

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені ІГОРЯ СІКОРСЬКОГО»
ФАКУЛЬТЕТ ІНФОРМАТИКИ ТА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ
Кафедра автоматизованих систем обробки інформації та управління

До захисту допущено:

В.о. завідувача кафедри

(підпис) Олександр ПАВЛОВ
(вл.ім'я, прізвище)

“ ____ ” _____ 2021 р.

Дипломний проєкт
на здобуття ступеня бакалавра

**за освітньо-професійною програмою «Інформаційні управляючі
системи та технології»
спеціальності 126 «Інформаційні системи та технології»**

*на тему: «Інформаційна система з підтримки діяльності служби
таксі »*

Виконав:

студент IV курсу, групи ІС-72

Цеюков Артур Володимирович
(прізвище, ім'я, по батькові)

(підпис)

Керівник

доц., к.т.н., доц. Тєлишева Тамара Олексіївна
(посада, науковий ступінь, вчене звання, прізвище, ім'я, по батькові)

(підпис)

**Консультант з
графічної
документації**

доц., к.т.н. Новінський Валерій Петрович
(посада, науковий ступінь, вчене звання, прізвище, ім'я, по батькові)

(підпис)

Рецензент

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

(підпис)

Засвідчую, що у цьому дипломному проєкті
немає запозичень з праць інших авторів без
відповідних посилань.

Студент (-ка) _____
(підпис)

Київ – 2021 року

**Національний технічний університет України
“Київський політехнічний інститут імені Ігоря Сікорського”**

Факультет (інститут) інформатики та обчислювальної техніки
(повна назва)

Кафедра автоматизованих систем обробки інформації та управління
(повна назва)

Рівень вищої освіти – перший (бакалаврський)

Спеціальність – 126 «Інформаційні системи та технології»

Освітньо-професійна програма «Інформаційні управляючі системи та технології»

ЗАТВЕРДЖУЮ

В.о. завідувача кафедри

_____ Олександр ПАВЛОВ
(підпис) (вл.ім'я, прізвище)

“ ” _____ 2021 р.

**ЗАВДАННЯ
на дипломний проєкт студенту**

Цеюкову Артуру Володимировичу
(прізвище, ім'я, по батькові)

1. Тема проєкту «Інформаційна система з підтримки діяльності
служби таксі»

керівник проєкту Телишева Тамара Олексіївна, к.т.н., доцент
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від “ 11 ” травня 2021 р. № 1139-с

2. Термін подання студентом проєкту “04”червня 2021 року

3. Вихідні дані до проєкту

Технічне завдання

4. Зміст пояснювальної записки

1. Загальні положення: основні визначення та терміни, опис предметного середовища, огляд ринку програмних продуктів, постановка задачі

2. Інформаційне забезпечення: вхідні дані, вихідні дані, опис структури бази даних

3. Математичне забезпечення: змістовна та математична постановки задачі, обґрунтування та опис методу розв'язання

4. Програмне та технічне забезпечення: засоби розробки, вимоги до технічного забезпечення, архітектура програмного забезпечення, побудова звітів

5. Технологічний розділ: керівництво користувача, методика випробувань програмного продукту

5. Перелік графічного матеріалу

1. Схема структурна варіантів використання

2. Схема структурна компонентів програмного забезпечення

3. Схема бази даних

4. Схема структурна класів програмного забезпечення

5. Схема структурна активності

6. Креслення вигляду екранних форм

6. Консультанти розділів проєкту

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання «7» квітня 2021 року

Календарний план

№ з/п	Назва етапів виконання дипломного проєкту	Термін виконання етапів проєкту	Примітка
1.	Вивчення рекомендованої літератури	17.04.2021	
2.	Аналіз існуючих методів розв'язання задачі	23.04.2021	
3.	Постановка та формалізація задачі	23.04.2021	
4.	Розробка інформаційного забезпечення	30.04.2021	
5.	Алгоритмізація задачі	04.05.2021	
6.	Обґрунтування використовуваних технічних засобів	10.05.2021	
7.	Розробка програмного забезпечення	14.05.2021	
8.	Налагодження програми	01.06.2021	
9.	Виконання графічних документів	01.06.2021	
10.	Оформлення пояснювальної записки	15.05.2021	
11.	Подання ДП на попередній захист	14.05.2021	
12.	Подання ДП на основний захист	04.06.2021	
13.	Подання ДП рецензенту	07.06.2021	

Студент

Артур ЦЕЮКОВ

Керівник

Тамара ТЄЛИШЕВА

[illegible]

Пояснювальна записка до дипломного проєкту

на тему: Інформаційна система з підтримки діяльності служби таксі

Київ – 2021 року

АНОТАЦІЯ

Структура та обсяг роботи. Пояснювальна записка дипломного проекту складається з 5 розділів, містить 27 рисунків, 14 таблиць, 12 джерел.

Дипломний проект присвячений розробці комплексу задач інформаційної системи з підтримки діяльності служби таксі для підвищення якості обслуговування клієнтів.

В першому розділі наведено опис існуючого стану якості надання послуг з використанням інформаційних технологій та виявлені проблеми, які необхідно вирішити для підвищення якості обслуговування клієнтів з замовлення послуг таксі. Визначені цілі та завдання для їх вирішення.

У розділі з інформаційного забезпечення вказані вхідні та вихідні дані для системи з підтримки діяльності служби таксі та описано структуру бази даних.

Розділ математичного забезпечення присвячений алгоритмам, які потрібні для реалізації.

В проекті надані рішення з структури системи та архітектури розробленого програмного забезпечення, засоби розробки

У технологічному розділі надано керівництво користувача та результати тестування системи.

ТАКСІ, СЛУЖБА ТАКСІ, РЕЙТИНГ, КЛІЄНТ, ОБСЛУГОВУВАННЯ,
ВЕБ-РОЗРОБКА.

					ДП 7232.00.000 ПЗ				
		Прізвище	Підпис	Дата					
Розроб.	Цеюков А.В.				Інформаційна система з підтримки діяльності служби таксі	Літ.	Лист	Листів	
Перевірив.	Тєлшиєва Т.О.						2		
						КПІ ім. Ігоря Сікорського Каф. АСОІУ Гр.ІС-72			
Н. кон.	Новінський В.П.								
Затв.	Павлов О.А.								

ABSTRACT

Structure and scope of work. The explanatory note of the diploma project consists of 5 sections, contains 27 images, 14 tables, 12 sources.

The diploma project is devoted to the development of a set of tasks of the information system to support the activities of the taxi service to improve the quality of customer service.

The first section describes the current state of quality of services using information technology and identifies issues that need to be addressed to improve the quality of customer service for ordering taxi services. Goals and objectives for their solution are defined.

The information support section provides input and output data for the taxi service support system and describes the structure of the database.

The section of mathematical software is devoted to the algorithms required for implementation.

The project provides solutions for the structure of the system and the architecture of the developed software, development tools

The technology section provides a user guide and test results for the system.

TAXI, TAXI SERVICE, RATING, CUSTOMER, MAINTENANCE, WEB DEVELOPMENT.

					ДП 7232.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		2

ЗМІСТ

ВСТУП	5
1 ЗАГАЛЬНІ ПОЛОЖЕННЯ	7
1.1 ОПИС ПРЕДМЕТНОГО СЕРЕДОВИЩА	7
1.1.1 Опис процесудіяльності	9
1.1.2 Опис функціональної моделі	10
1.2 ОГЛЯД НАЯВНИХ АНАЛОГІВ	12
1.3 ПОСТАНОВКА ЗАДАЧІ	18
1.3.1 Призначення розробки	18
1.3.2 Цілі та задачі розробки	18
Висновок до розділу	18
2 ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ	19
2.1 ВХІДНІ ДАНІ	19
2.2 ВИХІДНІ ДАНІ	19
2.3 ОПИС СТРУКТУРИ БАЗИ ДАНИХ	19
Висновок до розділу	21
3 МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ	22
3.1 ЗМІСТОВНА ПОСТАНОВКА ЗАДАЧІ	22
3.2 МАТЕМАТИЧНА ПОСТАНОВКА ЗАДАЧІ	22
3.3 ОБҐРУНТУВАННЯ ТА ОПИС МЕТОДУ РОЗВ’ЯЗАННЯ	24
Висновок до розділу	25
4 ПРОГРАМНЕ ТА ТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ	26
4.1 ЗАСОБИ РОЗРОБКИ	26
4.2 ВИМОГИ ДО ТЕХНІЧНОГО ЗАБЕЗПЕЧЕННЯ	27
4.2.1 Загальні вимоги	27
4.3 АРХІТЕКТУРА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	27
4.3.1 Діаграма класів	29
4.3.2 Діаграма послідовності	30
4.3.3 Діаграма компонентів	31
4.3.4 Специфікація функцій	31
4.4 ОПИС ЗВІТІВ	32

Висновок до розділу	32
5 ТЕХНОЛОГІЧНИЙ РОЗДІЛ	33
5.1 Керівництво користувача	33
5.2 Випробування програмного продукту	41
5.2.1 Мета випробувань	41
5.2.2 Загальні положення	41
5.2.3 Результати випробувань	42
Висновок до розділу	46
ЗАГАЛЬНІ ВИСНОВКИ	47
ПЕРЕЛІК ПОСИЛАНЬ	48
ДОДАТОК Тексти програмного коду	49

ВСТУП

Таксі – це міський транспорт, зазвичай автомобіль, який клієнти використовують для подорожі в будь-яку точку. Оплата за проїзд вираховується згідно таксометру.

Історія таксі почалася на початку 17 століття в Лондоні. Тоді почали з'являтися карети, які перевозили пасажирів за гроші. Проте назва пішла від французького слова *hacquenée*, що в перекладі означало коня, якого беруть в оренду. На осі карети був закріплений «таксометр». Це був звичайний таз, в який кожні 200 метрів падав камінчик. В кінці поїздки підраховувалась кількість камінців і за їх кількістю розраховувалась вартість поїздки.

На 1988 рік в колишньому СРСР була прийнята система «такса за дистанцію»: 20 копійок за кілометр, при цьому відстань, яку треба було проїхати обговорювалася ще до початку поїздки. Тут з'являлася проблема, що водієві було невигідно працювати в час пік, проте в ті час пробок не було. Також були і переваги зі сторони пасажира тому, що вартість поїздки не залежить від дорожніх обставин, а водій може вибрати для себе оптимальний маршрут та рухатись з оптимальною швидкістю.

Існували також інші варіанти тарифікації: такса за хвилину або за кілометр по лічильнику. Проте такі способи розрахунку вартості поїздки були чреваті шахрайством зі сторони водія.

Послуги сервісу з автоперевезення – таксі в сучасному житті відіграє дуже важливу роль, яка виконує найрізноманітніші функції: від традиційної доставки пасажирів та допомоги їм в порятунку життя людини.

Таксі потрібно кожному з нас. Всі служби таксі знаходяться під пильним наглядом конкурентів. Найважливіше для цієї галузі – конкурентоспроможність, яка в свою чергу потребує найкращого обслуговування клієнтів. Тому компанії с автоперевезення, розуміючи це, покращують свій сервіс не тільки за рахунок технічного стану автопарку, а і висококваліфікованих водіїв.

					ДП 7232.00.000 ПЗ	Арк.
						5
Змн.	Арк.	№ докум.	Підпис	Дата		

На сьогоднішній день існують компанії, які можуть запропонувати клієнту додаток в мобільний додаток, через який він може здійснити замовлення. В таких компаніях як правило застосовують дорогі іномарки, які не мають вигляд таксі. Вони застосовуються для клієнтів, які б хотіли показати свій високий соціальний статус. Проте в додатку можна обрати і бюджетний варіант, в якому буде запропоновано машину нижче за класом

Актуальність теми.

Сумісні дії з автоматизації процесів сервісу та покращення рівня обслуговування за рахунок контролю рейтингу водіїв стає актуальним питанням з розвитку компаній, які надають послуги автоперевезень Тому тема дипломного проекту є актуальною.

Практичне значення:

Інформаційна система з підтримки служби таксі забезпечує реєстрацію водія адміністратором з характеристиками своєї машини як виконавця замовлення, а також дає можливість клієнту створити замовлення та обрати вільного водія згідно своїх побажань.

Моніторинг процесу обслуговування клієнтів надає додатковий стимул для водія обслуговувати пасажирів на високому рівні тому, що клієнт буде оцінювати услугу, яка буде надаватися компанією.

					ДП 7232.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		6

1 ЗАГАЛЬНІ ПОЛОЖЕННЯ

1.1 Опис предметного середовища

За оцінками на 2020 рік, ринок таксі в Україні становить близько 40 млрд гривень. Водіями є 220 тис. осіб, 98 % з яких працюють нелегально. Щодня таксистки здійснюють в середньому 1,5 млн поїздок. У Києві 50 % ринку займає Uber, 15 % — Uklon і 10 % — Bolt. Решту ділять між собою багато дрібних перевізників.

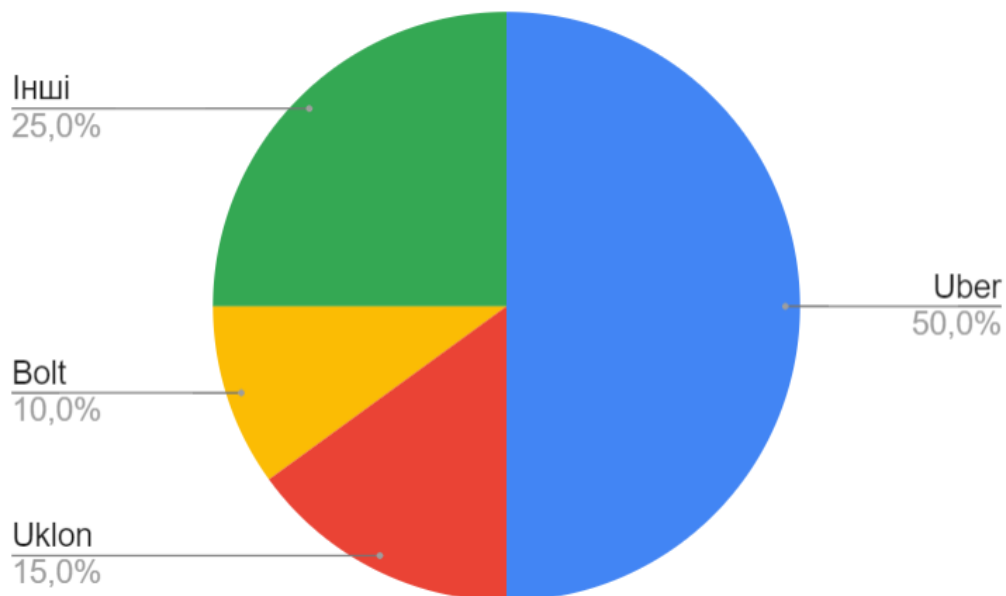


Рисунок 1.1 – Кругова діаграма порівняння

Великі служби онлайн-замовлень працюють лише у обмеженій кількості великих міст (наприклад, Uber у семи містах, Uklon — у 13), тому у менших містах ринок зазвичай контролюється дрібними операторами

Існують декілька способів замовлення таксі:

- вуличні таксі – такий спосіб передбачає водіїв, які їздять по місту та їх можна «спіймати» або чекають у місцях з активним потоком людей, наприклад, аеропорт або вокзал;

- служби таксі – такі таксі можна замовити по телефону. В таких компаніях є диспетчер, який обробить замовлення клієнта та підрахує вартість поїздки заздалегідь. Вони не мають власних машин, проте вони надають інформаційні послуги для водіїв і за це отримують комісію;
- інтернет-служби таксі – такі служби дають змогу клієнту зробити замовлення через мобільний додаток або на сайті. В таких компаніях використовується інтелектуальні алгоритми машинного навчання, які дозволяють розрахувати ціну поїздки, побудувати на карті маршрут для водія, зменшувати час очікування. Такі компанії також не мають власного автопарку.

Під час користування таксі є сильні сторони, проте існують і проблеми.

Таблиця 1.1 – Переваги і недоліки таксі.

Переваги	Недоліки
Доставка пасажирів в будь-яке місце призначення	Висока ціна послуг
Цілодобовий графік роботи	Низька пасажиромісткість
Комфорт	Відмова у замовленні з різних причин
	Велика чисельність нелегальних водіїв (приблизно 85% за даними за 2018 рік)
	Рівень обслуговування (адже водій може приїхати на іншу адресу, запізнитись, також бувають напади на пасажирів)

1.1.1 Опис процесу діяльності

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 з бази даних водіїв. Адміністратор також несе відповідальність за систему (рисунок 1.2).

