

## Зміст

1. Вступ.....	2
2. Завдання.....	2
3. Опис методів та засобів, які застосовуються при проектуванні.....	2
4. Специфікування проекту	
4.1 Встановлення вимог.....	2
4.2 Специфікування вимог.....	10
4.3 Моделювання поведінки системи.....	12
4.4 Проектування архітектури ПЗ.....	15
4.5 Детальне проектування.....	16
5. Генерація програмного коду.....	18
6. Висновок.....	20
7. Список літератури.....	20

## 1. Вступ

В період масової автоматизації надання послуг, більша кількість людей обирає найбільш просунуті способи користування послугами, які б задовольняли їх потреби та зменшували час на очікування.

Метою цього проекту є автоматизація системи «Виклик таксі», яка представляє собою більш швидший та безпечний метод надання послуги

## 2. Завдання

Завданням цього проекту є автоматизація системи «Виклик таксі». Система працює у вигляді додатку на телефон, до якого мають доступ Користувач, Водій, та Адміністратор. Дані Користувача та Водіїв зберігаються у базі даних додатку, що полегшує взаємодію з системою та забезпечує безпеку Користувачам та Водіям у разі злочинів.

Для проектування системи було обрано каскадну модель життєвого циклу ПЗ, так як замовником виступає фізична особа. Також ця модель має як і переваги, так і вагомні недоліки

Переваги:

- Послідовне виконання етапів проекту у строго фіксованому порядку.
- Дозволяє оцінити якість продукту на кожному етапі.

Недоліки:

- Відсутність зворотніх зв'язків між етапами
  - Не відповідає реальним умовам розробки програмного продукту
- Відноситься до першої групи моделей.

## 3. Опис методів та засобів, які застосовуються при проектуванні.

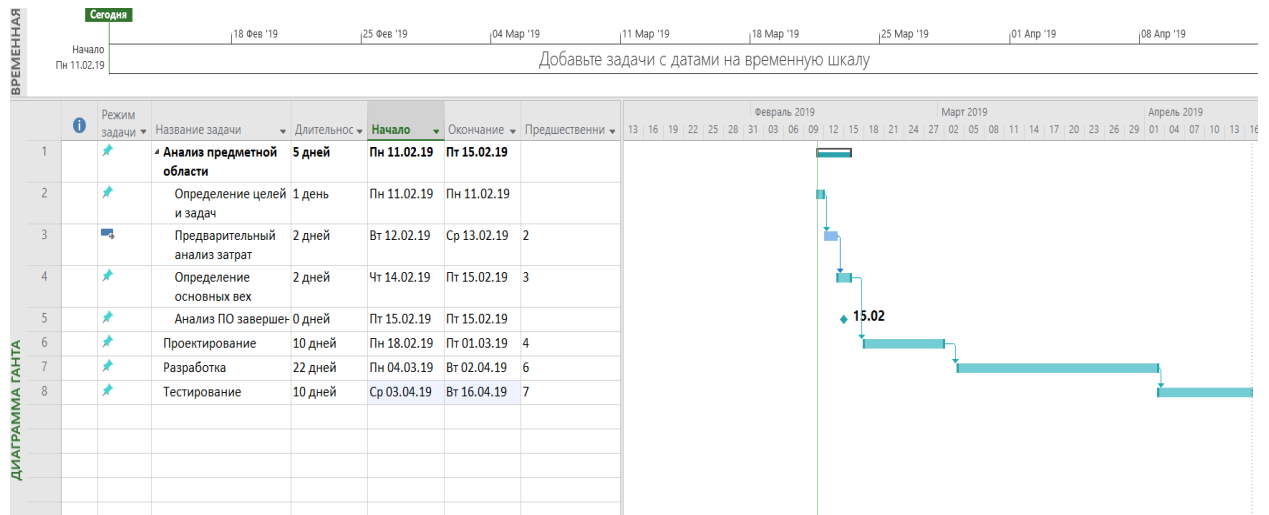
В ході розробки систему, були використані такі засоби:

- COCOMO calculator
- MicroSoft Project
- Unified Modeling Language

## 4. Специфікування вимог

### 4.1 Встановлення вимог

В процесі встановлення вимог було проведено календарне планування, та планування необхідних ресурсів у середовищі MS Project(рис. 1-2).



(рис. 1) Календарне планування

	Название ресурса	Тип	Единицы измерения	Краткое название	Группа	Макс. единиц	Стандартная ставка	Ставка сверхурочных	Затраты на исполн.	Начисление	Базовый календарь	Код	Зависит новый стол
1	Компьютер	Материальный	К				0,00 Р		0,00 Р	Пропорциональное	Стандартный		
2	Менеджер	Трудовой	М			100%	700,00 Р/ч	800,00 Р/ч	0,00 Р	Пропорциональное	Стандартный		
3	Бизнес аналитик	Трудовой	Б			100%	750,00 Р/ч	850,00 Р/ч	0,00 Р	Пропорциональное	Стандартный		
4	Backend разработчик	Трудовой	В			100%	400,00 Р/ч	500,00 Р/ч	0,00 Р	По окончании	Стандартный		
5	Frontend разработчик	Трудовой	Ф			100%	400,00 Р/ч	500,00 Р/ч	0,00 Р	По окончании	Стандартный		
6	Тестировщик	Трудовой	Т			100%	400,00 Р/ч	500,00 Р/ч	0,00 Р	Пропорциональное	Стандартный		
7	Дизайнер	Трудовой	Д			100%	350,00 Р/ч	450,00 Р/ч	0,00 Р	Пропорциональное	Стандартный		

(рис. 2) Планування ресурсів

В результат було створено перелік вимог до ПЗ.

## 1. Загальні відомості :

**1.1 Проект:** Автоматизована система «Виклик таксі».

**1.2 Замовник:** компанія Uber (www.uber.com).

**1.2.1 Реквізити:** 32008184801 в ГУ НБУ по м.Києву і Київській області(код банку 321024).

**1.3 Розробник:** Богатко корпорейшн(Bohatko corp.).

**1.3.1 Реквізити:** УК у Шевченківському районі(18050400), місто Київ.

**1.4 Планові терміни початку та закінчення роботи:** 11.02.2020-16.04.2020

**1.5 Відомості про джерела та порядок фінансування робіт:** роботи фінансуються замовником пропорційно вимогам розробника.

## 2. Призначення та цілі створення ПЗ:

**2.1 Вид діяльності:** громадська діяльність Визову таксі

**2.2 Об'єкти автоматизації:** алгоритм визову таксі

**2.3 Основні показники, що повинні бути досягнуті в результаті впровадження ПЗ:** як результат – автоматизована система «Виклик Таксі» у мобільному додатку.

