

РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

ОТЧЕТ ПО ПРОЕКТУ

На тему:

Сервис для аренды самокатов «Юрент»

дисциплина: Программная инженерия

Студент: Парфенова Елизавета Евгеньевна

Группа: НФИбд-02-21

Студ. билет:1032216437

МОСКВА

2023 г.

СОДЕРЖАНИЕ

СОДЕРЖАНИЕ	2
1. ОБЩИЕ ПОЛОЖЕНИЯ	3
1.1. Наименование и условное обозначение работы	3
1.2. Наименование предприятий Заказчика и Исполнителя.....	3
1.3. Сроки выполнения.....	3
1.4. Особые условия	3
2. НАЗНАЧЕНИЕ РАЗРАБОТКИ.....	4
3. ТРЕБОВАНИЯ К ПРОГРАММЕ ИЛИ ПРОГРАММНОМУ ИЗДЕЛИЮ.....	5-15
3.1. Требования к функциональным характеристикам	5-15
3.1.1. Общие требования к функциям ПО	5-13
3.1.2. Требования к функциям ПО	13-14
3.2. Требования к надежности	14
3.3. Требования к составу и параметрам технических средств.....	14-15
3.4. Требования к информационной и программной совместимости	15
4. ТРЕБОВАНИЯ К ПРОГРАММНОЙ ДОКУМЕНТАЦИИ.....	16

1. ОБЩИЕ ПОЛОЖЕНИЯ

1.1. Наименование и условное обозначение работы

1.1.1. Наименование работы – "Разработка сервиса аренды самокатов «Юрент»."

1.2. Наименования предприятий Заказчика и Исполнителя

1.2.1. Заказчиком на проведение работы является Транспортная корпорация

1.2.2. Исполнителем работы является Студия разработки мобильных приложений, г. Москва

1.3. Сроки выполнения

Начало работ: сентябрь 2023

Окончание работ: октябрь 2023

1.4. Особые условия

Настоящее Техническое задание (ТЗ) может уточняться и дополняться в процессе выполнения работ. Согласование и утверждение дополнений к настоящему Техническому заданию проводятся в порядке, установленном для ТЗ.

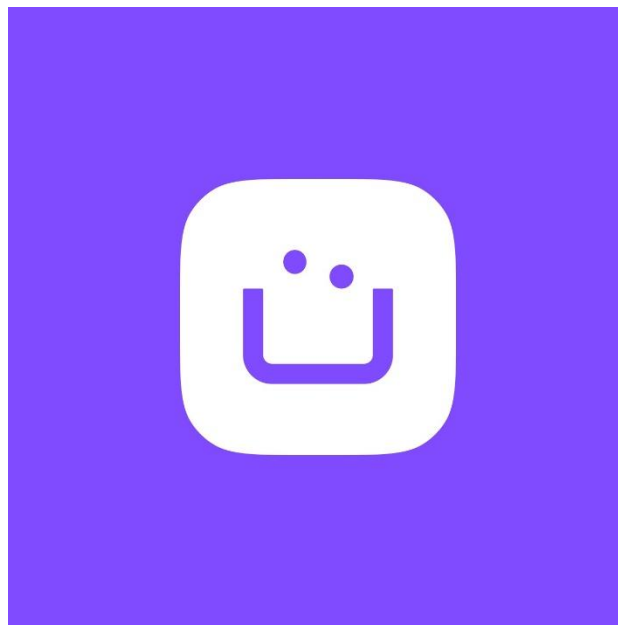
2. НАЗНАЧЕНИЕ РАЗРАБОТКИ

Целью данной разработки является создание сервиса для аренды самокатов «Юрент». Создание удобного и эффективного сервиса аренды самокатов поможет улучшить мобильность горожан и туристов, снизит количество автомобильных пробок и их длительность, а также улучшит экологию за счет уменьшения вредных выбросов в атмосферу.

Пользователями разрабатываемого сервиса могут стать жители и посетители разных городов, не имеющих собственного транспортного средства либо находящихся без него; курьеры в различных компаниях доставки.

Система предназначена для смартфонов ввиду удобства использования

Логотип разрабатываемого ПО:



3. ТРЕБОВАНИЯ К ПРОГРАММЕ ИЛИ ПРОГРАММНОМУ ИЗДЕЛИЮ

3.1. Требования к функциональным характеристикам

3.1.1. Общие требования к функциям ПО

3.1.1.1. Состав ПО

При реализации системы должны быть разработаны следующие модули:

Модуль поиска самоката

Модуль позволяет находить самокат в требуемой области с возможным использованием текущего местоположения

1. Определение местоположения

- Запросить разрешение на использование местоположения пользователя, которое будет включать в себя варианты: «только при использовании приложения», «всегда», «никогда».
- При выборе «никогда» уведомить пользователя о невозможности корректного поиска самокатов и снова запросить разрешение на использование местоположения.
- В остальных случаях определить местоположение пользователя

2. Поиск самоката

- Используя местоположение пользователя, найти самокаты поблизости
- Найденные самокаты указать в интерфейсе приложения, который представляет из себя карту местности
- Дать пользователю возможность просмотреть доступные самокаты в других районах, далеких от текущего местоположения

Модуль управления самокатами

Модуль позволяет пользователю арендовать или забронировать самокат, а администратору добавить или удалить самокат.