					ДП 7232.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		9

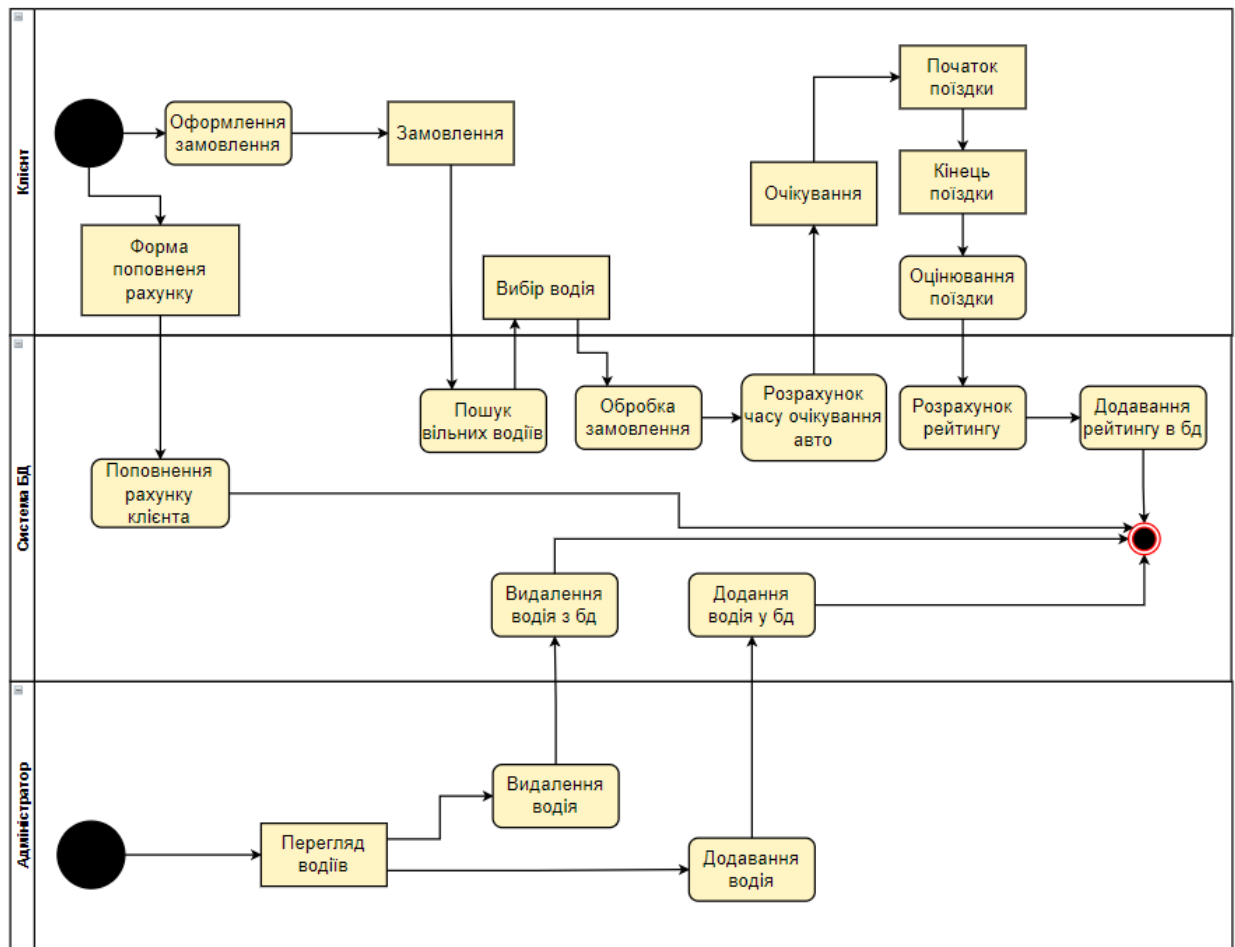


Рисунок 1.2 – Діаграма діяльності

1.1.2 Опис функціональної моделі

В цьому розділі буде ретельно розглянуто опис функцій акторів. На рисунку 1.3 наведено функціональна модель.

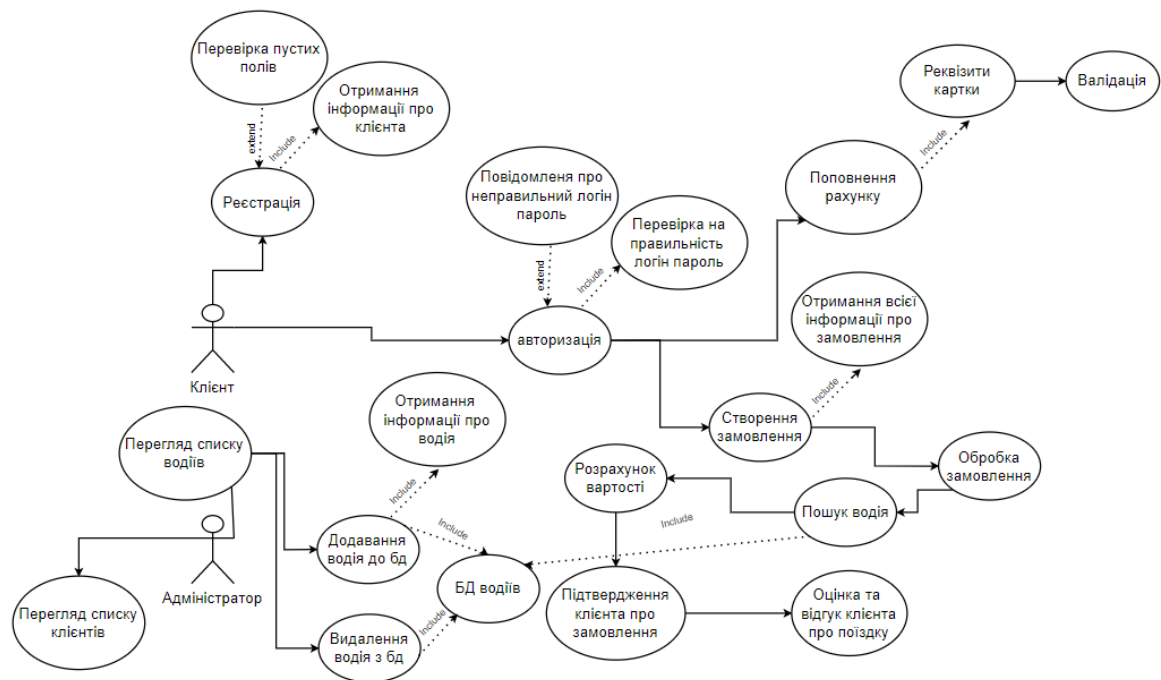


Рисунок 1.3 – Діаграма функціональної моделі

Таблиця 1.2 – Функції, які виконуватимуть актори

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

Продовження таблиці 1.2

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

1.2 Огляд наявних аналогів

1.2.1 Публічна компанія Uber

Uber - американська міжнародна публічна компанія з Сан-Франциско, яка створила однойменний мобільний додаток для пошуку, виклику і оплати таксі або приватних водіїв і доставки їжі[4]. За допомогою програми Uber замовник резервує машину з водієм і відстежує її переміщення до зазначеної точки. У більшості випадків водії використовують свої власні автомобілі, а також машини таксопарків або партнерів. У більшості країн 67% оплати переходять водієві, 33% перераховуються Uber.

У 2012 році Uber оголосила про плани розширення функціональних можливостей програми, після якого буде можливий пошук попутників для водіїв, які не надають послуги таксі.

У червні 2015 року Uber відкрив новий сервіс в Стамбулі UberBOAT, який здійснює переправу через протоку Босфор. Транспортування

здійснюватиметься за допомогою човнів. Вартість від 17 доларів, а місткість човни від 7 до 10 осіб.

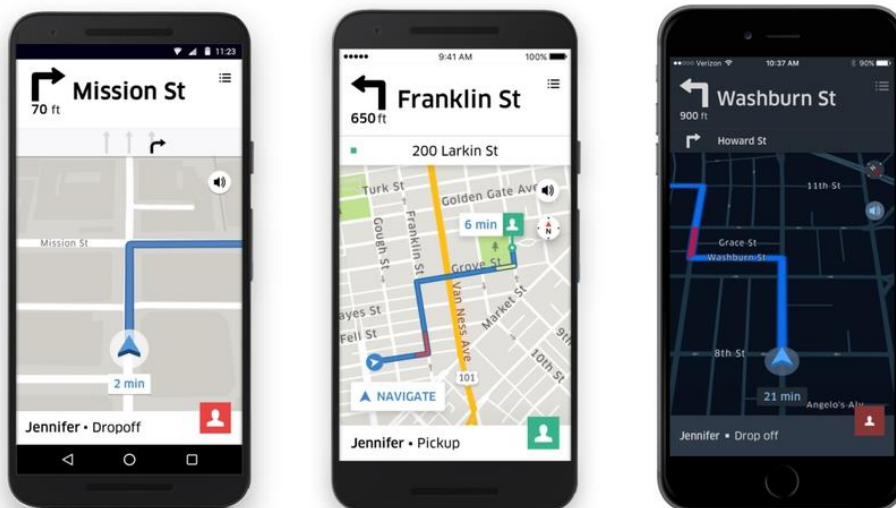


Рисунок 1.4 – Вигляд інтерфейсу мобільного додатку UBER

Переваги:

- безготівкова оплата через додаток;
- можливість відстежити переміщення автомобіля до вказаної адреси;
- доступність (у багатьох країнах Uber дешевше звичайного таксі).

Недоліки (з відгуків клієнтів [5]):

- ненадійні водії, що скасовують замовлення за 1 хв до прибуття;
- служба підтримки відсутній в Україні;
- відсутність належного контролю при підборі співробітників.

1.2.2 Сервіс Uklon

Uklon - перший на території України сервіс виклику авто через Інтернет. Uklon дозволяє замовляти автомобіль через додаток зі смартфона або на інтернет-сайті, без телефонних дзвінків в диспетчерські служби[6]. За деякими оцінками, належить до провідних в країні.

					ДП 7232.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		13

Безпосередньо Uklon не є службою таксі, це агрегатор замовлень, який розподіляє виклики між водіями-таксистами. Послуги з перевезення пасажирів надають в основному диспетчерські служби таксі.

Станом на жовтень 2019 року сервіс працює в Києві, Одесі, Харкові, Дніпрі, Львові, Запоріжжі, Вінниці, Сумах, Полтаві, Тернополі, Чернігові, Кривому Розі, Миколаєві, Білій Церкві, Маріуполі.

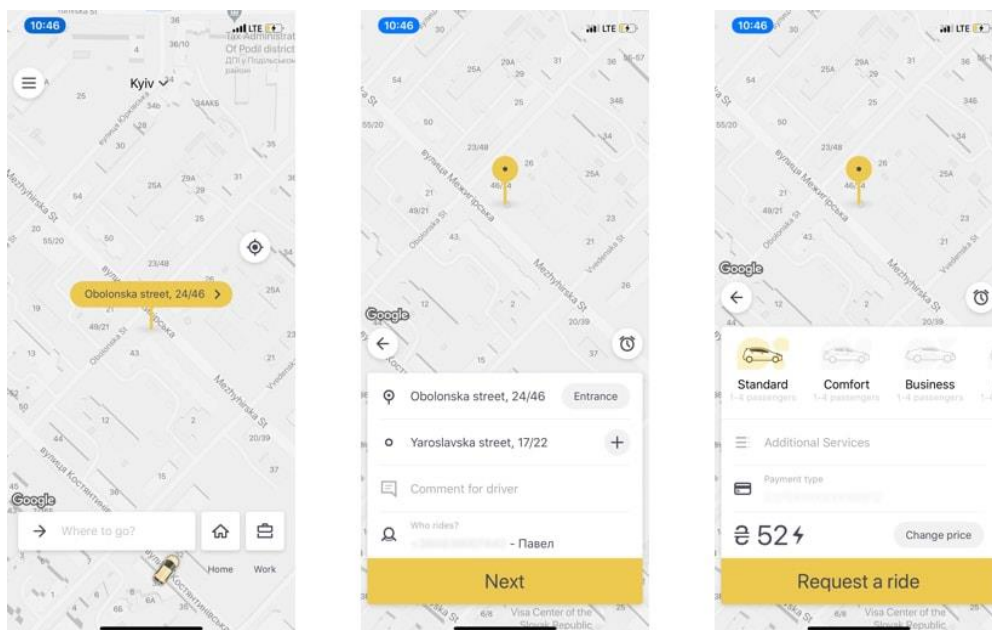


Рисунок 1.5 – Інтерфейс мобільного додатку Uklon

Переваги:

- безготівкова оплата через додаток;
- можливість відстежити переміщення автомобіля до вказаної адреси.

Недоліки (з відгуків клієнтів[7]):

- незручний інтерфейс;
- обман клієнтів;
- авто без маркування таксі.

					ДП 7232.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		14

1.2.3 Компанія Bolt

Bolt - естонська приватна компанія, творець мобільного застосування для виклику таксі та інших сервісів[8]. Офіс компанії знаходиться в Талліні.

Існують 2 застосунки: для пасажирів і водіїв. Станом на березень 2019, компанія працювала у 30 країнах Європи, Африки, Західної Азії, Центральної Америки та Австралії. Сервіс обслуговує більше 25 млн клієнтів. Комісія в Україні складає 25 %.

Компанія заснована 2013 року Маркусом Віллігом (на той час — 19-річним студентом) з метою об'єднати усі служби таксі Талліна і Риги в одну платформу. Сервіс було запущено в серпні 2013, а 2014 року почалося охоплення закордонних ринків.

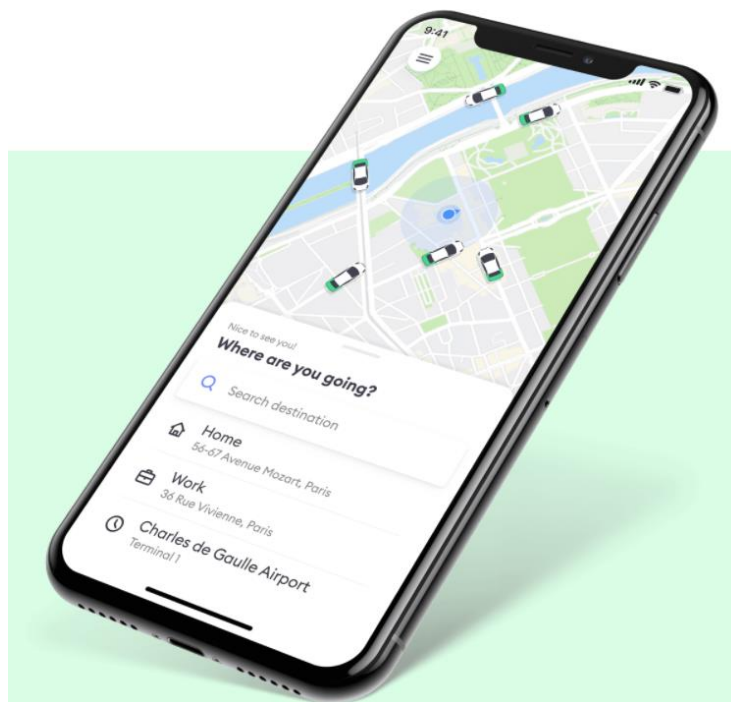


Рисунок 1.6 – Інтерфейс мобільного додатку Bolt

Переваги:

- можливість відстежити переміщення автомобіля до вказаної адреси;
- доступність;

					ДП 7232.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		15

- ціна фіксована.

Недоліки (з відгуків клієнтів [9]):

- сервіс не відповідає за водіїв;
- великі тарифи;
- водії, які абсолютно не володіють мовою;
- відсутнє замовлення з сайту.

1.2.4 Служба «Лайк таксі»

Лайк таксі пропонує послуги таксі в Києві та Київській області. Широкий вибір послуг і великий парк автомобілів дають можливість отримати автомобіль таксі в найкоротші терміни і за найвигіднішим тарифом[10]. Лайк таксі – молода служба, яка стрімко розвивається на таксомоторному ринку Києва. Велика кількість служб-партнерів дає можливість 100% подачі автомобіля навіть в час пік. Лайк таксі пропонує пасажиром таксі в Києві широкі інструменти при замовленні таксі, а також тонку тарифну політику. Пасажир сам може вибрати необхідний клас автомобіля і вказати особливості до замовлення.

Крім цього, адміністрація Лайк таксі уважно ставиться до якості обслуговування пасажирів, шляхом надання клієнтам можливості залишити відгук або питання, а також подяку після поїздки в таксі Києва.

					ДП 7232.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		16



Рисунок 1.7 – логотип служби «Лайк таксі»

Переваги:

- швидка подача авто;
- невисокі тарифи.

Недоліки (з відгуків клієнтів [11]) :

- некомфортні автомобілі;
- невідповідність автомобілів;
- відсутня безготівкова оплата;
- відсутній мобільний додаток.

Таблиця 1.3 – Порівняння аналогів

	Наявність безготівкової оплати	Наявність мобільного додатку	Наявність замовлення з сайту	Наявність власного автопарку	Зв'язок з оператором
Uber	+	+	+	+	-
Uklon	+	+	+	-	+
Bolt	+	+	-	-	+
Лайк таксі	-	-	-	-	+

1.3 Постановка задачі

1.3.1 Призначення розробки

Дана система забезпечує реєстрацію водія адміністратором, з характеристиками своєї машини, як виконавця замовлення, а також дає можливість клієнту створити замовлення та обрати вільного водія згідно своїх побажань.

1.3.2 Цілі та задачі розробки

Цілью інформаційної системи з підтримки служби діяльності таксі є підвищення якості обслуговування клієнтів служби таксі, за рахунок надання вибору послуги за його побажаннями.

Для досягнення поставленої цілі необхідно вирішити наступні задачі:

- розробити узагальнений алгоритм вирішення задачі пошуку клієнтом таксі, згідно заданих характеристик;
- створити функціональну модель системи;
- визначити та реалізувати інформаційне забезпечення системи;
- розробити інтерфейс з можливістю надання відгуку від клієнта;
- реалізувати систему в програмному коді, розробити керівництво користувача та адміністратора.

Висновок до розділу

У даному розділі було детально описане предметне середовище, також було описано процес діяльності і пояснено призначення цієї системи.

Були створені функціональні вимоги та описана функціональна модель.

Також у цьому розділі були описані та проаналізовані аналоги та виявлені їх переваги і недоліки.

Результатом є сформовані цілі і поставлені задачі.

					ДП 7232.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		18

2 ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ

2.1 Вхідні дані

Для того щоб зареєструватись клієнт повинен вказати:

- ім'я та прізвище;
- номер телефону;
- пароль.

Коли клієнт прагне зробити замовлення, треба передати системі такі дані:

- адреса місцезнаходження клієнта;
- тип машини (економ/стандарт/люкс/вантажна);
- адреса місцезнаходження кінцевої точки.

Після завершення поїздки, клієнт передає в систему дані про оцінку водія:

- оцінка (1-5);
- коментар.

2.2 Вихідні дані

Результатом роботи є наступний перелік вихідних даних:

1. список водіїв, які підбрала система по критеріям клієнта;
2. вартість проїзду.

2.3 Опис структури бази даних

Для розробки бази даних використаний SQL Server. База даних містить 4 таблиці, які наведені в таблицях 2.1.