## 3. Характеристика об'єктів автоматизації:

**3.1 Об'єкт автоматизації:** система пошуку водіїв та проведення поїздок.

**3.2 Експлуатація об'єкту:** користувачем є будь-яка людина.

**3.3 Вимоги до ПЗ:** Доступ до інформації відбувається через бази даних, які знаходяться у вільному доступі.

**3.4 Персонал:** Менеджер, бізнес-аналітик, Frontend-розробник, Backend-розробник, дизайнер, тестувальник.

**4.Структура ПЗ:** Складається з декількох основних етапів, деякі з яких розбиваються на підетапи.

**5.Основні компоненти та підсистеми:**

- 1) Аналіз предметної області.
  - а) Визначення цілей і задач.
  - б) Попередній аналіз витрат.
  - в) Визначення основних віх
- 2) Проектування
- 3) Розробка системи
- 4) Тестування

**6.Кваліфікація персоналу:** Середня

**7.Режим роботи:** 5 днів на тиждень по 8 годин на добу, включаючи святкові вихідні

**8.Вимоги до:**

- **Надійності:** вище середнього
- **Продуктивності:** середні
- **Ергономіки:** середні
- **Технічної естетики:** високі
- **Експлуатації:** середні
- **Технічного забезпечення:** середні
- **Збереження, захисту інформації від зовнішніх чинників:** середні
- **Патентної чистоти:** середні
- **Стандартизації:** середні
- **Уніфікації:** високі

**9.Склад та зміст робіт зі створенням системи:**

Отже, робота складалася з чотирьох основних етапів:

- 1)Аналіз предметної області; (3 дні)
- 2)Проектування додатку; (20 днів)
- 3)Розробка самої системи; (60 днів)
- 4)Тестування; (15 днів)

У результаті всі етапи завершилися успішно, як по термінам, так і результативно. Необхідно було розподілити роботу правильно між усіма співробітниками для того, аби досягти такого результату. Фінансування було надано Замовником даної автоматизованої системи.

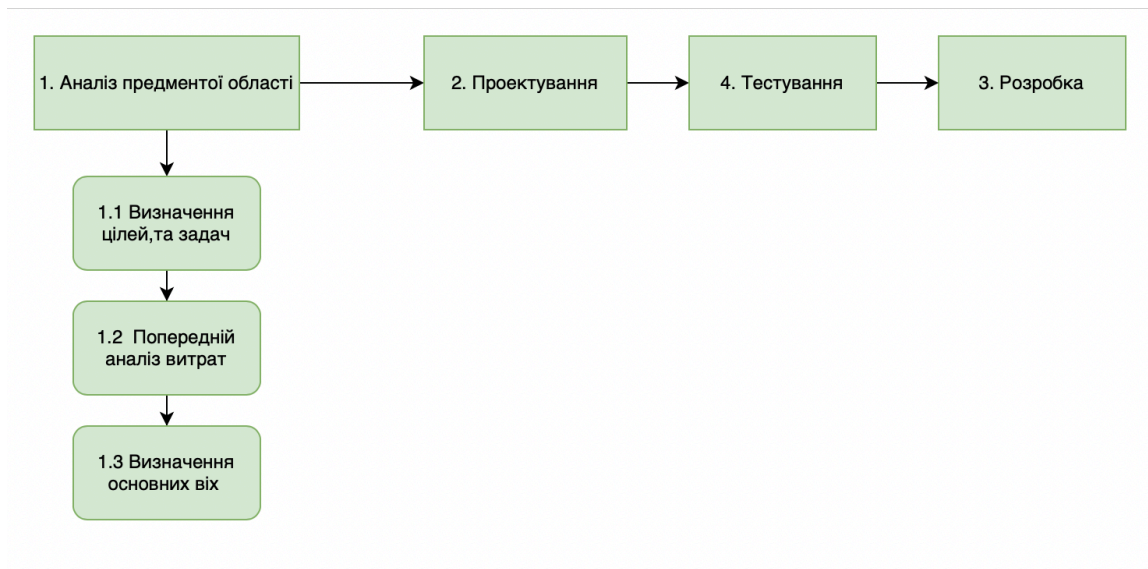
**10.Вимоги до документування:**

Свідоцтво про реєстрацію як суб'єкта підприємницької діяльності (ФОП) або (ТОВ).

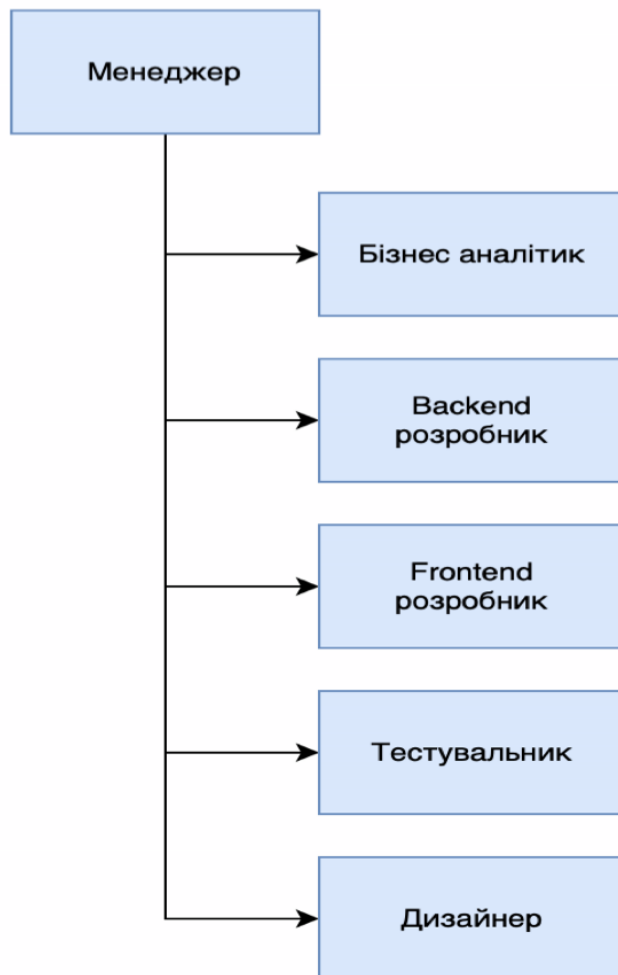
**10.1**

Для логічного детального розподілу усіх робіт на окремі операції, які плануються відповідно до обраної моделі життєвого циклу ,та внутрішній організації команди проекту було побудовано WBS(рис.3), OBS(рис. 4), та двонаправлена модель управління проектом(рис.5).