1. Аренда самоката

- Обеспечить возможность для пользователя выбрать тип аренды одним и способов, позволяющих интерфейсом
- Доступные типы аренды: «поминутная», «на 1 час», «пока не сядет»
- После выбора типа аренды сделать для пользователя опцию выбора одну из привязанных банковских карт, с которой произведется оплата, а также предложить доп.опцию: оплатить страховку.

- Перед арендой указать в интерфейсе тариф аренды и номер выбранного самоката.
- Если выбран поминутный тип аренды, поставить счетчик и автоматизировано подсчитывать сумму аренды по указанному тарифу. Показывать текущую сумму аренды в окне арендованного самоката
- Также показывать в окне арендованного самоката номер самоката и текущий заряд батареи
- Дать пользователю возможность арендовать несколько самокатов, оформив соответствующую кнопку в соответствии с дизайном
- Завершить аренду соответствующей кнопкой и предоставить пользователю полную информацию о поездке.

2. Бронирование самоката

- Дать возможность пользователю забронировать самокат только при наличии подписки.
- Резервировать самокат не более, чем на 10 минут
- Сделать лимит бронирования на неделю – не более 10 самокатов
- Дать пользователю возможность несколько раз бронировать один и тот же самокат
- Оформлять бронирование по нажатию на соответствующую кнопку и в окне забронированного самоката показывать оставшееся время бронирования в режиме реального времени

3. Администрирование самокатов

- Дать администратору возможность добавить новый самокат в общую базу данных. При этом начать отображать его на карте после его доставки на парковку
- Дать администратору возможность удалить самокат из общей базы данных. При этом больше не отображать самокат пользователю.

Модуль управления подпиской

Модуль позволяет пользователю приобрести подписку и пользоваться ее привилегиями

- Дать возможность пользователю приобрести подписку, которая будет обеспечивать ему «бесплатный старт» - то есть стоимость аренды будет состоять только лишь из суммы тарифа помноженного на минуты аренды, а

не составляться из этого и фиксированной цены старта, которая взимается за начало аренды каждого самоката

- Подписка кроме главного преимущества дает пользователю:
 - Заморозку подписку в независимости от срока действия подписки с 1 октября – то есть начала осенне-зимнего сезона, когда самокаты не могут являться средством передвижения – до открытия сезона весной
 - Скидку на страховку при каждой поездке
- Осуществлять деактивацию подписки при удалении аккаунта.
- Добавить три периода подписок: на месяц, на три месяца и на год.

Модуль управления платежами

Модуль, который позволяет осуществлять оплату аренды и подписки, а также привязать банковскую карту для этого. Данный модуль должен быть интегрирован с внешней платежной системой

1. Привязка карты

- При запросе на оплату от пользователя проверить, привязана ли банковская карта к приложению.
- Принять данные банковской карты пользователя
- Послать запрос во внешние системы на проверку действительности банковской карты по введенным данным

2. Оплата аренды

- Посылать запрос на резервацию фиксированной суммы при старте аренды и осуществлять возврат данной суммы по ее окончании.
- При сбое резервации по любой причине запретить пользователю арендовать самокат вплоть до успешной резервации.
- Посылать запрос на осуществление оплаты аренды в соответствии с высчитанной суммой

3. Оплата подписки

- Послать запрос на оплату фиксированной суммы, которая может меняться в соответствии с периодом подписки
- При сбое оплаты не активировать подписку пользователю

Модуль авторизации и регистрации

Модуль позволяет пользователю пройти регистрацию и аутентификацию

1. Регистрация

- При открытии приложения попросить пользователя ввести номер телефона, отобразив интерфейс в соответствии с дизайном
- Попросить пользователя ввести код подтверждения, высланный на введенный номер телефона. Позволить пользователю ввести его в соответствующее поле
- Код подтверждения должен совпадать с отправленным. Если нет – попросить пользователя ввести код повторно.
- Дать возможность пользователю запросить новый код подтверждения. При этом подтверждение осуществлять по последнему отправленному
- Проверить существование пользователя с данным номером телефона в базе данных
- Если пользователя не существует, то отобразить интерфейс, позволяющий ввести некоторые личные данные: Имя, Электронную почту
- Ознакомить пользователя с юридическим договором о присоединении. На интерфейсе отобразить слот для галочки и надпись рядом «Я принимаю условия соглашения».
- Не пропускать пользователя в основной интерфейс приложения, пока он не установит галочку в соответствующий слот
- Предложить пользователю купить подписку и переправить его на соответствующую страницу, если он согласился это сделать

