереження замовлення.

На наступному етапі активізується об'єкт менеджер договорів, який відповідає за здійснення конкретного замовлення в інформаційній системі. Менеджер договорів зберігає створене замовлення та створює порожнє замовлення та введе інформацію про замовника (ПІБ, №замовлення) марку та модель.

З метою візуального відображення структури запропонованої інформаційної системи автосалону, можна скористатися UML діаграмою компонентів (рис. 3.3). Згідно з цією діаграмою, запропонована ІС складається з трьох основних компонентів: пристрою користувача (для відображення інформації та взаємодії із системою), який представлений веб-браузером, веб сервера та бази даних.

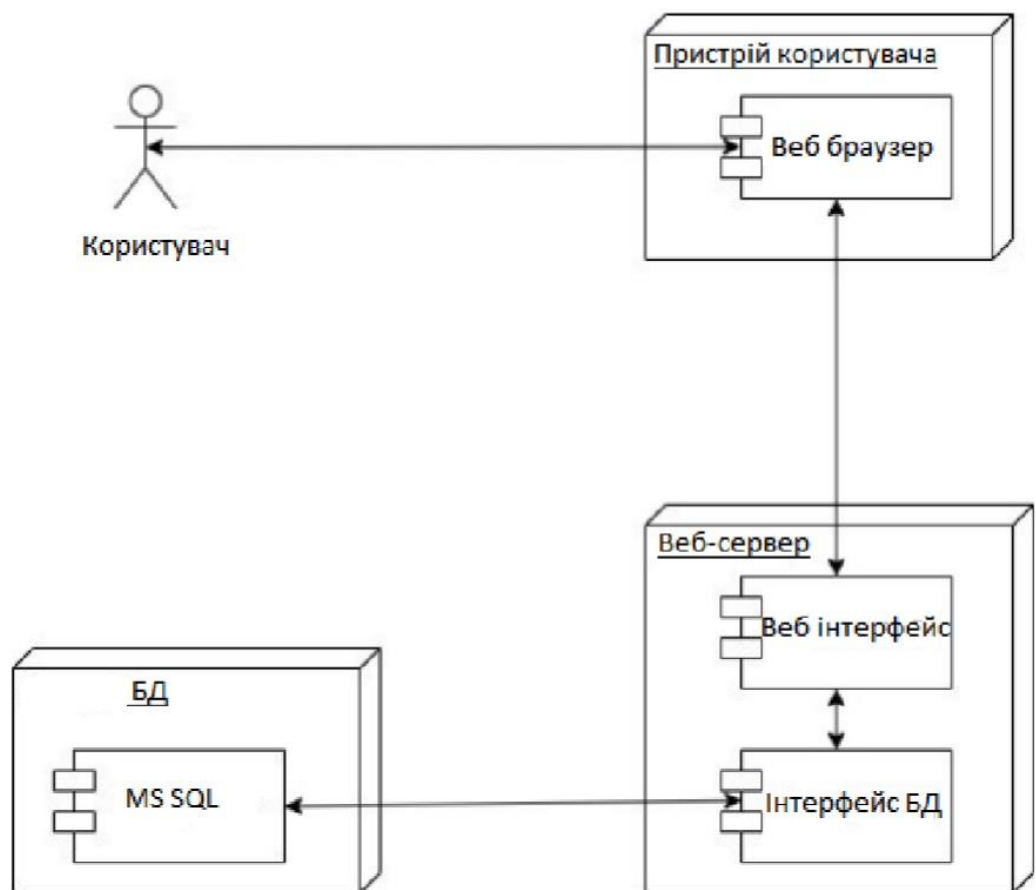


Рисунок 3.3 – UML-діаграма компонентів

3.2 Структура інформаційної системи автосалону

Інформаційна система автосалону є комплексом взаємопов'язаних компонентів, які допомагають управляти та оптимізувати різні аспекти діяльності автосалону. Вона включає в себе апаратне, програмне та організаційне забезпечення, що співпрацюють для збору, зберігання, обробки та передачі інформації, пов'язаної з продажем автомобілів. Визначимо структуру інформаційну систему автосалону у вигляді компонентів клієнтської та серверної частини, бази даних та API.

Клієнтська частина інформаційної системи автосалону складається з веб-сторінок, які користувачі бачать у своєму веб-браузері. Ця частина відповідає за збирання та відображення інформації, яка призначена для користувача.

Серверна частина інформаційної системи автосалону представляє собою програмне забезпечення, яке виконується на сервері і відповідає за зберігання, обробку та передачу даних між клієнтом та сервером. Вона включає у себе базу даних та веб сервер, які забезпечують зберігання даних та обробку запитів з боку клієнтської частини.

База даних у інформаційній системі автосалону використовується для зберігання даних та має наступні призначення:

– Зберігання інформації про автомобілі: База даних включає дані про доступні автомобілі, їх моделі, марки, технічні характеристики, ціни, наявність на складі та іншу важливу інформацію. Це дозволяє швидко отримати доступ до даних про автомобілі та ефективно керувати їх інвентаризацією.

– Клієнтська база даних: База даних містить інформацію про клієнтів, включаючи їх контактну інформацію, історію покупок, запити та додаткові дані. Це дозволяє автосалону зберігати і аналізувати дані про клієнтів, забезпечувати персоналізоване обслуговування та ефективно управляти взаємодією з ними.

– Управління продажами: База даних включає інформацію про продажі, включаючи деталі про угоди, статус замовлень, графіки доставки, оплату та інші

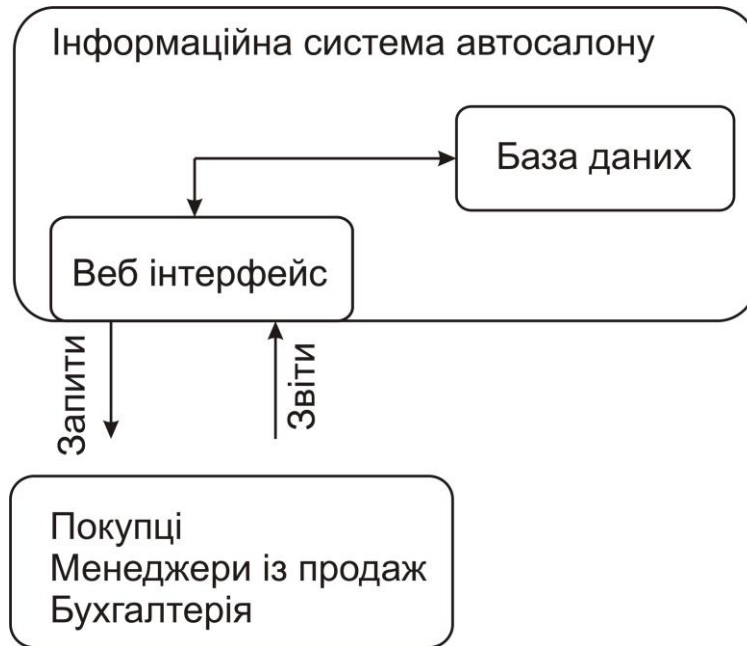


Рисунок 3.4 – Узагальнена структура інформаційної системи автосалону

Схема потоків даних спроектованої інформаційної системи автосалону подано на рис. 3.5.

