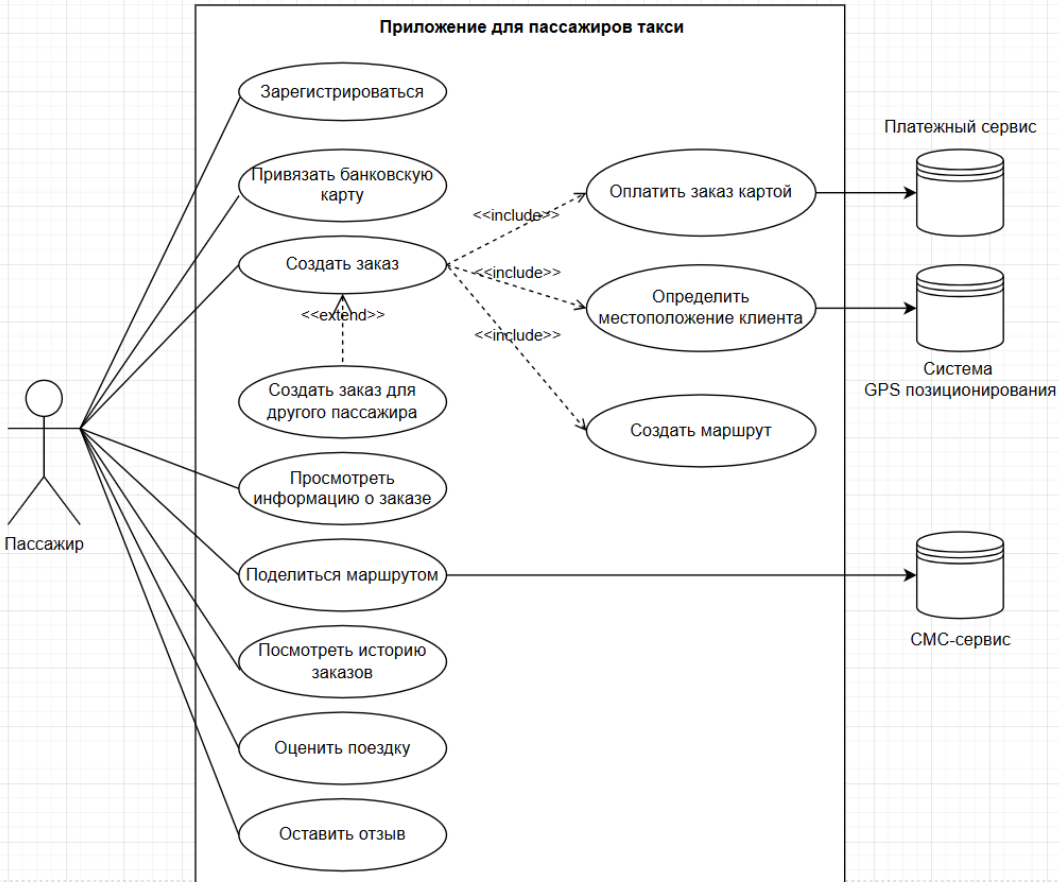
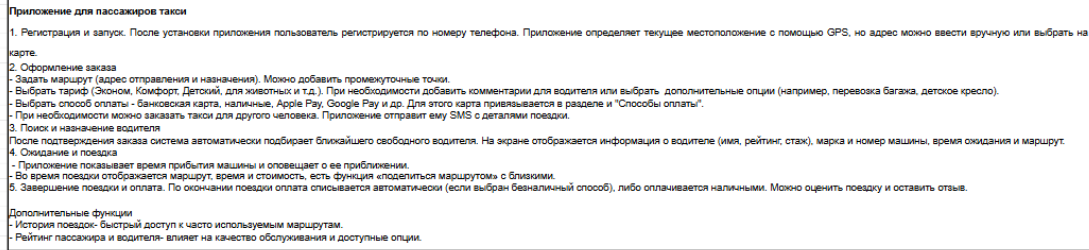
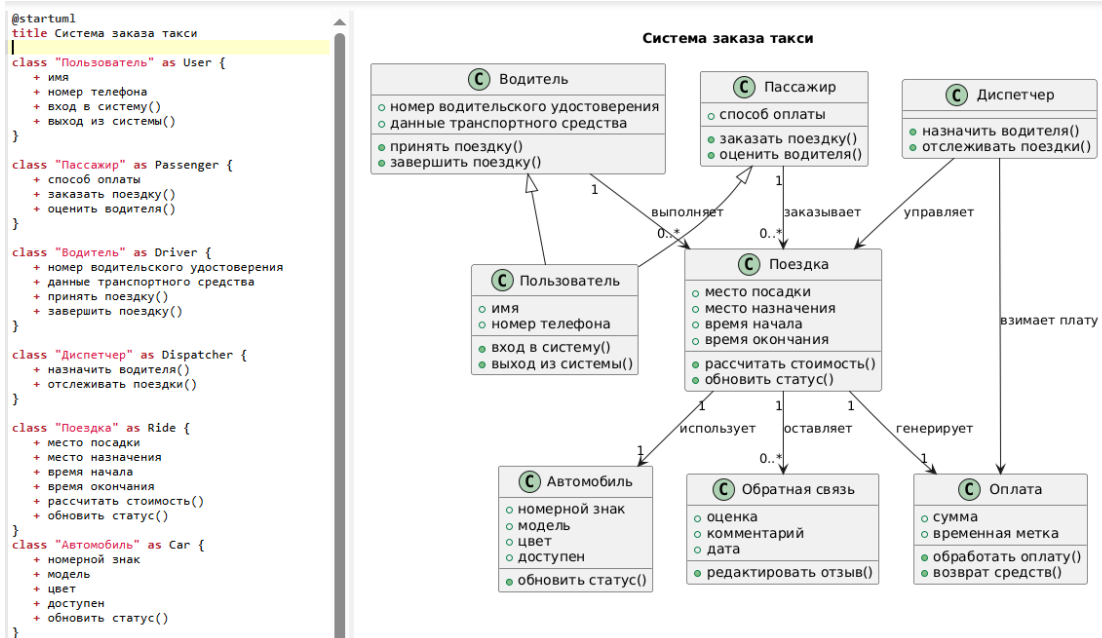