2. Авторизация

- При открытии приложения попросить пользователя ввести номер телефона, отобразив интерфейс в соответствии с дизайном
- Попросить пользователя ввести код подтверждения, высланный на введенный номер телефона. Позволить пользователю ввести его в соответствующее поле
- Код подтверждения должен совпадать с отправленным. Если нет – попросить пользователя ввести код повторно.
- Дать возможность пользователю запросить новый код подтверждения. При этом подтверждение осуществлять по последнему отправленному
- Если данный номер телефона обнаружен в базе данных пользователей, то пропустить пользователя в приложение и отобразить главный интерфейс

Модуль управления профилем

Модуль позволяет пользователю редактировать или удалить профиль и обеспечивает администратору управление профилем пользователя.

1. Редактирование профиля

- В интерфейсе обеспечить возможность пользователю перейти в свой личный профиль, где будет представлена основная информация.
- На странице профиля обеспечить возможность изменять поля, заполненные личной информацией, за исключением номера телефона.

2. Удаление профиля

- На странице профиля оформить кнопку, расположенную в соответствии с дизайном, которая позволит удалить аккаунт
- Удостовериться, точно ли пользователь этого хочет, высветив ему в интерфейсе предупреждение с соответствующим текстом.

3. Администрирование профиля

- Дать возможность администратору заблокировать профиль, если пользователь нарушил какой-либо из пунктов договора о присоединении
- Дать возможность администратору разблокировать профиль пользователя. При этом подписка, если она ее период не закончился за время блокировки, остается активной на оставшийся срок

Модуль управления уведомлениями

Модуль позволяет отправлять уведомления пользователям в соответствии с их действиями либо делать массовые рассылки по расписанию

- Дать возможность пользователю настроить отправку уведомление на странице профиля
- Отправлять уведомления при совершении резервации либо оплаты
- Отправлять уведомления про состояние самоката во время поездки
- Отправлять уведомления об акциях и предложениях в соответствии с актуальными требованиями компании

Модуль технической поддержки

Модуль позволяет пользователю получать помощь с техническими проблемами по запросу

- Дать пользователю возможность написать запрос в тех-поддержку и оформить это в интерфейсе в соответствии с дизайном
- Попросить пользователя обязательно классифицировать запрос. Добавить в перечень списка вариант «Другое» с обязательным комментарием (кратким пояснением причины запроса)
- Предоставить пользователю поле в интерфейсе для полного описания проблемы
- Создать запрос в базе
- Дать возможность администратору просмотреть запрос без ограничений
- Закрывать запрос после уведомления администратора в систему о предоставлении ответа по запросу.

Модуль доступа к базам данных

Модуль позволяет добавлять, изменять или удалять данные из различных БД, привязанных к системе, в зависимости от обстоятельств

- По запросу сравнить введенный номер телефона с номерами телефона в базе данных пользователей
- Добавить пользователя в базу данных пользователей при регистрации
- При удалении пользователем аккаунта удалить запись в базе данных пользователей
- Если пользователь оформил подписку, добавить запись об этом в базу данных подписок
- Изменить запись о состоянии самоката при его аренде в базе данных самокатов
- Изменить запись о состоянии самоката при его резервации в базе данных самокатов
- При аренде самоката добавить запись об этом в базу данных аренд
- При бронировании самоката добавить запись об этом в базу данных бронирования
- При запросе на отправку соответствующего уведомления воспользоваться уведомлением из базы данных уведомлений
- При добавлении администратором самоката добавить новую запись в базу данных самокатов
- При удалении администратором самоката удалить запись из базы данных самокатов

Модуль интерфейса пользователя

Модуль позволяет пользователю через интерфейс взаимодействовать с функциями ПО

1. Взаимодействие с самокатом

- Главный интерфейс приложения представляет из себя карту, определенную в некотором радиусе текущего местоположения, с доступными поблизости самокатами
- Обеспечить возможность пользователю отсканировать QR-cod самоката либо ввести номер найденного самоката для выбора типа аренды самоката. Необходимые кнопки встроены в интерфейс главного экрана в соответствии с дизайном
- Также дать возможность пользователю выбрать тип аренды самоката с помощью нажатия на соответствующий значок доступного самоката.
- Высветить всю информацию о выбранном самокате на экране, также указать тарифы аренды.
- В этом же окне встроить кнопку в соответствии с дизайном, которая позволит пользователю забронировать самокат
- Во время аренды встроить в интерфейс окно, которое будет содержать информацию об аренде, о выбранном самокате и текущей сумме аренды (если аренда поминутная)
- При завершении аренды высветить информацию об аренде, самокате и итоговой сумме (если аренда поминутная)