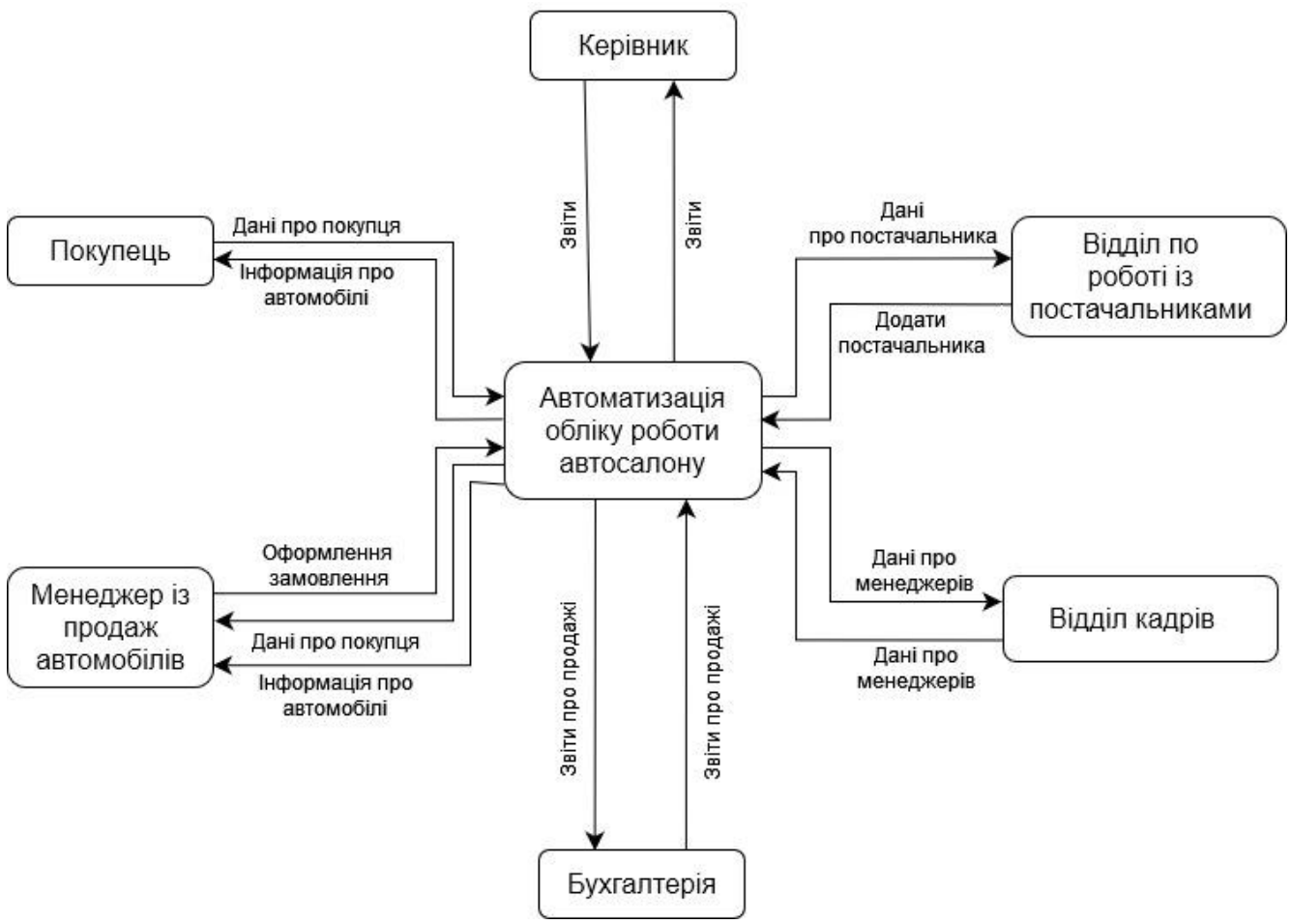


Рисунок 3.5 – Схема потоків даних

Зм.	Арк.	№докум.	Підпис	Дата



Рисунок 3.6 – Потоки даних інформаційної систем при продажу автомобіля

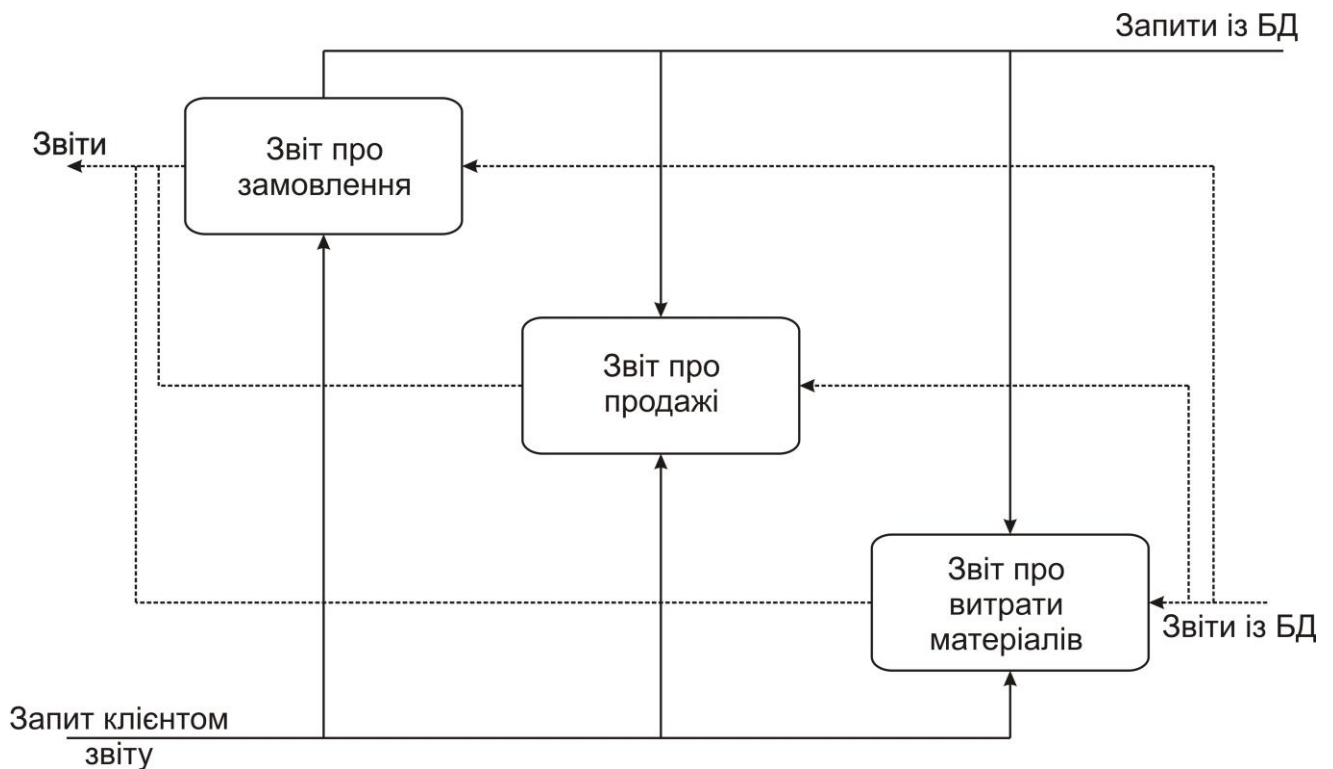


Рисунок 3.7 – Потоки даних для формування звітів

3.3 Реалізація інформаційної системи автосалону

Відповідно до запропонованої структури інформаційної системи, реалізація прототипу включала у себе створення програмного забезпечення у вигляді веб сайту та бази даних, що використовуватиметься для збереження даних про бізнес процеси функціонування автосалону. Таким чином реалізований прототип інформаційної системи автосалону характеризується таким функціоналом:

- авторизація для клієнтів та менеджерів автосалону;
- відображення інформації про доступні автомобілі та характеристики кожного автомобіля;
- Формування звіту про наявні автомобілі та їх характеристики.
- Додавання нового автомобіля та реєстрація клієнта.
- Видалення та редагування інформації про автомобілі.
- Створення договору про купівлю автомобіля.
- Формування звіту про клієнтів, що оформили купівлю автомобіля.
- Формування звіту про замовлення, які складені конкретним менеджером автосалону.
- Формування звіту про поставщиків автомобілів.
- Створення нового поставщика, а також редагування та видалення інформації про існуючих постачальників.

При реалізації прототипу інформаційної системи автосалону основна увага була приділена простоті та зручності дизайну, без розміщення надлишкової інформації.

При запуску веб сайту спочатку буде відображено форму авторизації (рис. 3.8). В системі передбачено чотири ролі: клієнт, менеджер, директор та працівник відділу бухгалтерії.



Log in

Login

Password

Рисунок 3.8 – Інтерфейсна форма форми авторизації

Після авторизації, клієнту буде доступно головна сторінка інформаційної системи (рис. 3.9).



Home Cars Clients Orders Managers Supplier About

Select car and enjoy



Show More>>>



Show More>>>