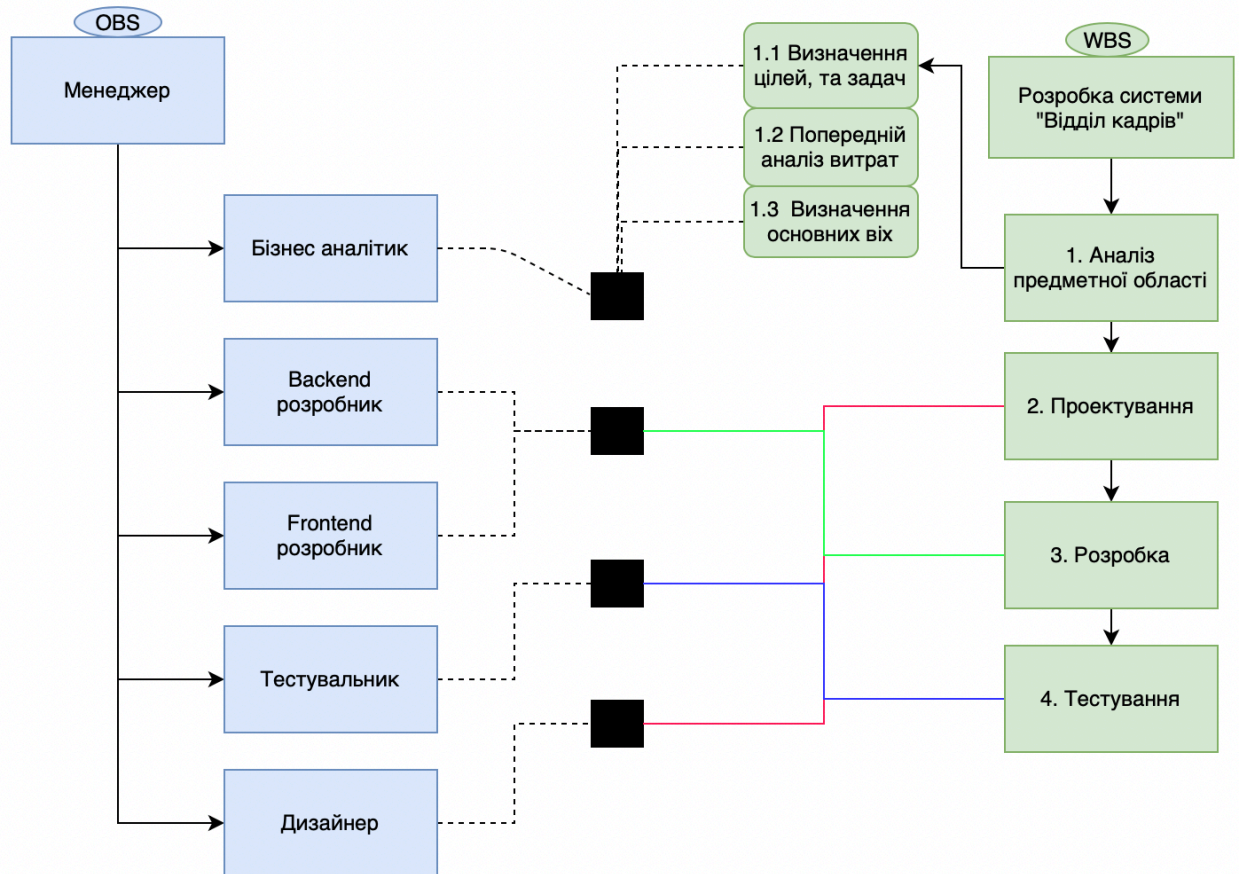
Також було побудовано Діаграму прецедентів(рис.6), яка стала основою для встановлення вимог до функціональності ПЗ.



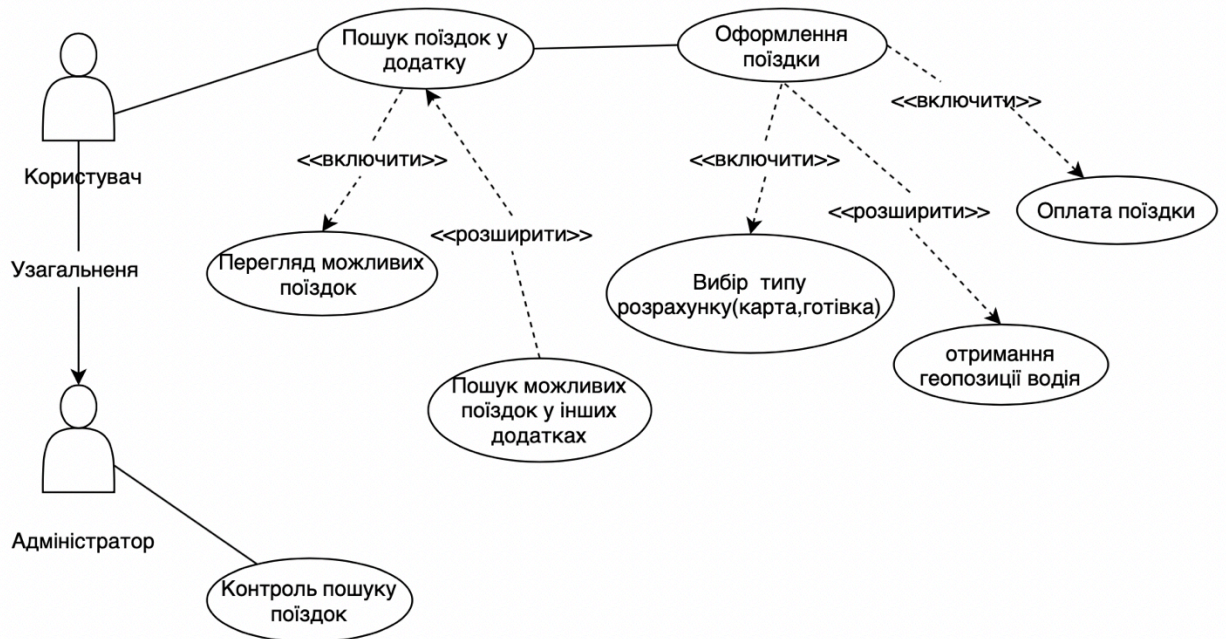
(рис.3) WBS модель



(рис.4) OBS модель



(рис. 5) Двонаправлена модель управління проектом



(рис.6) Діаграма прецедентів

Було складено детальний вербальний опис основних варіантів використання ПЗ:

1) Робота з додатком

**1.Короткий опис:** Користувач заходить у додаток та шукає поїздки

**2.Суб'єкт:** Користувач

**3.Передумова:** Користувач заходить у додаток

**4.Основний потік:** Користувач заходить додаток і вводить інформацію про поїздку – дату відправлення, місце відправлення місце призначення.

Якщо водіїв немає, виконується А1. Якщо додаток не завантажується, виконується А2.

**5.Альтернативні потоки:**

А1: Немає поряд водіїв. Користувач отримує повідомлення про те, що поряд немає водіїв.