					ДП 7232.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		19

Таблиця 2.1 – Перелік сутностей в базі даних

№	Назва таблиці	Сутність
1	Driver	Водій
2	User	Клієнт
3	Admin	Адміністратор
4	Comment	Оцінка/коментар

В таблиці 2.2 наведено детальний опис сутностей в базі даних

Таблиця 2.2 – Опис сутностей в базі даних

Назва таблиці	Назва стовпця	Тип даних	Опис поля
Driver	id	int	Первинний ключ
	FlName	varchar(100)	Ім'я водія
	RideCount	varchar(100)	Кількість поїздок
	Rating	int	Рейтинг
	VechicleClass	int	Клас автомобіля
	BabyChair	int	Дитяче крісло
	AnimalPosibility	int	Перевезення тварин
	CarantinPosibility	int	Карантинні заходи
	CountChairs	int	Кількість місць
User	id	int	Первинний ключ
	Phone	varchar(100)	Номер телефону
	FlName	varchar(100)	Ім'я клієнта
	Password	varchar(100)	Пароль
	Money	int	Кількість грошей
Administrator	id	int	Первинний ключ
	Phone	varchar(100)	Номер телефону
	FlName	varchar(100)	Ім'я адміністратора
	Password	varchar(100)	Пароль
Comment	id	int	Первинний ключ
	idDriver	varchar(100)	Первинний ключ водія
	Comment	varchar(100)	Коментар
VechicleClass	id	int	Первинний ключ
	Class	int	Клас автомобіля
	PricerPerKilometr	int	Вартість за км

Виходячи з таблиці 2.2, розроблена діаграма структури бази даних, яка показано на рисунку 2.1

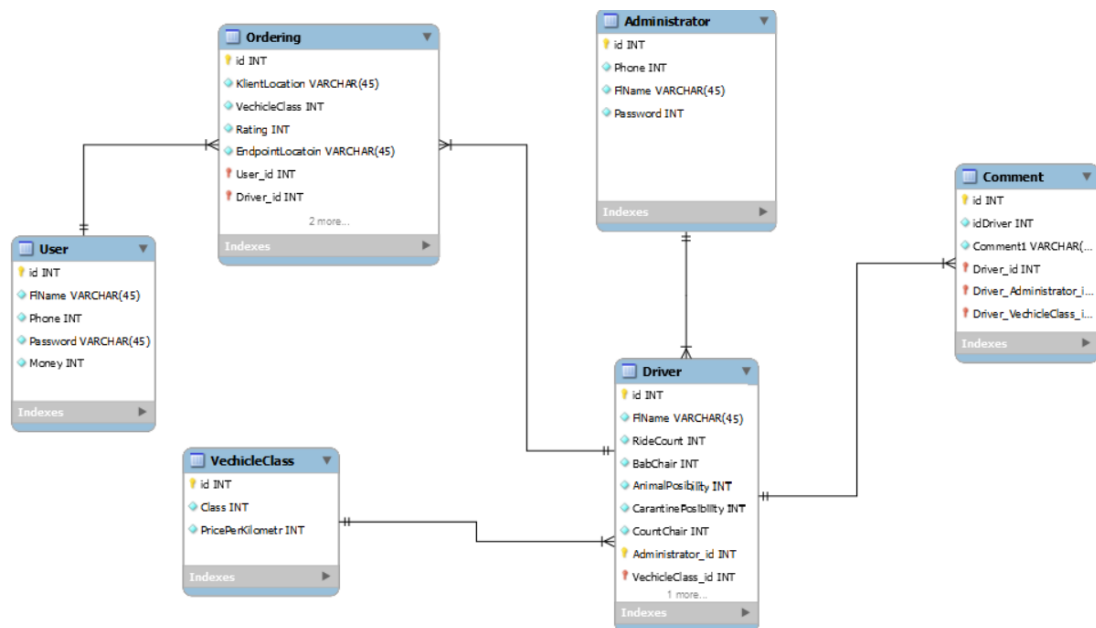


Рисунок 2.1 – Структура бази даних

Висновок до розділу

У цьому розділі було описано інформаційне забезпечення, яке відповідає за опис вхідних даних та вихідних даних. Розроблено базу даних та детально описано всі сутності в базі даних.

3 МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ

3.1 Змістовна постановка задачі

Задача, яку вирішено в дипломному проєкті наступна.

Інформаційна система з підтримки діяльності служби таксі повинна мати функції, які нададуть змогу клієнтам служби швидко знайти автомобіль за їхніми побажаннями за умови, що про якість обслуговування клієнт зможе залишити відгук и поставити оцінку за поїздки.

Для виконання задачу необхідно декомпонувати на наступні завдання:

- створити алгоритм пошуку автомобіля і водія згідно умов клієнта;
- створити алгоритм моніторингу рейтингу водіїв.

3.2 Алгоритм виконання задачі

Завдання створення алгоритму пошуку автомобіля і водія згідно умов клієнта вирішується за наступним алгоритмом (рисунок 3.1).

1. Створюємо базу даних, яка буде містити в собі необхідну інформацією про водіїв, автомобілів та клієнтів.
2. Вибір Entity Framework для пошуку автомобіля з водієм.
3. Створення інтерфейсу, для взаємодії з клієнтом.

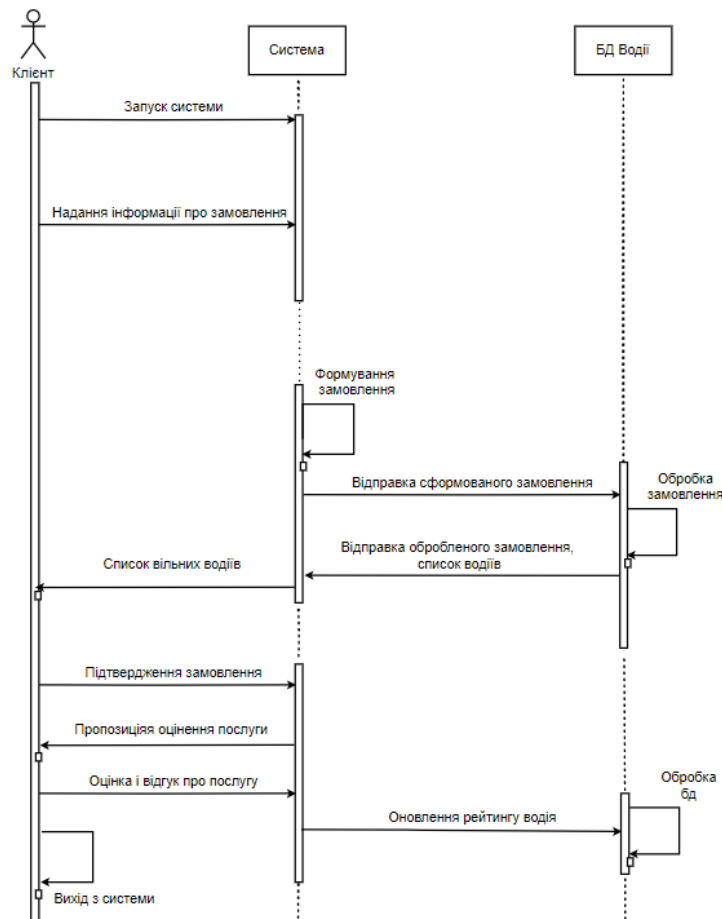


Рисунок 3.1 – Діаграма послідовності

Наступним завданням є отримання відгука та оцінки від клієта про поїздки та водія. Для цього потрібно:

1. використати Entity Framework для додавання в базу даних відгуку та оцінки;
2. скласти шкалу оцінювання водія;
3. надання можливості адміністратору контролювати діяльність водіїв за відгуками;
4. система оновлює рейтинг кожного водія після завершеної поїздки та контролює оцінки водіїв з інтервалом 5 поїздок для того, щоб видалити з бази даних водіїв, які мають середній рейтинг нижче 3. Після кожних 5 поїздок, буде перевірятися середній рейтинг кожного водія;
5. плата за проїзд залежить не тільки від дальності поїздки, а також від рейтингу водія та обраховується автоматично системою.

3.3 Обґрунтування та опис методу розв'язання

Розглянемо кожне завдання окремо.

«Пошук автомобіля і водія». Після надходження в систему побажань клієнта, по яким алгоритм буде шукати водіїв, система повертає список водіїв, які доступні і підходять по критеріям. Система надає змогу клієнту обрати зі списку водіїв. Після цього запускається таймер, який розраховує час, за який водій подолає відстань маршруту. В системі використано вибіркового запит, який дає змогу отримати лише потрібні дані по декількох критеріях.

«Завдання рейтинг водія» це означає підрахунок середньої оцінки кожного водія. Після кожної успішної поїздки клієнт повинен оцінити роботу водія по шкалі від 1 до 5. Підрахунок середньої оцінки рейтингу водія оновлюється після кожної успішної поїздки. Якщо водій здійснив 5 заказів і його середній рейтинг менше 3, тоді цього водія буде звільнено.

Середня оцінка підраховується за формулою (3.1):

$$S = \frac{\sum_{i=1}^N A}{N}, \quad (3.1)$$

де S – середня оцінка рейтингу;

A – оцінка за поїздку;

N – кількість поїздок.

Розрахунок плати за проїзд виконується наступним чином. Ця задача означає підрахунок плати за проїзд згідно відстані поїздки, рейтингу водія та класу його автомобіля. Рейтинг водія впливає на вартість поїздки таким чином, чим вищий рейтинг водія, тим більше плата і навпаки. Так само клас автомобіля водія впливає на вартість, тобто, чим вищий клас, тим більша плата.

Цей показник обраховується за формулою (3.2):

$$C = D \times (k + K), \quad (3.2)$$

де C – вартість поїздки;

					ДП 7232.00.000 ПЗ	Арк.
						24
Змн.	Арк.	№ докум.	Підпис	Дата		

D – дистанція;

k – коефіцієнт класу автомобіля (економ – 1, стандарт – 2, стандарт(вантажівка) – 2,5, бізнес – 3);

K – коефіцієнт рейтингу водія залежить від S.

Висновок до розділу

В даному розділі розроблені алгоритми вирішення завдань. У розділі описаний алгоритм, який наданий у вигляді діаграми діяльності, пояснені завдання пошуку водія, розрахунку плати за проїзд та визначення і оновлення рейтингу водія.

					ДП 7232.00.000 ПЗ	Арк.
						25
Змн.	Арк.	№ докум.	Підпис	Дата		

4 ПРОГРАМНЕ ТА ТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ

4.1 Засоби розробки

Програмний продукт був розроблений за допомогою таких програмно-технічних технологій таких, як C# та mSQL, та реалізовано за допомогою програмного редактору VisualStudioCode.

C# це мова програмування, призначений для розробки найрізноманітніших додатків, призначених для виконання в середовищі .NET Framework.

VisualStudioCode - програмний редактор коду, розроблений та оптимізований для проектування, тестування та впровадження сучасних веб-додатків. Підтримка Visual C # в Visual Studio забезпечується за допомогою повнофункціонального редактора коду, компілятора, шаблонів проектів, конструкторів, майстрів коду, потужного і зручного відладчика і багатьох інших засобів. Бібліотека класів .NET Framework надає доступ до багатьох служб операційної системи і іншим корисним, правильним класам, що істотно прискорює цикл розробки[13].

mSQL (вона ж MiniSQL) - легка клієнт-серверна реляційна СУБД, що випускається компанією Hughes Technologies.

Були використані такі бібліотеки на c#:

- WPF - є частина екосистеми платформи .NET і являє собою підсистему для побудови графічних інтерфейсів;
- EntityFramework - являє спеціальну об'єктно-орієнтовану технологію на базі фреймворка .NET для роботи з даними. Якщо традиційні засоби ADO.NET дозволяють створювати підключення, команди та інші об'єкти для взаємодії з базами даних, то Entity Framework являє собою більш високий рівень абстракції, який дозволяє абстрагуватися від самої бази даних і працювати з даними незалежно від типу сховища.,

- Google API – дозволяє з'єднуватися зі службами Google та інтегрувати їх в інші служби. Приклади таких служб: пошук, Gmail, Перекладач або Карти Google. Сторонні програми можуть використовувати ці API, щоб скористатися або розширити функціональність існуючих служб.

4.2 Вимоги до технічного забезпечення

Щоб програмний продукт працював без проблем, необхідний комп'ютер з доступом до інтернету.

Мінімальні вимоги ПК:

- об'єм оперативної пам'яті 1 Gb;
- процесор с тактовою частотою 2ГГц.

Підключені такі периферійні пристрої:

- монітор;
- мишка;
- клавіатура.

4.2.1 Загальні вимоги

Для повноцінної підтримки програмного забезпечення, потрібно щоб на комп'ютері були встановлені такі програми:

- Microsoft Visual C++ Redistributable for VisualStudio 2015, 2017 and 2019;
- Windows від 7;
- mSQL.

4.3 Архітектура програмного забезпечення

Архітектура програмного забезпечення складається з 2 частин:

1. Клієнтська частина системи проєкту, яка використовується для впровадження користувацького інтерфейсу.

					ДП 7232.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		27

2. Застосування для підтримки функціоналу змінення та отримання даних з бази даних, які будуть в подальшому оброблятися, а також збереження даних та передача до клієнтської частини проєкту. Це застосування в тому числі відповідає за створення інтерфейсу та обробку даних.

Особливості такої моделі полягають в тому, що користувач відправляє спеціальний запит на сервер, де той системно обробляється і кінцевий результат відсилається клієнту. У можливості сервера входить одночасне обслуговування відразу декількох клієнтів.

Якщо одночасно надходить більше одного запиту, то такі запити встановлюються в певній черзі і сервером виконуються по черзі. Часом запити можуть мати свої власні пріоритети. Частина запитів з більш високими пріоритетами будуть постійно працювати в першочерговому порядку.

Параметри, які можуть реалізуватися на стороні сервера:

- зберігання, захист і доступ до даних;
- робота зі вступниками клієнтськими запитами;
- процес відправки відповіді клієнту.

Параметри, які можуть реалізуватися на стороні клієнта:

- майданчик з надання графічного інтерфейсу;
- формулювання запиту до сервера і його подальша відправка;
- отримання підсумків запиту і відправка додаткової групи команд (запити на додавання, оновлення інформації, видалення групи даних).

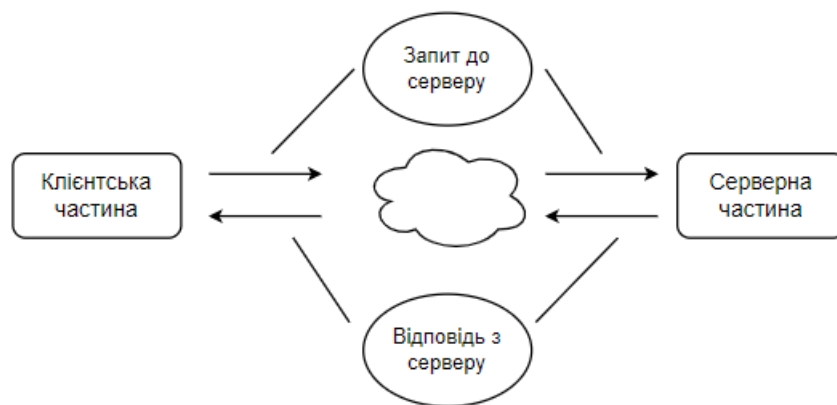


Рисунок 4.1 – Схема структури клієнт-серверної архітектури (дворівнева)

4.3.1 Діаграма класів

На рисунку 4.2 зображено діаграму класів.

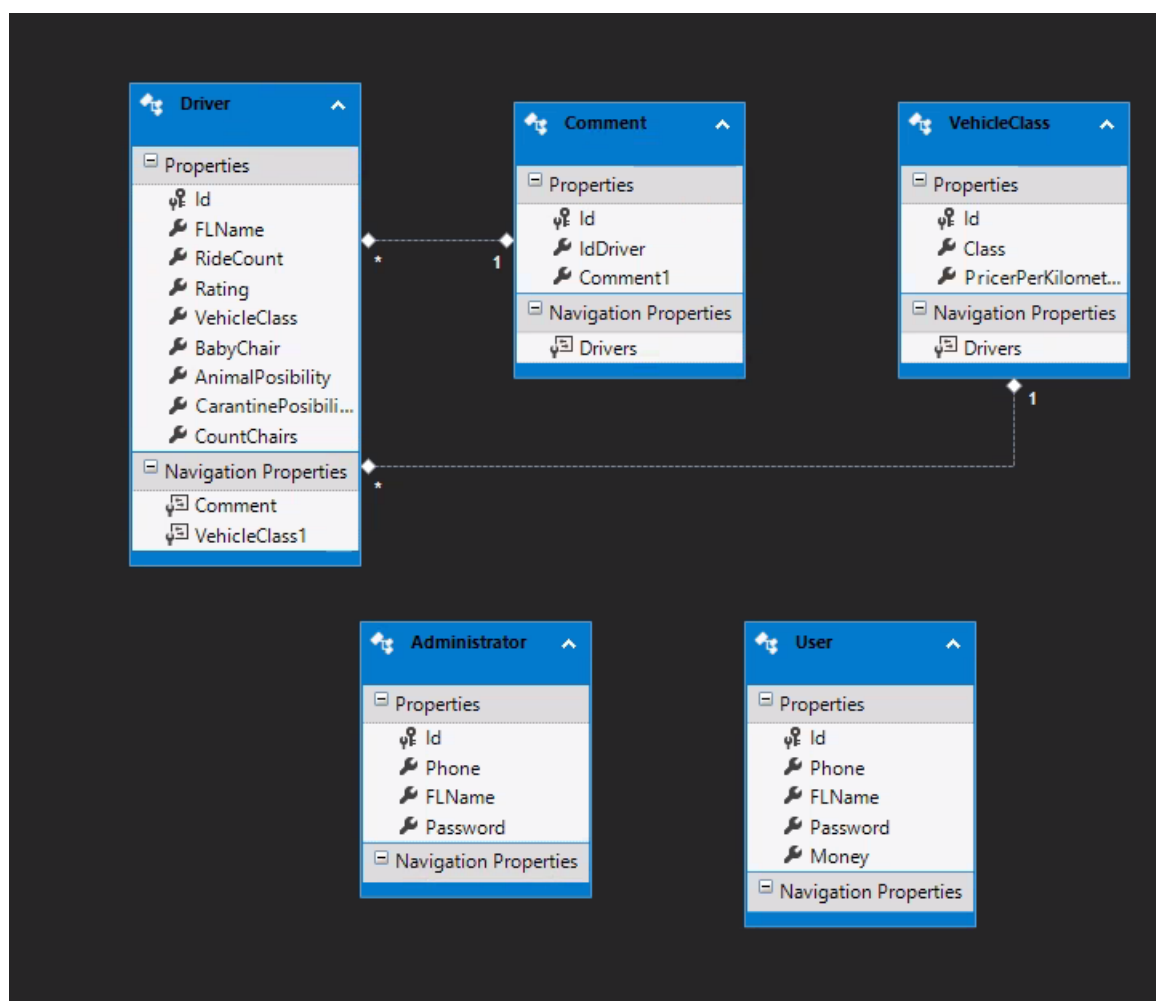


Рисунок 4.2 – Діаграма класів

4.3.2 Діаграма послідовності

На рисунку 4.3 показано діаграмі послідовності системи. Показано, як система взаємодіє з клієнтом.