2. Взаимодействие с подпиской

- Встроить в главный интерфейс кнопку, которая будет отображать дополнительное окно, в котором будет информация о различных периодах подписки
- Позволить пользователю купить определенный период подписки по клику. Оформить в соответствии с дизайном три периода подписки
- Позволить пользователю просматривать информацию о своей подписки в окне профиля
- Позволить пользователю включать и выключать «автопродление» подписки. Оформить эту функцию в виде ползунка в соответствии с дизайном

3. Взаимодействие с профилем

- Встроить в главный интерфейс кнопку, которая будет открывать дополнительное окно для взаимодействия с профилем
 - В этом окне обеспечить пользователю возможность изменить поля личной информации
 - Дать пользователю возможность удалить аккаунт по клику на соответствующую кнопку
4. Взаимодействие с тех – поддержкой
- Дать возможность пользователю в дополнительном окне обратиться в тех-поддержку, указав классификацию запроса
5. Взаимодействие с банковской картой в приложении
- Если банковская карта не привязана к приложению, то обновить интерфейс на соответствующий и позволить ввести пользователю данные банковской карты при запросе на оплату услуги
 - Дать пользователю возможность привязать несколько банковских карт приложению

Модуль интерфейса администратора

Модуль позволяет администратору через интерфейс взаимодействовать с функциями ПО

1. Управление профилем
- Добавить в интерфейс приложения для администратора дополнительные функции управления профилем
 - Дать возможность администратору просматривать нарушения пользователей и в соответствии с ними блокировать аккаунт пользователя по клику на соответствующую кнопку
 - Дать возможность администратору разблокировать аккаунт пользователей в определённых обстоятельствах по клику на кнопку в профиле
2. Управление самокатами
- Добавить в интерфейс приложения для администратора дополнительные функции управления самокатами
 - Дать возможность администратору добавить новый самокат в базу данных при его прибытии на парковку. После добавления самокат будет виден пользователям на карте

- Дать возможность администратору удалить из базы данных неподлежащий для пользования самокат. После удаления самокат не будет виден пользователям вне зависимости от того, где он находится

3. Обработка запросов в тех-поддержку

- Дать администратору возможность видеть список запросов пользователей в тех-поддержку в специальном окне отображения
- Дать администратору возможность отправить ответ на запрос пользователя в виде уведомления
- Дать администратору возможность закрывать запросы пользователей после отправки ответа в системе кликом на соответствующую кнопку

3.1.2. Требования функциям ПО

3.1.2.1. Разрабатываемая система должна обеспечивать высокую производительность и стабильную работу мобильного приложения, обеспечивая удовлетворение потребностей пользователей и минимизацию возможных сбоев.

Функции пользователя:

1. Зарегистрироваться
2. Авторизоваться
3. Ввести личные данные
4. Ввести код подтверждения
5. Ввести номер телефона
6. Просмотреть профиль
7. Купить подписку
8. Привязать карту
9. Редактировать профиль
10. Удалить профиль
11. Получать уведомления
12. Найти самокат
13. Арендовать самокат
14. Забронировать самокат
15. Просмотреть историю поездок
16. Просмотреть информацию о самокате
17. Обратиться в тех-поддержку

Функции администратора:

1. Авторизоваться

2. Обработать запрос тех-поддержки
3. Добавить самокат
4. Удалить самокат
5. Заблокировать аккаунт пользователя
6. Разблокировать аккаунт пользователя

В приложении 4 к настоящему ТЗ приводится подробное описание функциональных требований.

3.1.2.2. Входными данными разрабатываемого ПО должны быть:

1. Данные о пользователе:

- Номер телефона пользователя
- Местоположение пользователя

2. Запросы на поиск и аренду самоката:

- Запросы от пользователей для поиска доступных самокатов.
- Запросы от пользователей для аренды конкретного самоката.

3.1.2.3. Выходными данными разрабатываемого ПО должны быть:

1. Сведения о самокатах:

- Информация о доступных самокатах, включая их местоположение, уровень заряда и другие характеристики
- Тариф аренды для поминутного и сумма аренды для остальных типов.

2. Подтверждение аренды:

- Подтверждение аренды самоката пользователем
- Информация о сроке аренде и подсчет необходимой суммы в случае поминутной аренды

3.2. Требования к надежности

Надежность информационной системы определяется надежностью средств вычислительной техники и программного обеспечения, используемых в системе.

3.3. Требования к составу и параметрам технических средств

Для полноценного функционирования системы необходимо наличие двух серверов:

- **Сервер №1** предназначен для управления базами данных, включающих информацию о пользователях, самокатах, арендах, резервациях и так далее

- **Сервер №2** предназначен для обработки запросов клиентов в приложение, а также для обработки транзакций и интеграции с внешними системами (например, система платежей). Второй сервер должен иметь доступ к внешней сети Интернет.