А2: Не завантажується додаток. Користувач отримує повідомлення. Що потрібно почекати.

**6.Постумови:** Якщо поблизу є вільні водії, користувач оформлює поїздку.

2) Перегляд можливих поїздок.

**1.Короткий опис.**

Користувач здійснює перегляд всіх можливих поїздок для інформації яку вказав

**2. Суб'єкт – Користувач**

**3. Передумова – Пошук поїздок у додатку.**

**4. Основний потік.**

Користувач вибирає таксі, яке йому найкраще підходить із усіх, що відобразилися. Якщо вільних водіїв немає, виконується А1.

**5. Альтернативні потоки**

А1: Немає вільних водіїв. Користувач отримує повідомлення і пропозицію зачекати.

**6. Постумови**

Користувач оформлює поїздку.

3)Оформлення поїздки

**1. Короткий опис** Користувач оформлює поїздку.

**2. Суб'єкт – Користувач**

**3. Передумова – користувач обирає пункт призначення.**

**4. Основний потік.**

Користувач оформляє обрану поїздку – заповнює форму, оплачує, отримує електронний чек. Якщо неможливо оформити поїздку, виконується А1.

### **5. Альтернативні потоки**

A1: Неможливо оформити поїздку. Користувач отримує повідомлення про причину і пропозицію ще раз оформити поїздку.

### **6. Постумови**

Користувач отримує геопозицію водія

## **4) Вибір типу розрахунку.**

**1. Короткий опис.** Користувач вибирає вид таксі і починає оформлення: обирає тип розрахунку.

**2. Суб'єкт** – Користувач

**3. Передумова** – користувач обирає поїздку.

**4. Основний потік.**

Користувач заповнює форму, яка містить особисті дані для розрахунку(номер картки, або решта з готівки), та дані про поїздку. Після заповнення користувач відправляє форму і переходить до оплати. Якщо форма заповнена неправильно, виконується A1. Якщо форма заповнена не повністю, виконується A2.

### **5. Альтернативні потоки**

A1: Неправильно заповнена форма. Користувач отримує повідомлення та ще раз заповнює неправильно заповнене поле.

A2: Не повністю заповнена форма. Користувач отримує повідомлення і заповнює форму до кінця.

### **6. Постумови**

Користувач оплачує поїздку.

## **5) Оплата поїздки.**

**1. Короткий опис.** Користувач отримує данні о поїздки, для якої заповнив форму

**2. Суб'єкт** – Користувач

**3. Передумова** – користувач заповнює форму

**4. Основний потік.**

Користувач вводить дані карти та одноразовий пароль з смс. Якщо оплата успішна, він отримує електронний варіант квитка. Якщо оплата неуспішна, виконується A1.

### **5. Альтернативні потоки**

A1: Неуспішна оплата. Користувач отримує повідомлення та пропозицію спробувати оплатити ще раз.

### **6. Постумови**

Користувач отримує геопозицію водія.



## 6) Контроль пошуку поїздок

**1. Короткий опис.** Адміністратор контролює правильність заповнення

форм, вибору, оплати поїздки та її оформлення.

**2. Суб'єкт** – Адміністратор.

**3. Передумова** – немає.

**4. Основний потік.** Адміністратор контролює процеси, які відбуваються при замовленні поїздки. Якщо відбувається збій системи, відбувається А1.

**5. Альтернативні потоки**

А1: Збій системи. Відбувається резервне копіювання даних та перезавантаження системи.

**6. Постумови**

Немає.

## 7) Пошук можливих поїздок у інших додатках.

**1. Короткий опис.** Користувач отримує пропозицію замовити поїздку у іншому додатку.

**2. Суб'єкт** – Користувач

**3. Передумова** – користувач здійснює пошук водіїв.

**4. Основний потік.**

Користувач вводить дані, і якщо немає поблизу вільних водіїв, йому виводять пропозицію переглянути інші додатки. Якщо пропозицій немає, А1

**5. Альтернативні потоки**

А1: Немає пропозицій. Користувач отримує повідомлення.

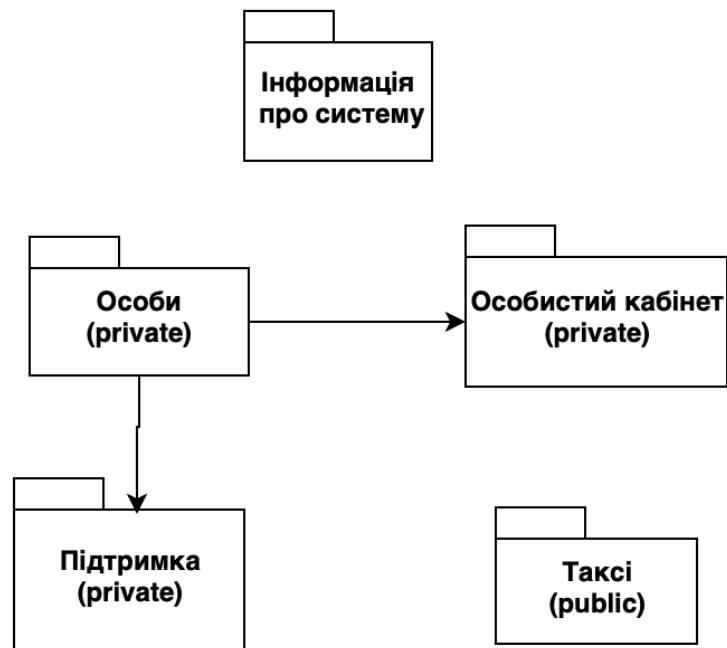
**6. Постумови**

Користувач продовжує пошук або залишає додаток.