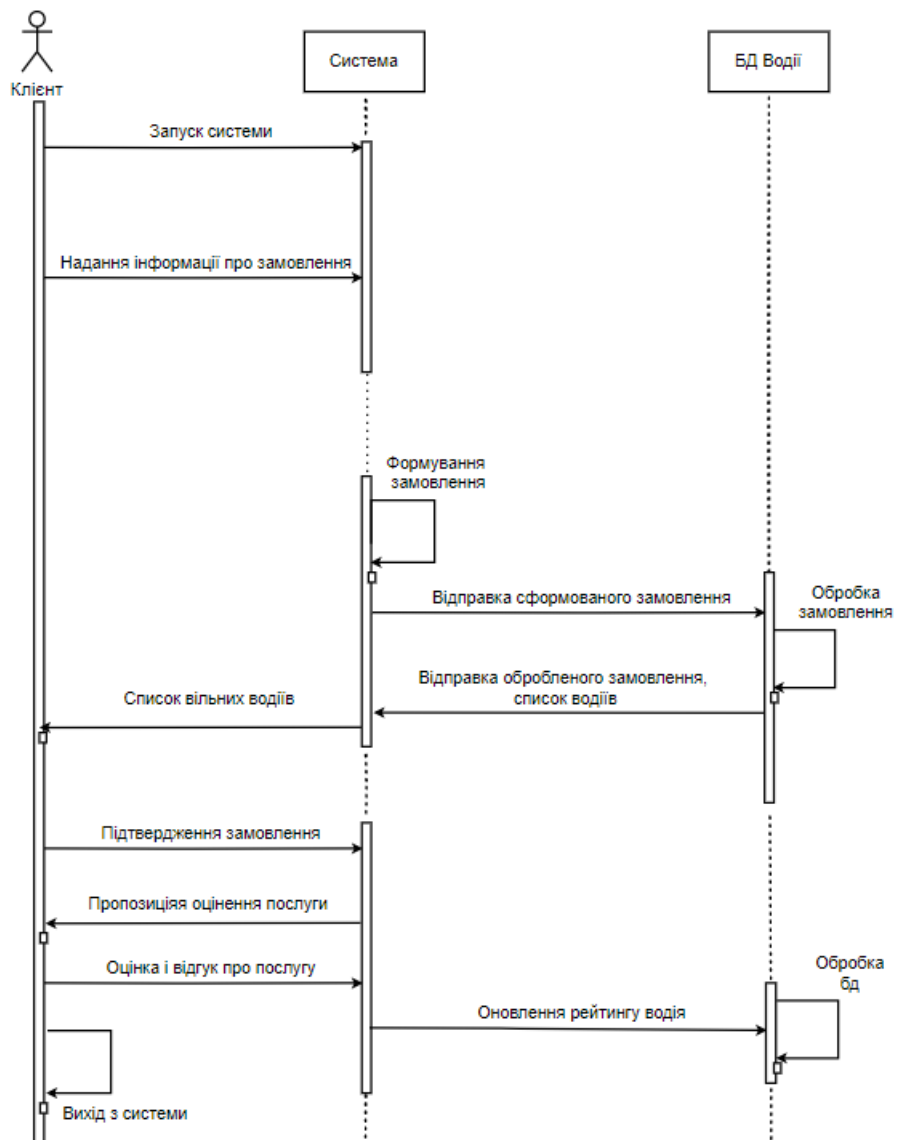


Рисунок 4.3 - Діаграма послідовності

4.3.3 Діаграма компонентів

На рисунку 4.3 зображено з яких компонентів складається програмний продукт.

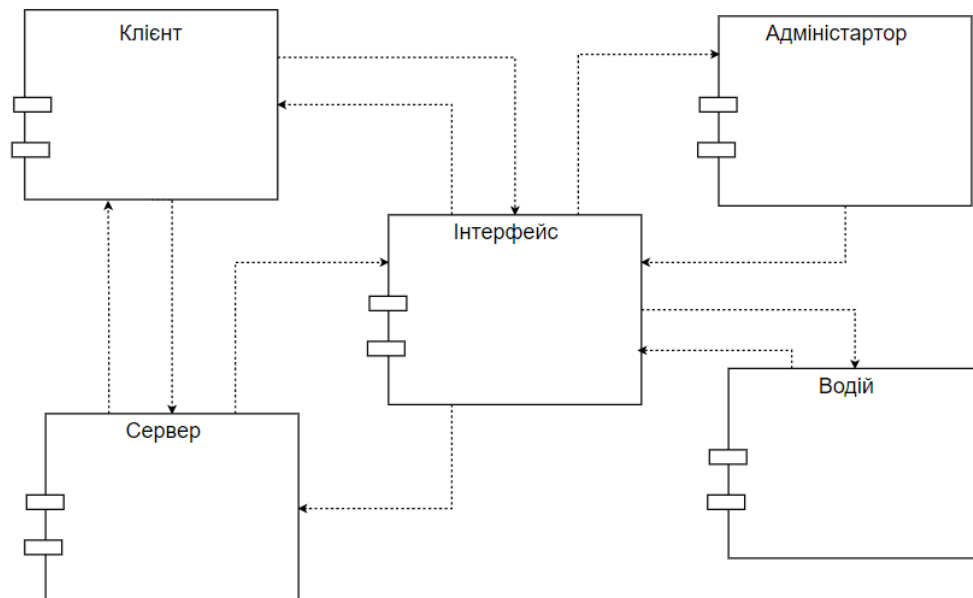


Рисунок 4.4 - Діаграма компонентів

4.3.4 Специфікація функцій

Специфікація функцій описано в таблиці 4.1.

Таблиця 4.1 – Таблиця основних функцій

Назва функції	Призначення
Register()	Додати новий акаунт до системи.
Login()	Вхід в систему та перевірка існуючого акаунту.
AddDriver()	Додавання водія в бд
Deposit()	Поповнення рахунку клієнта та оновлення бд
Gid_SelectionChanged()	Видаляє з бд водія за рейтингом
Offer()	Замовлення
ComboBox_SelectionChanged()	Підрахунок вартості проїзду

OfferDriver()

Пошук водія

Кінець таблиці 4.1.

Назва функції	Призначення
OfferRating()	Обробка рейтингу водія
OfferTime()	Запуск та зупинка таймеру поїздки

4.4 Опис звітів

Результатом поїздки є сформований чек, в якому вказано час поїздки, інформація про водія та вартість проїзду (рисунок 4.5).

Інформація

Водій: Taras Nishchuk

Час: 0:23

Ціна: 60 грн

Завершити поїздку

Рисунок 4.5 - Чек

Висновок до розділу

В цьому розділі наведено діаграма класів, діаграма послідовності, діаграма компонентів. Наведено список основних функцій, основне призначення цих функцій та в якому класі вони знаходяться.

					ДП 7232.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		32

5 ТЕХНОЛОГІЧНИЙ РОЗДІЛ

5.1 Керівництво користувача

На початку роботи, система відкриє для користувача вікно авторизації, яке потрібно для того, щоб клієнт зміг зайти в свій акаунт або зареєструватись, щоб почати роботу. Необхідно ввести номер телефону та пароль (рисунок 5.1).

Рисунок 5.1 – Форма авторизації

Якщо у користувача ще немає акаунту, тоді він повинен натиснути кнопку «Зареєструватись» та заповнити поля реєстрації. Клієнт повинен вказати свій номер телефону, ім'я та прізвище, та придумати пароль. Для

подальшої роботи повинні бути заповненими. На рисунку 5.2 зображена форма реєстрації.

Служба таксі

Номер телефону
0662838586

Прізвище, ім'я
Цеюков Артур

Пароль
1123456

Зареєструватися

Назад

Рисунок 5.2 – Форма реєстрації

Якщо користувач ввів правильно логін та пароль, тоді система відкриє йому головне меню, де буде інформація про його рахунок та дві кнопки: «Поповнити баланс» та «Створити замовлення (рисунок 5.3).

Вітаємо, Цеюков Артур

Ваш баланс: 0 грн

Поповнити баланс

Створити замовлення

Вийти з акаунту

Рисунок 5.3 – Меню клієнта

Коли клієнт хоче поповнити свій баланс, він повинен натиснути кнопку «Поповнити рахунок» в головному меню. Система відкриє йому форму, яку клієнт повинен заповнити своїми реквізитами, тобто клієнт повинен вказати номер своєї банківської карти, та ввести необхідні суму для поповнення балансу, щоб робити замовлення (рисуюнок 5.4).

Поповнення рахунку

Номер карти
2312337485736475

xx/yy
08 / 21

CVV-код
468

Сума
300

Поповнити

Назад

Рисуюнок 5.4 – Форма поповнення рахунку

Якщо клієнт готовий зробити замовлення, він натискає кнопку в головному меню «Зробити замовлення». Далі система відкриє гугл карту та запропонує клієнту ввести адресу свого місцезнаходження та кінцеву адресу. Після чого клієнт натискає кнопку «Вирахувати відстань» (рисуюнок 5.5).

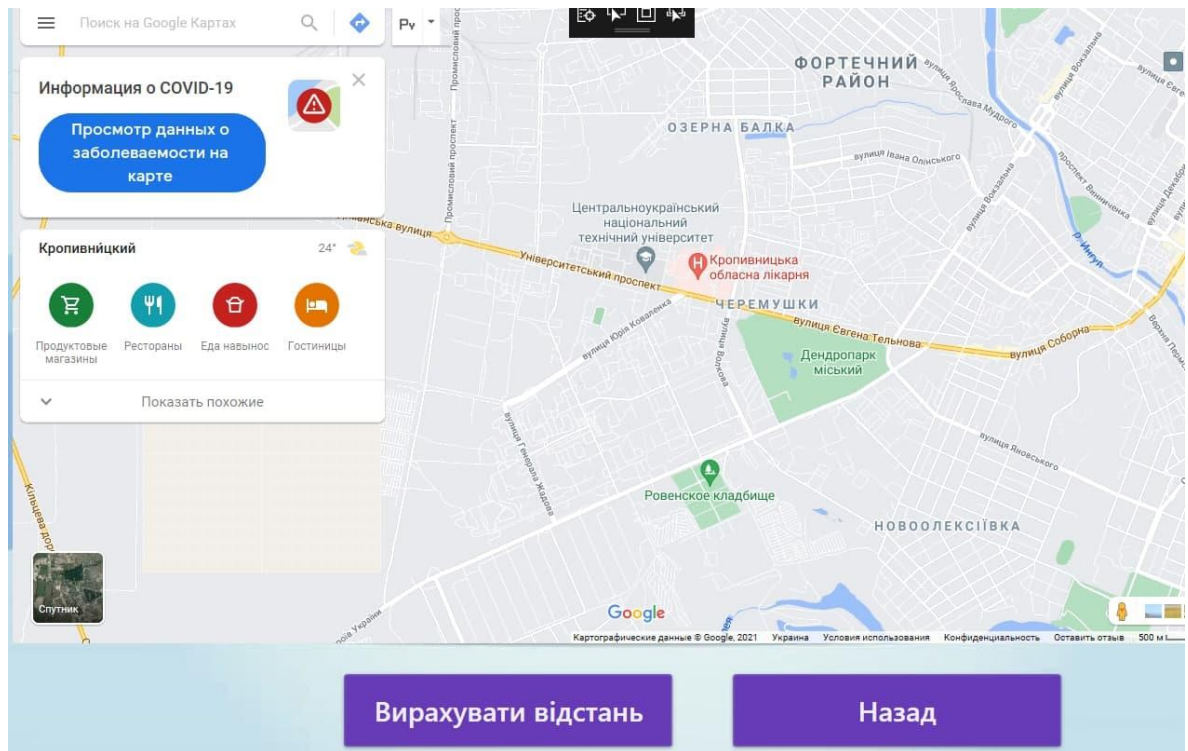


Рисунок 5.5 – Введення адреси

Далі система пропонує користувачу обрати клас автомобіля та ознайомлює його з вартістю проїзду, якщо клієнт підтверджує замовлення він натискає кнопку «Підтвердити» (рисунок 5.6).

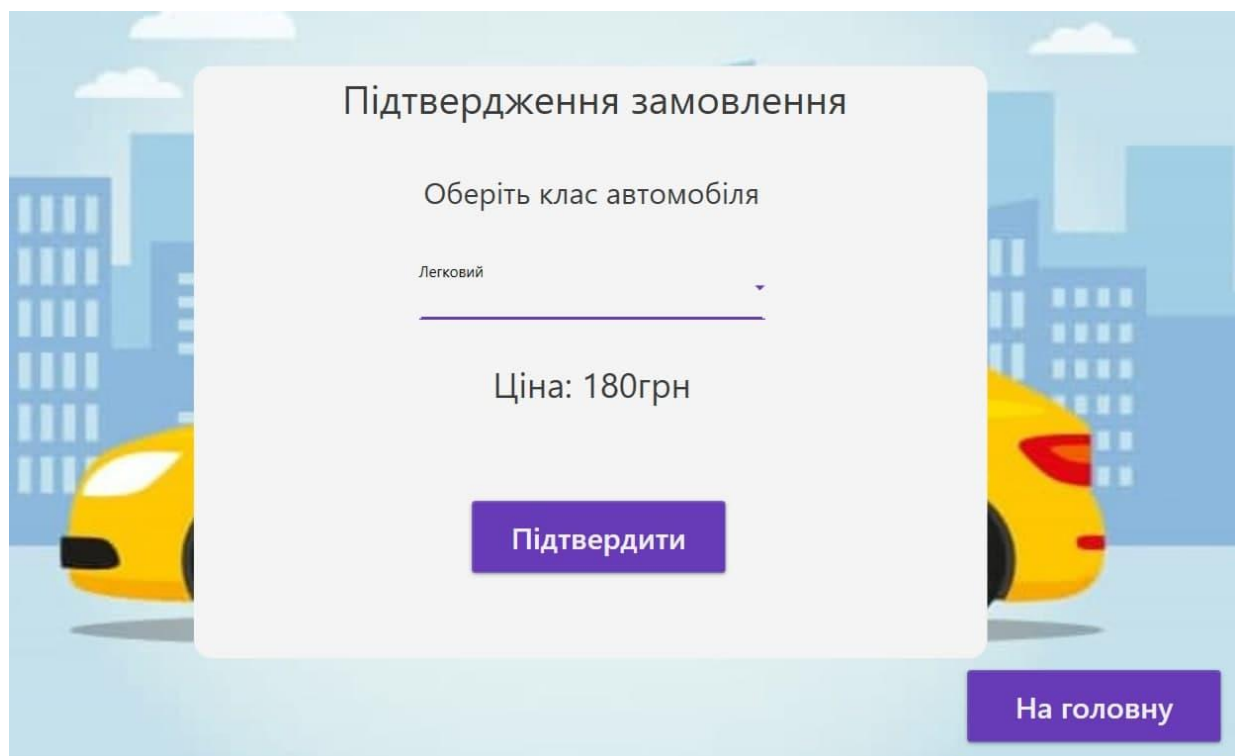


Рисунок 5.6 – Підтвердження замовлення.

Змн.	Арк.	№ докум.	Підпис	Дата

Наступним кроком є етап вибору водія. Клієнту надається вибір додаткових критеріїв, після чого користувач натискає кнопку «Знайти». Система видає клієнту список водіїв з класом автомобіля, який обрав клієнт та з додатковими критеріями. Далі потрібно обрати зі списку водія, який має вищий рейтинг (рисунки 5.7).

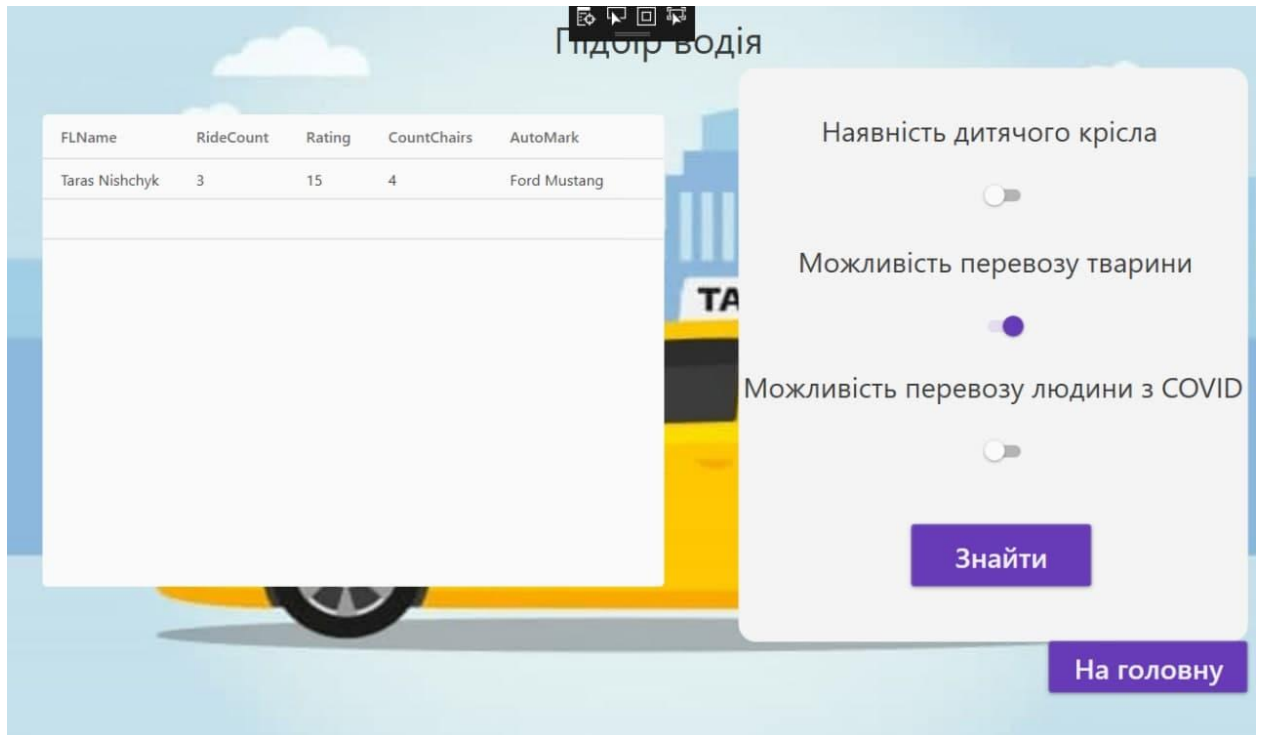


Рисунок 5.7 – Підбір водія

Далі система вираховує приблизний час, коли водій приїде. Коли водій приїхав клієнт натискає кнопку «Водій приїхав», і починається відлік часу самої поїздки, коли поїздка завершилась, клієнт натискає кнопку «Завершити замовлення» (рисунки 5.8).