Show More>>>

@2023 - Car sales

Рисунок 3.9 – Інтерфейсне вікно головної сторінки

Зм.	Арк.	№докум.	Підпис	Дата

КВРІСТ. 200197.01.09.09 ПЗ

Арк.
52

Головне меню сторінки складається із підпунктів вибору автомобілів (Cars), оформлення замовлення (Orders), відображення інформації про поставщиків автомобілів (Supplier). При вході у систему користувачу на сторінці буде відображено список автомобілів, доступних для продажу. Біля кожного автомобіля є посилання, перейшовши за яким є можливість відкрити сторінку із додатковими характеристиками вибраного автомобіля (рис. 3.10). Для повернення на головну форму передбачено підпункт меню Home (дозволяє повернутись із будь-якої сторінки).

Інформація про авто

CarID	2
SupplierID	Renault Inc
CarBodyID	Універсал
Mark	Renault
Model	Express
Color	Синій
ConfigurationID	Zen
Vin	VF1KTD90E52297704
Price	900000 грн
Date	11.04.2023
EngineID	Дизель




Рисунок 3.10 – Інтерфейсне вікно відображення додаткових характеристик автомобіля

Передбачена можливість сформувати зведені дані про всі доступні для продажу автомобілі у вигляді звіту. Звіт складається із полів марка, модель, колір, ціна, потужність двигуна, тип палива та вид трансмісії (рис. 3.11).



Доступні для продажу автомобілі

Mark	Model	Color	Price	EngineCapacity	FuelType	TransmissionNarr
Renault	Magane	Білий	950 000,00 €	1,5	Дизель	Автоматична
Renault	Express	Сірий	900 000,00 €	2	Дизель	Ручна
Peugeot	308	Чорний	870 000,00 €	2	Дизель	Автоматична
Renault	Magane	Синій	980 000,00 €	1,4	Бензин	Автоматична
Renault	Magane	Білий	945 000,00 €	1,5	Дизель	Ручна
Renault	Magane	Чорний	890 000,00 €	1,5	Дизель	Роботизована
Peugeot	2008	Помаранчевий	1 200 000,00 €	2	Дизель	Автоматична
Renault	Magane	Синій	980 000,00 €	1,5	Дизель	Автоматична

Page 1 из 1

Рисунок 3.11 – Інтерфейсне вікно відображення доступних для продажу автомобілів

Також є можливість перегляну звіт про доступні для продажу автомобілі, інформація про які згрупована за типом палива (рис.3.12).



