

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет водного господарства та природокористування
Навчально-науковий інститут кібернетики інформаційних технологій та
інженерії
Кафедра комп'ютерних технологій та економічної кібернетики

Допущено до захисту:

Завідувач кафедри
комп'ютерних технологій та
економічної кібернетики
д. е. н., проф. П. М. Грицюк

«_____» _____ 2024 р.

**КВАЛІФІКАЦІЙНА РОБОТА
НА ЗДОБУТТЯ ОСВІТНЬО-КВАЛІФІКАЦІЙНОГО РІВНЯ
«БАКАЛАВР»
«Розробка програмного забезпечення із застосуванням сучасних
технологій програмування»**

Виконав:

здобувач вищої освіти за ОПП
126 «Інформаційні системи та
технології»

Іванюк Дмитро Борисович

Керівник:

Старший викладач Шевченко І.М.

Рецензент:

Рівне – 2024

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет водного господарства
та природокористування
Навчально-науковий інститут кібернетики інформаційних технологій та
інженерії
Кафедра комп'ютерних технологій та економічної кібернетики
Освітньо-кваліфікаційний рівень - бакалавр
Освітньо-професійна програма «Інформаційні системи та технології»
Спеціальність 126 «Інформаційні системи та технології»

ЗАТВЕРДЖУЮ
Завідувач кафедри
комп'ютерних технологій та
економічної кібернетики
д.е.н., професор П.М. Грицюк

“ ___ ” _____ 2024 року

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ
Іванюк Дмитро Борисович

1. Тема роботи «Розробка програмного забезпечення із застосуванням
сучасних технологій програмування»

керівник роботи: старший викладач Шевченко І.М.

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджена наказом по університету від С№ _____ від _____ 2024 р.

2. Термін здачі студентом закінченої роботи 18.06.2024 року

3. Вихідні дані до роботи Посібники, наукові статті, інтернет джерела
за темою роботи.

4. Зміст розрахунково-пояснювальної записки (перелік питань, що їх належить
розробити)

АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ПОСТАНОВКА ЗАДАЧІ.

*Загальна характеристика сфери автопослуг. Суть послуг таксі, пов'язані
процеси та послуги . Огляд існуючих аналогів служб таксі. Сучасні технології
програмування.*

ПРОЕКТУВАННЯ ІНФОРМАЦІЙНО-ПОШУКОВОЇ СИСТЕМИ.

Опис предметної області. Програмна реалізація. Структура бази даних ІС

«Taxi».

5. Перелік графічного матеріалу

Презентація за матеріалами бакалаврської роботи.

1. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		Завдання видав	Завдання прийняв
I II	Шевченко І.М.		

Дата видачі завдання 20.02.2024р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів бакалаврської роботи	Термін виконання етапів (роботи)	Примітка
1	Робота над Розділом 1.	20.02.2024-09.04.2024	
2	Робота над Розділом 2.	10.04.2024-7.06.2024	
3	Підготовка пояснювальної записки	8.06.2024-14.06.2024	
4	Написання вступу, висновків, реферату	16.06.2024	
5	Попередній захист роботи	17.06.2024	
6	Підготовка презентації роботи	18.06.2024	
7	Відгук керівника, рецензування роботи, перевірка на плагіат	20.06.2024	
8	Допуск до захисту	21.06.2024	

Студент _____ ..Іванюк Д.Б.
(підпис) (прізвище і ініціали)

Керівник кваліфікаційної роботи
_____ Шевченко І.М.
(підпис) (прізвище і ініціали)

ЗМІСТ

ВСТУП	6
1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ПОСТАНОВКА ЗАДАЧІ	8
1.1 Загальна характеристика сфери автопослуг	8
1.2 Суть послуг таксі, пов'язані процеси та послуги	12
1.3 Огляд існуючих аналогів служб таксі	18
1.4 Сучасні технології програмування	24
2 РОЗДІЛ ПРОЕКТУВАННЯ ІНФОРМАЦІЙНО-ПОШУКОВОЇ СИСТЕМИ	31
2.1 Опис предметної області	31
2.2 Програмна реалізація	38
2.3 Структура бази даних ІС «Taxi»	45
ВИСНОВКИ	53
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ	56
ДОДАТКИ	58

РЕФЕРАТ

Кваліфікаційна робота бакалавра: 63с., 36 рис., 3 табл., 16 літературних джерел, 5 додатків.

Об'єкт розробки: інформаційна система автоматизації обліку діяльності замовлень автопослуг приватної фірми.

Мета кваліфікаційної роботи: підвищення ефективності роботи з обліку діяльності транспортного підприємства за рахунок розробки та використання відповідної інформаційної системи.

У вступі розглядається аналіз та сучасний стан проблеми, конкретизується мета кваліфікаційної роботи та галузь її застосування, наведено обґрунтування актуальності теми та уточнюється постановка завдання.

У першому розділі проведено аналіз предметної області, визначено актуальність завдання та призначення розробки, розроблена постановка завдання, задані вимоги до програмної реалізації, технологій та програмних засобів.

У другому розділі виконано аналіз існуючих рішень, обрано вибір платформи для розробки, виконано проектування і розробка програми, наведено опис алгоритму і структури функціонування підсистеми, визначені вхідні і вихідні дані, наведені характеристики складу параметрів технічних засобів, описаний виклик та завантаження застосунку, описана робота програми.

Практичне значення полягає в автоматизації процесу обліку діяльності замовлень автопослуг приватної фірми, збору інформації у загальну базу даних та її систематизацію. Актуальність підсистеми маршрутизації запитів визначається великим попитом на подібні розробки.

Ключові слова: ІНФОРМАЦІЙНА СИСТЕМА, ТРАНСПОРТНА КОМПАНІЯ, MYSQL, БАЗА ДАНИХ

ВСТУП

Актуальність теми дослідження

В сучасному світі, де темп життя постійно зростає, все більшого значення набувають послуги, які економлять час та зусилля. Одним із прикладів таких послуг є замовлення автопослуг. Інформаційно-пошукові системи (ІПС) замовлень автопослуг дозволяють користувачам швидко та зручно знайти необхідну їм послугу, порівняти ціни та умови різних компаній, а також забронювати авто онлайн.

Розробка ефективних ІПС замовлень автопослуг потребує використання сучасних технологій проектування програмного забезпечення (ПЗ). Це дозволяє створити системи, які відповідають потребам користувачів, мають високу продуктивність та надійність.

ІПС замовлень автопослуг дозволяють автоматизувати багато рутинних завдань, пов'язаних з прийомом та обробкою замовлень, веденням бази даних клієнтів та історії їх обслуговування, а також з наданням клієнтам доступу до інформації про послуги та ціни. Завдяки цьому ІПС можуть допомогти приватним фірмам:

- Знизити витрати на обслуговування клієнтів.

- Покращити якість обслуговування клієнтів.

- Збільшити обсяг продажів.

- Підвищити конкурентоспроможність на ринку.

Метою даної бакалаврської роботи є розробка інформаційно-пошукової системи замовлень автопослуг для приватної фірми з використанням сучасних технологій проектування програмного забезпечення.

Це дозволить:

- Автоматизувати процес прийому та обробки замовлень.

- Створити базу даних клієнтів та історії їх обслуговування.

- Надати клієнтам доступ до інформації про послуги та ціни.

- Знизити витрати на обслуговування клієнтів.

- Покращити якість обслуговування клієнтів.

Збільшити обсяг продажів.

Підвищити конкурентоспроможність фірми на ринку.

Об'єкт дослідження

Об'єктом дослідження даної дипломної роботи є процес надання автопослуг приватною фірмою.

Предметом дослідження даної дипломної роботи є інформаційно-пошукова система замовлень автопослуг, що розробляється для приватної фірми.

Для досягнення мети дипломної роботи використовувалися такі методи дослідження: Аналіз літературних джерел з даної теми, системний аналіз та проектування інформаційно-пошукової системи, розробка програмного забезпечення ПС, тестування та впровадження ПС.

В роботі використані сучасні технології проектування ПЗ: Microsoft Visual Studio та MySQL Workbench.

Microsoft Visual Studio було використано як інтегроване середовище розробки (IDE) для написання коду програми на мові програмування [мова програмування]. Visual Studio надав потужний редактор коду, відладчик та інструменти для тестування, що значно спростило процес розробки.

MySQL Workbench використовувався як графічний інтерфейс користувача (GUI) для роботи з базою даних MySQL. Завдяки MySQL Workbench можна було легко створювати та редагувати об'єкти бази даних, виконувати SQL-запити та переглядати дані.

Практична значимість даної дипломної роботи полягає в тому, що розроблена ПС може бути використана приватними фірмами для автоматизації процесу надання автопослуг, підвищення якості обслуговування клієнтів та збільшення обсягу продажів.

Робота складається з вступу, двох розділів, висновків, списку використаних джерел з 16 найменувань та додатків.

Робота викладена на 63 сторінках.

1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Загальна характеристика сфери автопослуг

Сфера автопослуг в Україні охоплює широкий спектр послуг, пов'язаних з експлуатацією, обслуговуванням та ремонтом транспортних засобів. До основних видів автопослуг належать:

Перевезення пасажирів та вантажів: таксі, автобусні перевезення, вантажні перевезення, перевезення спецтехніки тощо.

Технічне обслуговування та ремонт: станції технічного обслуговування (СТО), шиномонтаж, послуги автоелектрика, кузовний ремонт тощо.

Продаж запчастин та аксесуарів: автомагазини, спеціалізовані магазини, інтернет-магазини тощо.

Інші послуги: мийка авто, евакуація авто, прокат авто, автосклади, паркінги тощо.

Підприємства сфери автопослуг можна класифікувати за різними ознаками, зокрема:

За формою власності: державні, приватні, комунальні.

За розміром: малі, середні, великі.

За видом діяльності: спеціалізовані (надають один вид послуг) та універсальні (надають декілька видів послуг).

За територією обслуговування: місцеві, регіональні, національні.

Ринок автопослуг в Україні є одним з найбільших та динамічно розвиваючихся секторів економіки. Обсяг ринку перевезень таксі, за оцінками BRDO, складає близько 40 млрд грн на рік (0,5% від загального обсягу «білої» виручки), інформаційно-диспетчерського обслуговування — близько 4 млрд грн.

Динаміці розвитку ринку автопослуг сприяють такі фактори:

Зростання парку транспортних засобів в Україні.

Покращення купівельної спроможності населення.

Розвиток інфраструктури автошляхів.

Зростання популярності онлайн-сервісів для замовлення автопослуг.

На рис.1.1 представлена структура ринку автопослуг

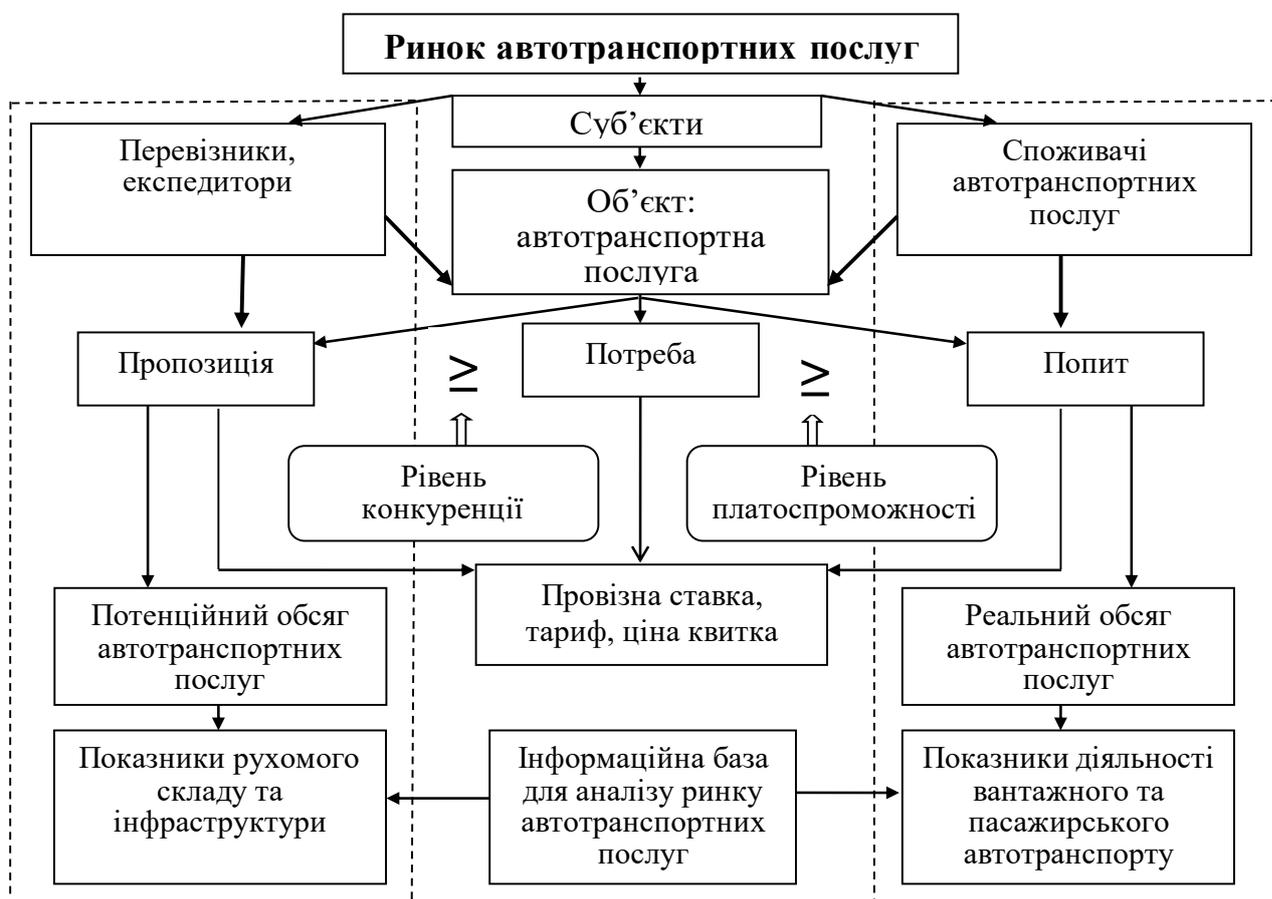


Рис. 1.1 Структура ринку автопослуг

Аналіз основних параметрів ринку автотранспортних послуг

1. Попит:

Визначення: Попит на автотранспортні послуги – це частина потреби в цих послугах, яка забезпечена платоспроможністю споживачів.

Важливість: Попит є ключовим фактором, що зв'язує виробництво автотранспортних послуг з їх споживанням. Вивчення, аналіз та прогнозування попиту є невід'ємною частиною комплексного дослідження ринку.

Формування попиту: Споживачі автотранспортних послуг – це підприємства, установи, організації та населення. Їхні вимоги та платоспроможність визначають обсяг попиту.

Характеристика попиту: Попит на автотранспортні послуги відображає реальний обсяг роботи автотранспорту, адже на цьому виді транспорту виробництво та споживання є єдиним цілим.

Оцінка попиту: Оцінка попиту здійснюється за допомогою системи статистичних показників, що включають дані про діяльність вантажного та пасажирського автотранспорту.