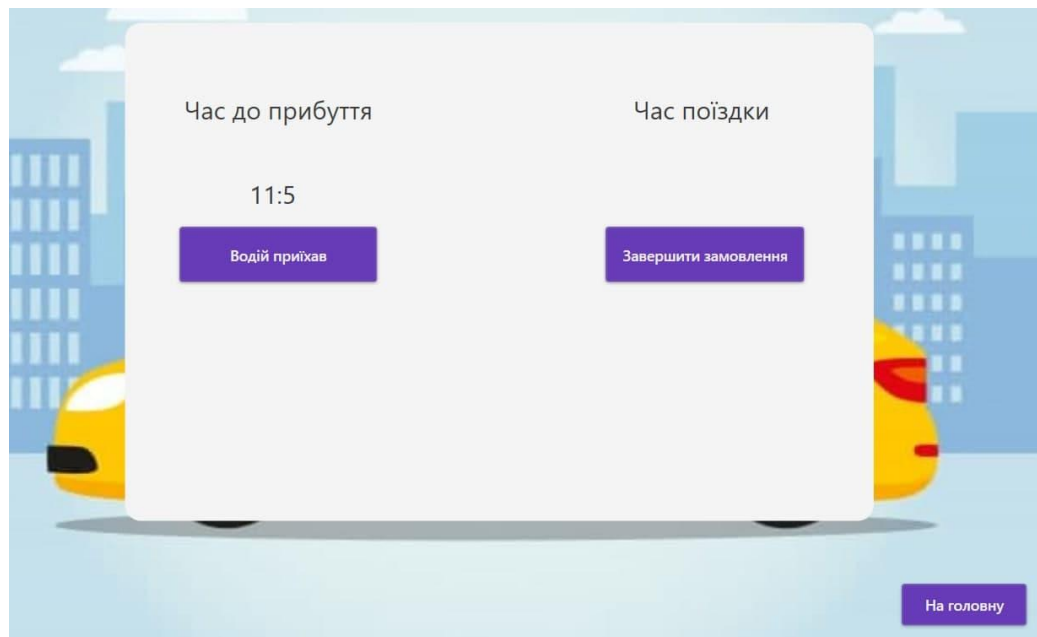


Рисунок 5.8 – Час поїздки і час до прибуття.

Потім клієнту надається можливість залишити відгук та поставити оцінку водію. Після натискання кнопки «Завершити замовлення», клієнт знов потрапить на головне меню (рисунок 5.9).

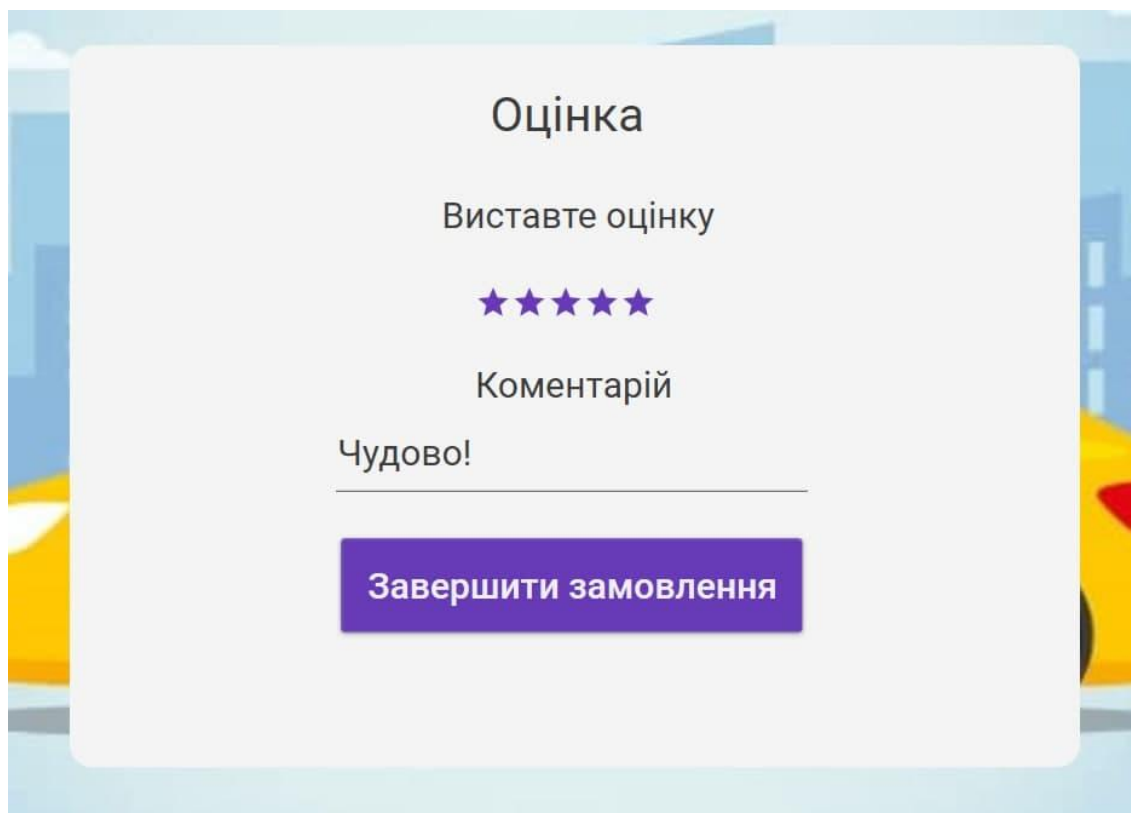


Рисунок 5.9 – Форма для відгуку

Також в системі є акаунт адміністратора, який може переглядати список користувачів та водіїв. Також адміністратор має можливість додавати або видаляти водія. В головному меню адміністратора можемо побачити кнопки, які зображені на рисунку 5.10.

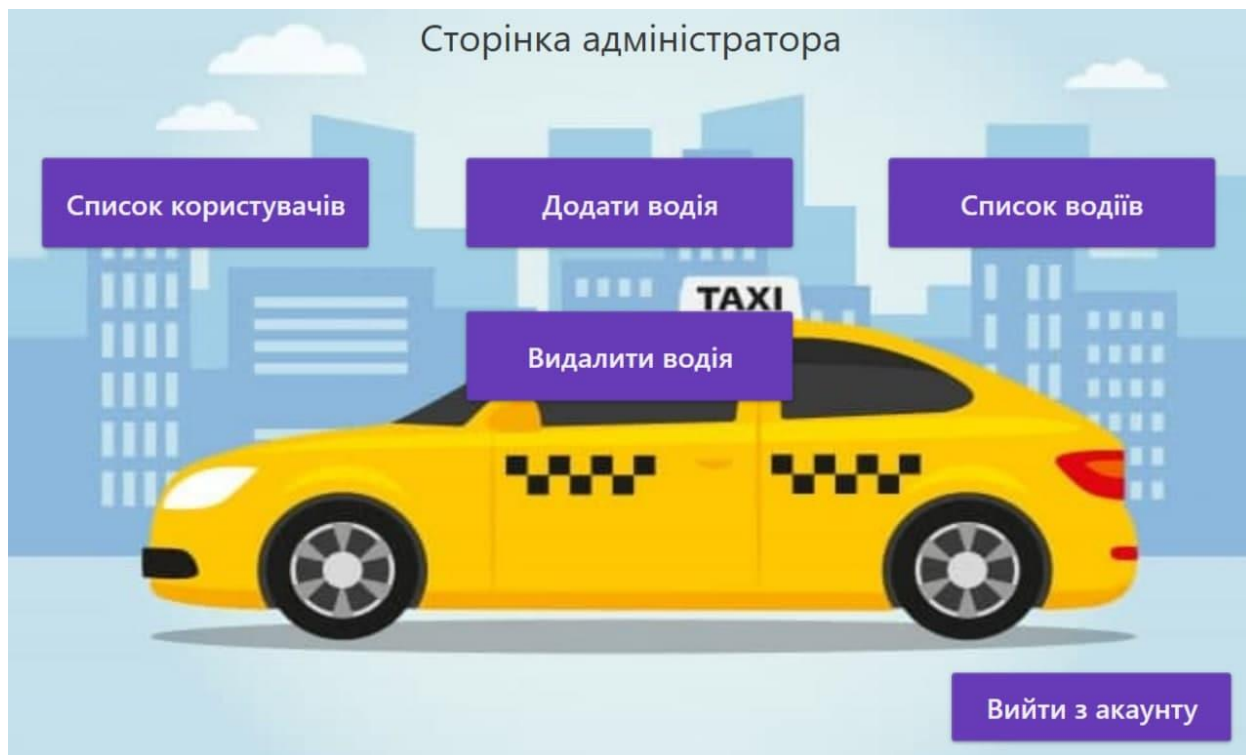


Рисунок 5.10 – Сторінка адміністратора

Натиснувши на кнопку «Список користувачів» адміністратор бачить такий список (рисунок 5.11).

Рисунок 5.11 – Список користувачів

[illegible]

Рисунок 5.12 - Список водіїв

Також адміністратор має можливість додавати водія. Для цього йому потрібно натиснути кнопку «Додати водія» та заповнити форму для

додавання, в якій він повинен вказати характеристики автомобіля, та данні водія. Це показано на рисунку 5.13.

Рисунок 5.13 – Форма реєстрації водія

5.2 Випробування програмного продукту

5.2.1 Мета випробувань

Метою випробувань являється перевірка відповідності функцій комплексу задач інформаційної системи з підтримки служби діяльності таксі вимогам технічного завдання.

5.2.2 Загальні положення

Випробування проводяться на основі наступних документів:

- ГОСТ 34.603–92. Інформаційна технологія. Види випробувань автоматизованих систем;
- ГОСТ РД 50-34.698-90. Автоматизовані системи вимог до змісту документів.

					ДП 7232.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		41

5.2.3 Результати випробувань

Таблиця 5.1 – Перевірка випробувань авторизації

Назва випробування	Випробування авторизації
Стан системи перед випробуванням	Система відкриває форму авторизації
Вхідні данні	Логін та пароль
Вихідні данні	Якщо користувач ввів неправильно логін або пароль , система показує повідомлення щодо неправильного вводу даних. Якщо пароль і логін введено правильно, система переходить до наступної форми.
Стан системи після випробування	Система відкриває форму головного меню клієнта.

Таблиця 5.2 – Перевірка випробувань реєстрації

Назва випробування	Випробування реєстрації
Стан системи перед випробуванням	Система відкриває форму реєстрації
Вхідні данні	Ім'я та прізвище, номер телефону, пароль.
Вихідні данні	Якщо користувач заповнив поле неправильно або не заповнив одне з полів , система показує повідомлення щодо неправильного вводу даних. Якщо дані введено правильно, тоді реєстрація пройшла успішно.
Стан системи після випробування	Система відкриває форму авторизації.

Таблиця 5.3 – Перевірка випробувань додавання водія адміністратором

Назва випробування	Випробування додавання водія
Стан системи перед випробуванням	Система відкриває форму додавання водія
Вхідні данні	Ім'я та прізвище, клас автомобіля, марка автомобіля, кількість місць, наявність дитячого крісла, можливість перевозу тварин, можливість перевозу людини з COVID.
Вихідні данні	Якщо адміністратор заповнив поле неправильно або не заповнив одне з полів, система показує повідомлення щодо неправильного вводу даних. Якщо дані введені правильно, тоді реєстрація пройшла успішно.
Стан системи після випробування	Система відкриває форму головного меню адміністратора.

Таблиця 5.4 – Перевірка випробувань поповнення рахунку

Назва випробування	Випробування поповнення рахунку
Стан системи перед випробуванням	Система відкриває форму поповнення рахунку
Вхідні данні	Реквізити банківської карти
Вихідні данні	Якщо клієнт заповнив поле неправильно або не заповнив одне з полів , система показує повідомлення щодо неправильного вводу даних. Якщо дані введені правильно, тоді поповнення рахунку пройшло успішно.
Стан системи після випробування	Система відкриває форму головного меню клієнта.

Таблиця 5.5 – Перевірка випробувань замовлення таксі (1)

Назва випробування	Випробування замовлення таксі
Стан системи перед випробуванням	Система відкриває форму введення адрес.
Вхідні данні	Адреса місцезнаходження клієнта та кінцевої адреси.
Вихідні данні	Якщо клієнт заповнив поле неправильно або не заповнив одне з полів , система показує повідомлення щодо неправильного вводу даних. Якщо дані введені правильно, тоді перехід до наступного форми.
Стан системи після випробування	Система відкриває форму підтвердження замовлення.

Таблиця 5.6 – Перевірка випробувань замовлення таксі (2)

Назва випробування	Випробування замовлення таксі
Стан системи перед випробуванням	Система відкриває форму вибору класу автомобіля
Вхідні данні	Вибір класу автомобіля, підтвердження замовлення.
Вихідні данні	Ціна за проїзд. Якщо клієнт не має достатньо коштів на рахунку, система показує повідомлення. Якщо коштів достатньо переходить на наступну форму.
Стан системи після випробування	Система відкриває форму підбір водія.

Таблиця 5.7 – Перевірка випробувань замовлення таксі (3)

Назва випробування	Випробування замовлення таксі
Стан системи перед випробуванням	Система відкриває форму підбір водія
Вхідні данні	Вибір водія, додавання критеріїв автомобіля: наявність дитячого крісла, можливість перевозу тварин, можливість перевозу людини з COVID
Вихідні данні	Якщо система не знайшла водіїв, буде виведено повідомлення. Якщо клієнт обрав тоді замовлення таксі виконано успішно.
Стан системи після випробування	Система відкриває форму оцінка

Таблиця 5.8 – Перевірка випробувань оцінка водія

Назва випробування	Випробування оцінка водія
Стан системи перед випробуванням	Система відкриває форму оцінка
Вхідні данні	Оцінка водія і коментар
Вихідні данні	Якщо клієнт не поставив оцінку, система виведе повідомлення. Якщо клієнт поставив оцінку, то оцінювання виконано успішно. Зняття плати за проїзд з рахунка клієнта.
Стан системи після випробування	Система відкриває форму головного меню

Висновок до розділу

В цьому розділі було розглянуто основні функції та інструкції роботи з системою. Також було наведено скріншоти інтерфейсу програмного продукту. Проведено випробування основних частин системи, було розглянуто стан системи до замовлення та після, вхідні та вихідні дані.

ЗАГАЛЬНІ ВИСНОВКИ

В першому розділі проведений аналіз становища в галузі сервісів таксі та виявлені проблеми, обґрунтовано доцільність розробки інформаційної системи з підтримки діяльності служби таксі. Сформульовано ціль та задачі розробки системи. Розроблена діаграма послідовності для подальшого проектування системи та сформована функціональна модель, яка описана за варіантами використання та діями акторів.

Інформаційне забезпечення містить опис вхідних та вихідних даних, перелік та опис сутностей в базі даних, а також структуру бази даних

Змістовна постановка задачі математичного забезпечення полягає в наданні змоги клієнтам служби швидко знайти автомобіль за їхніми побажаннями за умови, що про якість обслуговування клієнт зможе залишити відгук и поставити оцінку за поїздку. Виконання задачі розглянуто як створення алгоритмів, за якими система відпрацьовує завдання пошуку автомобіля і водія, а також завдання отримання відгука та оцінки від клієнта про поїздку та водія і розрахунку плати за поїздку.

Для реалізації проєкту виконані необхідні діаграми: послідовності, діяльності, варіантів використання, компонентів системи. Розроблено програмне та технічне забезпечення для системи, діаграма класів.

В технологічному розділі проєкту розроблено керівництво для користувача та системного адміністратора для детально описано основні функції та інструкції для процесу роботи з системою.

Проведено випробування функцій інформаційної системи з підтримки служби діяльності таксі, які показали відповідність вимогам технічного завдання.