Фільтр за типом палива

Model	Mark	FuelType
Express	Renault	Дизель
Magane	Renault	Дизель
308	Peugeot	Дизель
Magane	Renault	Дизель
		4

Page 1 из 1

Тип палива

Дизель

Рисунок 3.12 – Інтерфейсне вікно фільтрації доступних автомобілів за типом палива

Зм.	Арк.	№докум.	Підпис	Дата
-----	------	---------	--------	------

КВРІСТ. 200197.01.09.09 ПЗ

Арк.
54

В інформаційній системі передбачено також роль менеджера із продажу автомобіля. Реалізований прототип інформаційної системи надає менеджеру із продажу автомобілів наступний функціонал:

- Створення нового замовлення на купівлю автомобіля.
- Додавання у систему нового клієнта.
- Фільтрації автомобілів по поставщиках.

Форма створення нового замовлення на купівлю автомобіля (рис. 3.13) передбачає введення інформації про ідентифікатор автомобіля, його марку та модель, ідентифікатор клієнта, ідентифікатор менеджера з продажу, спосіб оплати замовлення, а також статус замовлення. Статус замовлення передбачає два значення «завершено» та «у процесі». Після сплати всієї суми, статус замовлення маркується як «завершений».

Створення договору купівлі

Mark

Model

CarID

ClientID

ManagerID

Payment

Status

Рисунок 3.13 – Інтерфейсне вікно створення нового замовлення на купівлю автомобіля

Також менеджером із продажу заповнюються дані про клієнта, шляхом заповнення відповідної форми (рис. 3.14).

Додавання клієнта

ClientSurname


ClientName

Phone

PlaceResidence

Рисунок 3.14 – Інтерфейсне вікно додавання нового клієнта

Окрім роботи із клієнтами у менеджера є функція перегляду інформації та формування звіту про поставщиків автомобілів. Інтерфейсне вікно фільтрації автомобілів по поставщиках наведено на рис. 3.15.



Фільтрація авто по поставщиках


FirmName	City	Mark	Model
Renault Inc	Київ	Renault	Magane
Renault Inc	Київ	Renault	Express
Renault Inc	Київ	Renault	Magane
Renault Inc	Київ	Renault	Magane
Renault Inc	Київ	Peugeot	2008
Renault Inc	Київ	Renault	Magane

Company

Page 1 из 1

Рисунок 3.15 – Інтерфейсне вікно фільтрації автомобілів по поставщиках

Також окрім ролі клієнта та менеджера передбачено роль працівника відділу бухгалтерії (або керуючого автосалоном). Основною функцією даної ролі, є надання керівнику можливості моніторингу процесу складання договорів про продаж автомобілів менеджерами із продажу. Інтерфейсне відображення звіту про замовлення, які складені менеджером наведено на рис. 3.16.



Замовлення складені менеджером

ManagerSurname	OrderDate	OrderID	Status	Mark	Model
Романюк	02.05.2023	4 <input type="checkbox"/>		Peugeot	308
Романюк	05.05.2023	5 <input type="checkbox"/>		Renault	Magane

Page 1 из 1

Manager Name

Рисунок 3.16 – Інтерфейсне відображення звіту про замовлення, які складені менеджером

Таким чином запропоновано прототип інформаційної системи автосалону, що представлений у формі веб сайату. Запропонована інформаційна система має наступні можливості: авторизація для клієнтів та менеджерів автосалону; відображення інформації про доступні автомобілі та характеристики кожного автомобіля; формування звіту про наявні автомобілі та їх характеристики; додавання нового автомобіля та реєстрація клієнта; створення договору про купівлю автомобіля; формування звіту про клієнтів, що оформили купівлю автомобіля; формування звіту про замовлення, які складені конкретним менеджером автосалону; формування звіту про поставщиків автомобілів. Реалізована інформаційна система автосалону має досить простий та зрозумілий інтерфейс, що дозволяє досить швидко зорієнтуватись при роботі

із нею. В подальшому дану інформаційну систему можна розширити, додавши можливості, наприклад, по оренді автомобіля, продажу у кредит, технічного обслуговування, тощо.

Висновки за розділом 3

Таким чином представлено модель вимог та модель проектування інформаційної системи автосалону у вигляді UML діаграм. Представлено структуру інформаційної системи автосалону. Запропоновано прототип інформаційної системи автосалону, який володіє наступними функціональними можливостями: а) авторизація для клієнтів та менеджерів автосалону; відображення інформації про доступні автомобілі та характеристики кожного автомобіля; формування звіту про наявні автомобілі та їх характеристики; додавання нового автомобіля та реєстрація клієнта; видалення та редагування інформації про автомобілі; створення договору про купівлю автомобіля; формування звіту про клієнтів, що оформили купівлю автомобіля; формування звіту про замовлення, які складені конкретним менеджером автосалону; формування звіту про поставщиків автомобілів; створення нового поставщика, а також редагування та видалення інформації про існуючих постачальників.

					КВРІСТ. 200197.01.09.09 ПЗ	Арк.
						58
Зм.	Арк.	№докум.	Підпис	Дата		

відкритими, що означає, що дозволило швидко й якісно розробити прототип інформаційної системи для автосалону.

В третьому розділі представлено модель вимог та модель проектування інформаційної системи автосалону у вигляді UML діаграм. Представлено структуру інформаційної системи автосалону. Виділено вхідні та вихідні потоки даних. Запропоновано прототип інформаційної системи автосалону, який володіє наступними функціональними можливостями: а авторизація для клієнтів та менеджерів автосалону; відображення інформації про доступні автомобілі та характеристики кожного автомобіля; формування звіту про наявні автомобілі та їх характеристики; додавання нового автомобіля та реєстрація клієнта; видалення та редагування інформації про автомобілі; створення договору про купівлю автомобіля; формування звіту про клієнтів, що оформили купівлю автомобіля; формування звіту про замовлення, які складені конкретним менеджером автосалону; формування звіту про поставщиків автомобілів; створення нового поставщика, а також редагування та видалення інформації про існуючих постачальників.