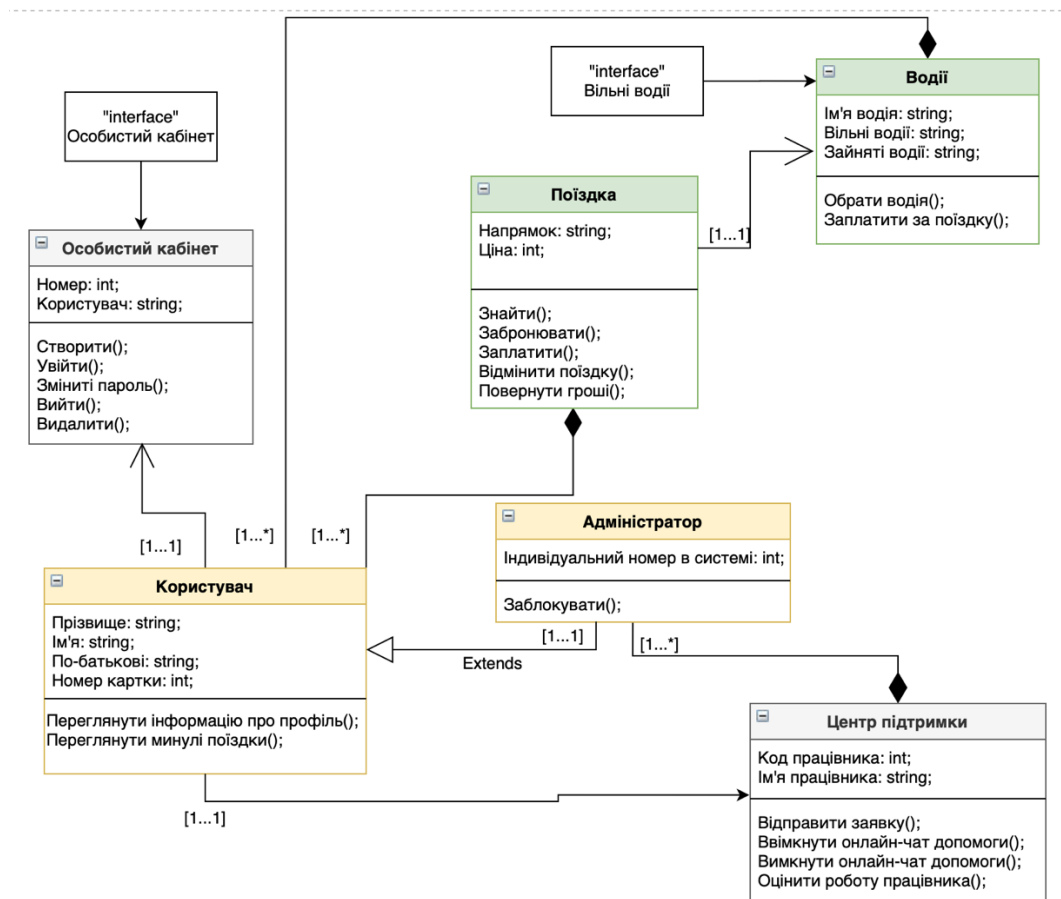
## 4.2 Специфікування вимог

В рамках специфікування вимог до ПЗ було побудовано діаграму класів(рис.8), що описує структуру системи, демонструє класи системи, їх атрибути, методи та залежності між класами

Також було побудовано діаграму пакетів(рис.7), яка ілюструє організацію елементів в групі за певною ознакою з метою спрощення структури і організацією роботи з моделлю системи.



(рис.7) Діаграма Пакетів



(рис.8) Діаграма Класів

Нижче наведена вербальна специфікація кожного пакета у формі: пакет → опис вмісту:

#### I. Клас Користувач

- Атрибути:
  - public Прізвище : string;
  - public Ім'я: string;
  - public По батькові: string;
  - private Номер карти: int;
- Операції:
  - Переглянути інформацію про профіль();
  - Переглянути минулі поїздки();

#### II. Клас Особистий кабінет

- Атрибути:
  - private Номер: int;
  - public Користувач: string;
- Операції:
  - Створити();
  - Увійти();
  - Змінити пароль();
  - Вийти();
  - Видалити();

#### III. Клас Адміністратор

- Атрибути:
  - private Індивідуальний номер в системі: int;
- Операції:

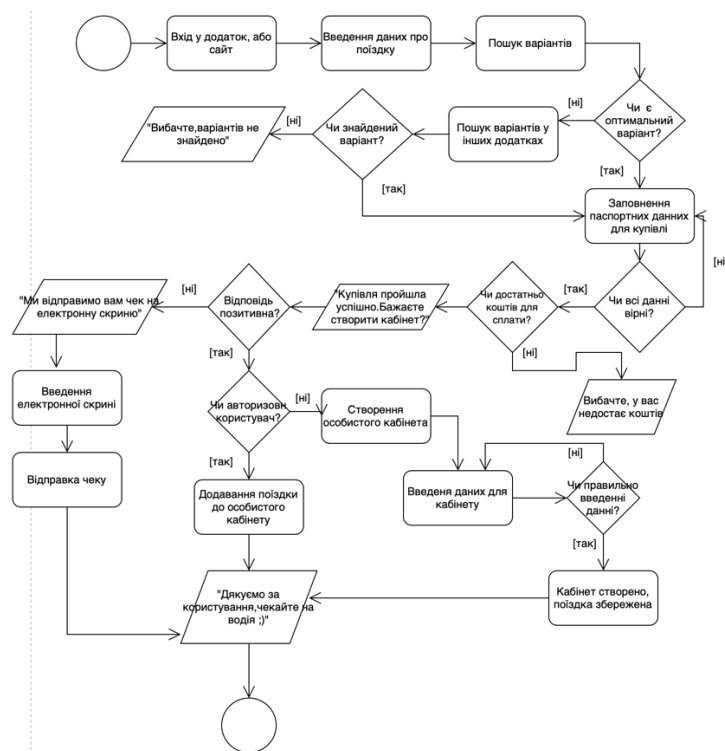
- Заблокувати();
- IV. Клас Водії
  - 1. Атрибути:
    - public Ім'я водія: string;
    - public Вільні водії: int;
    - public Зайняті водії: int;
  - 2. Операції:
    - Обрати водія();
    - Заплатити за поїздку();
- V. Клас Поїздка
  - 1. Атрибути:
    - public Напрямок: string;
    - public Ціна: int;
  - 2. Операції:
    - Знайти();
    - Забронювати();
    - Заплатити();
    - Відмінити поїздку();
  - Повернути гроші();
- VI. Клас Центр підтримки
  - 1. Атрибути:
    - private Код працівника: int;
    - public Ім'я працівника: string;
  - 2. Операції:
    - Відправити заявку();
    - Ввімкнути онлайн-чат допомоги();
    - Вимкнути онлайн чат допомоги();
    - Оцінити роботу працівника();

## 4.3 Моделювання поведінки системи

Для моделювання поведінки системи було побудовано діаграми діяльності(рис.9), взаємодії(рис.9), стану(рис.10).

Діаграма діяльності показує розкладання деякої діяльності на її складові частини. Під діяльністю розуміється специфікація виконуваної поведінки у вигляді координованого послідовного і паралельного виконання підпорядкованих елементів.