2. Пропозиція:

Визначення: Пропозиція на ринку автотранспортних послуг формується перевізниками та експедиторами. Вона характеризує потенційний обсяг роботи автотранспорту, який, в свою чергу, залежить від рівня конкуренції на ринку.

Оцінка пропозиції: Оцінити пропозицію на ринку автотранспортних послуг складніше, ніж попит. Для цього використовуються експертні оцінки, а також дані вибіркового обстеження та опитувань суб'єктів ринку.

3. Взаємодія попиту та пропозиції:

Вплив факторів: Співвідношення попиту та пропозиції на ринку автотранспортних послуг формується під впливом соціально-економічних, правових та інших факторів.

Визначення ціни: Саме співвідношення попиту та пропозиції в кожний конкретний момент визначає рівень тарифів на транспортні послуги (ринкову ціну).

Ціна на транспорті: Ціна на транспортні послуги виступає у специфічному вигляді провізної ставки, тарифу, ціни квитка тощо.

Вплив на ціну товарів: Ціна наданої транспортної послуги стає частиною ціни товару, що існувала до його переміщення.

Аналіз основних параметрів ринку автотранспортних послуг – це важливий інструмент для розуміння його функціонування та прогнозування майбутніх тенденцій. Вивчення попиту, пропозиції та їх взаємодії дозволяє оцінити стан ринку, визначити оптимальні ціни на транспортні послуги та розробити ефективні стратегії розвитку для різних учасників ринку.

До основних тенденцій розвитку сфери автопослуг в Україні належать:

Впровадження нових технологій: використання штучного інтелекту, робототехніки, систем телеметрії тощо.

Розвиток онлайн-сервісів: замовлення автопослуг через мобільні додатки, онлайн-оплата, онлайн-запис на СТО тощо.

Зростання попиту на екологічні автопослуги: електромобілі, гібридні авто, альтернативні види палива тощо.

Зміна структури попиту на автопослуги: зростання попиту на комплексні послуги (наприклад, абонементне обслуговування), послуги з доставки запчастин та аксесуарів тощо.

Сучасні інформаційні технології значно покращили управління та експлуатацію вантажного та пасажирського транспорту. Наведемо декілька прикладів їх використання у сфері вантажних та пасажирських перевезень:

1. Програмне забезпечення для управління рухомим складом автотранспортного підприємства: програмне забезпечення для керування автопарком дозволяє компаніям відстежувати місцезнаходження та статус своїх транспортних засобів у режимі реального часу. Це допомагає оптимізувати маршрути, зменшити споживання палива та підвищити безпеку праці водія.

2. Телематика: телематичні системи використовують GPS та інші датчики для збору даних про місцезнаходження, швидкість і продуктивність транспортних засобів.

Ці дані можна використовувати для підвищення показників ефективності функціонування автомобільного транспорту та безпеки руху.

3. Електронні пристрої реєстрації (ELD): ELD — це пристрої, які реєструють години роботи водія в електронному вигляді. Це допомагає переконатися, що водії дотримуються нормативних вимог щодо безпеки руху та умов праці і відпочинку, а також забезпечує зменшення адміністративного навантаження на персонал АТП.

4. Транспортні послуги на вимогу: транспортні послуги на вимогу, такі як Uber і Lyft, кардинально змінили традиційну індустрію таксі та спільних поїздок. Ці сервіси дозволяють пасажиром швидко та легко знаходити поїздки за допомогою своїх смартфонів, а також дозволяють водіям працювати за гнучким графіком.

5. Інтелектуальні транспортні системи (ITS): ІТС-технології, такі як системи управління дорожнім рухом, системи інформації для мандрівників і системи управління вантажами, можуть допомогти підвищити ефективність і безпеку транспортної системи в цілому.

Загалом сучасні інформаційні технології допомагають підвищити ефективність, безпеку та стійкість вантажних і пасажирських перевезень. Ці технології також створюють нові бізнес-моделі та можливості для інновацій у галузі автомобільних перевезень.

1.2 Суть послуг таксі, пов'язані процеси та послуги

Ринок внутрішніх перевезень на таксі є елементом транспортної інфраструктури та важливим для розвитку мікро- та малого підприємництва в Україні, а також з точки зору тимчасового працевлаштування громадян які втратили роботу з тих чи інших причин. За оцінками УТМА, кількість осіб, задіяних в перевезеннях на таксі, складає понад 220 тис. осіб, що становить близько 1% від загальної кількості економічно активного населення.

Для пасажирів таксі ринок надає послуги з локальної мобільності та підвищеного комфорту пересування.

З точки зору територіальних громад та міст таксі є частиною транспортної системи.

Перевезення пасажирів на таксі – це перевезення пасажирів на таксі в межах України, часто в межах одного населеного пункту. Послуги з перевезення на таксі надаються громадянам у порядку черги на стоянках таксі та на шляху прямування, а також на замовлення (звичайне або термінове, усне, письмове чи за телефоном).

Учасниками ринку є:

Пасажири таксі;

Перевізники;

Юридичні особи;

Фізичні особи-підприємці (ФОП, автомобільні перевізники);

Фізичні особи, які здійснюють перевезення без реєстрації підприємницької діяльності;

Інформаційно-диспетчерські служби (ІДС);

Власники стоянок таксі;

Страхові компанії;

Лізингові компанії.

ЮРИДИЧНІ ОСОБИ (КОМПАНІЇ-ПЕРЕВІЗНИКИ):

Як правило, мають ліцензію;

Інколи мають власну ІДС, контактну інформацію якої рекламують;

Виконують перевезення на власних або орендованих авто (лізинг);

Можуть наймати для перевезень водіїв на власні та орендовані авто;

Перевезення таксі може бути не основним бізнесом;

Типовими представниками є: Еліт таксі, Оксі таксі, Уклон.

ФІЗИЧНІ ОСОБИ-ПІДПРИЄМЦІ (САМОЗАЙНЯТІ АВТОМОБІЛЬНІ ПЕРЕВІЗНИКИ):

Виконують перевезення пасажирів на власних або орендованих авто;

Мають ліцензію на відповідний вид діяльності.

ФІЗИЧНІ ОСОБИ, ЯКІ ЗДІЙСНЮЮТЬ ПЕРЕВЕЗЕННЯ БЕЗ РЕЄСТРАЦІЇ ПІДПРИЄМНИЦЬКОЇ ДІЯЛЬНОСТІ

Виконують перевезення на власних або орендованих авто без реєстрації суб'єктами господарювання та без отримання відповідної ліцензії. Займаються перевезенням пасажирів на таксі як основним видом діяльності або заради додаткового заробітку.

ІНФОРМАЦІЙНО-ДИСПЕТЧЕРСЬКІ СЛУЖБИ (ІДС):

Надають інформаційно-диспетчерські послуги перевізникам.

Допомагають пасажиром знайти перевізника та навпаки;

ІДС можуть бути:

Офіційно зареєстровані суб'єктами підприємницької діяльності;

Не зареєстровані (учасники тіньового ринку) або зареєстровані за кордоном;

Деякі перевізники мають власні ІДС (часто це окрема юридична особа), дрібні перевізники укладають з ІДС договори про інформаційне диспетчерське обслуговування.

ВЛАСНИКИ СТОЯНОК ТАКСІ

Місцеві ради через відповідні комунальні підприємства;

Власники торговельних центрів та інших об'єктів інфраструктури.

СТРАХОВІ КОМПАНІЇ

Здійснюють страхування транспортних засобів та цивільної відповідальності водіїв.

ЛІЗИНГОВІ КОМПАНІЇ

Надають перевізникам автомобілі в лізинг.

Кількісні параметри ринку перевезень на таксі за даними УТМА наведені в таблиці 1.1

Таблиця 1.1

Кількісні параметри ринку перевезень на таксі

Показник по Україні	Обсяг
Авто задіяних у перевезеннях	200 тис.
Водіїв задіяних в перевезеннях	220 тис.
Поїздок, Україна	280 млн шт./рік
% перевізників part-time	50%
Обсяг ринку перевезень	40 млрд грн
Обсяг ринку перевезень	36 млрд грн
Обсяг ринку ІДС	4 млрд грн
Ринок ліцензованих перевезень	
Авто задіяних у перевезеннях	28 тис.
Авто задіяних у перевезеннях, %	14%
Поїздок в день	100 тис.
Обсяг ринку перевезень згідно стат. даних (ліцензованих)	0,6 млрд грн
% перевезень ліцензіатами в загальному обсязі	3%
Обсяг ринку ліцензованих перевезень (через серед. дохід), %	5,1%
% перевезень ліцензіатами в загальному обсязі (через серед. дохід)	30%
% тінізації ринку ліцензованих перевезень	95%

Дані, Uklon, розрахунки BRDO

Значна частина постійних користувачів послуг перевезень не вважає замовлення таксі за допомогою телефону зручним через такі фактори:

- проблеми з мобільним зв'язком: оператор може неправильно почути адресу посадки або час, на який замовлено таксі;
- диспетчери часто говорять дуже нерозбірливо, що ускладнює процес замовлення;

- деякі люди (зокрема інтроверти) відчувають дискомфорт під час особистого спілкування, відповідно розмова з диспетчером при замовленнях також є неприйнятною для них;
- перевантажені лінії call-центрів служб замовлення таксі унеможливають швидке прийняття замовлень, тому доводиться довго очікувати відповіді.

Також варто враховувати, що є група людей з вадами слуху та мовлення, які фізично не можуть замовити таксі за телефоном. В свою чергу, дзвінки водіїв про прибуття на місце посадки часто бувають недоречними або заважають.

Замовлення таксі через мобільні додатки:

Хоча мобільні додатки для замовлення таксі зручні, вони мають свої недоліки. Їх потрібно встановлювати, вони займають місце на телефоні (що не всім подобається), а ваш акаунт прив'язаний до конкретного пристрою. До того ж, корпоративним клієнтам, які замовляють таксі для співробітників, потрібна можливість робити це з комп'ютера.

Як працює звичайна служба таксі з диспетчером:

Клієнт телефонує або залишає заявку.

Диспетчер отримує адреси посадки та висадки, іноді озвучує орієнтовну вартість.

Замовлення надсилається водіям.

Вільний водій приймає замовлення, система повідомляє про це клієнту SMS.

Водій виконує замовлення, повідомляє про це диспетчеру.

Чому для веб-застосунку не підходить алгоритм з диспетчером:

Цей проект розробляється як автоматизована система, тому диспетчер не потрібен.

Алгоритм для веб-застосунку:

Клієнт залишає заявку з адресами посадки та висадки, система одразу повідомляє орієнтовну вартість.

Водії бачать замовлення на спеціальній сторінці, обирають та приймають найвигідніше.

Клієнт бачить зміну статусу замовлення.

Водій виконує замовлення, позначає його як завершене. Система автоматично нараховує оплату.

Переваги нового алгоритму:

Автоматизація процесу.

Зручність для клієнтів та водіїв.

Прозорість та контроль над замовленнями.

Зниження витрат на диспетчерів.

Цей алгоритм робить веб-застосунок для замовлення таксі сучасним, зручним та ефективним.

Варто зазначити, актуальність системи що розробляється значно виросла через світову пандемію COVID-19, адже громадський транспорт на території України було заборонено для загального користування, тому попит на таксі у цілому значно підвищився, адже інших шляхів пересування містом не існує, якщо немає особистого транспорту.

Отже, виникає потреба в універсальному веб-застосуванні для замовлення таксі, що не вимагатиме особистого спілкування з диспетчером, не потребуватиме завантажування додаткового програмного забезпечення на телефон та буде зручним для користування з будь-якого пристрою. При цьому на різних пристроях повинен бути незалежний доступ до єдиного акаунту користувача.

1.3 Огляд існуючих аналогів служб таксі

Сфера замовлень таксі онлайн надзвичайно велика, зокрема, в Україні є декілька великих конкуруючих компаній, що пропонують послуги онлайн замовлення таксі, серед них Uber, Uklon, Bolt.

Вибір служби таксі залежить від індивідуальних потреб та вподобань клієнта. Uklon може бути гарним вибором для тих, хто шукає доступні ціни та зручний мобільний додаток. Bolt може бути гарним вибором для тих, хто шукає швидку подачу таксі та широкий спектр додаткових послуг. Uber може бути гарним вибором для тих, хто шукає міжнародний бренд та високий рівень обслуговування.

- Uber – один з перших сервісів, що популяризував замовлення таксі онлайн не лише в Україні.

Історія компанії: Uber заснований у 2010 році в США. Зараз компанія працює в 70 країнах (включаючи Україну) та має понад 100 мільйонів користувачів.

Бізнес-модель: Uber використовує агрегаторну модель, тобто не володіє власним автопарком, а співпрацює з водіями-партнерами.

Технологічні рішення: Uber має мобільний додаток для iOS та Android, а також веб-сайт. Додаток дозволяє користувачам замовити таксі, відстежувати його розташування, оцінити водія та сплатити за поїздку.

Парк автомобілів: Uber не володіє власним автопарком, а співпрацює з водіями-партнерами, які використовують власні автомобілі.

Цінова політика: Uber використовує динамічну ціну, яка залежить від попиту та пропозиції на таксі.

Відгуки клієнтів: Uber має переважно позитивні відгуки клієнтів. Користувачі цінують зручність мобільного додатку, швидкість подачі таксі та доступні ціни.

Компанія пропонує замовлення таксі телефоном, у мобільному додатку або через сайт. Зокрема, замовити послугу через сайт достатньо важко, адже

сторінка замовлення погано відображується пошуковими системами [1], а іноді після замовлення через сайт користувачам можуть телефонувати для підтвердження замовлення. Замовлення через мобільний додаток дуже зручне та містить багато додаткових опцій, декілька способів оплати та можливість зв'язатись з водієм через додаток за необхідності.