					КВРІСТ. 200197.01.09.09 ПЗ	Арк.
						60
Зм.	Арк.	№докум.	Підпис	Дата		

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Навчальні ресурси СумДУ, Проектування інформаційних систем, URL: https://elearning.sumdu.edu.ua/free_content/lectured:de1c9452f2a161439391120eef364dd8ce4d8e5e/20160217112601/165292/index.html.
2. Проектування інформаційних систем: Загальні питання теорії проектування ІС (конспект лекцій) [Електронний ресурс]: навч. посіб. для студ. спеціальності 122 «Комп'ютерні науки» / КПІ ім. Ігоря Сікорського; уклад.: О. С. Коваленко, Л. М. Добровська, Київ : КПІ ім. Ігоря Сікорського, 2020, 192с.
3. Якість та тестування інформаційних систем. Навчальний посібник для самостійної роботи студентів вищих навчальних закладів. Київ: ННІТ ДУТ, 2020. –128 с.
4. Підручники, Проектування інформаційних систем. CASE – технології, URL : https://pidru4niki.com/18580318/informatika/proektuvannya_informatsiynih_sistem_case_tehnologiyi
5. Cybersecurity and Secure Information Systems. Challenges and Solutions in Smart Environments, *Springer Cham*, 2019, p. 314.
6. Єсін В.І. Безпека інформаційних систем і технологій : навчальний посібник, Х.: ХНУ імені В. Н. Каразіна, 2013, 632 с. ISBN 978-966-623-927-6.
7. Chapple M. CISSP Certified Information Systems Security Professional, *Sybex*, 2018, 1616 p.
8. Kim D. Fundamentals of Information Systems Security 3rd Edition, *Jones & Bartlett Learning*, 2016, 571 p.
9. Mayer-Schönberger V. Big Data: A Revolution That Will Transform How We Live, Work, and Think, *Eamon Dolan.Mariner Books*, 2014, 272 p.
10. Antoniou J. Quality of Experience and Learning in Information Systems, *Springer Cham*, 2019, 110 p.
11. Jena A. K. Automated Software Testing: Foundations, Applications and Challenges (Services and Business Process Reengineering), *Springer*; 1st ed., 2020. – 179 p.

					КВРІСТ. 200197.01.09.09 ПЗ	Арк. 61
Зм.	Арк.	№докум.	Підпис	Дата		

12. Проектування інформаційних систем: Загальні питання теорії проектування ІС (конспект лекцій) [Електронний ресурс]: навч. посіб. для студ. спеціальності 122 «Комп'ютерні науки» / КПП ім. Ігоря Сікорського; уклад.: О. С. Коваленко, Л. М. Добровська, Київ : КПП ім. Ігоря Сікорського, 2020, 192с

13. Мінухін С. В. Методи і моделі проектування на основі сучасних CASE– засобів. Навчальний посібник, Харків: ХНЕУ, 2008, 272 с.

14. Проектування інформаційних систем: навчальний посібник, Черкаси: Черкаський національний університет ім. Б. Хмельницького, 2017, 434 с.

15. Методичні вказівки для студентів спеціальності 8.05010102 «Інформаційні технології проектування» ОКР Магістр, Таврійський державний агротехнологічний університет, 2013, 21 с.

16. Ушакова І. О. Основи системного аналізу об'єктів та процесів комп'ютеризації : навчальний посібник. Ч. 2, Х. : Вид. ХНЕУ, 2008, 324 с.

17. Ушакова І. О. Практикум з навчальної дисципліни Основи системного аналізу об'єктів і процесів комп'ютеризації: навчальний посіб-ник., Х. : Вид. ХНЕУ, 2010. – 344 с.

18. Muhammad K., Maryati M.. Lean IT Transformation Plan for Information Systems Development. *International Journal of Advanced Computer Science and Applications*. 2020, Vol 11 (8).

19. Mikalsen, M., Moe, Nils, B., Wong, Sut I., Stray, Viktoria. Agile Information System Development Organizations Transforming to Large-Scale Collaboration. *In proceedings of the Forty-Second International Conference on Information Systems (ICIS 2021)*, Austin, USA. 2021.

20. Isyaku. I. Iterative and Incremental Development Analysis Study of Vocational Career Information Systems. *International Journal of Software Engineering & Applications*. 11. 2020, pp. 13-24. doi: 10.5121/ijsea.2020.11502.

21. Chan Ою. SQL: Learn SQL (using MySQL) in One Day and Learn It Well. SQL for Beginners with Hands-on Project, 2018, 166р.

22. Мулеса О.Ю. Інформаційні системи та реляційні бази даних. Навч.посібник, Електронне видання, 2018, 118 с

					КВРІСТ. 200197.01.09.09 ПЗ	Арк. 62
Зм.	Арк.	№докум.	Підпис	Дата		

23. Корнієнко С.К. Проектування інформаційного забезпечення автоматизованих систем. Навч. Посібник, Запоріжжя: ЗНТУ, 2015, 210 с.
24. Карпуша В.Д., Панченко Б.Є. Моделювання та проектування реляційних баз даних : навч. посіб, Суми : СДУ, 2010.
25. Сенів М. М., Яковина В. С. Безпека програм та даних : навч. посіб. Львів : Видавництво Львівської політехніки, 2015.
26. Завадський І.О. Основи баз даних: [Навч. посіб.], К. Видавець І.О. Завадський, 2011, 192 с.
27. Балик Н., Мандзюк В. Базы даних MySQL: Навчальний посібник. – Тернопіль: Навчальна книга, Богдан, 2010, 160 с.
28. Гайдаржи В.І., Дацюк О.А. Основи проектування та використання баз даних : Навч. посібник Київ : Політехніка, 2004, 166с.
29. Адміністрування баз даними // Режим доступу: <http://firebirdsql.org/manual/ru/migration-mssql-db-admin-ru.html>.
30. Ben Forta. SQL in 10 Minutes a Day, Sams Teach Yourself: *Sams Publishing*; 5th edition, 18 Aug. 2020, 256p.
31. Гайдаржи В., Ізварін І. Базы даних в інформаційних системах: Навчальний посібник. – Тернопіль: Навчальна книга, 2018, 418 с.
32. Костенко О. Б. Організація баз даних та знань : конспект лекцій (для студентів денної та заочної форм навчання першого (бакалаврського) рівня вищої освіти за спеціальністю 126 – Інформаційні системи та технології), Харків. нац. ун-т міськ. госп-ва ім. О. М. Бекетова, Харків : ХНУМГ ім. О. М. Бекетова, 2021, 92 с.
33. Берко А. Ю. Системи баз даних та знань : підручник, Львів : Магнолія, 2015. Книга 1, 440 с.
34. Павлиш В. А. Основи інформаційних технологій і систем : підручник / В. А. Павлиш, Л. К. Гліненко, Н. Б. Шаховська, Львів : Львівська політехніка, 2018, 619 с.
35. Шаховська Н. Б. Системи штучного інтелекту : навч. посібник , Львів : Львівська політехніка, 2018. – 391 с.