					ДП 7232.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		47

ПЕРЕЛІК ПОСИЛАНЬ

1. «URL:
METANIT.COM »
2. «URL:
<https://testmatick.com/ru/osnovnye-ponyatiya-i-osobennosti-klient-servernoj-arhitektury/> »
3. «URL:
<https://uk.wikipedia.org/wiki/%D0%A2%D0%B0%D0%BA%D1%81%D1%96#%D0%86%D1%81%D1%82%D0%BE%D1%80%D1%96%D1%8F> »
4. «URL:
<https://www.uber.com/ua/ru/about/> »
5. «URL:
<https://www.otzyvua.net/uber/plus-i-minus> »
6. «URL:
<https://uklon.com.ua/> »
7. «URL:
<https://www.otzyvua.net/onlayn-zakaz-taksi-ukloncomua/plus-i-minus> »
8. «URL:
<https://bolt.eu/ru/> »
9. «URL:
<https://www.otzyvua.net/bolt-bolt/plus-i-minus> »
10. «URL:
<https://catalog.taxiservice.com.ua/taxi-kiev/lajk-taksi-221> »
11. «URL:
<https://www.otzyvua.net/uk/lajk-taksi/plus-i-minus> »
12. «URL:
https://skillbox.ru/media/code/entity_framework/ »
13. «URL:
https://flexberry.github.io/ru/gbt_csharp.html »

					ДП 7232.00.000 ПЗ	Арк.
						48
Змн.	Арк.	№ докум.	Підпис	Дата		

Додаток А

Тексти програмного коду**Інформаційної системи з підтримки діяльності служби таксі**

(Найменування програми (документа))

DVD-R

(Вид носія даних)

33 арк, 144 Кб

(Обсяг програми (документа) , арк., Кб)

Київ – 2021 року

					ДП 7232.00.000 ПЗ	Арк.
						49
Змн.	Арк.	№ докум.	Підпис	Дата		

AdminMain

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Navigation;
using System.Windows.Shapes;

namespace TaxiService.Pages {
    /// <summary>
    /// Interaction logic for AdminMain.xaml
    /// </summary>
    public partial class AdminMain : Page {
        public Frame F { get; set; }
        public AdminMain(Frame F) {
            InitializeComponent();
            this.F = F;
        }
        // Переход на страницу с таблицей пользователей
        private void Button_Click(object sender,
RoutedEventArgs e) {
            Classes.Pages.SetDUListPage(F, 0);
            F.Navigate(Classes.Pages.DUListPage);
        }
        // Переход на страницу с таблицей водителей
        private void Button_Click_1(object sender,
RoutedEventArgs e) {
            Classes.Pages.SetDUListPage(F, 1);
            F.Navigate(Classes.Pages.DUListPage);
        }
        // Переход на страницу с таблицей выдаления
водителей
        private void Button_Click_2(object sender,
RoutedEventArgs e) {
            Classes.Pages.SetDUListPage(F, 2);
        }
    }
}
```

```

        F.Navigate(Classes.Pages.DUListPage);
    }
    // Переход на страницу добавления водителя
    private void Button_Click_3(object sender,
RoutedEventArgs e) {
        Classes.Pages.SetAddDriverPage(F);
        F.Navigate(Classes.Pages.AddDriverPage);
    }
    // Переход на страницу логинации
    private void Button_Click_4(object sender,
RoutedEventArgs e) {
        Classes.Pages.SetLoginPage(F);
        F.Navigate(Classes.Pages.LoginPage);
    }
}
}

```

Deposit

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Navigation;
using System.Windows.Shapes;
using TaxiService.Classes;

namespace TaxiService.Pages {
    /// <summary>
    /// Interaction logic for Deposit.xaml
    /// </summary>
    public partial class Deposit : Page {
        public Frame F { get; set; }
        public Deposit(Frame F) {
            InitializeComponent();
        }
    }
}

```



```

        this.F = F;
    }
    // Пополнение суммы и апдейт БД
    private void Button_Click(object sender,
RoutedEventArgs e) {
        if(textBox1.Text.Length == 16 &&
textBox2.Text.Length == 2 && textBox3.Text.Length == 2 &&
textBox4.Text.Length == 3) {

            if(Validator.ValidateNumbers(textBox1.Text) &&
Validator.ValidateNumbers(textBox2.Text) &&
Validator.ValidateNumbers(textBox3.Text) &&
Validator.ValidateNumbers(textBox4.Text) &&
Validator.ValidateNumbers(textBox5.Text)) {
                if (Convert.ToInt32(textBox2.Text)
<= 12 && Convert.ToInt32(textBox2.Text) >= 1 &&
Convert.ToInt32(textBox3.Text) >= 21) {
                    List<User> L =
MainWindow.DB.Users.ToList();
                    for (int i = 0; i < L.Count;
i++) {
                        if (MainWindow.U.Phone ==
L[i].Phone) {
                            User U =
MainWindow.DB.Users.SingleOrDefault(u => u.Phone ==
MainWindow.U.Phone);
                            if (U != null) {
                                U.Money +=
Convert.ToInt32(textBox5.Text);

                                MainWindow.DB.SaveChanges();

                                Classes.Pages.SetUserMainPage(F);

                                F.Navigate(Classes.Pages.UserMainPage);

                                return;
                            }
                        }
                    }
                }
            }
            MessageBox.Show("Введено неправильно
дані");
            return;
        }
    }
}

```

```

        MessageBox.Show("Введено      неправильно
дані");
        return;
    }
    MessageBox.Show("Введено неправильно дані");
    return;
}
// Возвращает на главную страницу
private void Button_Click1(object sender,
RoutedEventArgs e) {
    Classes.Pages.SetUserMainPage(F);
    F.Navigate(Classes.Pages.UserMainPage);
}
}
}

```

DUList

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Navigation;
using System.Windows.Shapes;
using TaxiService.Classes;

namespace TaxiService.Pages {
    /// <summary>
    /// Interaction logic for DUList.xaml
    /// </summary>
    public partial class DUList : Page {
        public Frame F { get; set; }
        public int T { get; set; }
    }
}

```

```

private List<DriverTable> PDTL { get; set; }
public DUList(Frame F, int T) {
    this.F = F;
    this.T = T;
    InitializeComponent();
}
// Инициализует таблицу
private void Page_Initialized(object sender,
EventArgs e) {
    if(T == 0) {
        List<User> L =
MainWindow.DB.Users.ToList();
        List<UserTable> UL = new
List<UserTable>();
        for(int i = 0; i < L.Count; i++) {
            UserTable UT = new
UserTable(L[i].Phone, L[i].FLName, L[i].Money);
            UL.Add(UT);
        }
        grid.ItemsSource = UL;
    }
    if(T == 1) {
        List<Driver> DL =
MainWindow.DB.Drivers.ToList();
        List<DriverTable> DTL = new
List<DriverTable>();
        PDTL = DTL;
        for(int i = 0; i < DL.Count; i++) {
            string Vclass = "";
            char AP = '-';
            char BP = '-';
            char CP = '-';
            if(DL[i].VehicleClass == 1) {
                Vclass = "Легковий";
            }
            else if(DL[i].VehicleClass == 2) {
                Vclass = "Мінівен";
            }
            else if (DL[i].VehicleClass == 3) {
                Vclass = "Грузовий";
            }
            else if (DL[i].VehicleClass == 4) {
                Vclass = "Автобус";
            }
        }
    }
}

```

```

        if(DL[i].AnimalPosibility == 1) {
            AP = '+';
        }
        if(DL[i].BabyChair == 1) {
            BP = '+';
        }
        if(DL[i].CarantinePosibility == 1) {
            CP = '+';
        }
        DriverTable DT = new
DriverTable(DL[i].FLName, DL[i].RideCount, DL[i].Rating,
Vclass, BP, AP, CP, DL[i].CountChairs, DL[i].AutoMark);
        DTL.Add(DT);
    }
    grid.ItemsSource = DTL;
}
if(T == 2) {
    List<Driver> DL =
MainWindow.DB.Drivers.ToList();
    List<DriverTable> DTL = new
List<DriverTable>();
    PDTL = DTL;
    for (int i = 0; i < DL.Count; i++) {
        string Vclass = "";
        char AP = '-';
        char BP = '-';
        char CP = '-';
        if (DL[i].VehicleClass == 1) {
            Vclass = "Легковий";
        }
        else if (DL[i].VehicleClass == 2) {
            Vclass = "Мінівен";
        }
        else if (DL[i].VehicleClass == 3) {
            Vclass = "Грузовий";
        }
        else if (DL[i].VehicleClass == 4) {
            Vclass = "Автобус";
        }
        if (DL[i].AnimalPosibility == 1) {
            AP = '+';
        }
        if (DL[i].BabyChair == 1) {
            BP = '+';
        }
    }
}

```

```

        }
        if (DL[i].CarantinePosibility == 1)
    {
        CP = '+';
    }
    DriverTable DT = new
DriverTable(DL[i].FLName, DL[i].RideCount, DL[i].Rating,
Vclass, BP, AP, CP, DL[i].CountChairs, DL[i].AutoMark);
    DTL.Add(DT);
    }
    grid.ItemsSource = DTL;
    }
    }
    // Возвращает на главную страницу
    private void Button_Click(object sender,
RoutedEventArgs e) {
        Classes.Pages.SetAdminMainPage(F);
        F.Navigate(Classes.Pages.AdminMainPage);
    }
    // При выборе удаляет из БД, если значение 2
    private void grid_SelectionChanged(object
sender, SelectionChangedEventArgs e) {
        if (T == 2) {
            if (grid.SelectedIndex != -1) {
                string T =
(grid.Items[grid.SelectedIndex] as DriverTable).FLName;
                List<Driver> DL =
MainWindow.DB.Drivers.ToList();
                for (int i = 0; i < DL.Count; i++) {
                    if (DL[i].FLName == T) {
                        MainWindow.DB.Drivers.Remove(DL[i]);
                        MainWindow.DB.SaveChanges();
                        PDTL.RemoveAt(i);
                        grid.ItemsSource = null;
                        grid.ItemsSource = PDTL;
                        return;
                    }
                }
            }
        }
    }
}

```

Login

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Navigation;
using System.Windows.Shapes;
using TaxiService.Classes;

namespace TaxiService.Pages {
    /// <summary>
    /// Interaction logic for Login.xaml
    /// </summary>
    public partial class Login : Page {
        public Frame F { get; set; }
        public Login(Frame F) {
            InitializeComponent();
            this.F = F;
        }
        // Проверка с БД авторизации
        private void Button_Click(object sender,
RoutedEventArgs e) {
            if(!string.IsNullOrEmpty(textBox1.Text) &&
!string.IsNullOrEmpty(textBox2.Password)) {

if(Validator.ValidateNumbers(textBox1.Text) &&
textBox1.Text.Length == 10) {
                List<Administrator> AL =
MainWindow.DB.Administrators.ToList();
                for (int i = 0; i < AL.Count; i++) {
                    if (AL[i].Phone == textBox1.Text
&& AL[i].Password == textBox2.Password) {
```

```

Classes.Pages.SetAdminMainPage(F);

F.Navigate(Classes.Pages.AdminMainPage);
    return;
}
}
List<User> L =
MainWindow.DB.Users.ToList();
    for(int i = 0; i < L.Count; i++) {
        if(L[i].Phone == textBox1.Text
        && L[i].Password == textBox2.Password) {
            MainWindow.U = L[i];

Classes.Pages.SetUserMainPage(F);

F.Navigate(Classes.Pages.UserMainPage);
    return;
}
}
MessageBox.Show("Такого користувача
не існує");
    return;
}
}
MessageBox.Show("Неправильно введенні
дані");
    return;
}
// Переход на страницу регистрации
private void
Label_PreviewMouseDown(object sender,
MouseButtonEventArgs e) {
    Classes.Pages.SetRegisterPage(F);
    F.Navigate(Classes.Pages.RegisterPage);
}
// Выход из программы
private void Button_Click_4(object sender,
RoutedEventArgs e) {
    MainWindow.Exited = 1;
}
}
}
}

```

Offer

```
using System;
using System.Collections.Generic;
using System.Configuration;
using System.Linq;
using System.Reflection;
using System.Runtime.InteropServices;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Navigation;
using System.Windows.Shapes;

namespace TaxiService.Pages {
    /// <summary>
    /// Interaction logic for Offer.xaml
    /// </summary>
    public partial class Offer : Page {
        public Frame F { get; set; }
        public Offer(Frame F) {
            InitializeComponent();
            this.F = F;
        }
        // Реализация отключения джаваскриптовских
ошибок
        public static void SetSilent(WebBrowser browser,
bool silent) {
            if (browser == null)
                throw new
ArgumentNullException("browser");

            // get an IWebBrowser2 from the document
            IOleServiceProvider sp = browser.Document as
IOleServiceProvider;
            if (sp != null) {
                Guid IID_IWebBrowserApp = new
Guid("0002DF05-0000-0000-C000-000000000046");
```



```

        Guid IID_IWebBrowser2 = new
Guid("D30C1661-CDAF-11d0-8A3E-00C04FC9E26E");

        object webBrowser;
        sp.QueryService(ref IID_IWebBrowserApp,
ref IID_IWebBrowser2, out webBrowser);
        if (webBrowser != null) {

webBrowser.GetType().InvokeMember("Silent",
BindingFlags.Instance | BindingFlags.Public |
BindingFlags.PutDispProperty, null, webBrowser, new object[]
{ silent });

        }
    }

    [ComImport, Guid("6D5140C1-7436-11CE-8034-
00AA006009FA"),
InterfaceType(ComInterfaceType.InterfaceIsIUnknown)]
    private interface IOleServiceProvider {
        [PreserveSig]
        int QueryService([In] ref Guid guidService,
[In] ref Guid riid, [MarshalAs(UnmanagedType.IDispatch)] out
object ppvObject);
    }

    // Переход на страницу подтверждения заказа
    private void Button_Click(object sender,
RoutedEventArgs e) {
        Classes.Pages.SetOfferConfirmPage(F);
        F.Navigate(Classes.Pages.OfferConfirmPage);
    }

    // Отключает джаваскриптовские ошибки
    private void Web_Navigated(object sender,
NavigationEventArgs e) {
        SetSilent(Web, true);
    }

    // Переход на главную страницу
    private void Button_Click_1(object sender,
RoutedEventArgs e) {
        Classes.Pages.SetUserMainPage(F);
        F.Navigate(Classes.Pages.UserMainPage);
    }
}
}

```

OfferConfirm

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Navigation;
using System.Windows.Shapes;

namespace TaxiService.Pages {
    /// <summary>
    /// Interaction logic for OfferConfirm.xaml
    /// </summary>
    public partial class OfferConfirm : Page {
        public Frame F { get; set; }
        public int Price { get; set; }
        public int T { get; set; }
        public OfferConfirm(Frame F) {
            InitializeComponent();
            combo.Items.Add("Легковий");
            combo.Items.Add("Мінівен");
            combo.Items.Add("Грузовий");
            combo.Items.Add("Автобус");
            this.F = F;
            Random R = new Random();
            T = R.Next(1, 15);
        }

        private void ComboBox_SelectionChanged(object sender, SelectionChangedEventArgs e) {
            if(combo.SelectedIndex != -1) {

                if(combo.SelectedIndex == 0) {
                    label1.Content = "Ціна: " +
Convert.ToString(T * 30) + "грн";
```

```

        Price = T * 30;
        MainWindow.VehicleClass = 1;
    }
    if(combo.SelectedIndex == 1) {
        label1.Content = "Ціна: " +
Convert.ToString(T * 25) + "грн";
        Price = T * 25;
        MainWindow.VehicleClass = 2;
    }
    if (combo.SelectedIndex == 2) {
        label1.Content = "Ціна: " +
Convert.ToString(T * 40) + "грн";
        Price = T * 40;
        MainWindow.VehicleClass = 3;
    }
    if (combo.SelectedIndex == 3) {
        label1.Content = "Ціна: " +
Convert.ToString(T * 20) + "грн";
        Price = T * 20;
        MainWindow.VehicleClass = 4;
    }
    MainWindow.Price = Price;
    MainWindow.Time = T * 90;
}
}
// Подтверждение заказа
private void Button_Click(object sender,
RoutedEventArgs e) {
    if(combo.SelectedIndex != -1) {
        Classes.Pages.SetOfferDriverPage(F);

F.Navigate(Classes.Pages.OfferDriverPage);
    }
}
// Переход на главную страницу
private void Button_Click_1(object sender,
RoutedEventArgs e) {
    Classes.Pages.SetUserMainPage(F);
    F.Navigate(Classes.Pages.UserMainPage);
}
}
}

```

OfferDriver

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Navigation;
using System.Windows.Shapes;
using TaxiService.Classes;

namespace TaxiService.Pages {
    /// <summary>
    /// Interaction logic for OfferDriver.xaml
    /// </summary>
    public partial class OfferDriver : Page {
        public Frame F { get; set; }
        public OfferDriver(Frame F) {
            InitializeComponent();
            this.F = F;
        }
        // Поиск водителя с БД
        private void Button_Click(object sender,
RoutedEventArgs e) {
            List<Driver> L =
MainWindow.DB.Drivers.ToList();
            List<DriverTableOrder> DTOL = new
List<DriverTableOrder>();
            for (int i = 0; i < L.Count; i++) {
                if (toogle1.IsChecked == true) {
                    if (L[i].BabyChair == 0) {
                        continue;
                    }
                }
                if (toogle2.IsChecked == true) {
                    if (L[i].AnimalPosibility == 0) {
                        continue;
                    }
                }
            }
        }
    }
}
```


OfferRating

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Navigation;
using System.Windows.Shapes;
using TaxiService.Classes;

namespace TaxiService.Pages {
    /// <summary>
    /// Interaction logic for OfferRating.xaml
    /// </summary>
    public partial class OfferRating : Page {
        public Frame F { get; set; }
        public OfferRating(Frame F) {
            InitializeComponent();
            this.F = F;
        }
        // Переход на главную страницу и добавление в БД
        private void Button_Click(object sender,
RoutedEventArgs e) {
            if (textBox1.Text.Length != 0) {
                List<User> L =
MainWindow.DB.Users.ToList();
                for (int i = 0; i < L.Count; i++) {
                    if (MainWindow.U.Phone ==
L[i].Phone) {
                        User U =
MainWindow.DB.Users.SingleOrDefault(u => u.Phone ==
MainWindow.U.Phone);
                        if (U != null) {
                            U.Money -= MainWindow.Price;
                            MainWindow.DB.SaveChanges();
                        }
                    }
                }
            }
        }
    }
}
```

```

        }
    }
    Comment C = new
Comment(MainWindow.IdDriver, textBox1.Text);
    MainWindow.DB.Comments.Add(C);
    MainWindow.DB.SaveChanges();
    List<Driver> DL =
MainWindow.DB.Drivers.ToList();
    for (int i = 0; i < DL.Count; i++) {
        if (MainWindow.IdDriver == DL[i].Id)
    {
        Driver D =
MainWindow.DB.Drivers.SingleOrDefault(d => d.Id ==
MainWindow.IdDriver);
        if (D != null) {
            if (bar.Value == 1) {
                D.Rating += 0.6 *
bar.Value;
            }
            else if (bar.Value == 2) {
                D.Rating += 0.7 *
bar.Value;
            }
            else if (bar.Value == 3) {
                D.Rating += 0.8 *
bar.Value;
            }
            else if (bar.Value == 4) {
                D.Rating += 0.9 *
bar.Value;
            }
            else if (bar.Value == 5) {
                D.Rating += bar.Value;
            }
            D.RideCount++;
            MainWindow.DB.SaveChanges();
        }
    }
}
Classes.Pages.SetUserMainPage(F);
F.Navigate(Classes.Pages.UserMainPage);
return;
}
MessageBox.Show("Неправильно введені дані");

```

```

        return;
    }
}

```

OfferTime

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Navigation;
using System.Windows.Shapes;
using System.Windows.Threading;

namespace TaxiService.Pages {
    /// <summary>
    /// Interaction logic for OfferTime.xaml
    /// </summary>
    public partial class OfferTime : Page {
        public Frame F { get; set; }
        private int Time { get; set; }
        private DispatcherTimer timer1 = new
DispatcherTimer();
        private DispatcherTimer timer2 = new
DispatcherTimer();
        public OfferTime(Frame F) {
            InitializeComponent();
            this.F = F;
            timer1.Interval = TimeSpan.FromSeconds(1);
            timer1.IsEnabled = true;
            timer1.Tick += new EventHandler(TimerTick1);
            timer1.Start();
            timer2.Stop();
        }
    }
}

```



```

        private void Grid_Initialized(object sender,
EventArgs e) {

        }

        private void Page_Initialized(object sender,
EventArgs e) {
            Time = new Random().Next(220, 800);
        }

        private void TimerTick1(object sender, EventArgs
e) {
            Time--;
            if(Time == 0) {
                MessageBox.Show("Перевищений ліміт
часу");
                Classes.Pages.SetUserMainPage(F);
                F.Navigate(Classes.Pages.UserMainPage);
            }
            label1.Content = Time / 60 + ":" + (Time -
(Time / 60) * 60);
        }
        private void TimerTick2(object sender, EventArgs
e) {
            if (MainWindow.Time >= 1) {
                MainWindow.Time--;
            }
            if(MainWindow.Time == 0) {
                MessageBox.Show("Перевищений ліміт
часу");
            }
            label2.Content = MainWindow.Time / 60 + ":"
+ (MainWindow.Time - (MainWindow.Time / 60) * 60);
        }
        // Остановление таймера
        private void Button_Click(object sender,
RoutedEventArgs e) {
            timer1.Stop();
            timer2.Interval = TimeSpan.FromSeconds(1);
            timer2.IsEnabled = true;
            timer2.Tick += new EventHandler(TimerTick2);
            timer2.Start();
        }
        // Переход на страницу рейтинга