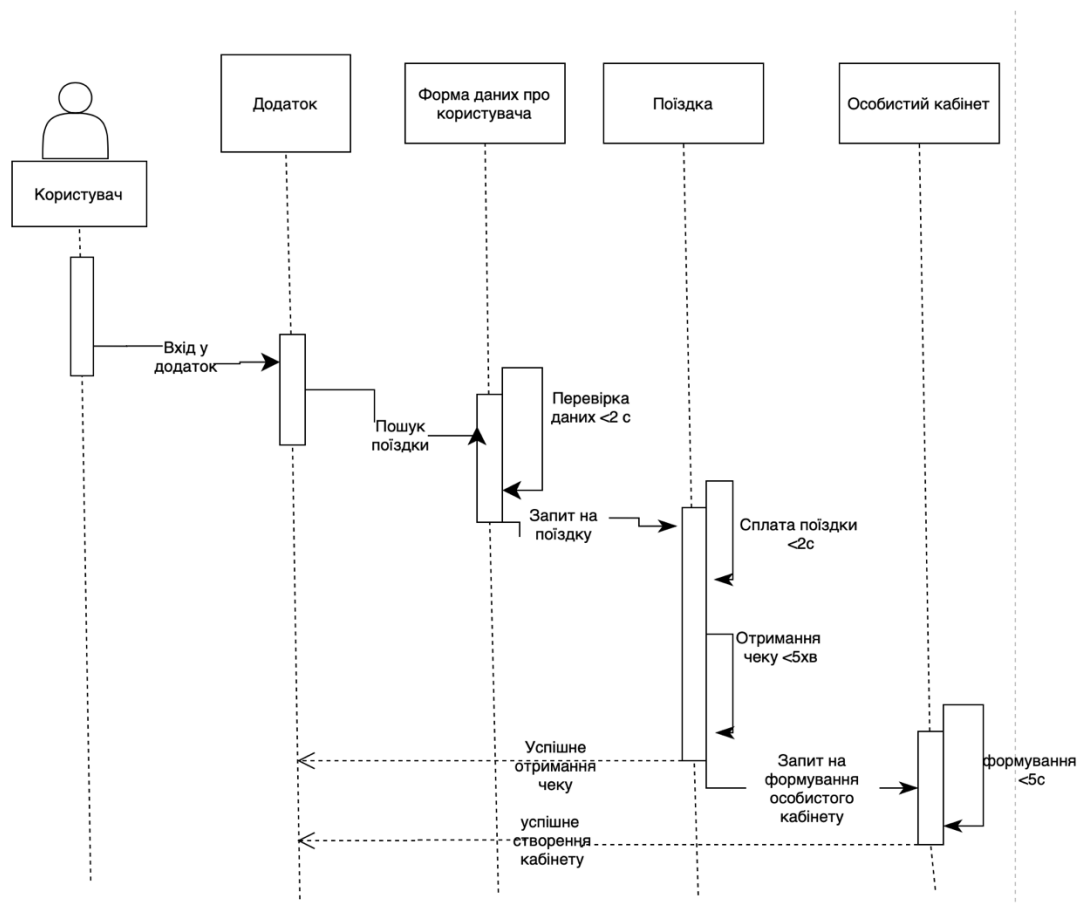
Дана діаграма діяльності ілюструє процес роботи з додатком



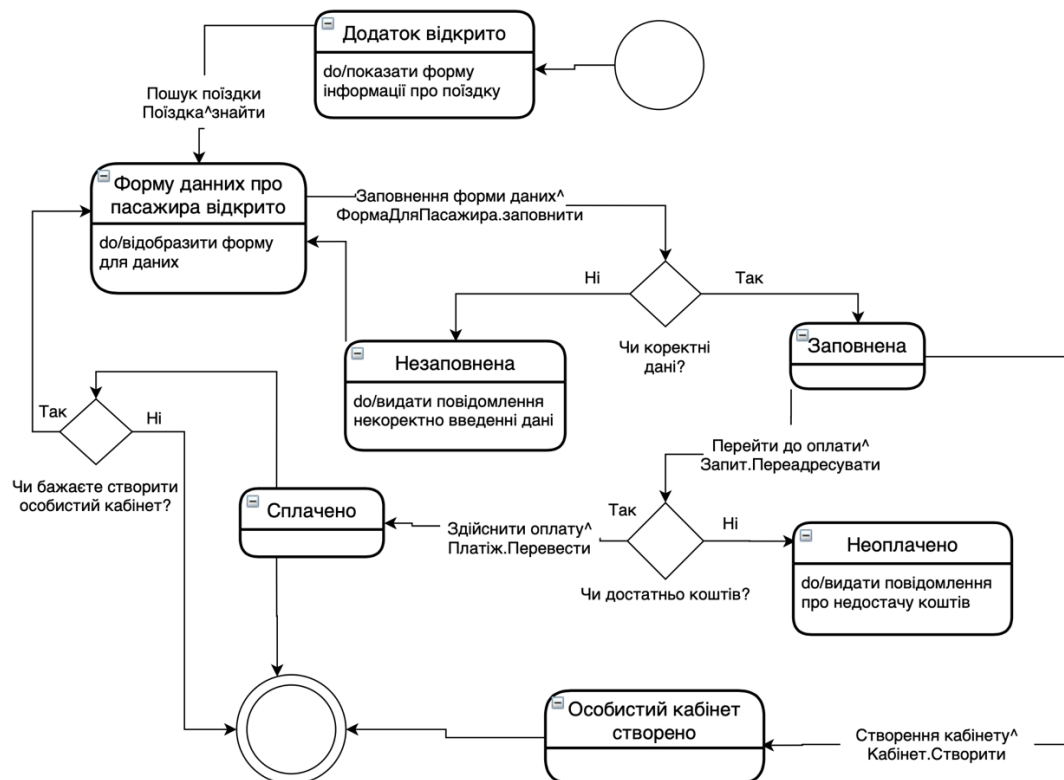
(рис.9) Діаграма Діяльності

Формулювання прецеденту	Стан виду діяльності
1. Вхід у додаток чи на сайт виклику таксі	Користувач здійснює вхід, відкриваючи його
2. Введення бажаних даних	Користувач заповнює форму, яка містить дані про бажану поїздку
3. Пошук варіантів з існуючої бази даних	Система виконує пошук по раніше заданим параметрам
4. Пошук варіантів у інших додатках	Система виконує розширений пошук по базам даних інших додатків
5. Заповнення паспортних даних для купівлі	Користувач заповнює форму, яка містить паспортні дані, необхідні для сплати поїздки
6. Введення електронної пошти	Користувач вводить електронну пошту, куди буде відіслано квиток
7. Відправка чеку	Система відправляє чек на пошту, яку ввів користувач
8. Введення бажаних даних	Користувач заповнює форму, яка містить дані про бажану поїздку
9. Додавання поїздки до особистого кабінету	Система автоматично додає поїздку до особистого кабінету
10. Створення особистого кабінету	Система надає запит на створення особистого кабінету
11. Введення даних для кабінету	Користувач заповнює форму, яка містить дані, необхідні для створення кабінету
12. Кабінет створено, поїздка збережена	Система створює особистий кабінет, відправляє чек в особистий кабінет