36. Костріков С. В. Інформаційні технології в БД. Навчально-методичний посібник, Харків : *PBB ХНУ*, 2015, 56 с.
37. Цеслів О. В. Технологія проєктування та адміністрування баз даних і сховищ даних : навч. посібник, Нац. техн. ун-т України Київ. політехн. ін-т ім. Ігоря Сікорського, Київ : КПІ ім. І. Сікорського : Політехніка, 2017, 281 с.
38. Ушакова І. О. Практикум з навчальної дисципліни Основи системного аналізу об'єктів і процесів комп'ютеризації: навчальний посібник, Х. : Вид. ХНЕУ, 2010. – 344 с.
39. Ушакова І. О. Основи системного аналізу об'єктів та процесів комп'ютеризації : навчальний посібник. Ч. 2, Х. : Вид. ХНЕУ, 2008. – 324 с.
40. Nachenberg, C. Understanding heuristics: Symantec's bloodhound technology, *Symantec White Paper Series*, 1998, Vol. 34, P. 17.
41. Perriot F. Principles and practise of xraying, *The 14-th Virus Bulletin International Conference*, September 12, 2004: Proceedings, 2004, Pp. 51-56.
42. Rotshtein A. Identification of a nonlinear dependence by a fuzzy knowledgebase in the case of a fuzzy training set, *Cybernetics and Systems Analysis*, 2006, Vol. 42, Issue 2, Pp.176-182.
43. Isyaku. I. Iterative and Incremental Development Analysis Study of Vocational Career Information Systems. *International Journal of Software Engineering & Applications*. 11. 2020, pp. 13-24. doi: 10.5121/ijsea.2020.11502.
44. Jena A. K. Automated Software Testing: Foundations, Applications and Challenges (Services and Business Process Reengineering), *Springer*; 1st ed., 2020, 179 p.
45. Проєктування інформаційних систем: Загальні питання теорії проєктування ІС (конспект лекцій) [Електронний ресурс]: навч. посіб. для студ. спеціальності 122 «Комп'ютерні науки» / КПІ ім. Ігоря Сікорського; уклад.: О. С. Коваленко, Л. М. Добровська, Київ : КПІ ім. Ігоря Сікорського, 2020, 192с
46. Мінухін С. В. Методи і моделі проєктування на основі сучасних CASE– засобів. Навчальний посібник, Харків: Вид. ХНЕУ, 2008, 272 с.

47. Falliere N. Sality: Story of a Peer-to-Peer Viral Network, *Symantec Labs*, 2011, 21 p.
48. Костріков С. В. Інформаційні технології в БД. Навчально-методичний посібник, Харків : *PBB ХНУ*, 2015, 56 с.
49. Chan Ою. SQL: Learn SQL (using MySQL) in One Day and Learn It Well. SQL for Beginners with Hands-on Project, 2018, 166p.
50. Мулеса О.Ю. Інформаційні системи та реляційні бази даних. Навч.посібник, Електронне видання, 2018, 118 с
51. Entity Framework URL: Core<https://learn.microsoft.com/en-us/ef/core/>
52. Zetter K. Countdown to Zero Day: Stuxnet and the Launch of the World's First Digital Weapon, *Broadway Books*, 2015, 448 p.
53. Solomon M.G. Fundamentals of Communications and Networking, *Jones & Bartlett Learning*, 2014, 512 p.
54. Azad S. Practical Cryptography: Algorithms and Implementations Using C++ 1st Edition, *Auerbach Publications*, 2014, 365 p.
55. Марченко А.В. Проектування інформаційних систем, *Держваний університет телекомунікацій*, 2016, 89 с.
56. Павлиш В. А. Основи інформаційних технологій і систем : підручник / В. А. Павлиш, Л. К. Гліненко, Н. Б. Шаховська, Львів : Львівська політехніка, 2018, 619 с.
57. Chapple M. CISSP Certified Information Systems Security Professional, *Sybex*, 2018, 1616 p.
58. Lucidchart What is an Entity Relationship Diagram (ERD), URL: <https://www.lucidchart.com/pages/er-diagrams>
59. Jena A. K. Automated Software Testing: Foundations, Applications and Challenges (Services and Business Process Reengineering), *Springer*; 1st ed., 2020, 179 p.
60. Корнієнко С.К. Проектування інформаційного забезпечення автоматизованих систем. Навч. Посібник., Запоріжжя: *ЗНТУ*, 2015, 210 с.

ДОДАТОК А

Копія креслення «Екранні форми»

