*Архитектура приложения.*

*Таблица классов*

|  |  |  |
| --- | --- | --- |
| *Номер* | *Название* | *Пояснение* |
| 1. | *User* | Это класс пользователя, который запоминает необходимую информацию о нем и своевременно отдает ее. |
| 2. | *Request* | Это класс регистрации водителя, в нем реализованы запросы неоюходимые для проведения этой операции. |
| 3. | *ClientRequest* | Это класс функций клиента:   * Создание нового заказа * Информация о заказе * Закрытие заказа |
| 4. | *DriverRequest* | Это класс функций водителя:   * Закрытие зааза * Информация о заказе |

*Таблица методов 1-ого класса {class User}*

|  |  |
| --- | --- |
| *Функция* | *Пояснение* |
| def check\_data\_base | ***Проверяет наличие пользователя в базе данных(водитель/клиент)*** |
| def add\_to\_data\_base | ***Производит добавление пользователя в базу*** |

*Таблица методов 2-ого класса {class Request}*

|  |  |
| --- | --- |
| *Функция* | *Пояснение* |
| def find\_address | ***Находит адрес по запросу(передается адрес)*** |
| def add\_driver\_info | ***Добавляет информацию о водителе в базу*** |
| def get\_pos | ***Находит координаты по запросу(передается адрес)*** |

*Таблица методов 3-ого класса {class ClientRequest}*

|  |  |
| --- | --- |
| *Функция* | *Пояснение* |
| def first\_status | ***Меотод реализует первую функцию клиента*** |
| def second\_status | ***Метод реализует вторую функцию клиента*** |
| def third\_status | ***Метод реализует третью функцию клиента*** |
| def get\_receipt\_info | ***Метод реализует получение информации о заказе (входит во вторую функцию).*** |
| def find\_driver | ***Метод реализуюет подбор водителя по указанным требованиям (водит в третью фйнкцию).*** |
| def take\_status\_driver | ***Метод находит смену в кторорую работает водитель.*** |
| def calculate\_cost | ***Метод реализует подщет стоимости поездки.*** |
| def check\_address | ***Метод реализует проверку адреса.*** |
| def check\_class\_car | ***Метод реализует проверку класса автомобиля*** |
| def get\_coordinates | ***Метод реализует получение координат по адресу*** |
| def add\_new\_receipt | ***Метод реализует добавление нового заказа в базу данных*** |
| def change\_status\_in\_receipt | ***Метод реализует изменение статуса у выбранного водителя*** |
| def receipt\_not\_closed | ***Метод реализует проверку заказа на закрытость*** |
| def find\_address | ***Метод реализует поиск адреса по координатам*** |
| def calculate\_distance | ***Метод реализует подсчет расстояния между двумя точками***  ***на поверхности земли*** |
| def check\_date | ***Метод реализует проверку даты на правильность*** |

*Таблица методов 4-ого класса {class DriverRequest}*

|  |  |
| --- | --- |
| *Функция* | *Пояснение* |
| def first\_status | ***Меотод реализует первую функцию клиента*** |
| def second\_status | ***Метод реализует вторую функцию клиента*** |
| def find\_address | ***Метод реализует поиск адреса по его координатам*** |
| def receipt\_not\_closed | ***Метод реализует проверку заказа на закрытость*** |
| def get\_info\_about\_receipt | ***Метод реализует получение информации о заказе*** |