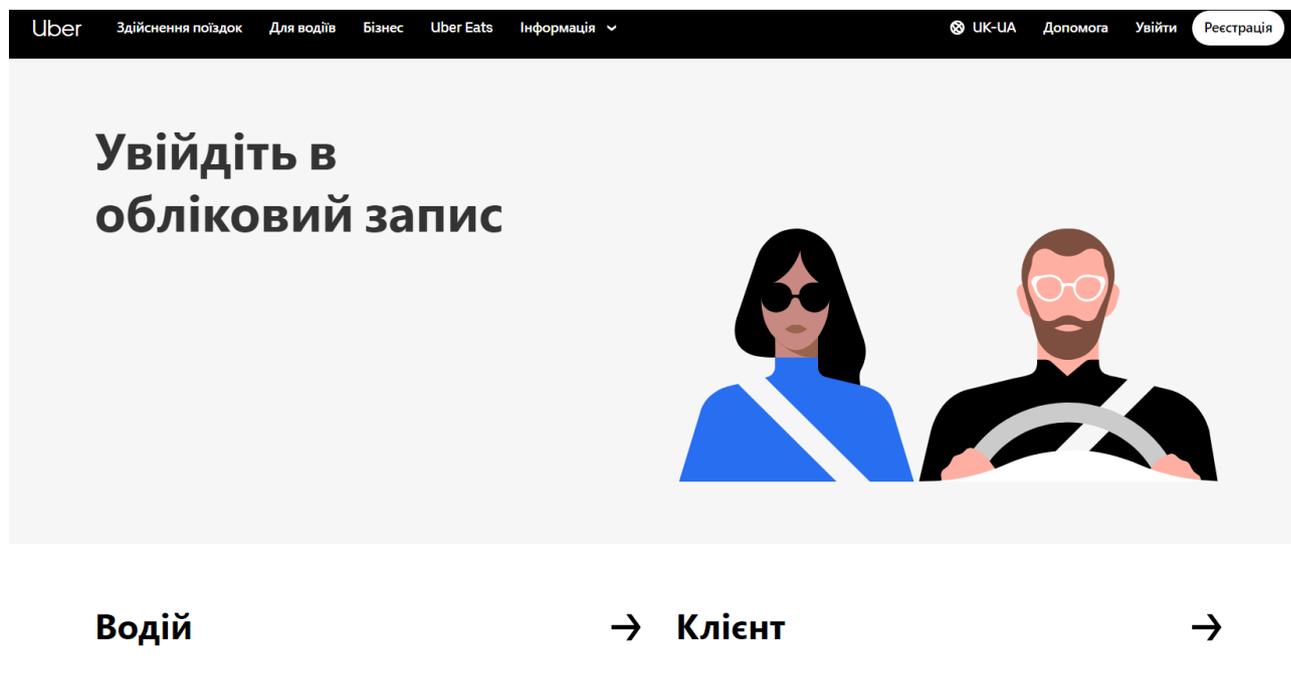


Рис. 1.2 – Сайт Uber

Uklon – перший український сервіс для замовлення таксі онлайн [2].

Історія компанії: Uklon заснований у 2014 році в Україні. Зараз компанія працює в 6 країнах (Україна, Білорусь, Казахстан, Грузія, Молдова, Киргизстан) та має понад 35 мільйонів користувачів.

Бізнес-модель: Uklon використовує агрегаторну модель, тобто не володіє власним автопарком, а співпрацює з водіями-партнерами.

Технологічні рішення: Uklon має мобільний додаток для iOS та Android, а також веб-сайт. Додаток дозволяє користувачам замовити таксі, відстежувати його розташування, оцінити водія та сплатити за поїздку.

Парк автомобілів: Uklon не володіє власним автопарком, а співпрацює з водіями-партнерами, які використовують власні автомобілі.

Цінова політика: Uklon використовує динамічну ціну, яка залежить від попиту та пропозиції на таксі.

Відгуки клієнтів: Uklon має переважно позитивні відгуки клієнтів. Користувачі цінують зручність мобільного додатку, швидкість подачі таксі та доступні ціни.

В Uklon повністю відсутня можливість замовити таксі через телефон, зробити замовлення можна виключно в мобільному додатку. Додаток має зручний інтерфейс, декілька можливих способів оплати, а також можливість змінювати ціну на поїздку. Проте, з особистого досвіду користування можна відмітити проблеми з визначенням адреси при замовленні поїздки, яка полягає в тому, що пошуковик додатку не завжди знаходить потрібну вулицю за частиною назви.

Uklon започаткувався як проєкт компанії Evos, яка займалася розробкою ПЗ для служб таксі. 2006 року Evos заснував Дмитро Дубровський. На той час в Evos було близько 20 працівників.

Початкові інвестиції в проєкт склали \$10 тис., за пів року було створено сайт з замовленням авто[7]. За пів року був розроблений багатофункціональний вебпортал, де для онлайн-замовлення авто виділялася одна із вкладок. На цій сторінці можна було викликати таксі без «розмови з диспетчером».[6] Uklon розпочав роботу 25 березня 2010 року. Відтоді сервіс виклику авто трансформувалася з порталу з вкладкою виклику таксі у великий хмарний продукт. Станом на 2013 рік сервіс співпрацював із більш ніж 100 службами таксі, тоді через Uklon приходило 500—1000 викликів на добу. Уже у 2014 році агрегатор став самоокупним. Evos та Uklon розпочали працювати як дві окремі компанії з 2015 року.[6]

Назва компанії є скороченням від «Ukraine online» — інтернет-порталу, який створений засновниками сервісу.[9] У 2012-му був випущений перший додаток для Android, у 2013 — для iOS. У 2016 році компанія впровадила оплату проїзду банківською карткою.

У 2017—2018 роках Uklon повністю переробив застосунок, систему рейтингу та боротьби з шахрайством, покращив карти, алгоритми подачі авто, ввів страхування пасажирів і водіїв, а також здійснив ребрендинг.[6]

У 2019 Uklon спільно зі страховою компанією ARX збільшили суму страхування пасажирів до 1 млн грн, попередньо послугу страхування запустили у 2017 році. У травні 2022-го Uklon оголосив про запуск франшизи для глобальної експансії на нові ринки.[13]

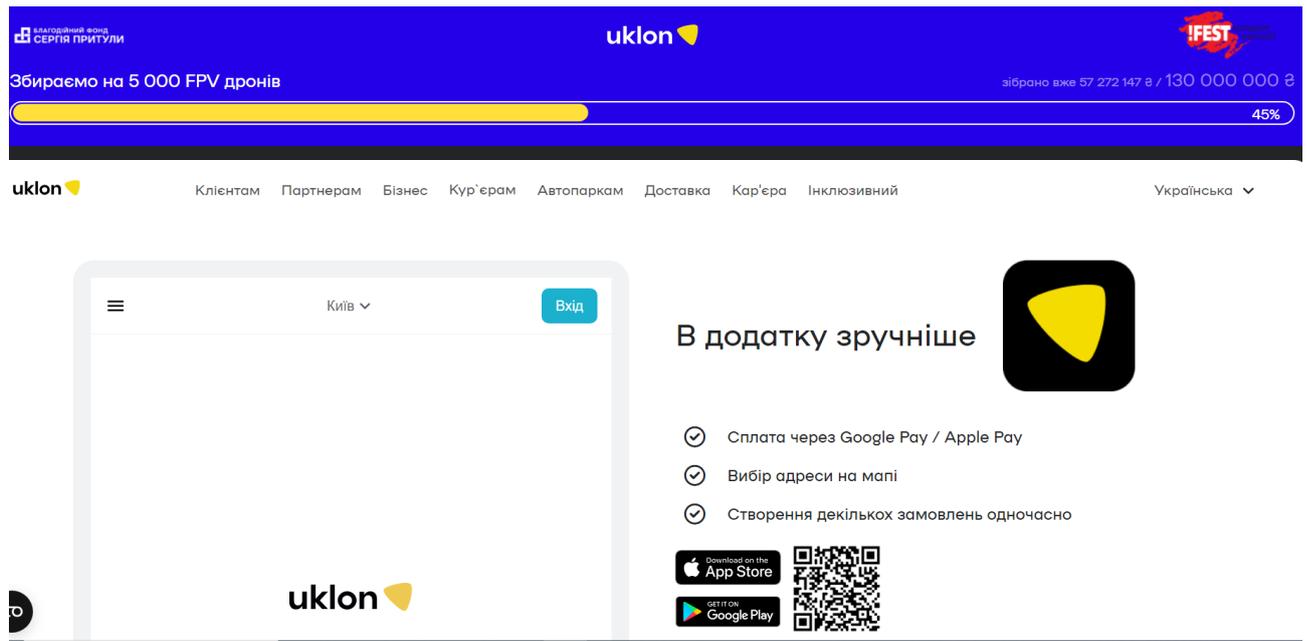


Рис.1.3. Сайт компанії Uklon

Bolt

Історія компанії: Bolt заснований у 2013 році в Естонії. Зараз компанія працює в 45 країнах (включаючи Україну) та має понад 100 мільйонів користувачів.

Бізнес-модель: Bolt використовує агрегаторну модель, тобто не володіє власним автопарком, а співпрацює з водіями-партнерами.

Технологічні рішення: Bolt має мобільний додаток для iOS та Android, а також веб-сайт. Додаток дозволяє користувачам замовити таксі, відстежувати його розташування, оцінити водія та сплатити за поїздку.

Парк автомобілів: Bolt не володіє власним автопарком, а співпрацює з водіями-партнерами, які використовують власні автомобілі.

Цінова політика: Bolt використовує динамічну ціну, яка залежить від попиту та пропозиції на таксі.

Відгуки клієнтів: Bolt має переважно позитивні відгуки клієнтів. Користувачі цінують зручність мобільного додатку, швидкість подачі таксі та доступні ціни.

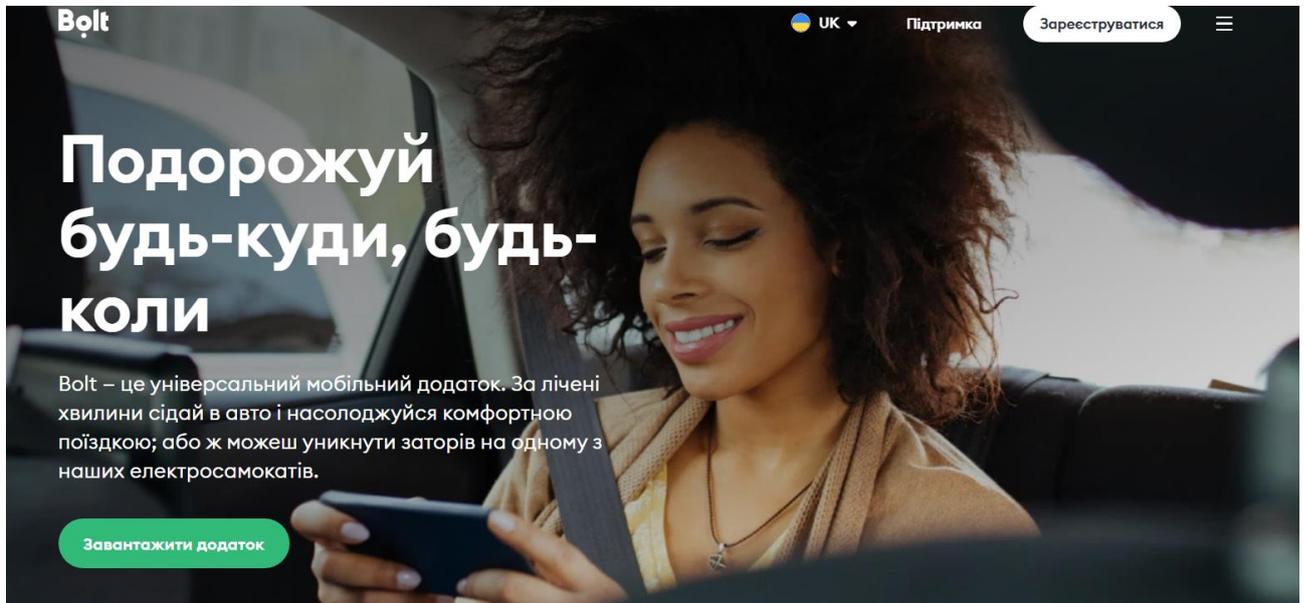


Рис.1.4. Сайт компанії Bolt

Bolt – міжнародна компанія, що надає послуги замовлення таксі виключно в додатку, набула популярності в Україні останньою з вищезгаданих [3]. Компанія має найбільший вибір типів авто, які можна замовити, але порівняно незрозумілий інтерфейс з іншими мобільними додатками, натомість пропонує багато бонусів для своїх користувачів.

Існують також інші достатньо популярні сервіси, що пропонують послуги замовлення таксі за телефоном та онлайн. За останні роки більшість приватних компаній на ринку пасажирських перевезень намагаються створити та популяризувати свої способи замовлення таксі онлайн або в мобільних застосуваннях. Вдалим прикладом є служба замовлення таксі 838 [4], яка лише у 2019 році запустила повноцінний мобільний додаток для замовлення таксі. Хоча ця компанія на ринку досить давно, проте її популярність залишається порівняно низькою з іншими сервісами.

Таблиця 1.2.

Порівняльний аналіз служб таксі

Характеристика	Uklon	Bolt	Uber
Країна заснування	Україна	Естонія	США
Рік заснування	2014	2013	2010
Кількість країн	6	45	70
Кількість користувачів	35 мільйонів	100 мільйонів	100 мільйонів
Бізнес-модель	Агрегатор	Агрегатор	Агрегатор
Парк автомобілів	Власні автомобілі та автомобілі партнерів	Власні автомобілі та автомобілі партнерів	Власні автомобілі та автомобілі партнерів
Цінова політика	Динамічна ціна	Динамічна ціна	Динамічна ціна
Додаткові послуги	Дитяче крісло, кондиціонер, Wi-Fi, англомовний водій	Дитяче крісло, кондиціонер, Wi-Fi, англомовний водій	Дитяче крісло, кондиціонер, Wi-Fi, англомовний водій
Відгуки клієнтів	Переважно позитивні	Переважно позитивні	Переважно позитивні

Також існує безліч інших непопулярних онлайн сервісів, які дозволяють «замовити» таксі онлайн. Але насправді після залишеної заявки телефонує диспетчер для уточнення адреси посадки та на вимогу називає приблизну вартість поїздки телефоном, що знову ж таки є незручним для більшості користувачів. Необхідно ще раз звернути увагу на те, що веб-сервіс буде розроблений таким чином, щоб зробити систему повністю автоматизованою, забезпечити замовлення таксі та реалізацію замовлень без участі диспетчера (лише клієнт та водій). Тому можна вважати, що зараз для більшості компаній в галузі замовлень таксі, аналогових до розробки, що стала об'єктом

дослідження, реальним конкурентом є лише сервіс замовлень таксі онлайн від Uber. Проте, варто враховувати, що конкурентними є також і мобільні додатки, адже є багато прихильників таких способів замовлення таксі. Тому оцінка кількості конкурентів може варіюватися в залежності від погляду на проблему: а саме, чи враховувати служби, що пропонують замовлення таксі в мобільних додатках, як конкурентів? Отже, наразі на ринку надання послуг пасажирських перевезень існує достатньо аналогів розробки, проте більшість з них не задовольняють повний перелік вимог користувачів до таких систем, особливо онлайн-сервіси, що лише імітують автоматизоване замовлення таксі. Частина користувачів цілком задоволена мобільними додатками, проте також існує пласт невдоволених користувачів. Таким чином, це ще раз доводить доцільність розробки такої інформаційної системи.

1.4 Сучасні технології програмування

Сучасні технології проектування ПЗ можна класифікувати за різними ознаками:

За методологією:

Структуровані методології: передбачають чітку послідовність етапів розробки ПЗ та використання формальних методів моделювання та опису системи. Прикладами структурованих методологій є каскадна модель, V-подібна модель, ітеративна модель.

Об'єктно-орієнтовані методології: фокусуються на розбитті системи на об'єкти, які мають власні властивості та методи. Прикладами об'єктно-орієнтованих методологій є UML, OMT, RUP.

Адаптивні методології: передбачають гнучкий підхід до розробки ПЗ, який дозволяє змінювати плани та пріоритети в міру просування проекту. Прикладами адаптивних методологій є Scrum, Kanban, Extreme Programming.

За рівнем абстракції:

Методології загального рівня: описують загальні принципи та підходи до розробки ПЗ, не даючи детальних інструкцій щодо реалізації. Прикладами методологій загального рівня є CMMI, ITIL.