Рабочие станции, с которых осуществляется доступ к системе, должны быть подключены к Интернет.

3.4. Требования к информационной программной совместимости

3.4.1. Для разработки ПО должны использоваться следующие языки программирования высокого уровня и разметки данных:

- а) С спецификации стандарт ISO C99, GNU Coding Standards;
- б) C++ спецификации стандарт ISO C99, GNU Coding Standards;
- в) HTML 4.0 и выше;
- г) JavaScript 1.0 и выше.

Допускается использование вспомогательных библиотек ПО с открытым исходным кодом.

3.4.2. Разрабатываемое ПО не должно основываться на программных продуктах, требующих лицензионных отчислений от пользователей.

3.4.3. Разрабатываемое ПО должно обеспечивать доступ пользователей к своей функциональности посредством HTML-браузеров.

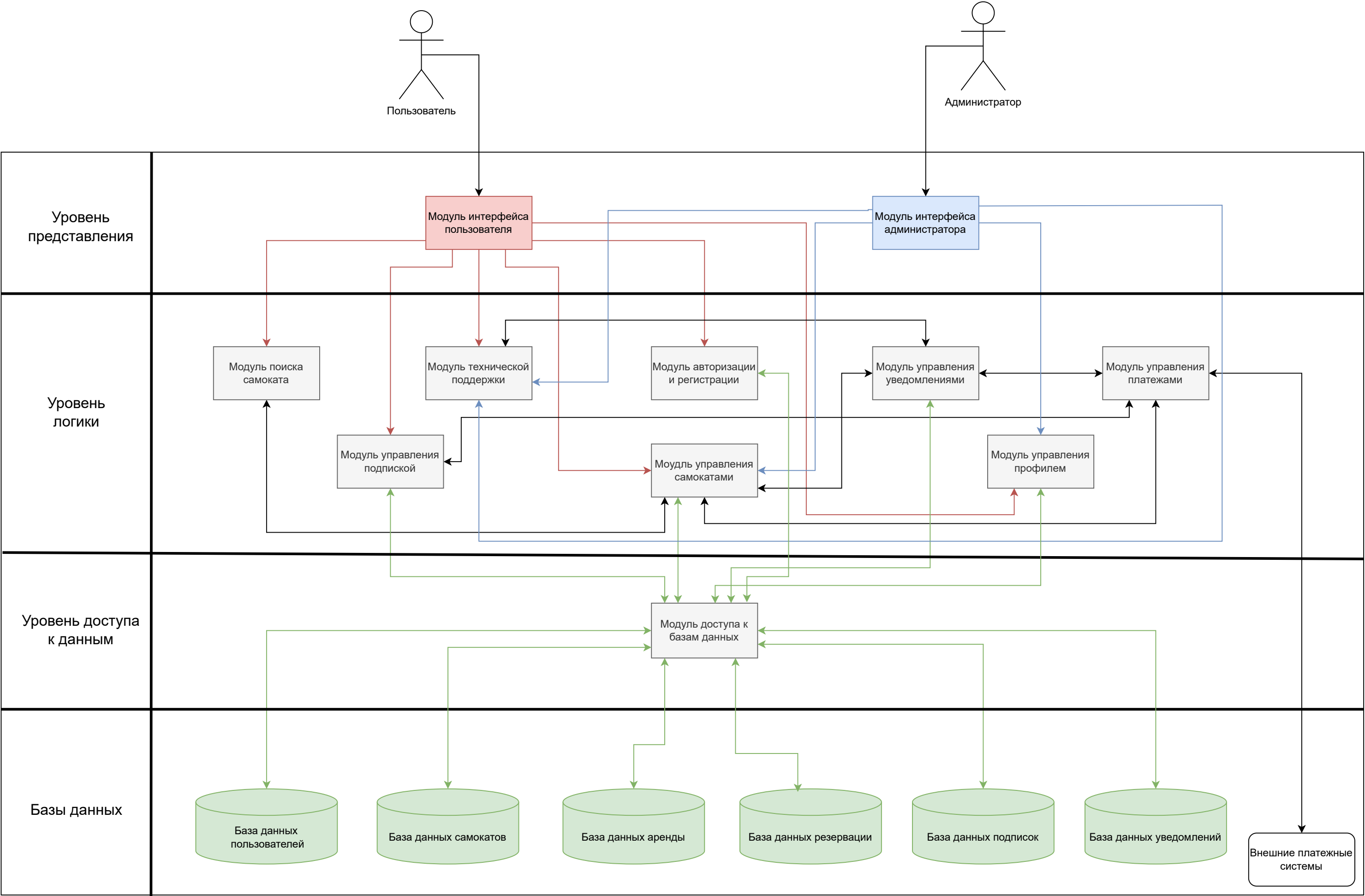
4. ТРЕБОВАНИЯ К ПРОГРАММНОЙ ДОКУМЕНТАЦИИ

При создании ПО должны быть оформлены следующие документы:

1. Руководство системного программиста (ГОСТ 19.503-79);
2. Руководство оператора (ГОСТ 19.505-79);
3. Программа и методика испытаний (ГОСТ 19.301-79).