Інтерфейсна форма форми авторизації

Інтерфейсне вікно головної сторінки

Інтерфейсне вікно відображення додаткових характеристик автомобіля

Інтерфейсне вікно відображення доступних для продажу автомобілів

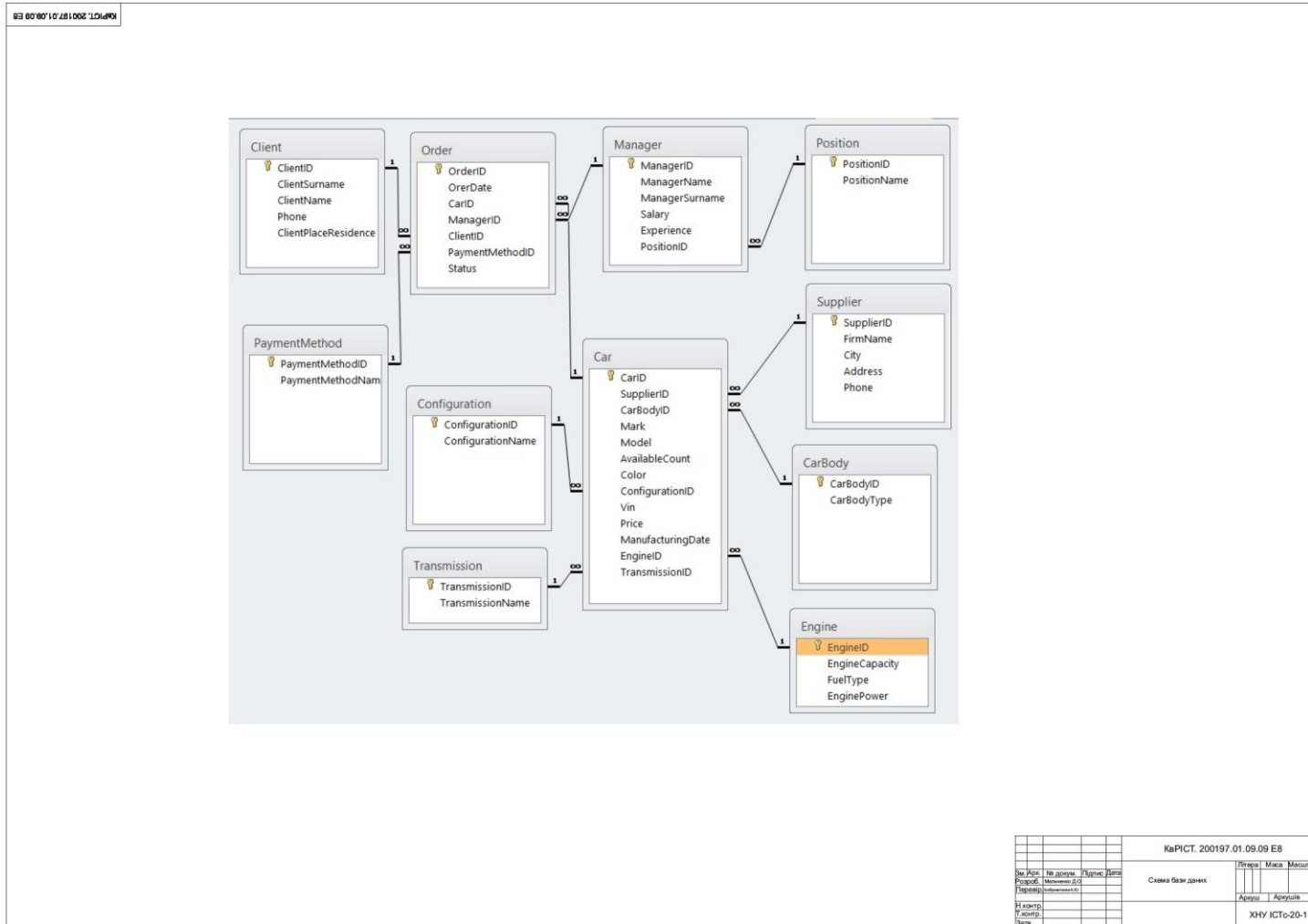
Інтерфейсне вікно фільтрації доступних автомобілів за типом палива

Інтерфейсне вікно створення нового замовлення на купівлю автомобіля

Карт. 200197.01.09.09 Е8			
№ Акт. № докум. Підпис Дата	Креслення виготовлено за замовленням	Ім'я	Місце
Розроб. Інженер Д.С. Білоус		Архит.	Архит.
І.в.контр.		ХНУ ІСТО-20-1	
Дата:			

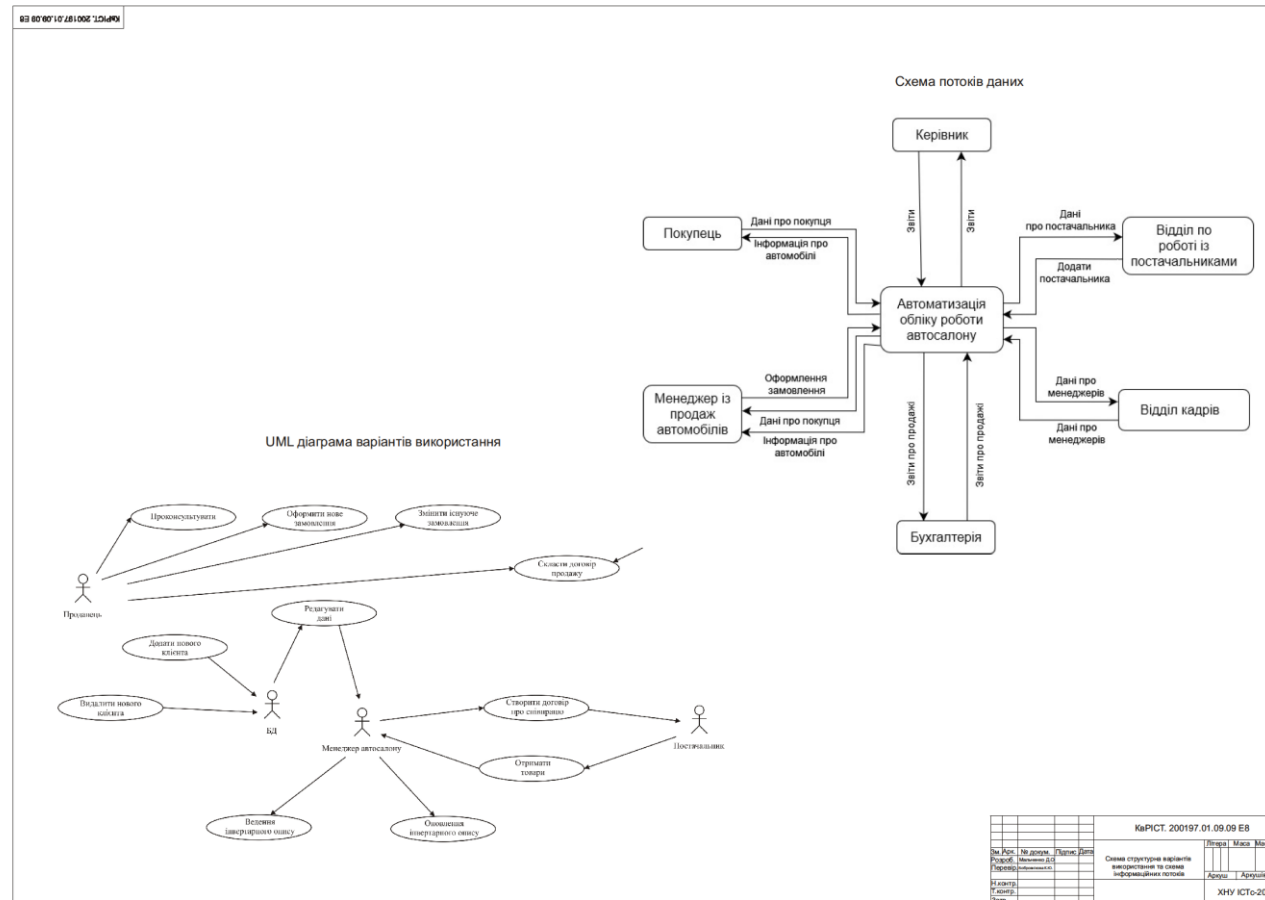
ДОДАТОК Б

Копія креслення «Схема бази даних»