Методології архітектурного рівня: описують принципи проектування архітектури ПЗ, включаючи визначення компонентів системи, їх взаємодії та інтерфейсів. Прикладами методологій архітектурного рівня є TOGAF, Zachman Framework.

Методології детального рівня: описують деталі реалізації ПЗ, включаючи вибір мови програмування, інструментів розробки, шаблонів проектування тощо. Прикладами методологій детального рівня є Clean Code, Design Patterns.

За сферою застосування:

Методології розробки веб-додатків: призначені для розробки веб-сайтів та веб-застосунків. Прикладами методологій розробки веб-додатків є Agile Web Development, Model-View-Controller (MVC).

Методології розробки мобільних додатків: призначені для розробки мобільних застосунків для смартфонів та планшетів. Прикладами методологій розробки мобільних додатків є Native Development, Cross-Platform Development.

Методології розробки вбудованих систем: призначені для розробки програмного забезпечення для вбудованих систем, таких як мікроконтролери та промислові комп'ютери. Прикладами методологій розробки вбудованих систем є Real-Time Operating Systems (RTOS), Embedded Software Development Lifecycle (ESDLC).

Важливо зазначити, що не існує єдиної "універсальної" методології проектування ПЗ, яка б підходила для всіх випадків. Вибір методології залежить від багатьох факторів, таких як розмір та складність проекту, бюджет, терміни, навички команди розробників тощо.

В роботі будуть застосовані наступні технології:

Microsoft Visual Studio.

Microsoft Visual Studio – це комплексна інтегрована середовище розробки (IDE), що використовується для створення програмного забезпечення. Вона популярна серед програмістів завдяки своїм широким можливостям та зручному інтерфейсу. Visual Studio використовується для розробки різноманітних програм, включаючи:

Веб-сайти та веб-додатки: ASP.NET, JavaScript, TypeScript, C#

Десктопні програми: C#, C++, F#

Ігри: C++, DirectX

Мобільні програми: C#, Xamarin

Хмарні служби: Azure

Основні можливості Visual Studio:

Редактор коду: Visual Studio пропонує потужний редактор коду з підтримкою синтаксичного підсвічування, автозаповнення коду, рефакторингу та інших функцій, що полегшують написання та читання коду.

Компілятор та інструменти розробки: Visual Studio включає компілятори та інші інструменти для різних мов програмування, що дозволяє компілювати, тестувати та відлагоджувати код.

Відладчик: Visual Studio має вбудований відладчик, який дозволяє програмістам покроково виконувати код, встановлювати точки зупинки та переглядати значення змінних.

Інтегроване середовище розробки (IDE): Visual Studio об'єднує редактор коду, компілятор, відладчик та інші інструменти в одному середовищі, що робить процес розробки більш ефективним.

Розширення: Visual Studio підтримує розширення, які додають нові функції та можливості до IDE. Існує безліч безкоштовних та платних розширень, які можна використовувати для налаштування Visual Studio під свої потреби.

Переваги використання Visual Studio:

Збільшена продуктивність: Visual Studio пропонує безліч функцій, які допомагають програмістам писати код швидше та ефективніше.

Покращена якість коду: Visual Studio допомагає писати чистий, добре структурований та надійний код.

Легше тестування та відладка: Visual Studio має вбудовані інструменти для тестування та відладки коду, що полегшує виявлення та виправлення помилок.

Широкий спектр можливостей: Visual Studio підтримує широкий спектр мов програмування та платформ, що робить її універсальним інструментом для розробки програмного забезпечення.

Велика спільнота: Visual Studio має велику та активну спільноту користувачів, що значить, що ви можете легко знайти допомогу та підтримку онлайн.

Види Visual Studio:

Існує два основних види Visual Studio:

Visual Studio Community: Безкоштовна версія IDE з базовими можливостями. Підходить для студентів, початківців та невеликих проектів.

Visual Studio Professional: Платна версія IDE з розширеними можливостями. Підходить для професійних розробників та великих проектів.

Загальний вигляд вікна, яке відображається при завантаженні Visual Studio, початок роботи та створення нового проекту відображено на рис. 1.4.

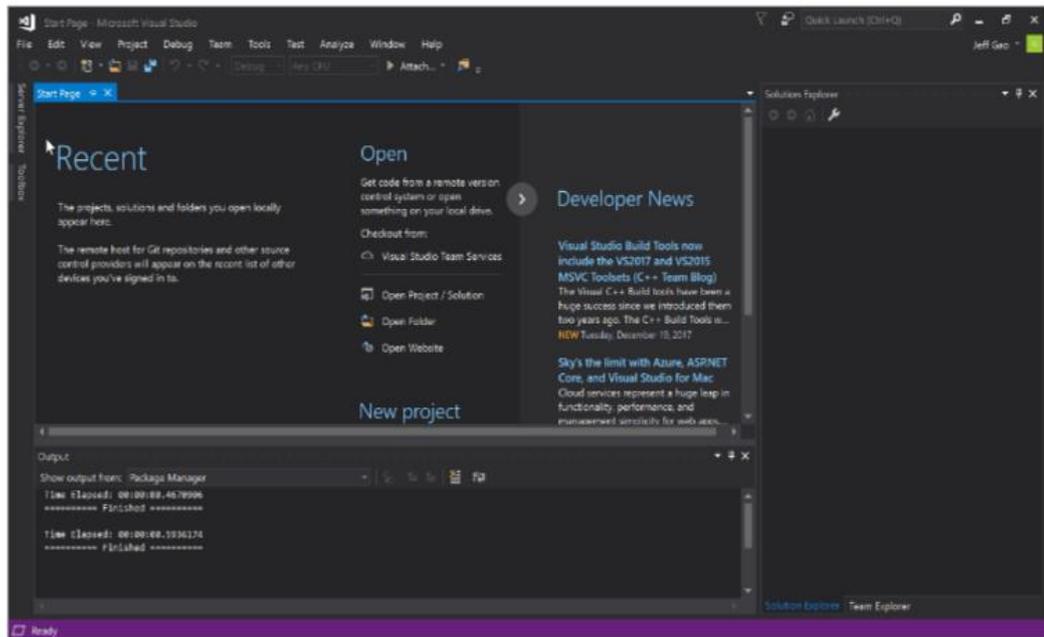


Рис.1.4. Visual Studio

Microsoft Visual Studio – це потужний та універсальний інструмент для розробки програмного забезпечення, який використовується мільйонами програмістів у всьому світі. Дієкі своїм широким можливостям та зручному інтерфейсу Visual Studio може допомогти вам писати код швидше, ефективніше та створювати програмне забезпечення кращої якості[8].

MYSQL WORKBENCH

MySQL Workbench – це універсальний візуальний інструмент, розроблений для архітекторів баз даних, розробників та адміністраторів баз даних (DBA). Він пропонує широкий спектр функцій для спрощення різних завдань, пов'язаних з базами даних MySQL, включаючи:

Моделювання даних: Візуально проектуйте та створюйте структури баз даних за допомогою ER-діаграм (Entity-Relationship Diagrams).

Розробка SQL: Пишіть, редагуйте та виконуйте SQL-запити та сценарії для взаємодії з вашою базою даних.

Адміністрування бази даних: Керуйте серверами MySQL, користувачами, дозволами, резервними копіями та виконуйте інші адміністративні завдання.

Візуальні інструменти: Використовуйте візуальні інструменти для налаштування сервера, керування користувачами, моніторингу продуктивності та імпорту/експорту даних.

Основні переваги MySQL Workbench:

Уніфікована платформа: Об'єднує різні функціональні можливості в одному інтерфейсі, підвищуючи ефективність робочого процесу.

Візуальний дизайн: Забезпечує інтуїтивно зрозуміле проектування бази даних за допомогою ER-діаграм, спрощуючи процес.

Функції розробки SQL: Пропонує автозаповнення коду, підсвітку синтаксису та інструменти налагодження для написання ефективних SQL-запитів.

Спрощене адміністрування: Надає зручний інтерфейс для керування серверами та базами даних MySQL.

Сумісність з кількома платформами: Працює без проблем на операційних системах Windows, macOS та Linux.

Хто може скористатися MySQL Workbench?

Розробники баз даних: Спростить робочі процеси розробки за допомогою візуальних інструментів проектування та функцій розробки SQL.

Адміністратори баз даних: Ефективно керуйте серверами MySQL, користувачами та виконуйте адміністративні завдання.

Архітектори баз даних: Візуально проектуйте та моделюйте складні структури баз даних.

На рисунку 1.3 показано вікно розробки SQL для MySQL Workbench.

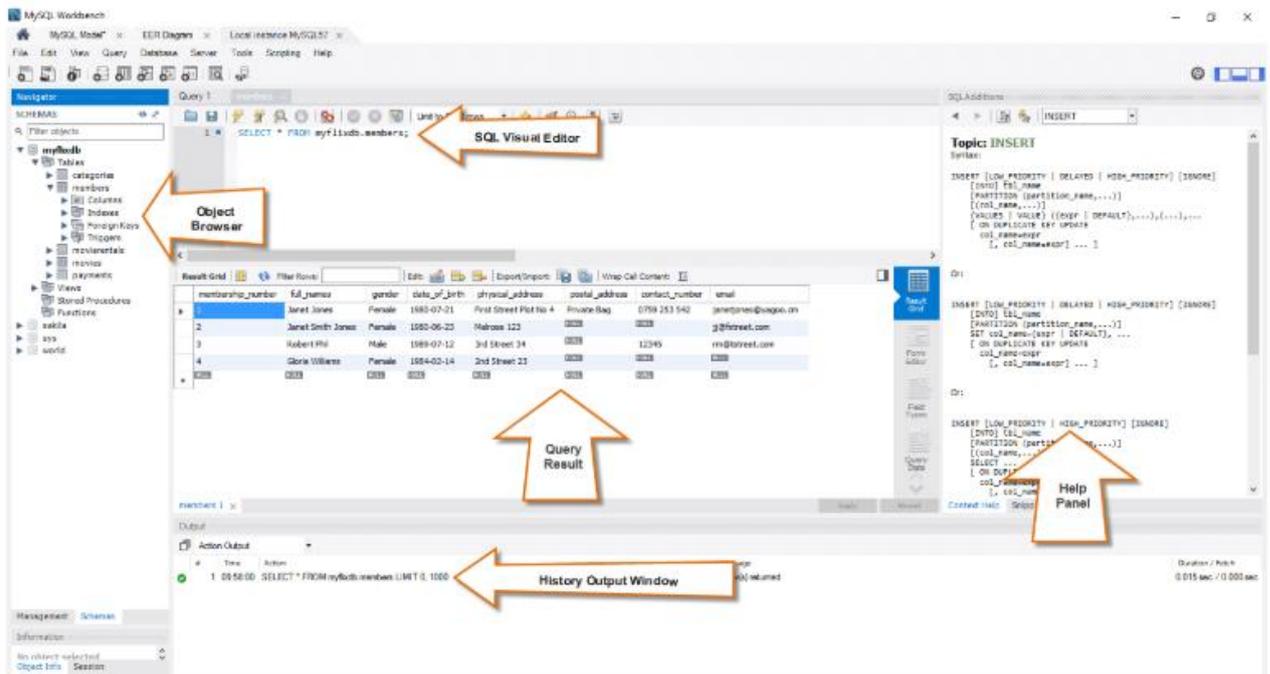


Рис.1.3 Вікно розробки MySQL Workbench

Підсумовуючи, MySQL Workbench – це потужний інструмент, який дає змогу фахівцям з баз даних ефективно проектувати, розробляти та адмініструвати бази даних MySQL.

2 РОЗДІЛ ПРОЕКТУВАННЯ ІНФОРМАЦІЙНО-ПОШУКОВОЇ СИСТЕМИ

2.1 Опис предметної області

Впровадження інформаційних систем у будь-яку сферу людської діяльності залежить від масштабу та складності завдань, які потребують автоматизації. Головною ціллю є створення єдиної інформаційної структури, яка б полегшувала внесення, пошук, зберігання та обмін інформацією для користувачів.

Завдяки впровадженню комп'ютеризації процеси взаємодії персоналу стають більш зручними, зрозумілими та контрольованими, що значно економить час. Тому інвестиції в комп'ютеризацію повинні ґрунтуватися на чітко окресленій меті, досягнення якої зробить певний процес максимально ефективним.

Предметною областю бакалаврської роботи є автоматизація діяльності служби таксі.

Процес пошуку автомобіля, який відповідає побажанням клієнта відбувається наступним чином.

Клієнт, якого ще не має в базі даних, повинен пройти реєстрацію, вказавши свій номер телефону, ПІБ та пароль.

Якщо клієнт є зареєстрованим, тоді він повинен пройти авторизацію, щоб зайти в свій акаунт. Після чого клієнт буде мати змогу поповнити свій баланс коштів або здійснити замовлення.

Якщо клієнт обере поле для поповнення рахунку, система попросить його ввести реквізити картки.

Якщо клієнт обере поле для замовлення автомобіля, система попросить його заповнити поля для замовлення. У випадку якщо у клієнта мало коштів на

рахунку система про це повідомить і не дасть здійснити замовлення. Якщо коштів достатньо, він отримає інформацію про те, коли приїде автомобіль.

Коли почнеться подорож буде активовано час поїздки, який в кінці зупиниться. В кінці поїздки клієнту буде дана змога виставити оцінку і написати коментар про водія і поїздку.

Також в системі є ще один актор – адміністратор. Він має змогу передивлятися список всіх клієнтів і всіх водіїв. Також адміністратор може додавати або видаляти с бази даних водіїв. Адміністратор також несе відповідальність за систему (рисунок 2.1).