Информативность документации должна соответствовать потребностям жизненного цикла ПС.

Трёхуровневая архитектура сервиса аренды самокатов "Юрент"



Лабораторная работа №1 представляет из себя три диаграммы для сервиса аренды самокатов «Юрент».

Диаграмма Use Case

Диаграмма включает в себя возможности пользователя (разделено на два актера для удобства) и возможности администратора

Диаграмма классов

Диаграмма включает в себя, как классы, которые являются вместилищем каких-то данных (Rental, Reservation и т.д), так и классы, содержащие в себе только методы

Связи между классами:

Основная связь, которую я использовала, - ассоциация. Также на диаграмме присутствуют агрегация, зависимость и композиция.

Поясню:

- User и Identities – зависимость. Класс Identities требует доступ к определенным данным, содержащимся в User, для корректной работы его методов. Также изменения в User могут так или иначе повлиять на Identities
- User и Admin – агрегация. Эти классы существуют независимо друг от друга, но класс Admin может вносить изменения в User, то есть агрегирует его (например, заблокировать аккаунт пользователя)
- Subscription и Reservatuion_Service – композиция. Сервис устроен так, что пользователь не может забронировать самокат, пока у него нет подписки. Соответственно, как только подписка куплена (создан экземпляр Subscription). То пользователь может использовать метод Reservatuion_Service.
- Subscription и Subscribe_manage, Reservation и Reservatuion_Service, Payment_information и Payment_Service, Notification и Notification_manage, Rental и Rental_Service – композиция. Все эти классы связаны по, можно сказать, одинаковому принципу: запись в базе данных не создастся, если не будет выполнено действие (то есть пока существует, например, экземпляр класса, методом которого является начать бронирование, будет существовать и экземпляр класса, где записывается информация о конкретной брони)

Диаграмма последовательности действий

Включает в себя три сценария:

1. Поиск самоката, его бронирование и аренда самоката с последующей оплатой поездки
2. Покупка подписки пользователем
3. Запрос в тех-поддержку (от пользователя админу)

Ниже представлены три диаграммы в том порядке, который описан выше.

Диаграмма "Use Case"