```

```

        private void Button_Click_1(object sender,
RoutedEventArgs e) {
            timer1.Stop();
            Classes.Pages.SetOfferRatingPage(F);
            F.Navigate(Classes.Pages.OfferRatingPage);
        }
        // Переход на главную страницу
        private void Button_Click_2(object sender,
RoutedEventArgs e) {
            Classes.Pages.SetUserMainPage(F);
            F.Navigate(Classes.Pages.UserMainPage);
        }
    }
}

```

Register

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Navigation;
using System.Windows.Shapes;
using TaxiService.Classes;

namespace TaxiService.Pages {
    /// <summary>
    /// Interaction logic for Register.xaml
    /// </summary>
    public partial class Register : Page {
        public Frame F { get; set; }
        public Register(Frame F) {
            InitializeComponent();
            this.F = F;
        }
        // Добавление в БД пользователя
    }
}

```

```

        private void Button_Click(object sender,
RoutedEventArgs e) {
            if(!string.IsNullOrEmpty(textBox1.Text)    &&
!string.IsNullOrEmpty(textBox2.Text)                &&
!string.IsNullOrEmpty(textBox3.Password)) {

if(Validator.ValidateNumbers(textBox1.Text)           &&
Validator.ValidateText(textBox2.Text)                 &&
textBox1.Text.Length == 10) {
                User U = new User(textBox1.Text,
textBox2.Text, textBox3.Password);
                MainWindow.DB.Users.Add(U);
                MainWindow.DB.SaveChanges();
                Classes.Pages.SetLoginPage(F);
                F.Navigate(Classes.Pages.LoginPage);
                return;
            }
        }
        MessageBox.Show("Неправильно введені
дані");
    }
    // Переход на страницу логинации
    private void Button_Click_1(object sender,
RoutedEventArgs e) {
        Classes.Pages.SetLoginPage(F);
        F.Navigate(Classes.Pages.LoginPage);
    }
}
}

```

UserMain

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;

```

```

using System.Windows.Navigation;
using System.Windows.Shapes;
using System.Windows.Threading;

namespace TaxiService.Pages {
    /// <summary>
    /// Interaction logic for UserMain.xaml
    /// </summary>
    public partial class UserMain : Page {
        public Frame F { get; set; }
        public UserMain(Frame F) {
            InitializeComponent();
            this.F = F;
            label1.Content = "Вітаємо, " +
MainWindow.U.FLName;
            label2.Content = "Ваш баланс: " +
MainWindow.U.Money + " грн";
        }
        // Переход на страницу пополнения
        private void Button_Click(object sender,
RoutedEventArgs e) {
            Classes.Pages.SetDepositPage(F);
            F.Navigate(Classes.Pages.DepositPage);
        }
        // Переход на страницу заказа
        private void Button_Click_1(object sender,
RoutedEventArgs e) {
            Classes.Pages.SetOfferPage(F);
            F.Navigate(Classes.Pages.OfferPage);
        }
        // Переход на страницу логинации
        private void Button_Click_2(object sender,
RoutedEventArgs e) {
            Classes.Pages.SetLoginPage(F);
            F.Navigate(Classes.Pages.LoginPage);
        }
    }
}

```

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені ІГОРЯ СІКОРСЬКОГО”
Кафедра автоматизованих систем обробки інформації та управління

УЗГОДЖЕНО

Керівник проєкту

_____ Тамара ТЄЛИШЕВА

(підпис)

(вл. ім'я, прізвище)

“05” квітня 2021 р.

ЗАТВЕРДЖУЮ

В.о. завідувача кафедри

_____ Олександр ПАВЛОВ

(підпис)

(вл. ім'я, прізвище)

“06” квітня 2021 р.

Інформаційна система підтримки діяльності служби таксі

ТЕХНІЧНЕ ЗАВДАННЯ

Шифр *ДП 7232.01.000 ТЗ*

на 10 сторінках

Київ – 2021 року

ЗМІСТ

1 ЗАГАЛЬНІ ПОЛОЖЕННЯ.....	2
1.1 Повне найменування системи та її умовне позначення.....	2
1.2 Найменування організації-замовника та організацій-учасників робіт.....	2
1.3 Перелік документів, на підставі яких створюється система (Завдання на ДП).....	2
1.4 Планові терміни початку і закінчення роботи зі створення системи.....	3
2 ПРИЗНАЧЕННЯ І ЦІЛІ СТВОРЕННЯ СИСТЕМИ.....	4
2.1 Призначення системи.....	4
2.2 Цілі створення системи.....	4
3 ХАРАКТЕРИСТИКА ОБ'ЄКТА АВТОМАТИЗАЦІЇ.....	5
4 ВИМОГИ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	6
4.1 Вимоги до функціональних характеристик.....	6
4.2 Вимоги до надійності.....	6
4.3 Вимоги до складу і параметрів технічних засобів.....	6
5 СТАДІЇ І ЕТАПИ РОЗРОБКИ.....	8
6 ПОРЯДОК КОНТРОЛЮ ТА ПРИЙМАННЯ.....	9
6.1 Види випробувань.....	9

					ДП 7232.01.000 ТЗ			
Зм.	Арк.	Прізвище	Підпис	Дата				
Розроб.		Цеюков А.В.			Інформаційна система підтримки діяльності служби таксі	Літ.	Лист	Листів
Перевірив.		Тєлишева Т.О.					2	10
						КПІ ім. Ігоря Сікорського Каф. АСОІУ Гр. ІС-72		
Н. кон.		Проскура С.Л.						
Затв.		Павлов О.А.						

1 ЗАГАЛЬНІ ПОЛОЖЕННЯ

1.1 Повне найменування системи та її умовне позначення

Повна назва системи: Інформаційна система з підтримки діяльності служби таксі.

Коротке найменування системи: «Система служби таксі».

1.2 Найменування організації-замовника та організації-учасника робіт

Замовником є кафедра автоматизованих систем обробки інформації та управління Національного технічного університету України "Київський політехнічний інститут ім. Ігоря Сікорського" (далі за текстом — Замовник).
Адреса замовника: м. Київ, п. Перемоги 37, 18 корпус ФІОТ.

Розробник сервісу — студент групи ІС-72 кафедри автоматизованих систем обробки інформації та управління Національного технічного університету України «Київський політехнічний інститут ім. Ігоря Сікорського» Цеюков Артур Володимирович.

1.3 Перелік документів, на підставі яких створюється система

При розробці системи і створенні проектно-експлуатаційної документації Виконавець повинен керуватися вимогами наступних нормативних документів:

- ДСТУ 19.201-78. Технічне завдання. Вимоги до змісту і оформлення;
- ДСТУ 34.601-90. Комплекс стандартів на автоматизовані системи. Автоматизовані системи. Стадії створення;
- ДСТУ 34.201-89. Інформаційні технології. Комплекс стандартів на автоматизовані системи. Види, комплексність і позначення документів при створенні автоматизованих систем.

					ДП 7232.01.000 ТЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		2

1.4 Планові терміни початку і закінчення роботи зі створення системи

Плановий термін початку роботи над створенням інформаційної системи з підтримки діяльності служби таксі – 1 грудня 2021 року.

Плановий термін по закінченню роботи над створенням інформаційної системи з підтримки діяльності служби таксі – 25 травня 2021 року.

					ДП 7232.01.000 ТЗ	Арк.
						3
Змн.	Арк.	№ докум.	Підпис	Дата		

2 ПРИЗНАЧЕННЯ І ЦІЛІ СТВОРЕННЯ СИСТЕМИ

2.1 Призначення системи

Система з підтримки діяльності служби таксі призначена для Автоматизації процесів надання послуги з замовлення таксі.

Цілі створення системи

Ціль: підвищення якості обслуговування клієнтів з використання послуг таксі, за допомогою системи оцінювання послуг і водіїв.

Для досягнення цілі необхідно розв'язати наступні задачі:

- визначити функції системи оцінювання послуг і водіїв;
- обґрунтувати вибір вхідних і вихідних даних та створити базу даних;
- розробити архітектуру системи;
- спроектувати інтерфейс;
- реалізувати систему оцінювання послуг і водіїв;
- провести шляхом тестування системи.

					ДП 7232.01.000 ТЗ	Арк.
						4
Змн.	Арк.	№ докум.	Підпис	Дата		

3 ХАРАКТЕРИСТИКА ОБ'ЄКТА АВТОМАТИЗАЦІЇ

Для того, щоб користуватися системою, користувач повинен бути зареєстрованим і авторизованим.

Об'єктами автоматизації є процеси, побудова рейтингу та управління відбору водіїв.

Клієнт має змогу:

- реєстрації;
- авторизації;
- введення адресу місцезнаходження;
- введення кінцевого адресу;
- вибір класу автомобіля;
- підтвердження замовлення згідно предоставленою ціною проїзду;
- виставлення оцінки водію.

					ДП 7232.01.000 ТЗ	Арк.
						5
Змн.	Арк.	№ докум.	Підпис	Дата		

4 ВИМОГИ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

4.1 Вимоги до функціональних характеристик

Актор «клієнт» повинен зробити наступні кроки:

- реєстрація користувача;
- авторизація користувача;
- додавання адреси місцезнаходження клієнта;
- додавання адреси кінцевої точки;
- вибір класу автомобіля;
- підтвердження замовлення згідно предоставленою ціною проїзду;
- виставлення оцінки водію.

Алгоритм здійснює пошук вільного водія за обраними критеріями, розраховує вартість поїздки

Вимоги до надійності

Система служби таксі повинна функціонувати без відмінно, навіть якщо вона буде мати ймовірні дефекти, які можуть бути виявлені під час експлуатації. На етапі бета-тестування та розробки - дефекти повинні бути виправлені.

Помилки в програмному забезпеченні не повинні призводити до руйнувань бази даних.

Всі аварійні ситуації повинні бути задокументовані і відправлені розробникам, щоб в майбутньому усунути причини збою системної роботи.

4.2 Вимоги до складу і параметрів технічних засобів

Мінімальні вимоги для технічного забезпечення клієнту:

- оперативна пам'ять – 512 Мб;
- місткість жорсткого диску – 1 Гб;
- процесор – одноядерний, 1 ГГц;
- монітор, HD-графічна карта.

					ДП 7232.01.000 ТЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		6

Рекомендовані вимоги для технічного забезпечення клієнту:

- оперативна пам'ять – 2 Гб;
- місткість жорсткого диску – 1 Гб;
- процесор – двоядерний, 1 ГГц;
- монітор, HD-графічна карта.

Мінімальні вимоги для технічного забезпечення серверу:

- оперативна пам'ять – 1 Гб;
- місткість жорсткого диску – 5 Гб;
- процесор – двоядерний, 1 ГГц.

Рекомендовані вимоги для технічного забезпечення серверу:

- оперативна пам'ять – 4 Гб;
- місткість жорсткого диску – 20 Гб;
- процесор – восьмиядерний, 3 ГГц.

					ДП 7232.01.000 ТЗ	Арк.
						7
Змн.	Арк.	№ докум.	Підпис	Дата		

5 СТАДІЇ І ЕТАПИ РОЗРОБКИ

Основні етапи виконання робіт з розробки системи підтримки діяльності служби таксі:

№ з/п	Назва етапів виконання дипломного проєкту	Термін виконання етапів проєкту	Примітка
1.	Вивчення рекомендованої літератури	17.04.2021	
2.	Аналіз існуючих методів розв'язання задачі	23.04.2021	
3.	Постановка та формалізація задачі	23.04.2021	
4.	Розробка інформаційного забезпечення	30.04.2021	
5.	Алгоритмізація задачі	04.05.2021	
6.	Обґрунтування використовуваних технічних засобів	10.05.2021	
7.	Розробка програмного забезпечення	14.05.2021	
8.	Налагодження програми	01.06.2021	
9.	Виконання графічних документів	01.06.2021	

6 ПОРЯДОК КОНТРОЛЮ ТА ПРИЙМАННЯ СИСТЕМИ

6.1 Види випробувань

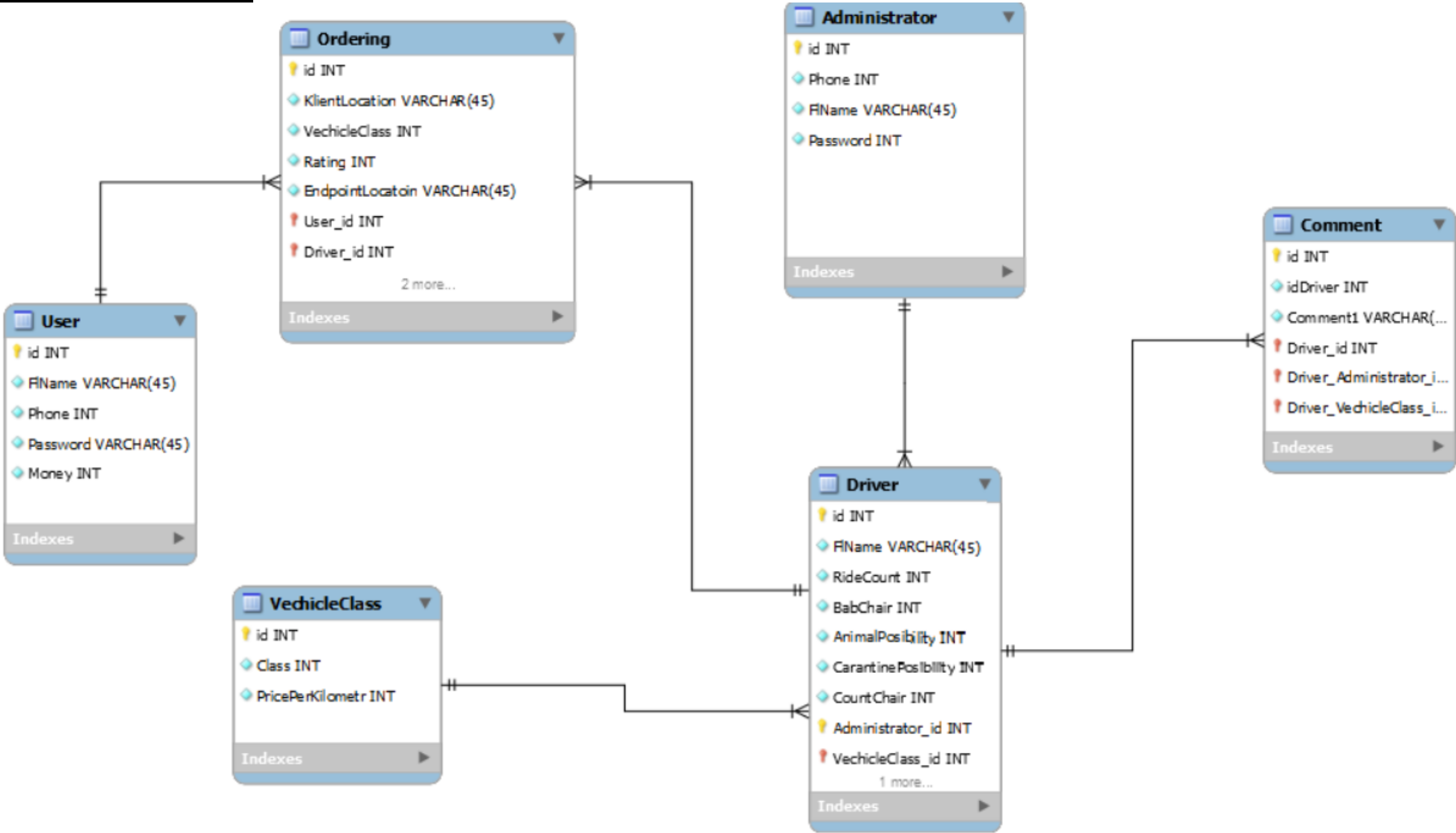
Для того щоб програмний продукт працював коректно буде проведено ряд функціональних випробувань. В ході тестування буде проведено перевірку роботи всіх функцій, з метою усунення недоліків. Система, також, буде проходити тестування на коректну роботу шляхом виконання некоректних кроків клієнтом. Окремо система буде проходити перевірку безпеки, окремими тестам.