Табл.1. Специфікація діаграми діяльності



(рис.9)Діаграма послідовності



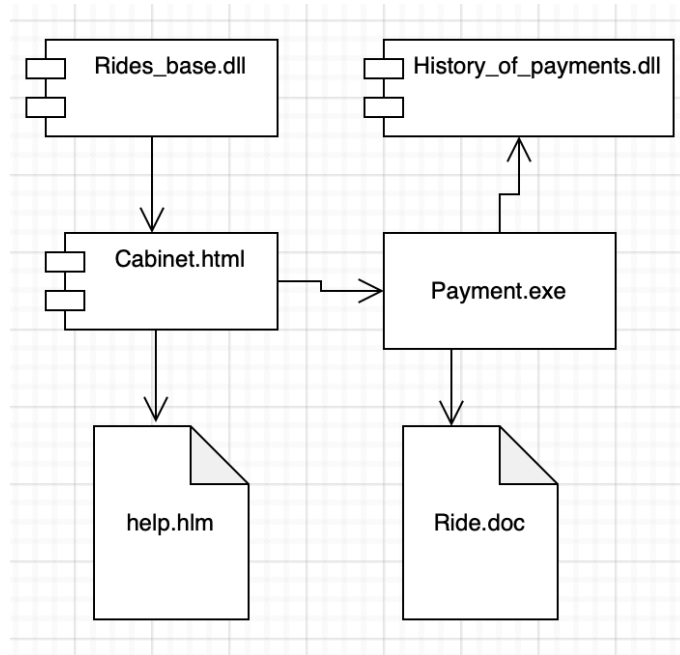
(рис.10)Діаграма станів

## 4.4 Проектування архітектури ПЗ

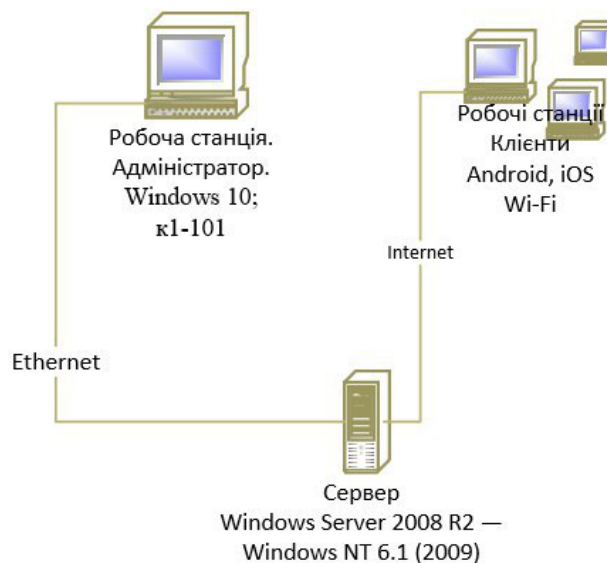
На етапі проектування архітектури ПЗ, було використано такі види діаграм як діаграми компонентів(рис.11) та діаграми розгортання(рис.12)

Діаграми компонентів показують розбиття програмної системи на структурні компоненти і зв'язки між компонентами. В якості фізичних комопнентів можуть виступати файли, бібліотеки, модулі, виконувані файли, пакети тощо.

Діаграми розгортання служать для моделювання працюючих вузлів і артефактів, розгорнутих в них.



(рис. 11)Діаграма компонентів



(рис. 12) Діаграма розгортання

## 4.5 Детальне проектування

В даному розділі відбувається одна з фінальних частин проектування ПЗ, а саме проектування інтерфейсу та баз даних.

Відповідно до вимог ергономіки та технічної естетики ,інтерфейс додатку повинен бути зрозумілим та простим, передбачається що додаток можна буде запускати навіть на застарілих телефонах.

Робота інтерфейсу:

Інтерфейс є інформаційним сповіщенням для користувача щодо вибору поїздки, можливих помилок та варіантів їх виправлення.

В першу чергу, користувач вводить дані про поїздку, яку бажав би замовити. Якщо всі дані вірні, з'являються варіанти і користувач може обрати будь-який автомобіль з бажаних. (рис.13-14)

У разі затвердження поїздки, користувач вводить дані щодо оплати: якщо дані вірні, вони не потребують змін. Якщо ж ні, користувач бачить сповіщення щодо можливості їх змінити у окремому вікні.(рис 15-16)

У дочірньому вікні користувач має змогу вибрати спосіб оплати:дефолтний, ввести новий номер карти, або заплатити готівкою. Після затвердження обраного способу оплати, користувач повертається назад до батьківського вікна.

У дочірньому вікні користувач має змогу обрати місце посадки: по геолокації, або обрати інше місце.(рис 13)

На батьківському вікні вказана інформація щодо суми поїздки.(рис. 14)

Якщо всі деталі замовлення задовільняють користувача, він може оформити поїздку натиснувши на кнопку “Pay”, що переведе його до дочірнього вікна, що сповістить про успішну сплату. Якщо ж умови його не задовільняють, завжди є можливість скасувати оформлення, натиснувши на кнопку “Cancel”.

Нижче наведено прототип інтерфейсу додатка:



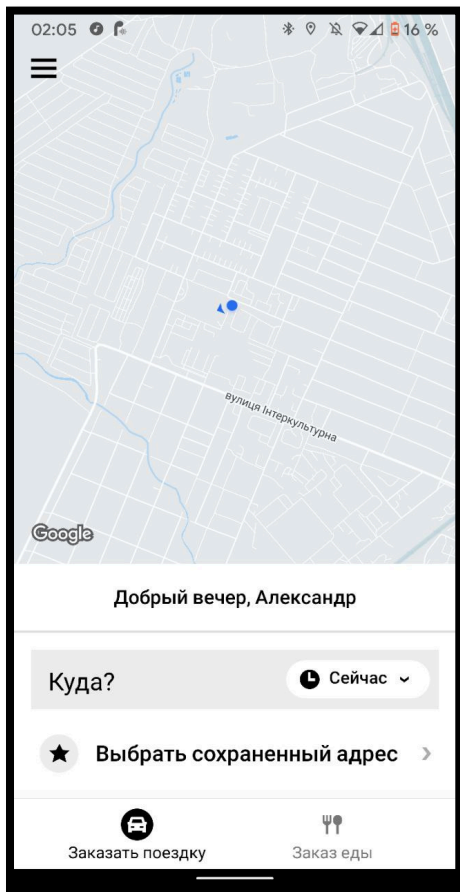


Рис.13

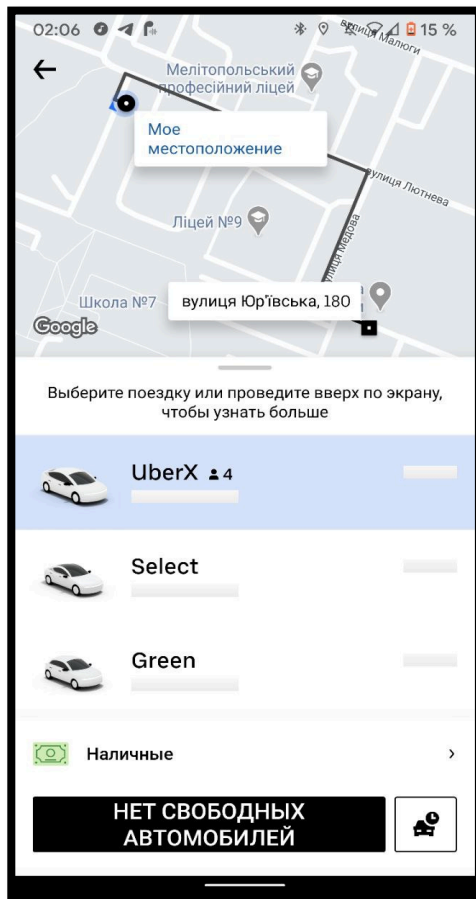


Рис.14

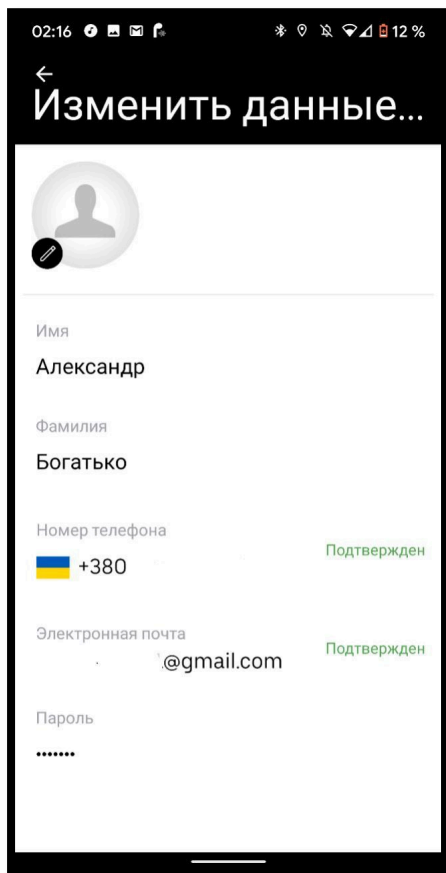


Рис.15

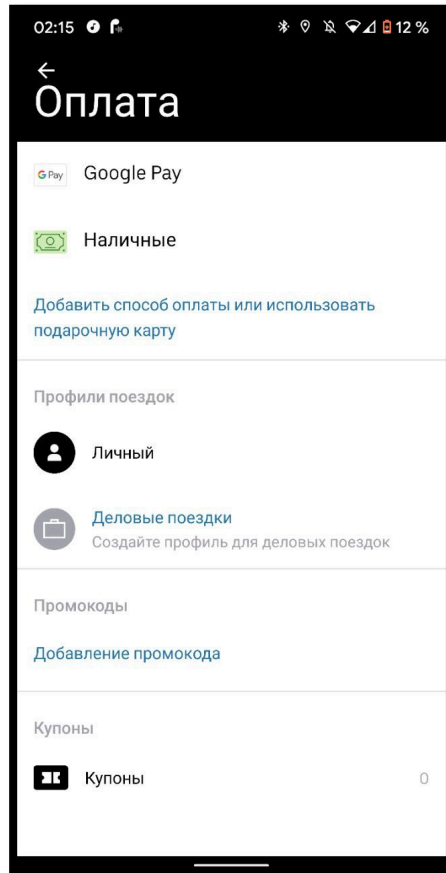


Рис.16

Також було спроектовано бази даних Clients.db(рис.17), та Payments.db(рис.18) за допомогою SQL

Реляція бази даних clients:

$R(S) = \{client - id, name, surname, date - of - birth, city, country, adress, card - number\}, \{\gamma\}$ , де  $\gamma$  – це множина всіх кортежів бази даних.

Реляція бази даних payments:

$R(S) = \{payment - id, name, surname, transaction, location, balance\}, \{\beta\}$ , де

$\beta$  – це множина всіх кортежів бази даних.

В базі даних clients ключем є значення *client - id*, в базі payments це *payment - id*.

```
DROP DATABASE IF EXISTS clients;
CREATE DATABASE clients;
USE clients;

CREATE TABLE client (
    client_id INT PRIMARY KEY,
    name VARCHAR(50),
    surname VARCHAR(50),
    date_of_birth DATE,
    city VARCHAR(50),
    country VARCHAR(50),
    adress VARCHAR(50),
    card_number INT UNIQUE
);
```

(рис.17) База даних Clients

```
DROP DATABASE IF EXISTS payments;
CREATE DATABASE payments;
USE payments;

CREATE TABLE payment (
    payment_id INT PRIMARY KEY,
    name VARCHAR(50),
    surname VARCHAR(50),
    transaction INT UNIQUE,
    location VARCHAR(50),
    balance INT
);
```

(рис.18) База даних Payments

## 5. Генерація програмного коду

Було згенеровано вхідний код програмного забезпечення на основі діаграми класів(рис. 8):

```
public class User{
    public string name;
    public string midname;
    public string surname;
```

```

        public int card_id;

        public void ShowProfile(){}
        public void LastRides(){}
    }
    public class Cabinet{
        private int Num;
        public string User;

        public void CreateProf(){}
        public void Enter(){}
        public void ChangePass(){}
        public void Exit(){}
        public void Delete (){}
    }
    public class Admin extends User{
        private int id;

        public void Ban(){}
    }
    public class Drivers{
        public string Destination;
        public int Cost;

        public void Find(){}
        public void Book(){}
        public void Transaction(){}
        public void CancelRide(){}
        public void ReturnMoney(){}
    }
    public class SupportCenter{
        private int id;
        public string Name;

        public void SendRequest(){}
        public void TurnOnChat(){}
        public void TurnOffChat(){}
        public void RateWorker(){}
    }
}

```

## 6. Висновок

Результатом виконаної роботи є автоматизована система «Виклик таксі», у вигляді кросплатформеного мобільного додатку, яка виконує функцію спрощення роботи водіїв, та забезпечує комфортні та безпечні поїздки для користувачів. В системі налагоджено документування всіх дій користувачів. В розпорядженні системи база даних, яка містить інформацію про користувачів та історію поїздок.

## 7. Список літератури

- Основи програмної інженерії. Лабораторний практикум / уклад. С. В. Поперешняк, І. А. Юрчук, 2019. – 52 с.
- <https://m.habr.com/ru/post/111674/>
- <https://ru.m.wikipedia.org/wiki/UML>