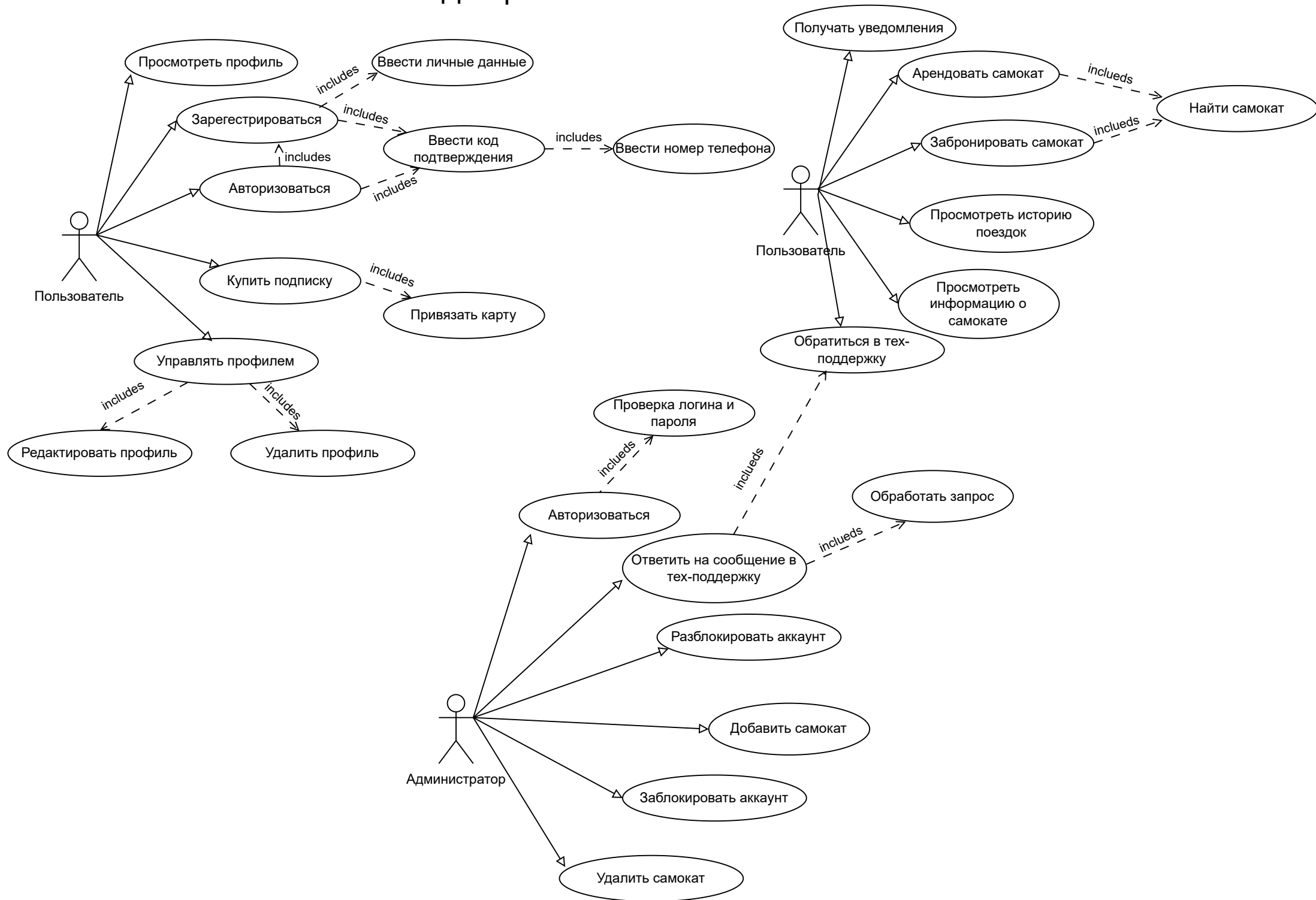


Диаграмма классов