ДОДАТОК В

Копія креслення «Схема структурна варіантів використання та схема інформаційних потоків»



РІШЕННЯ ЕКСПЕРТНОЇ КОМІСІЇ
КАФЕДРИ КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ
ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

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

Назва: Інформаційна система автосалону

Автор: Мальченко Денис Олександрович

Спеціальність: 126 – Інформаційні системи та технології

Освітня програма: освітньо-професійна

Науковий керівник: Бобровнікова Кіра Юліївна, к.т.н, доцент

Після аналізу звіту подібності зроблено такий висновок:

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

Підтвердження:

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

- 1) запозичення розміщені в розділі аналізу існуючих аналогів та відомих рішень, які не описують безпосередньо авторське дослідження і не стосуються результатів роботи;
- 2) усі запозичення фрагментарні, або мають належним чином оформленні посилання;
- 3) частина списку використаних джерел було визначено як плагіат.

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ідентичності/схожості, складає 25,5% і адресується до 402 першоджерела, що, з урахуванням наведених обґрунтувань, відповідає характеру наукового дослідження і свідчить на користь кваліфікаційної роботи.

Керівник роботи

Гарант ОП

Завідувач кафедри КІС



К.Ю. Бобровнікова

Є.Г. Гнатчук

Т. О. Говорущенко

Завідувачу кафедри КІС

д-р.техн.наук, проф. Говорушенко Т. О.


ПІБ здобувача вищої освіти

ФІТ, 3 курсу, групи ІСТс-20-1

ЗАЯВА

З правилами чинного Положення «Про дотримання академічної доброчесності в Хмельницькому національному університеті» від 26.09.2020 (зі змінами від 26.11.2020), згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування заходів дисциплінарної та академічної відповідальності, ознайомлений (а). Про використання програмно-технічних засобів для перевірки кваліфікаційних робіт здобувачів вищої освіти на плагіатоповіщений (а) та надаю свою згоду на обробку та збереження університетом моєї роботи в інституційному репозитарії університету.

Також надаю університету право на передачу моєї роботи для обробки та збереження в базах даних програмно-технічних засобів (Unicheck та Anti-Plagiarism) та використання роботи для виявлення плагіату в інших роботах, які перевіряються програмно-технічними засобами та користувачами, що мають доступ до цих програмно-технічних засобів, виключно в обмежених цілях для виявлення плагіату в текстах робіт.

Робота для перевірки університетом надається в друкованому та електронному варіанті. Електронна версія моєї роботи збігається (ідентична) з друкованою.

20.05.2023

дата



підпис

Ім'я користувача:
Кафедра КІ

ID перевірки:
1015065790

Дата перевірки:
13.05.2023 21:20:34 EEST

Тип перевірки:
Doc vs Internet + Library

Дата звіту:
13.05.2023 21:28:55 EEST

ID користувача:
100005591

Назва документа: Мальченко_Інформаційна система автосалону

Кількість сторінок: 69 Кількість слів: 10507 Кількість символів: 82765 Розмір файлу: 5.67 MB ID файлу: 1014751793

1510 слів позначені як "вилучені" та не враховуються у підрахунку слів

Виявлено модифікації тексту (можуть впливати на відсоток схожості)

25.5% Схожість

Найбільша схожість: 8.64% з Інтернет-джерелом (<http://elar.khmn.edu.ua/jspui/bitstream/123456789/11997/1/%d0%9e>).

24% Джерела з Інтернету 326 Сторінка 71

11.2% Джерела з Бібліотеки 56 Сторінка 74

1.28% Цитат

Цитати 9 Сторінка 75

Не знайдено жодних посилань

0% Вилучень

Немає вилучених джерел

Модифікації

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

Підозріле форматування 15 сторінок

Anti-Plagiarism v-15.257

Максимальне співпадіння з одним документом 5.0%

Словники перевірки: en_US, ru_RU, ua_UA. Помилки в документах: 10%

ID: 113329 Назва: БКР Інформаційна система автосалону Додано в БД: 2023-05-13 Автора: Д.О. Мальченко Керівники: К.Ю. Бобровнікова Консультанти: Опоненти:	Документ		Сумарний збіг по Базі Даних	
	Символи	Лексеми	Символи	Лексеми
	135439	586	9379 (7%)	78 (13%)

Джерело плагіату

ID	Опис	Наявність плагіату в документі	
		Символи	Лексеми
112607	Назва: Інформаційна web-система для IT школи Додано в БД: 2023-04-26 Автора: К.А. Костенко Керівники: К.Ю. Бобровнікова Консультанти: Опоненти:	6885 (5.0%)	92 (16.0%)

РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Дипломник: Мальченко Денис Олександрович

Тема: Інформаційна система автосалону

Спеціальність: 126 «Інформаційні системи та технології»

Обсяг дипломної роботи:

Кількість листів креслень 3; кількість сторінок записки 55

1. Короткий зміст роботи та прийнятих рішень У роботі запропоновано інформаційну систему автосалону

2. Висновок про відповідність роботи дипломному завданню _____
Дипломний проект відповідає виданому завданню

3. Характеристика виконання кожного розділу, ступінь використання останніх досягнень науки і техніки і передових методів роботи: У першому розділі проведено аналіз підходів до проектування інформаційних систем. У другому розділі проведено вибір засобів реалізації та спроектовано концептуальну модель бази даних. У третьому розділі проведено реалізацію прототипу інформаційної системи автосалону

4. Позитивні сторони роботи: Запропоновано концептуальну модель бази даних автосалону та спроектовано інформаційну систему автосалону.

5. Негативні сторони роботи: Запропонований прототип інформаційної системи автосалону володіє досить обмеженим функціоналом.

6. Оцінка графічного оформлення та пояснювальної записки роботи:

пояснювальна записка та листи креслення виконані згідно діючих вимог

7. Відгук про роботу в цілому: В загальному робота виконана на задовільному рівні.

8. Інші зауваження: -

9. Оцінка дипломної роботи:

Розглянувши позитивні та негативні сторони представленої дипломної роботи вважаю, що робота заслуговує оцінки «задовільно» 3,0 (Е)

Рецензент (прізвище, ім'я, по батькові, посада, місце роботи)

Радельчук Таліна Іванівна, доцент кафедри інженерії програмного забезпечення ХНУ

“26” 05 2023р.