					ДП 7232.01.000 ТЗ	Арк.
						9
Змн.	Арк.	№ докум.	Підпис	Дата		

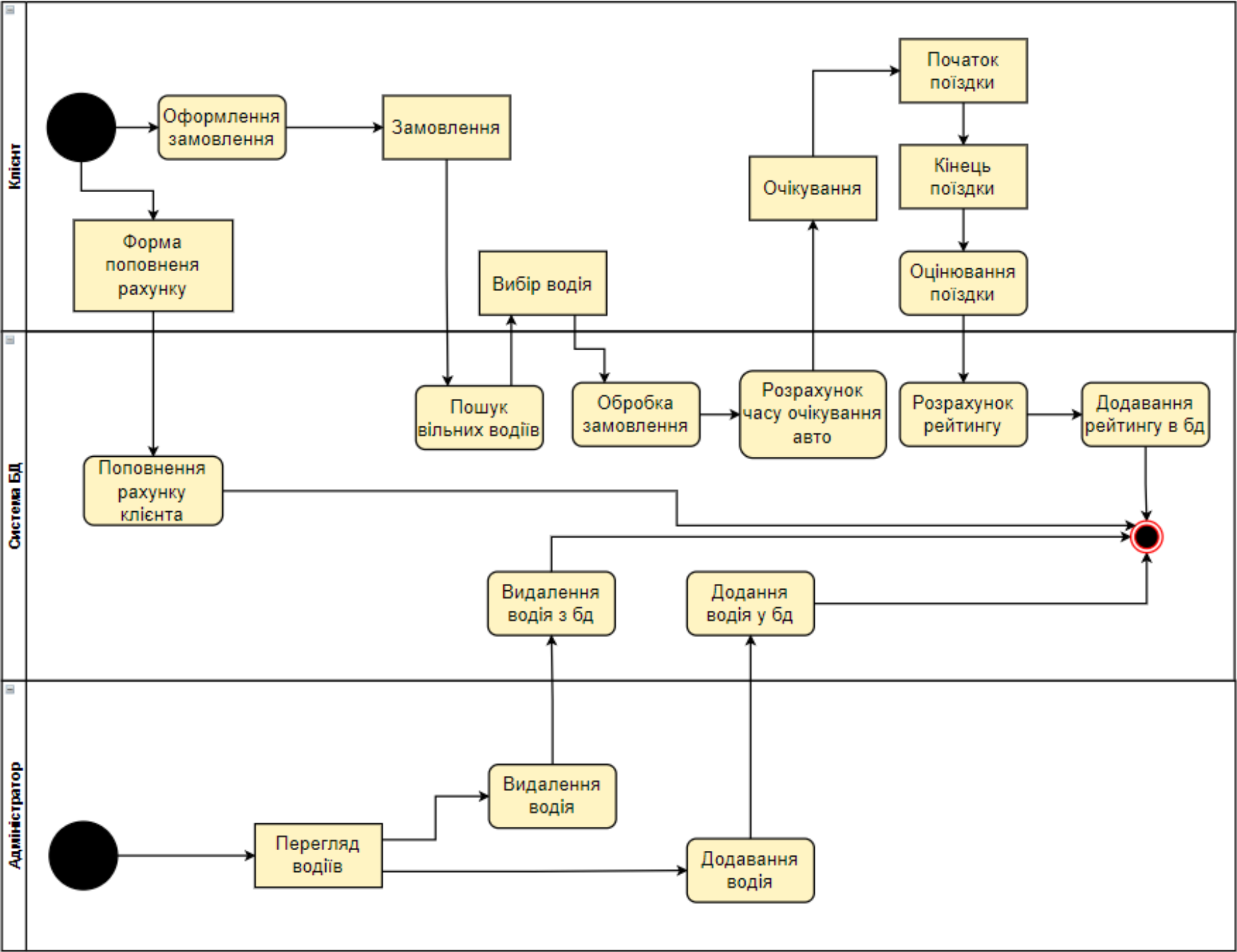
Графічний матеріал до дипломного проєкту

на тему: Інформаційна система з підтримки діяльності служби таксі

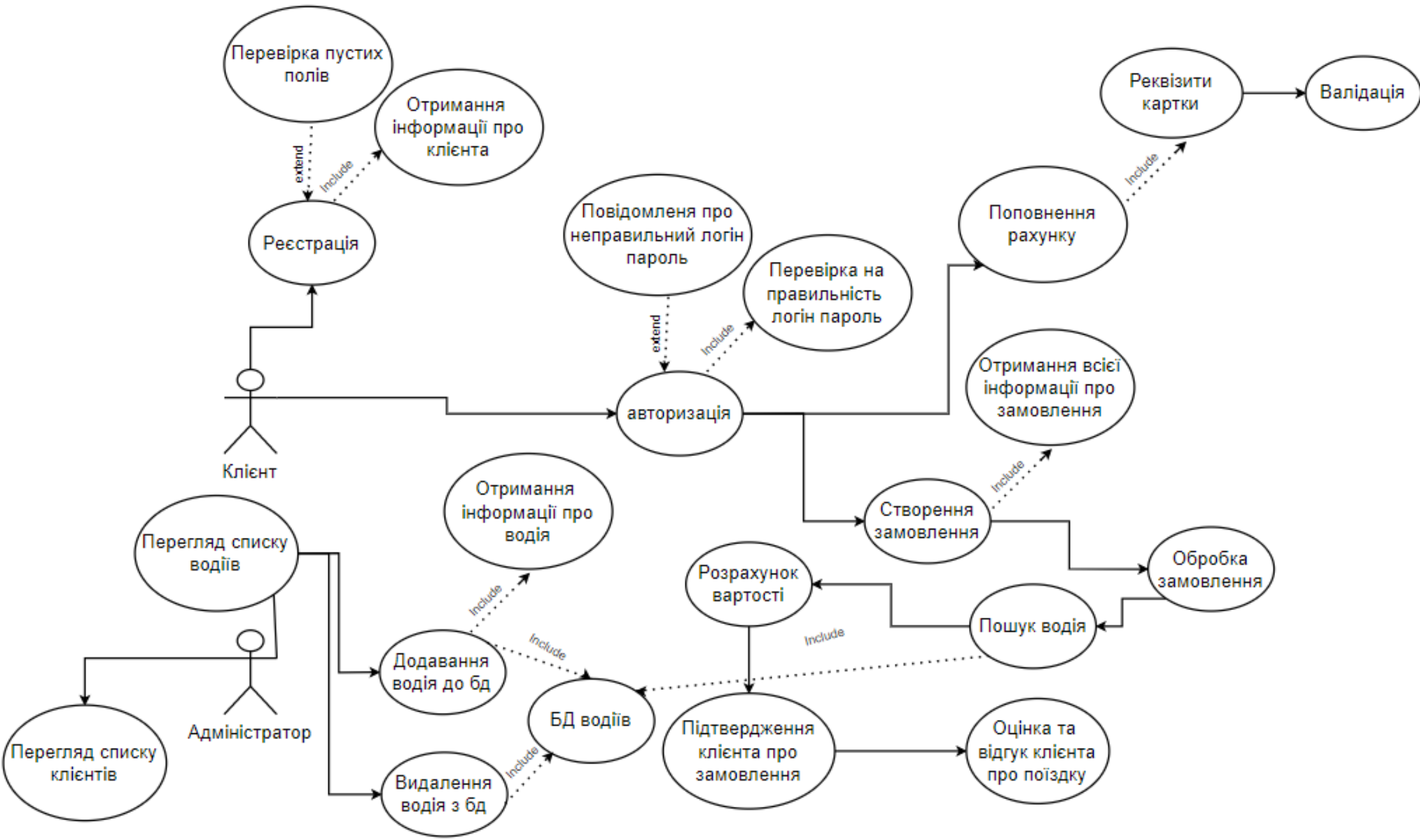
Київ – 2021 року



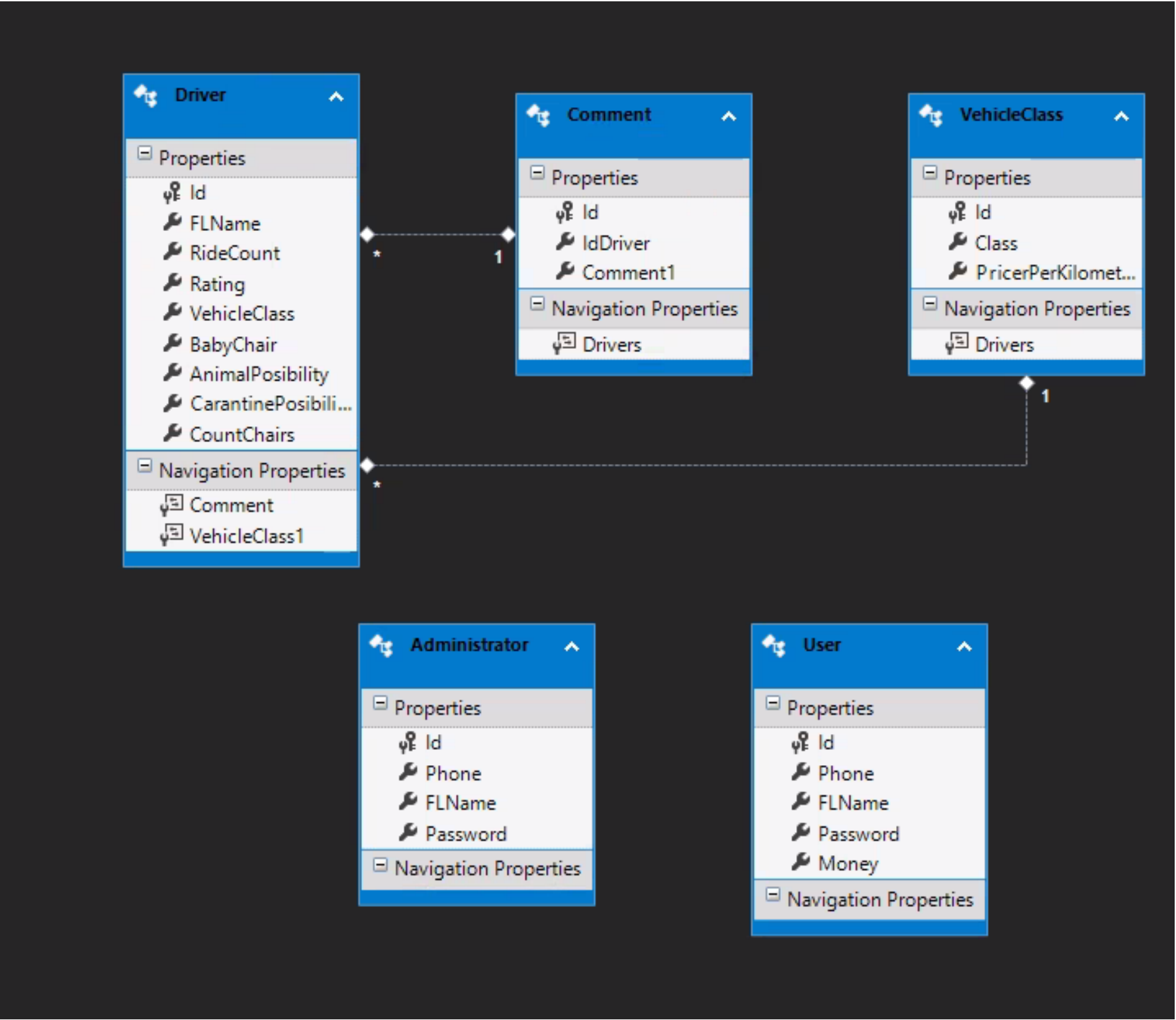
					ДП 7232.03.000 СБД							
					СТруктура бази даних			Літера		Маса	Масштаб	
Зм.	Арк.	№ документа	Підпис	Дата								
Розробив		Цеюков А.В.										
Перевірів		Телишева Т.О.						Аркуш 1		Аркушів 1		
Консульт.								Інформаційна система з підтримки діяльності служби таксі				
Н. кон.		Новінський В.П.										
Затвердив		Телишева Т.О.										
							КПІ ім. Ігоря Сікорського каф. АСОІУ ар. ІС-72					



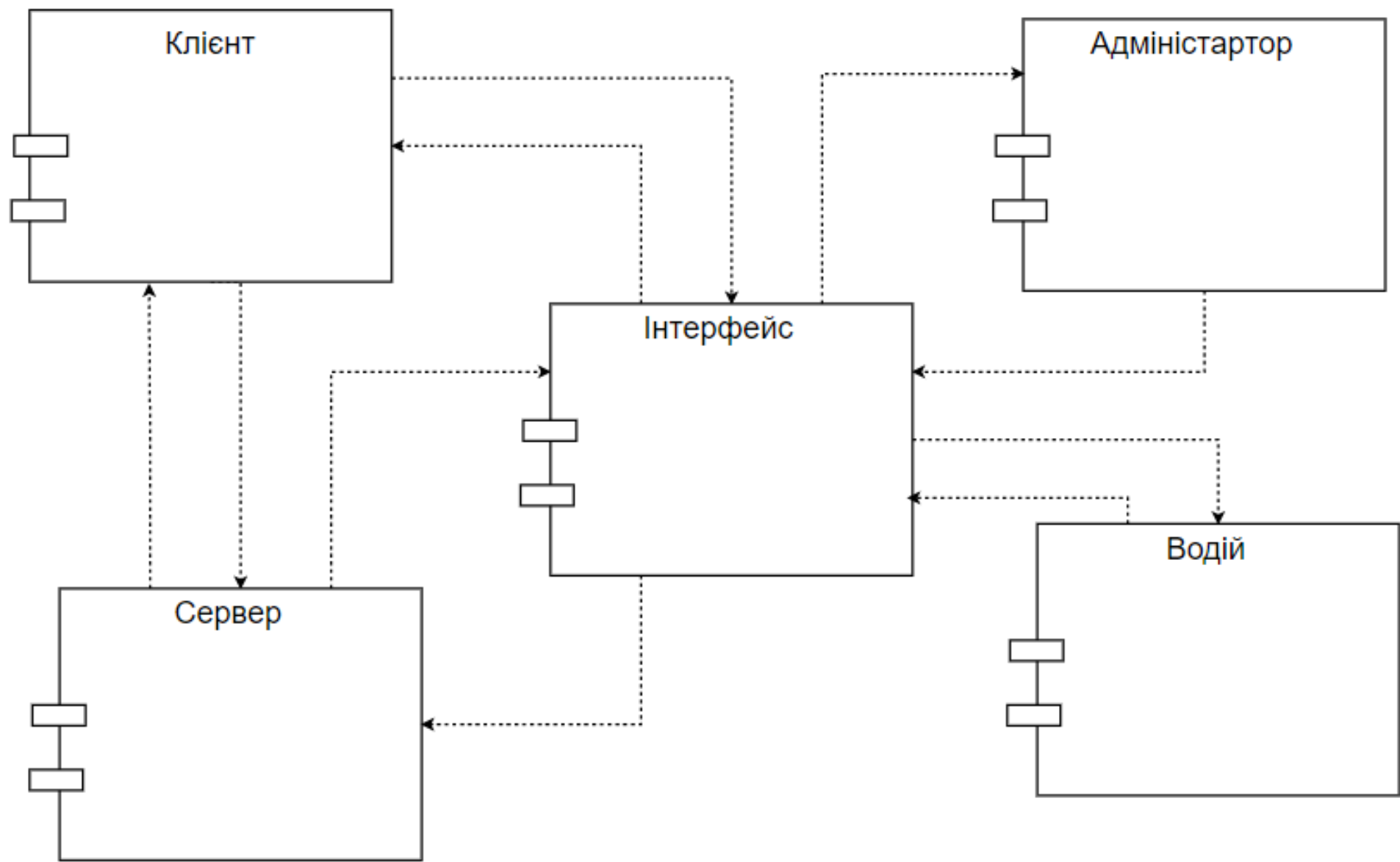
					ДП 7232.05.000 ССД			
					Схема діаграма діяльності Інформаційна система з підтримки діяльності служби таксі	Літера	Маса	Масштаб
Зм.	Арк.	№ документа	Підпис	Дата				
Розробив	Цеюков А.В.							
Перевірів	Телишева Т.О.					Аркуш 1	Аркушів 1	
Консульт.						КПІ ім. Ігоря Сікорського Каф. АСОІУ Гр. ІС-72		
Н. кон.	Новінський В.П.							
Затвердив	Телишева Т.О.							



					ДП 7232.02.000 ССВ				
					Схема структурна функціональної моделі		Літера		Маса
									Масштаб
Зм.	Арк.	№ документа	Підпис	Дата					
Розробив		Цеюков А.В.							
Перевірів		Телишева Т.О.					Аркуш 1		Аркушів 1
Консульт.					Інформаційна система з підтримки діяльності служби таксі		КПІ ім. Ігоря Сікорського Каф. АСОІУ Гр. ІС-72		
Н. кон.		Новінський В.П.							
Затвердив		Телишева Т.О.							



					ДП 7232.04.000 ССК									
					Схема структурна класів програмного забезпечення					Літера		Маса	Масштаб	
										Аркуш 1		Аркушів 1		
										Інформаційна система з підтримки діяльності служби таксі				
Зм.	Арк.	№ документа	Підпис	Дата										
Розробив	Цеюков А.В.													
Перевірів	Телишева Т.О.													
Консульт.					Інформаційна система з підтримки діяльності служби таксі					КПІ ім. Ігоря Сікорського каф. АСОІУ гр. ІС-72				
Н. кон.	Новінський В.П.													
Затвердив	Телишева Т.О.													



					ДП 7232.06.000 ССК						
					Схема структурна компонентів	Літера			Маса	Масштаб	
Зм.	Арк.	№ документа	Підпис	Дата							
Розробив	Цеюков А.В.										
Перевірів	Телишева Т.О.					Аркуш 1			Аркушів 1		
Консульт.											
Н. кон.	Новінський В.П.				Інформаційна система з підтримки діяльності служби таксі	КПІ ім. Ігоря Сікорського Каф. АСОІУ Гр. ІС-72					
Затвердив	Телишева Т.О.										

Служба таксі

Номер телефону

10 цифр

Пароль

Пароль

Авторизуватися

Зареєструватися

Поиск на Google Картах

Информация о COVID-19

Просмотр данных о заболеваемости на карте

Кропивницький 24°

Продуктовые магазины Рестораны Еда на вынос Гостиницы

Показать похожие

Вирахувати відстань

Назад

Підтвердження замовлення

Оберіть клас автомобіля

Легковий

Ціна: 180грн

Підтвердити

На головну

Підбір водія

FLName	RideCount	Rating	CountChairs	AutoMark
Taras Nishchuk	3	15	4	Ford Mustang

Наявність дитячого крісла

Можливість перевезу тварини

Можливість перевезу людини з COVID

Знайти

На головну

					ДП 7232.07.000 КЕ					
					Креслення вигляду екранних форм	Літера			Маса	Масштаб
Зм.	Арк.	№ документа	Підпис	Дата						
Розробив	Цеюков А.В.									
Перевірів	Телишева Т.О.					Аркуш 1			Аркушів 1	
Консульт.						КПІ ім. Ігоря Сікорського Каф. АСОІУ Гр. ІС-72				
Н. кон.	Новінський В.П.				Інформаційна система з підтримки діяльності служби таксі					
Затвердив	Телишева Т.О.									

Ім'я користувача:
Попенко Володимир Дмитрович

ID перевірки:
1008291106

Дата перевірки:
14.06.2021 13:22:25 EEST

Тип перевірки:
Doc vs Internet + Library

Дата звіту:
14.06.2021 13:59:04 EEST

ID користувача:
77149

Назва документа: Chejukov_bachelor_is72_2

Кількість сторінок: 48 Кількість слів: 5236 Кількість символів: 37964 Розмір файлу: 2.10 MB ID файлу: 1008359999

Виявлено модифікації тексту (можуть впливати на відсоток схожості)

18%
Схожість

Найбільша схожість: 4.56% з джерелом з Бібліотеки (ID файлу: 1000066113)

7.91% Джерела з Інтернету

92

Сторінка 50

17.4% Джерела з Бібліотеки

521

Сторінка 51

0% Цитат

Не знайдено жодних цитат

Вилучення списку бібліографічних посилань вимкнене

0%
Вилучень

Немає вилучених джерел

Модифікації

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

Підозріле форматування

14
сторінок