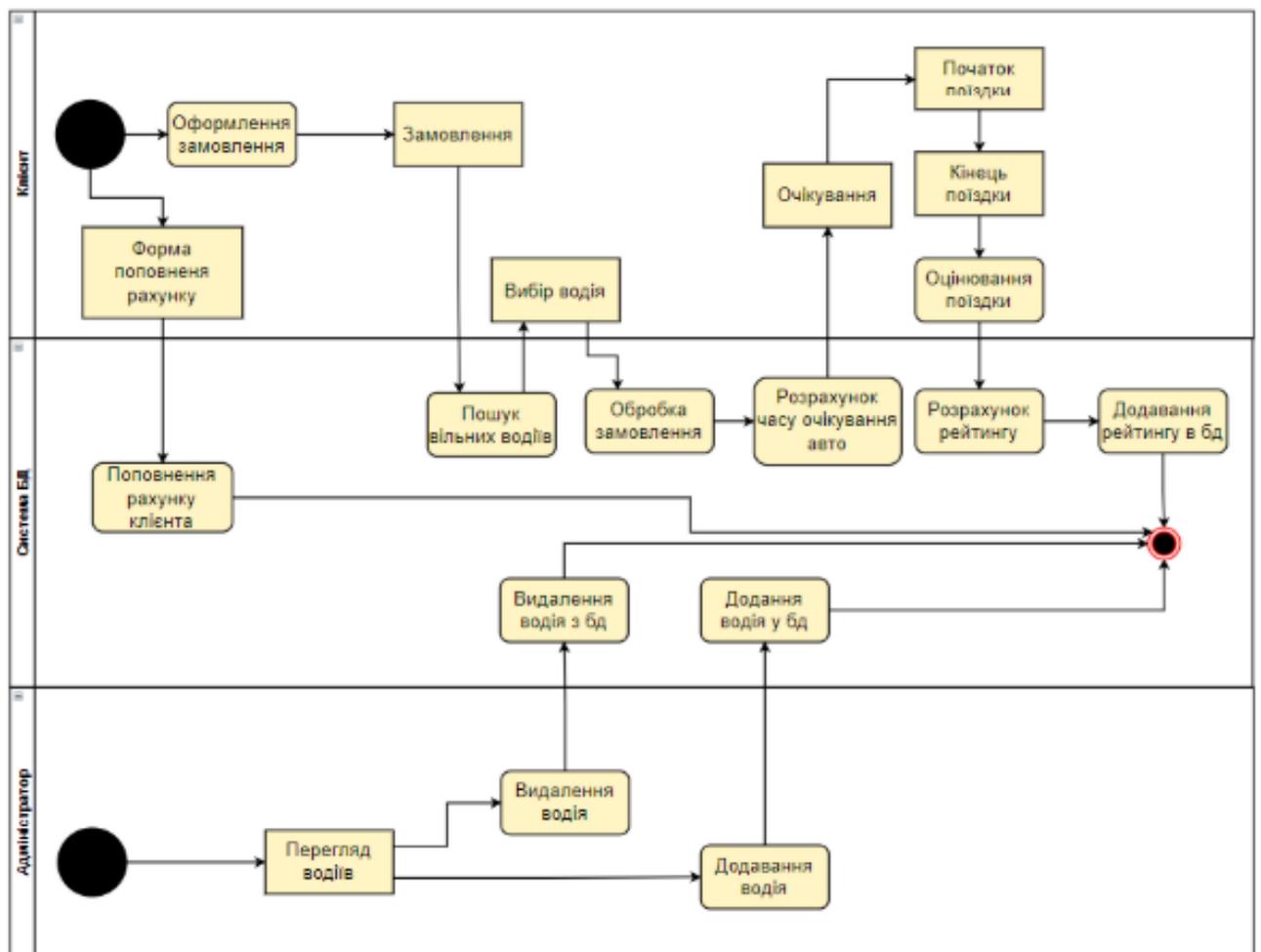


Рис. 2.1 Діаграма діяльності

На рисунку 2.2 показано діаграмі послідовності системи. Показано, як система взаємодіє з клієнтом.

Діаграма послідовності (Sequence Diagram) - це тип діаграми UML, яка використовується для моделювання взаємодії між об'єктами в системі. Вона візуалізує послідовність обміну повідомленнями між об'єктами, що допомагає зрозуміти динаміку роботи системи.

На діаграмі послідовності можна виділити наступні елементи:

Об'єкти: Об'єкти представлені вертикальними лініями, які називаються життєвими лініями. Кожна життєва лінія відповідає одному об'єкту.

Повідомлення: Повідомлення - це дії, які об'єкти надсилають один одному. Вони представлені горизонтальними стрілками, що з'єднують життєві лінії.

Час: Діаграма послідовності відображає послідовність подій, тому час розміщується зверху вниз. Чим вище розташоване повідомлення, тим раніше воно надсилається.

Активація: Об'єкт вважається активним, коли він виконує якесь повідомлення. Активний об'єкт зображується зафарбованою життєвою лінією.

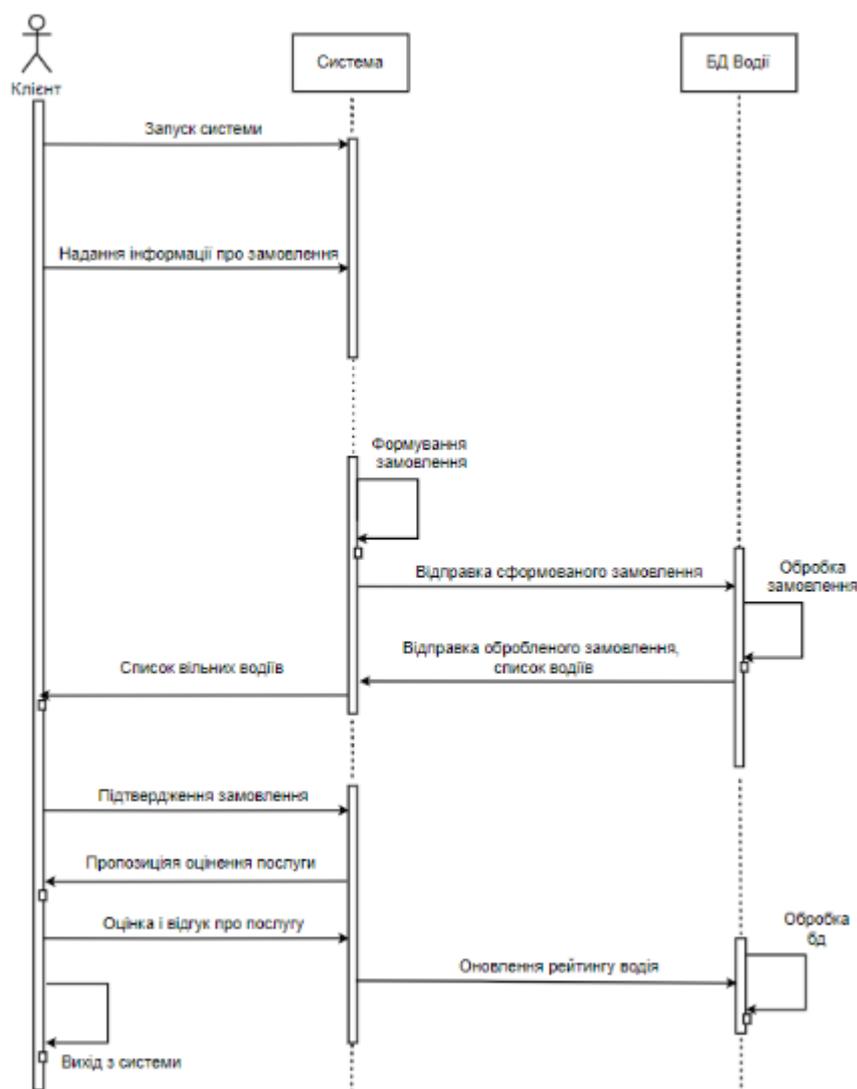


Рис. 2.2 Діаграма послідовності

Опис функціональної моделі

На рисунку 2.3. показана діаграма функціональної моделі.

Діаграма функціональної моделі - це тип діаграми потоку даних, яка використовується для моделювання функцій системи та їх взаємозв'язків. Її можна використовувати для опису бізнес-процесів, інформаційних систем та інших складних систем.

Діаграми функціональних моделей складаються з наступних компонентів:

Функції: Функції - це дії, які виконує система. Вони представлені прямокутниками.

Потоки даних: Потоки даних - це дані, які передаються між функціями. Вони представлені стрілками.

Зовнішні сутності: Зовнішні сутності - це особи або системи, які взаємодіють із системою. Вони представлені прямокутниками з пунктирною лінією.

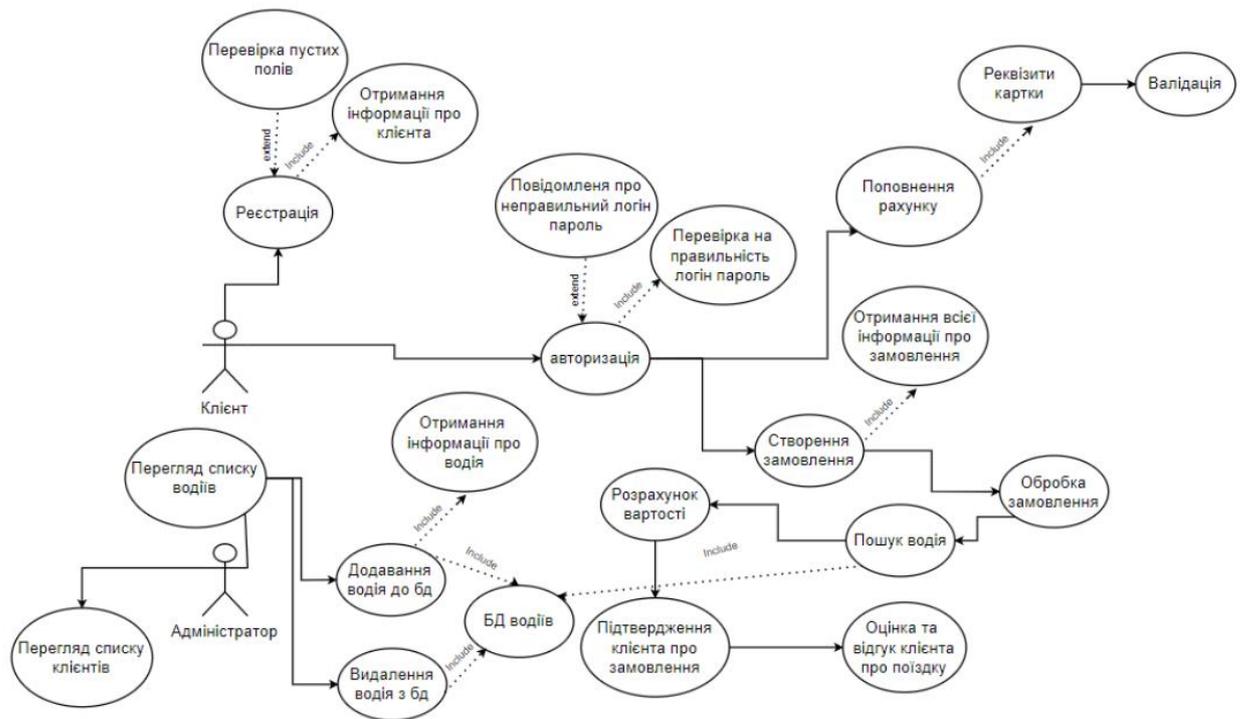


Рис. 2.3. Діаграма функціональної моделі

Діаграми функціональних моделей мають ряд переваг, таких як:
 Вони допомагають візуалізувати функції системи та їх взаємозв'язки.
 Вони допомагають визначити та усунути дублювання функцій.
 Вони допомагають покращити комунікацію між зацікавленими сторонами.
 Вони допомагають документувати вимоги до системи.

Функції системи

Актор	Варіант використання	Опис функцій варіантів використання	Пріоритет
Клієнт	Реєстрація	Користувач реєструється вводячи логін та пароль в отриману екранну форму.	Високий
Клієнт	Вхід в систему	Користувач отримує екранну форму, що складається з полів, які треба заповнити для потрапляння у систему. Ввести логін та пароль.	Високий
Клієнт	Заповнення анкети пошуку авто	Користувач заповняє форму необхідною інформацією, яка потрібна для пошуку авто	Високий
Клієнт	Виклик операції пошуку автомобіля	Користувач викликає операцію, натиснувши на необхідну кнопку. Користувач отримує список водіїв, які йому підходять	Високий
Клієнт	Заповнення полів поповнення рахунку	Користувач вводить у форму необхідні дані, пов'язані з його реквізитами картки.	Високий
Клієнт	Поповнення рахунку	Користувач викликає операцію, натиснувши на необхідну кнопку.	Високий
Адміністратор	Заповнення анкети додавання водія	Адміністратор заповнює анкету для додавання водія в базу даних	Високий
Адміністратор	Додавання водія	Адміністратор додає водія в базу даних, натиснувши на необхідну кнопку	Високий

На рис. 2.4 представлена структурна схема компонентів ІС

Структурна схема компонентів ІС (Information System Component Diagram) - це тип діаграми UML, яка використовується для візуалізації

архітектури інформаційної системи та взаємозв'язків між її компонентами. Вона показує, як різні компоненти системи взаємодіють один з одним, щоб досягти загальної мети.

На структурній схемі компонентів ІС можна виділити наступні елементи:

Компоненти - це основні модулі системи, які виконують певні функції. Вони представлені прямокутниками.

Інтерфейси - це точки доступу до компонентів, які дозволяють іншим компонентам взаємодіяти з ними. Вони представлені кружечками або овалами.

Залежності - показують зв'язки між компонентами. Вони представлені лініями, які з'єднують компоненти.

Тип зв'язку - може бути представлений різними типами стрілок, наприклад:

Спрямована залежність: Одна компонента залежить від іншої.

Неспрямована залежність: Компоненти взаємозалежні.

Асоціація: Компоненти пов'язані між собою, але не залежать один від одного

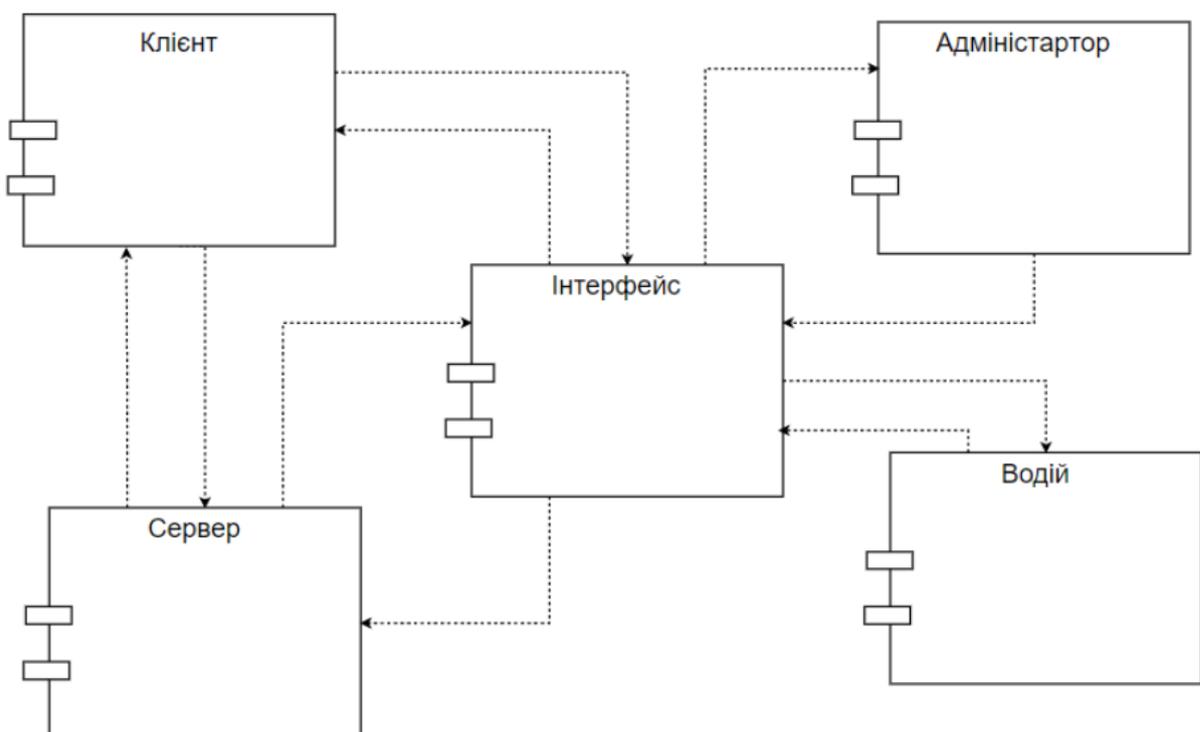


Рис. 2.4 Структурна схема компонентів ІС.