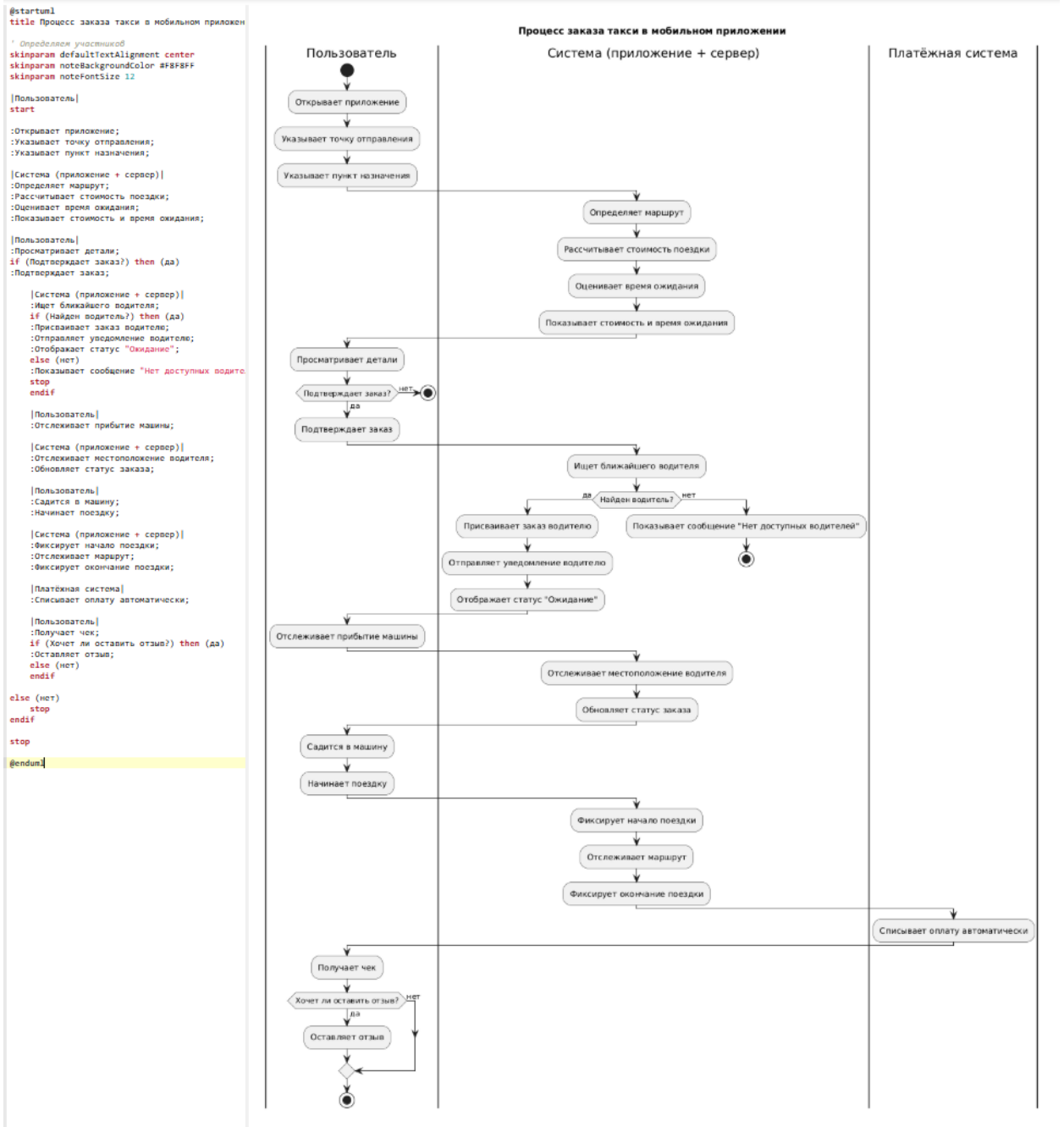
Use Case Diagram



Class Diagram



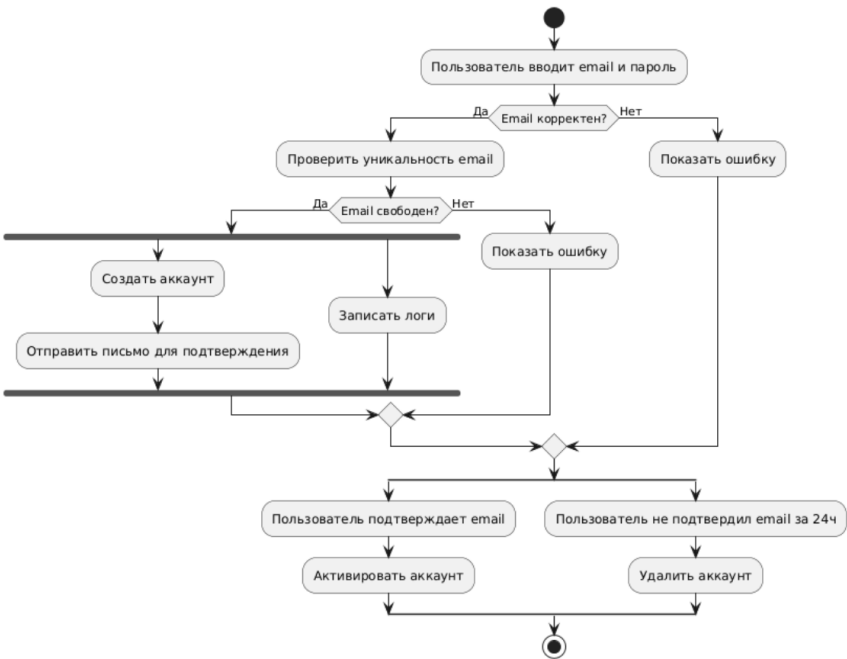
Activity Diagram



Регистрация пользователя с верификацией

```
@startuml
start
:Пользователь вводит email и пароль;
if (Email корректен?) then (Да)
:Проверить уникальность email;
if (Email свободен?) then (Да)
fork
:Создать аккаунт;
:Отправить письмо для подтверждения;
fork again
:Записать логи;
end fork
else (Нет)
:Показать ошибку;
endif
else (Нет)
:Показать ошибку;
endif

split
:Пользователь подтверждает email;
:Активировать аккаунт;
split again
:Пользователь не подтвердил email за 24ч;
:Удалить аккаунт;
end split
stop
@enduml
```



Процесс оплаты заказа с альтернативами

```
@startuml
actor User
participant "Клиентская часть" as Client
participant "Сервер" as Server
participant "Платёжный шлюз" as Gateway
participant "База данных" as DB

User -> Client: Нажимает "Оплатить"
Client -> Server: Синхронный запрос /create_order (POST)
Server -> DB: Сохранить заказ
DB --> Server: OK
Server --> Client: ID заказа

alt [Успешная оплата]
Client -> Server: **Асинхронный** запрос /proc
Server -> Gateway: Проверить платёжные данные
Gateway -> Server: Успех
Server -> DB: Обновить статус заказа
DB --> Server: OK
Server --> Client: "Оплата прошла"
else [Ошибка оплаты]
Client -> Server: **Асинхронный** запрос /proc
Server -> Gateway: Проверить платёжные данные
Gateway -> Server: Ошибка
Server --> Client: "Платёж отклонён"
end

opt [Повторная попытка]
Client -> Server: /retry_payment
Server -> Gateway: Повторная проверка
Gateway --> Server: Успех
end

@enduml
```