2.2 Програмна реалізація

Створюємо новий проект інформаційної системи за допомогою пункту головного меню «File» – «New» – «Project» – «Form (Windows Forms)» програмного забезпечення Microsoft Visual Studio.

Створюємо нову форму проекту «Form1» за допомогою пункту головного меню «Project» – «Add form...», яка буде відображати дані про вироби. У властивостях форми задаємо наступні параметри:

BackgroundImage - фонове зображення форми;

Text (Інформаційна система «Таксі») - встановлює ім'я форми;

Size (600; 500) - задає ширину та висоту модального вікна форми;

StartPosition (CenterScreen) - вказує на початкову позицію, з якою форма з'являється на екрані – по центру;

MinimizeBox (False) - вказує, чи буде доступна кнопка мінімізації вікна;

MaximizeBox (False) - вказує, чи буде доступна кнопка максимізації вікна в заголовку форми.



Рис. 2.5 Головне меню ІС «Таксі»

На створеній формі розміщуємо компонент MenuStrip з панелі інструментів. Він дозволяє створити головне меню для запуску інших створених форм.

Головне меню складається з наступних пунктів:

«Файл» включає в себе два підпункти:

Вхід в систему – забезпечує відкриття форми «*Вхід в систему*» у вигляді модального вікна (рис. 2.6).

Рис. 2.6 Форма «Вхід в систему» ІС «Таксі»

Дана форма дозволяє ввести ім'я користувача, пароль та сервер для підключення до БД. При натисненні на кнопку «ОК», відбувається перевірка на правильність введення даних та підключення до бази даних «**taxi**». Після успішного підключення на екран виводиться модальне вікно із повідомленням «Підключення відбулось успішно».

Вихід – забезпечує закриття ІС та з'єднання із БД. Для повторного входу потрібно авторизуватися, за допомогою підпункту «Вхід в систему».

«Довідники» включає в себе чотири підпункти:

Водії – забезпечує відкриття форми «Водії» у вигляді модального вікна.

	Код водія	Ім'я	Номер ліцензії	Модель авто	Номер авто	Кількість місць
▶	9	Fong Wai Man	663	nissan	QR4583BN	5
	2	Jason Phillips	6	audi	DL6253MB	4
	3	Jery Reed	960	opel	OK5389IJ	3
	5	Kwok Wing Fat	664	toyota	GI4573KX	5
	4	Lin Zhennan	378	ford	VD5959FP	4
	8	Lion Limbra	798	fiat	SF7110QS	7

Код водія: Ім'я та прізвище: Номер ліцензії:

Модель: Номер: Кількість місць:

Рис. 2.7 Форма «Водії» ІС «Таксі»

Форма дозволяє переглядати, додавати, редагувати, видаляти дані про водіїв.

Для редагування або видалення даних необхідно натиснути ПКМ та вибрати відповідний пункт із контекстного меню: «Редагувати», «Видалити».

	Код водія	Ім'я	Номер ліцензії	Модель авто	Номер авто	Кількість місць
▶	9	Fong Wai Man	663	nissan	QR4583BN	5
	2	Jason Phillips	6	audi	DL6253MB	4
	3	Jery Reed	960	opel	OK5389IJ	3

Редагувати
Видалити

Рис. 2.8 Контекстне меню

При натисненні на пункт меню «Редагувати», дані з'являються у відповідних полях нижче, а після внесення змін потрібно натиснути на кнопку «Зберегти/додати». Також за допомогою даної кнопки, можна додати новий запис до бази даних після внесення потрібних даних.

При натисненні на пункт меню «Видалити», обраний рядок із даними буде видалено із БД.

При натисненні на кнопку «Очистити», всі поля які містять дані будуть очищені.

Клієнти – забезпечує відкриття форми «Клієнти» у вигляді модального вікна.

The screenshot shows a modal window titled 'Клієнти' with a close button in the top right corner. Inside the window, there is a table with the following data:

	Код клієнта	Телефон	Ім'я	Email	Пароль
▶	42	555	aaa	@	22
	9	70-2537-4105	Anita West	westan@hotmail.com	Emvw10IAi
	38	330-322-5809	Chang Hui Mei	hmch331@gmail.com	8rgNU9MdGS
	31	70-0737-2947	Chris Nguyen	ngc@gmail.com	qCef8YIGKV
	22	5989 743649	Eddie Hill	hill9@gmail.com	kLDYUzmTgS
	32	158-3685-7534	Feng Jialun	fengjialu@gmail.com	VKSnSjpnB

Below the table, there are input fields for adding or editing a record:

Код клієнта: Телефон: Ім'я та прізвище:

Email: Пароль:

At the bottom, there are two buttons: 'Зберегти / додати' and 'Очистити'.

Рис. 2.9 Форма «Клієнти» ІС «Таксі»

Форма дозволяє переглядати, додавати, редагувати, видаляти дані про клієнтів.

Знижки – забезпечує відкриття форми «Знижки» у вигляді модального вікна.

Знижки

Виберіть промокод: TAXI100

	Код знижки	Ім'я	Промокод	Сума знижки
▶	16	Peng Zitao	TAXI100	10
	17	Jacob Olson	TAXI100	10
	18	Lai Chieh Lun	TAXI100	10
	19	Yokoyama Sakura	TAXI100	10
	20	Hao Zhiyuan	TAXI100	10

Виберіть клієнта:

Промокод:

Сума знижки:

Зберегти / додати Очистити

Рис.2.10 Форма «Знижки» ІС «Таксі»

Форма дозволяє переглядати, додавати, редагувати, видаляти дані про знижки.

Відгуки – забезпечує відкриття форми «Відгуки» у вигляді модального вікна.

	Код відгуку	Код поїздки	Рейтинг	Коментар
▶	1	1	242	Швидко
	2	7	566	Вчасно
	3	19	664	Повільно
	4	17	747	Безпечно
	5	8	199	Швидко, вчасно

Виберіть поїзду:

Рейтинг:

Коментар:

Рис. 2.11 Форма «Відгуки» ІС «Таксі»

Форма дозволяє переглядати, додавати, редагувати, видаляти дані про відгуки.

«Поїздки» – забезпечує відкриття форми «Поїздки» у вигляді модального вікна.

Поїздки

Код поїздки	Код клієнта	Код водія	Початкова точка	Кінцева точка	Статус поїздки
1	3	4	462 Ginza, Chuo-ku	935 Ginza, Chuo-ku	pending
2	7	10	651 Wall Street	687 Wall Street	in progress
3	5	3	254 Saidaiji Akodacho	567 Saidaiji Akodacho	completed

Клієнт: Водій: Статус:

Початкова точка: Кінцева точка:

Код оплати	Код поїздки	Вартість	Статус оплати
4	20	675	pending
30	20	10	cancelled
32	20	100	paid

Код оплати: Вартість:

Код поїздки: Статус:

Рис. 2.12 Форма «Відгуки» ІС «Таксі»

Форма дозволяє переглядати, додавати, редагувати, видаляти дані про поїздки.

За допомогою форми можна бачити початкову та кінцеву точки поїздки, а також її статус оплати та вартість.

«Довідка» – забезпечує відкриття форми «Довідка» у вигляді модального вікна.

Довідка

Інформаційна система "Taxi"

@ 2024 All rights reserved

Рис. 2.13 Форма «Відгуки» ІС «Таксі»

Форма дозволяє переглянути основну інформацію про ІС «Таксі».

2.3 Структура бази даних ІС «Taxi»

Для створення бази даних, використовуємо інструмент MySQL Workbench.

MySQL Workbench — інструмент для візуального проектування баз даних, що інтегрує проектування, моделювання, створення й експлуатацію БД в єдине безкоштовне оточення для системи баз даних MySQL. Є наступником DBDesigner 4 з FabForce.

MySQL Workbench пропонується в двох редакціях:[1]

Community Edition — поширюється під вільною ліцензією GNU GPL Standard Edition — доступна за щорічною передплатою. Ця версія включає в себе додаткові функції, які підвищують продуктивність розробників та адміністраторів БД.

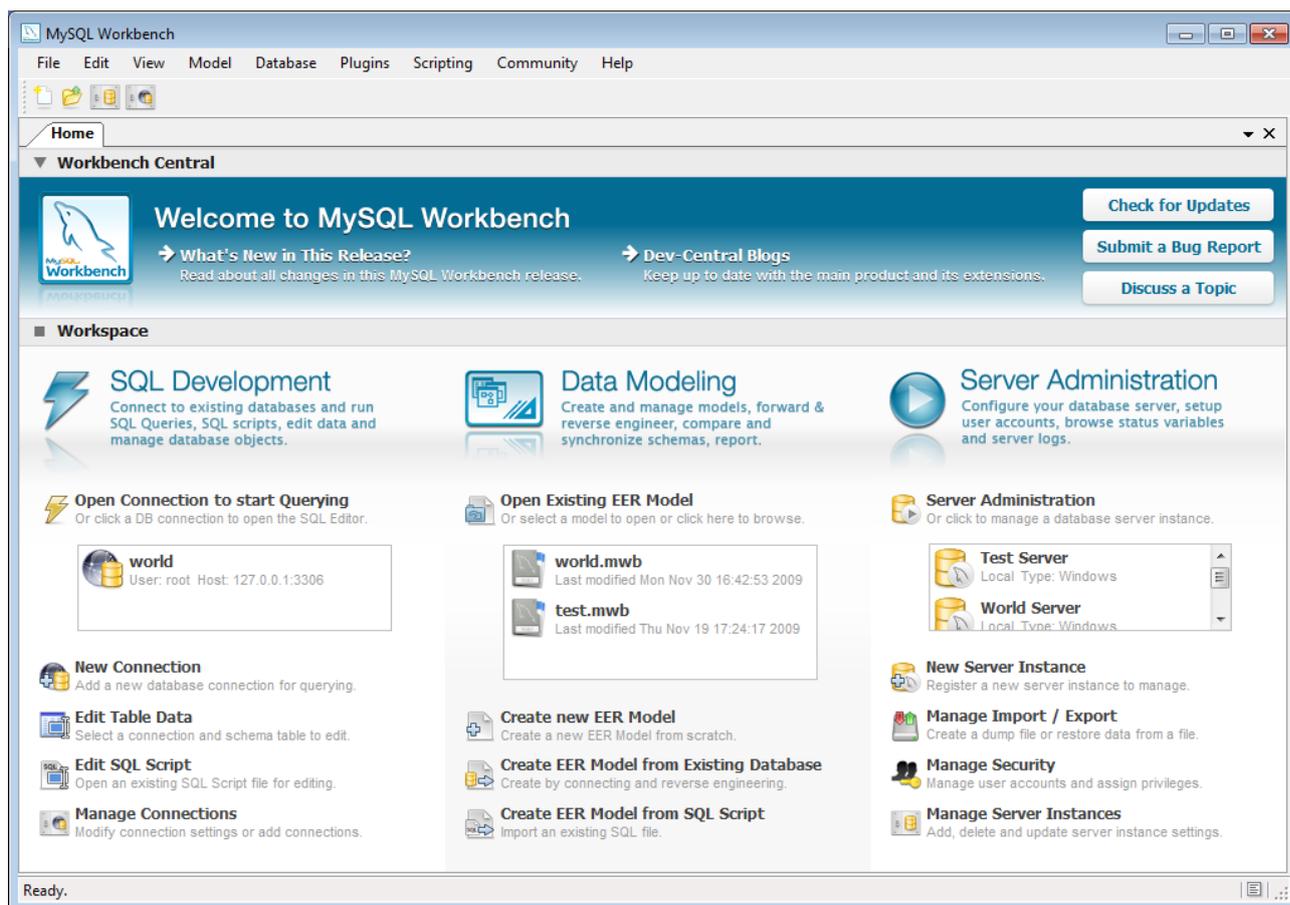


Рис.2.14 Загальний вигляд MySQL Workbench.

Створюємо нове підключення, для цього у вікні, що з'явилося вводимо наступні параметри: назву підключення, ім'я користувача, пароль та ім'я хоста.

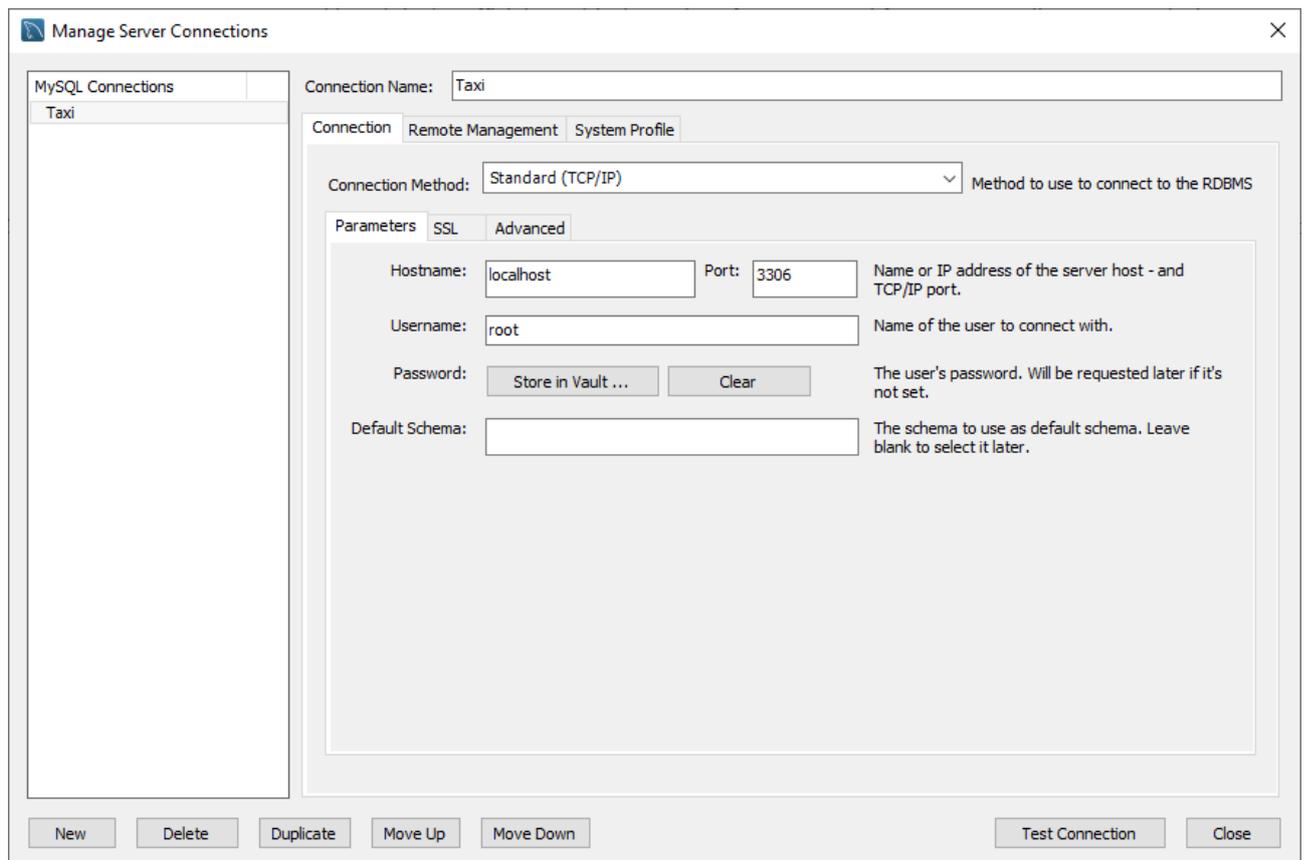


Рис 2.15 Вікно створення підключення до сервера бази даних MySQL

Після цього створюємо нову базу даних «**taxi**», використовуючи кнопку головного меню «**Create a new schema...**». У вікні, що з'явилося вводимо БД та обираємо кодування.



Рис 2.16 Створення бази даних

Далі створюємо таблицю «**customers**». Для цього виділяємо створену базу даних «**taxi**» із контекстного меню, натискаємо ПКМ та виконуємо команду «**Create table...**». У вікні, що з'явилося вводимо наступні дані.

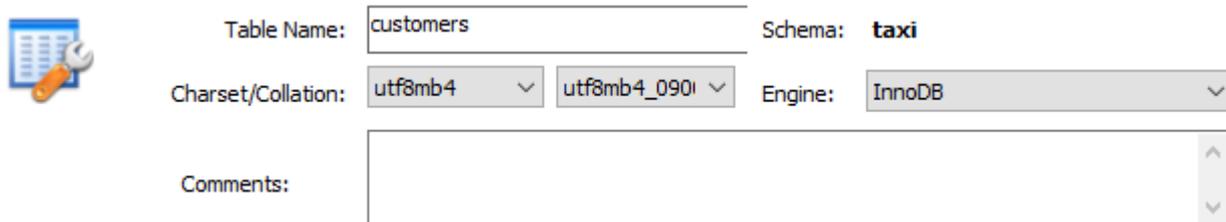


Table Name: customers Schema: taxi

Charset/Collation: utf8mb4 utf8mb4_0901 Engine: InnoDB

Comments:

Рис 2.17 Створення таблиці

Далі у таблиці «customers» створюємо наступні поля.

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
idCustomer	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>					
Phone	VARCHAR(50)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
Name	VARCHAR(50)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
Email	VARCHAR(50)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
Password	VARCHAR(50)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL

Рис 2.18 Поля таблиці «customers»

Для візуалізації логічної моделі за допомогою MySQL Workbench, виконаємо наступну команду: Database - Reverse Engineer... та слідуємо вказівкам.

Відповідно до наведеної схеми даних, створюємо ще 5 таблиць з необхідними полями (рис. 19).

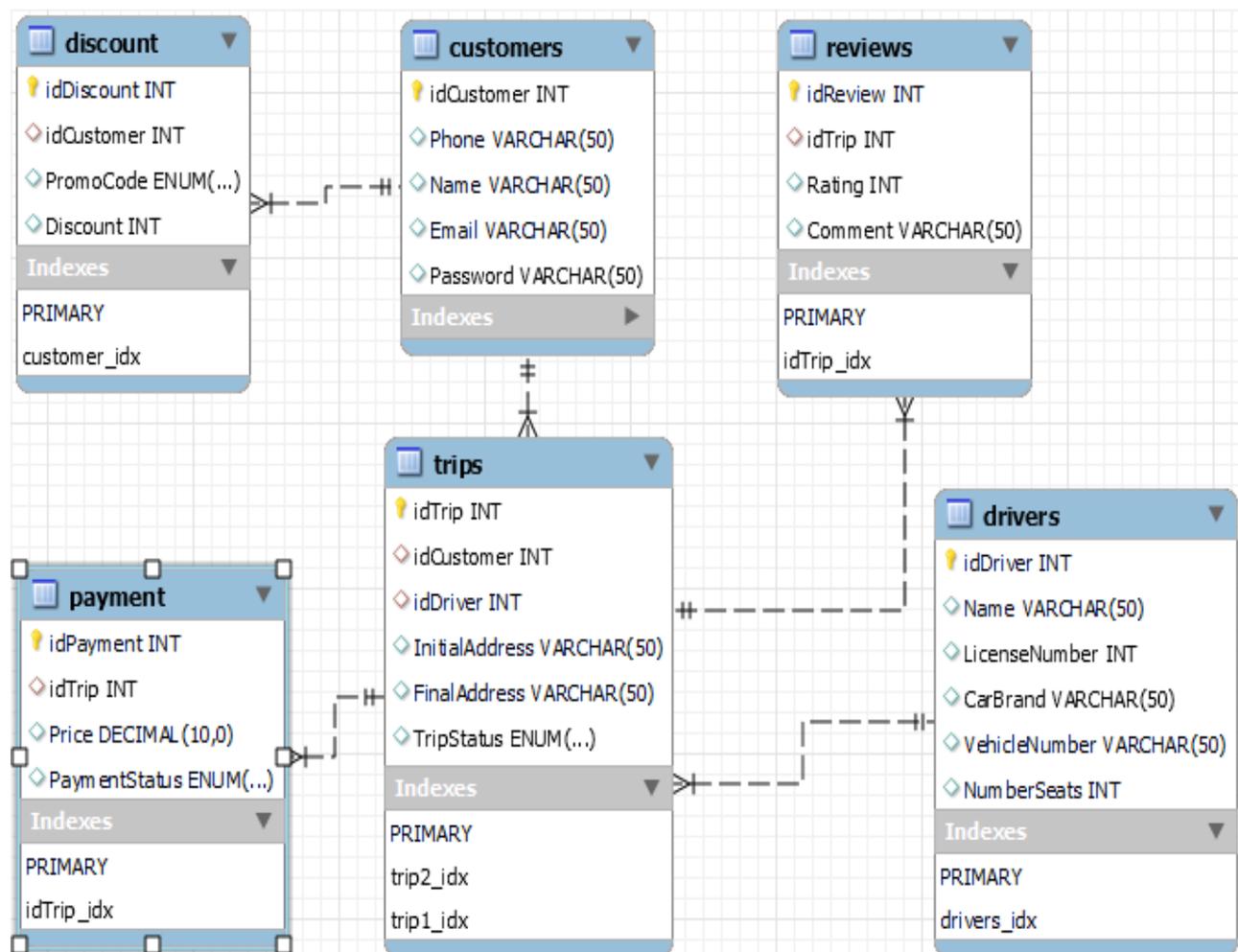


Рис. 2.19 Схема даних ІС «Таксі»

Після створення таблиць, їх структура буде наступною (рис. 6-10).

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
idDiscount	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>					
idCustomer	INT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
PromoCode	ENUM('TAXI40', 'TAX...)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
Discount	INT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL

Рис 2.20 Поля таблиці «discount»

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
idDriver	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>					
Name	VARCHAR(50)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
LicenseNumber	INT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
CarBrand	VARCHAR(50)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
VehicleNumber	VARCHAR(50)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
NumberSeats	INT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL

Рис 2.21 Поля таблиці «drivers»

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
idPayment	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>					
idTrip	INT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
Price	DECIMAL(10,0)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
PaymentStatus	ENUM('pending','pai...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	'pending'

Рис 2.22 Поля таблиці «payment»

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
idReview	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>					
idTrip	INT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
Rating	INT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
Comment	VARCHAR(50)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL

Рис 2.23 Поля таблиці «reviews»

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
idTrip	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>					
idCustomer	INT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
idDriver	INT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
InitialAddress	VARCHAR(50)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
FinalAddress	VARCHAR(50)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
TripStatus	ENUM('pending','acc...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	'pending'

Рис 2.24 Поля таблиці «trips»

Для заповнення таблиць даними використовуємо програмний продукт Navicat. Заповнюємо дані таблиці «customers», використовуючи переваги програми (рис. 2.25).

	idCustomer	Phone	Name	Email	Password
▶	1	614-365-6357	Martha Patel	patemarta@icloud.com	TSTwQ112kH
	2	70-0495-5125	Su Shihan	shihansu@icloud.com	TgW0CzaYj8
	3	(20) 0440 2771	Gong Zhennan	gongz815@outlook.com	L0u00HsYX
	4	838-099-7893	Yamashita Kasumi	ky1972@icloud.com	E90ZuWAlFP
	5	80-3052-3287	Shao Xiaoming	shaox501@gmail.com	CtTdc69FH5
	6	212-744-8001	Miyazaki Ikki	miyazaki84@outlook.com	HfoOQ9sZky
	7	28-051-8943	Hao Zhiyuan	zhiyuanhao@gmail.com	WKFFSTEOiv
	8	5075 733975	Zeng Zitao	zitao6@gmail.com	ouInRbNnF0
	9	70-2537-4105	Anita West	westan@hotmail.com	Ermvw 1OIAi
	10	312-118-7656	Yeow On Kay	yeonkay@icloud.com	bLcxeo5LgR

Рис 2.25 Дані таблиці «customers»

Відповідно до логічної моделі даних ІС «Taxi» (рис. 5), створюємо зв'язок між таблицями «trips» та «customers».

Для цього відкриваємо властивості таблиці «trips» (пункт контекстного меню «Alter Table») та переходимо на вкладку «Foreign Keys». У вікні, що з'явилося вводимо наступні параметри (рис. 2.26).

Foreign Key Name	Referenced Table	Column	Referenced Column	Foreign Key Options
trip1	`taxi`.`drivers`	<input type="checkbox"/> idTrip		On Update: CASCADE
trip2	`taxi`.`customers`	<input checked="" type="checkbox"/> idCustomer	idCustomer	On Delete: RESTRICT
		<input type="checkbox"/> idDriver		<input type="checkbox"/> Skip in SQL generation
		<input type="checkbox"/> InitialAddress		
		<input type="checkbox"/> FinalAddress		
		<input type="checkbox"/> TripStatus		

Рис 2.26 Зв'язок між таблицями «trips» та «customers»

Відповідно до схеми даних ІС «Taxi» (рис. 5) створюємо аналогічні зв'язки між іншими таблицями (рис. 13-16).

Foreign Key Name	Referenced Table	Column	Referenced Column	Foreign Key Options
review	`taxi`.`trips`	<input type="checkbox"/> idReview		On Update: CASCADE
		<input checked="" type="checkbox"/> idTrip	idTrip	On Delete: RESTRICT
		<input type="checkbox"/> Rating		<input type="checkbox"/> Skip in SQL generation
		<input type="checkbox"/> Comment		

Рис 2.27 Зв'язок між таблицями «reviews» та «trips»

Foreign Key Name	Referenced Table	Column	Referenced Column	Foreign Key Options
payment	`taxi`.`trips`	<input type="checkbox"/> idPayment		On Update: CASCADE
		<input checked="" type="checkbox"/> idTrip	idTrip	On Delete: RESTRICT
		<input type="checkbox"/> Price		<input type="checkbox"/> Skip in SQL generation
		<input type="checkbox"/> PaymentStatus		

Рис 2.28 Зв'язок між таблицями «payment» та «trips»

Foreign Key Name	Referenced Table	Column	Referenced Column	Foreign Key Options
discount	`taxi`.`customers`	<input type="checkbox"/> idDiscount		On Update: RESTRICT
		<input checked="" type="checkbox"/> idCustomer	idCustomer	On Delete: RESTRICT
		<input type="checkbox"/> PromoCode		<input type="checkbox"/> Skip in SQL generation
		<input type="checkbox"/> Discount		

Рис 2.29 Зв'язок між таблицями «discount» та «customers»

Foreign Key Name	Referenced Table	Column	Referenced Column	Foreign Key Options
trip1	`taxi`.`drivers`	<input type="checkbox"/> idTrip		On Update: CASCADE
trip2	`taxi`.`customers`	<input type="checkbox"/> idCustomer		On Delete: RESTRICT
		<input checked="" type="checkbox"/> idDriver	idDriver	<input type="checkbox"/> Skip in SQL generation
		<input type="checkbox"/> InitialAddress		
		<input type="checkbox"/> FinalAddress		
		<input type="checkbox"/> TripStatus		

Рис 2.30 Зв'язок між таблицями «trips» та «drivers»

Після створення зав'язків заповнюємо дані решти таблиць наступними записами (рис. 2.31-2.33).

	idDiscount	idCustomer	PromoCode	Discount		idPayment	idTrip	Price	PaymentStatus
▶	1	15	TAXI40	81	▶	1	10	598	pending
	2	5	TAXI40	82		2	8	991	paid
	3	1	TAXI40	83		3	4	0	paid
	4	6	TAXI40	84		4	20	675	pending
	5	10	TAXI40	85		5	1	120	cancelled
	6	12	TAXI60	91		6	2	493	pending
	7	14	TAXI60	92		7	11	150	paid
	8	4	TAXI60	93		8	15	510	pending
	9	3	TAXI60	94		9	7	865	cancelled
	10	19	TAXI60	95		10	5	8	pending

	idReview	idTrip	Rating	Comment
▶	1	1	242	Швидко
	2	7	566	Вчасно
	3	19	664	Повільно
	4	17	747	Безпечно
	5	8	199	Швидко, вчасно
	6	18	942	Запізнився
	7	4	811	Є оплата картою
	8	10	918	NULL
	9	5	728	NULL
	22	8	56	

Рис 2.31 Дані таблиць «discount», «payment» та «reviews»

	idDriver	Name	LicenseNumber	CarBrand	VehicleNumber	NumberSeats
▶	1	Tao Siu Wai	258	bmw	HO9523WA	10
	2	Jason Phillips	6	audi	DL6253MB	4
	3	Jerry Reed	960	opel	OK5389IJ	3
	4	Lin Zhennan	378	ford	VD5959FP	4
	5	Kwok Wing Fat	664	toyota	GI4573KX	5
	6	Tao Wai San	434	chery	FD5290JJ	4
	7	Stephen Snyder	67	kia	MQ5670JK	6
	8	Lion Limbra	798	fiat	SF7110QS	7
	9	Fong Wai Man	663	nissan	QR4583BN	5
	10	Yokoyama Yuto	49	dacia	UH5493BD	4

Рис 2.32 Дані таблиці «drivers»

	idTrip	idCustomer	idDriver	InitialAddress	FinalAddress	TripStatus
▶	1	3	4	462 Ginza, Chuo-ku	935 Ginza, Chuo-ku	pending
	2	7	10	651 Wall Street	687 Wall Street	in progress
	3	5	3	254 Saidaiji Akodacho	567 Saidaiji Akodacho	completed
	4	8	3	389 Nostrand Ave	345 Nostrand Ave	in progress
	5	10	8	20 Kawagishicho, Mizuho	15 Kawagishicho, Mizuho	pending
	6	3	8	371 Lower Temple Street	654 Lower Temple Street	in progress
	7	10	4	488 Grape Street	555 Grape Street5	in progress
	8	14	6	17 Kikusui 3 Jo, Shiroishi	5 Kikusui 3 Jo, Shiroishi	accepted
	9	16	2	766 Middle Huaihai Road	854 Middle Huaihai Road	completed
	10	25	8	800 Spring Gardens	245 Spring Gardens	accepted

Рис 2.33 Дані таблиці «trips»

ВИСНОВКИ

В ході виконання даної бакалаврської роботи було проведено проектування та реалізацію інформаційно-пошукової системи (ІПС) замовлень автопослуг для приватної фірми засобами сучасних технологій баз даних.

В рамках дослідження було:

Проведено аналіз існуючих ІПС замовлень автопослуг.

Розроблено концептуальну модель ІПС, яка враховує специфічні потреби та вимоги приватної фірми.

Вибрано програмне забезпечення та технології для реалізації ІПС.

Розроблено базу даних для ІПС.

Розроблено програмне забезпечення ІПС.

Проведено тестування та впровадження ІПС.

Сучасні технології проектування ПЗ дозволили створити чітку структуру та архітектуру системи, що полегшило її розробку та тестування.

Microsoft Visual Studio було використано як інтегроване середовище розробки (IDE) для написання коду програми на мові програмування [мова програмування]. Visual Studio надав потужний редактор коду, відладчик та інструменти для тестування, що значно спростило процес розробки.

MySQL Workbench використовувався як графічний інтерфейс користувача (GUI) для роботи з базою даних MySQL. Завдяки MySQL Workbench можна було легко створювати та редагувати об'єкти бази даних, виконувати SQL-запити та переглядати дані.

Переваги впровадження розробленої ІПС

Впровадження розробленої в рамках дипломної роботи ІПС дозволяє приватним фірмам, які надають автопослуги, отримати ряд суттєвих переваг, таких як:

Зниження витрат на обслуговування клієнтів: ІПС може автоматизувати багато рутинних завдань, пов'язаних з прийомом та обробкою замовлень, що дозволяє економити час та ресурси персоналу.

Підвищення якості обслуговування клієнтів: ІПС може забезпечити більш швидке та точне виконання замовлень, а також надати клієнтам можливість відстежувати статус своїх замовлень в режимі реального часу.

Збільшення конкурентоспроможності: Впровадження ІПС може дати приватній фірмі конкурентну перевагу на ринку, завдяки більш високому рівню обслуговування клієнтів та кращому контролю над процесом виконання замовлень.

Підвищення лояльності клієнтів: Завдяки кращому обслуговуванню та зручності користування ІПС, клієнти будуть більш лояльними до фірми.

Покращення збору та аналізу даних: ІПС збирає та зберігає дані про замовлення, клієнтів та послуги, що дозволяє фірмі проводити аналіз та приймати більш обґрунтовані управлінські рішення.

Розроблена ІПС може бути легко масштабована та адаптована до потреб різних приватних фірм, які надають автопослуги. Це пов'язано з тим, що ІПС має модульну структуру, що дозволяє додавати нові функціональні можливості та змінювати конфігурацію системи відповідно до потреб конкретної фірми.

Впровадження ІПС є економічно ефективним інвестицією для приватних фірм, які надають автопослуги. Це пов'язано з тим, що витрати на розробку та впровадження ІПС окупаються протягом короткого періоду часу за рахунок економії коштів на обслуговуванні клієнтів та підвищення конкурентоспроможності.

Практична значимість

Розроблена ІПС може бути використана в інших приватних фірмах, які надають автопослуги. Її впровадження дозволить їм підвищити рівень обслуговування клієнтів, знизити витрати та збільшити конкурентоспроможність.

Рекомендації щодо подальших досліджень

В подальшому дослідження в даній сфері можуть бути спрямовані на:

Розробку нових методів та технологій проектування та реалізації ІПС замовлень автопослуг.

Вдосконалення функціональних можливостей ІПС.

Інтеграцію ІПС з іншими системами, наприклад, з системами бухгалтерського обліку та управління запасами.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Лаурет А. Дизайн Web APIs / Arnaud Lauret. – США: Manning Publications Co., 2019.
2. Інформаційні системи і технології в економіці: Посібник. /За ред. В.С. Пономаренка. - К.: ВЦ. “Академія”, 2002.
3. Зацеркляний М.М., Мельников О.Ф. Інформаційні системи і технології у фінансово-кредитних установах: Посібник.- К.: Професіонал, 2006. -432 с.
4. Еміт С. SPA дизайн та архітектура: розуміння веб-додатків на одній сторінці / Скот Еміт. – Shelter Island: Manning Publications Co., 2016.
5. Monteiro F. Learning Single-page Web Application Developmen // Monteiro F. — Packt Publishing, 2014 – 214 с.
6. Masse M. REST API Design Rulebook: Designing Consistent RESTful Web Service Interfaces / Mark Masse. – Tokyo: O`Reilly Media Inc.. – 116 с.
7. Huber T. C. Getting Started with TypeScript: Includes Introduction to Angular / Thomas Claudius Huber., 2017.
8. Freeman A. Pro Angular 6 / Adam Freeman. – London: apress, 2018. – 804 с.
9. Freeman A. Essential TypeScript: From Beginner to Pro / Adam Freeman. – London: apress. – 546 с.
10. Cherny B. Programming TypeScript: Making Your JavaScript Applications Scale / Boris Cherny. – United States of America, 2019. – 324 с.
11. Туловский А. Інструкція розробника Angular, 2013.
12. «.NET Framework Guide» [Електронний ресурс] – Режим доступу до ресурсу: <https://docs.microsoft.com/en-us/dotnet/framework/>
13. «Common Language Runtime (CLR) overview» [Електронний ресурс] – Режим доступу до ресурсу: <https://docs.microsoft.com/en-us/dotnet/standard/clr>
14. «Visual Studio» [Електронний ресурс] – Режим доступу до ресурсу: <https://visualstudio.microsoft.com/>

15.«Windows Forms overview» [Электронный ресурс] – Режим доступа до ресурсу:

<https://docs.microsoft.com/enus/dotnet/framework/winforms/windows-forms-overview>

16.«Entity Framework Documentation» [Электронный ресурс] – Режим доступа до ресурсу: <https://docs.microsoft.com/en-us/ef/>

ДОДАТКИ

Додаток А

Відображення даних про водіїв у вигляді таблиці

```
private void LoadData()
{
    query = @"SELECT idDriver, Name, LicenseNumber, CarBrand, VehicleNumber,
NumberSeats FROM drivers ORDER BY Name ASC";

    try
    {
        Form1.OpenConnection();
        MySqlDataAdapter da = new MySqlDataAdapter(query, Form1.con);
        DataSet ds = new DataSet();
        da.Fill(ds, "drivers");
        dataGridView1.DataSource = ds;
        dataGridView1.DataMember = "drivers";

        dataGridView1.Columns[0].HeaderText = "Код водія";
        dataGridView1.Columns[1].HeaderText = "Ім'я";
        dataGridView1.Columns[2].HeaderText = "Номер ліцензії";
        dataGridView1.Columns[3].HeaderText = "Модель авто";
        dataGridView1.Columns[4].HeaderText = "Номер авто";
        dataGridView1.Columns[5].HeaderText = "Кількість місць";

        dataGridView1.Columns[0].Width = 60;
        dataGridView1.Columns[1].Width = 120;
        dataGridView1.Columns[2].Width = 60;
        dataGridView1.Columns[3].Width = 100;
        dataGridView1.Columns[4].Width = 80;
        dataGridView1.Columns[5].Width = 80;

        dataGridView1.Refresh();
    }
    catch (MySqlException ex)
    {
        MessageBox.Show("Error: " + ex.Message);
    }
    finally
    {
        Form1.CloseConnection(); // Закриття підключення тут
    }
}
```

Функціонування кнопки «Зберегти/додати»

```

private void button1_Click(object sender, EventArgs e)
{
    string query2;
    if (insert_mode)
    {
        query2 = "INSERT INTO drivers (Name, LicenseNumber, CarBrand, VehicleNumber,
NumberSeats) VALUES (@Name, @LicenseNumber, @CarBrand, @VehicleNumber, @NumberSeats)";
    }
    else
    {
        query2 = "UPDATE drivers SET Name = @Name, LicenseNumber = @LicenseNumber,
CarBrand = @CarBrand, VehicleNumber = @VehicleNumber, NumberSeats = @NumberSeats WHERE
idDriver = @idDriver";
    }
    try
    {
        Form1.OpenConnection();
        cmd = new MySqlCommand(query2, Form1.con);
        cmd.Parameters.AddWithValue("@Name", textBox2.Text);
        cmd.Parameters.AddWithValue("@LicenseNumber", textBox3.Text);
        cmd.Parameters.AddWithValue("@CarBrand", textBox4.Text);
        cmd.Parameters.AddWithValue("@VehicleNumber", textBox5.Text);
        cmd.Parameters.AddWithValue("@NumberSeats", textBox6.Text);
        if (!insert_mode)
        {
            cmd.Parameters.AddWithValue("@idDriver",
Convert.ToInt32(dataGridView1.CurrentRow.Cells[0].Value));
        }
        cmd.ExecuteNonQuery(); // Оновлення даних у DataGridView
        MySqlDataAdapter da = new MySqlDataAdapter(query, Form1.con);
        DataSet ds = new DataSet(); da.Fill(ds, "drivers");
        dataGridView1.DataSource = ds;
        dataGridView1.Refresh();
        // Вибір нової або оновленої строки
        for (int i = 0; i < dataGridView1.RowCount; i++)
        {
            dataGridView1.Rows[i].Selected = false;
            if
(dataGridView1.Rows[i].Cells[1].Value.ToString().Contains(textBox2.Text))
            {
                dataGridView1.Rows[i].Selected = true;
                break;
            }
        }
        dataGridView1.Enabled = true; // Очищення текстових полів
        textBox1.Text = ""; textBox2.Text = "";
        textBox3.Text = ""; textBox4.Text = "";
        textBox5.Text = ""; textBox6.Text = "";
        insert_mode = true; // Повернення в режим вставки
    }
    catch (MySqlException ex)
    {
        MessageBox.Show("Error: " + ex.Message);
    }
    finally
    {
        Form1.CloseConnection();
    }
}

```

Перевірка на правильність введення даних та підключення до БД «taxi»

```
private void button1_Click(object sender, EventArgs e)
{
    Form1.con.Close();
    string user_name = textBox1.Text;
    string user_password = textBox2.Text;
    string user_connection_string;
    string user_server = textBox3.Text;

    if (string.IsNullOrEmpty(user_password))
    {
        user_connection_string = $"server={user_server};User Id={user_name};Persist
Security Info=True; database=taxi";
    }
    else
    {
        user_connection_string = $"server={user_server};User
Id={user_name};password={user_password};Persist Security Info=True; database=taxi";
    }

    try
    {
        Form1.con.ConnectionString = user_connection_string;
        Form1.OpenConnection();
        MessageBox.Show("Підключення відбулось успішно", "Підключення",
        MessageBoxButtons.OK, MessageBoxIcon.Information);
        this.Close();
    }
    catch (MySqlException ex)
    {
        MessageBox.Show("Помилка при підключенні: " + ex.Message, "Увага",
        MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}
```

Відображення списку клієнтів та їх знижки за обраним промокодом

```
private void comboBox1_SelectionChangeCommitted(object sender, EventArgs e)
{
    LoadData();
}
private void LoadData()
{
    query = @"
        SELECT
            discount.idDiscount,
            customers.`Name`,
            discount.PromoCode,
            discount.Discount,
            discount.idCustomer
        FROM
            customers
            INNER JOIN
            discount
            ON
                customers.idCustomer = discount.idCustomer
        WHERE
            discount.PromoCode = @PromoCode";

    try
    {
        Form1.OpenConnection();
        MySqlCommand cmd = new MySqlCommand(query, Form1.con);
        cmd.Parameters.AddWithValue("@PromoCode", comboBox1.SelectedValue);
        MySqlDataAdapter da = new MySqlDataAdapter(cmd);
        DataSet ds = new DataSet();
        da.Fill(ds, "discount");
        dataGridView1.DataSource = ds.Tables["discount"];

        ConfigureDataGridView();
    }
    catch (MySqlException ex)
    {
        MessageBox.Show("Error: " + ex.Message);
    }
    finally
    {
        Form1.CloseConnection();
    }
}
private void ConfigureDataGridView()
{
    dataGridView1.Columns[0].HeaderText = "Код знижки";
    dataGridView1.Columns[1].HeaderText = "Ім'я";
    dataGridView1.Columns[2].HeaderText = "Промокод";
    dataGridView1.Columns[3].HeaderText = "Сума знижки";
    dataGridView1.Columns[4].Visible = false;
    dataGridView1.Columns[0].Width = 60;
    dataGridView1.Columns[1].Width = 120;
    dataGridView1.Columns[2].Width = 70;
    dataGridView1.Columns[3].Width = 70;

    dataGridView1.Refresh();
}
```

Відображення списку оплат клієнта при натисненні на відповідну поїзтку

```

private void dataGridView1_CellMouseClick(object sender,
DataGridViewCellMouseEventArgs e)
{
    if (e.RowIndex >= 0) // Переконайтеся, що користувач натиснув на дійсний рядок
    {
        int selectedTripId =
Convert.ToInt32(dataGridView1.Rows[e.RowIndex].Cells["idTrip"].Value);

        // Заповнить `textBox4` значенням `idTrip`
textBox4.Text = selectedTripId.ToString();
UpdateDataGV2(selectedTripId);
// Отримайте idPayment для вибраної поїздки
int idPayment = GetPaymentIdForTrip(selectedTripId);

        // Заповнить `textBox3` значенням idPayment
textBox3.Text = idPayment.ToString();
    }
}
private int GetPaymentIdForTrip(int tripId)
{
    int paymentId = 0;

    // Пошук idPayment для вибраної поїздки
string query = "SELECT idPayment FROM payment WHERE idTrip = @idTrip";

    try
    {
        Form1.OpenConnection();
        MySqlCommand cmd = new MySqlCommand(query, Form1.con);
        cmd.Parameters.AddWithValue("@idTrip", tripId);

        object result = cmd.ExecuteScalar();
        if (result != null)
        {
            paymentId = Convert.ToInt32(result);
        }
    }
    catch (MySqlException ex)
    {
        MessageBox.Show("Error: " + ex.Message);
    }
    finally
    {
        Form1.CloseConnection();
    }

    return paymentId;
}
private void UpdateDataGV2(int idTrip)
{
    string query10 = @"SELECT idPayment, idTrip, Price, PaymentStatus FROM payment
WHERE idTrip = @idTrip";

    try
    {
        Form1.OpenConnection();
        MySqlCommand cmd = new MySqlCommand(query10, Form1.con);
        cmd.Parameters.AddWithValue("@idTrip", idTrip);
    }
}

```

```
MySQLDataAdapter da = new MySQLDataAdapter(cmd);
DataSet ds = new DataSet();
da.Fill(ds, "payment");
dataGridView2.DataSource = ds;
dataGridView2.DataMember = "payment";

dataGridView2.Columns[0].HeaderText = "Код оплати";
dataGridView2.Columns[1].HeaderText = "Код поїздки";
dataGridView2.Columns[2].HeaderText = "Вартість";
dataGridView2.Columns[3].HeaderText = "Статус оплати";

dataGridView2.Columns[0].Width = 60;
dataGridView2.Columns[1].Width = 60;
dataGridView2.Columns[2].Width = 60;
dataGridView2.Columns[3].Width = 60;

dataGridView2.Refresh();
}
catch (MySqlException ex)
{
    MessageBox.Show("Error: " + ex.Message);
}
finally
{
    Form1.CloseConnection();
}
}
```