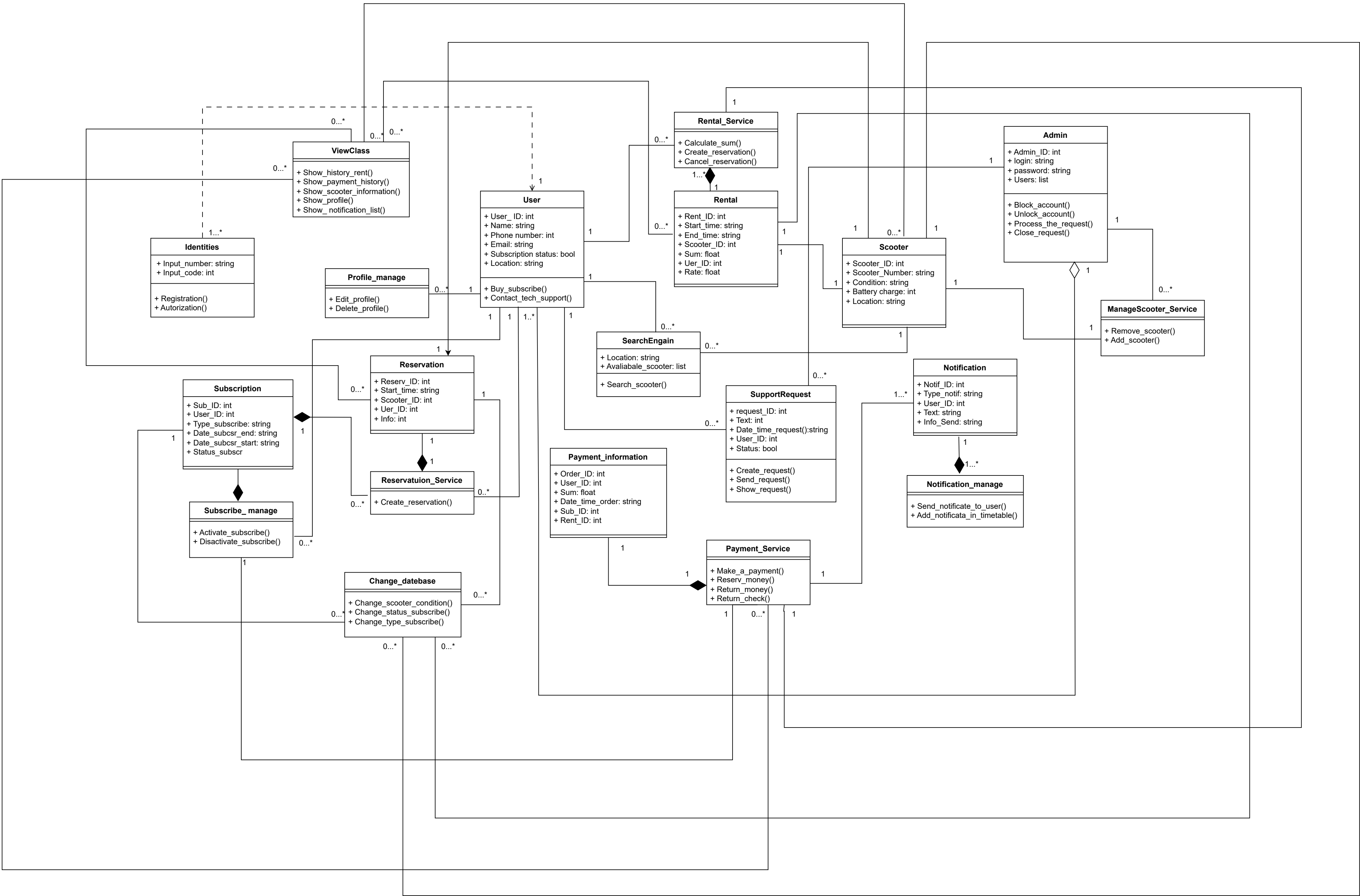
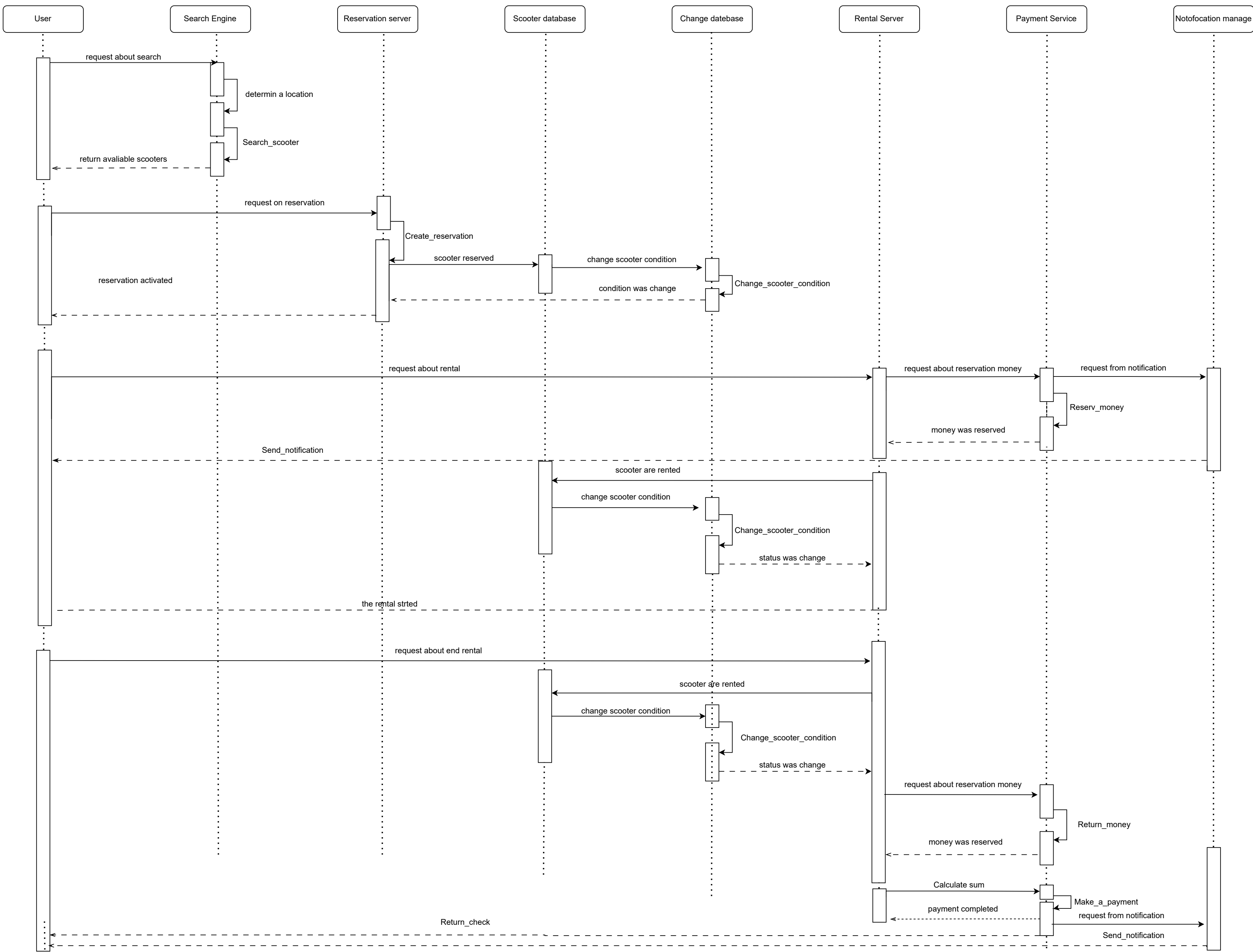
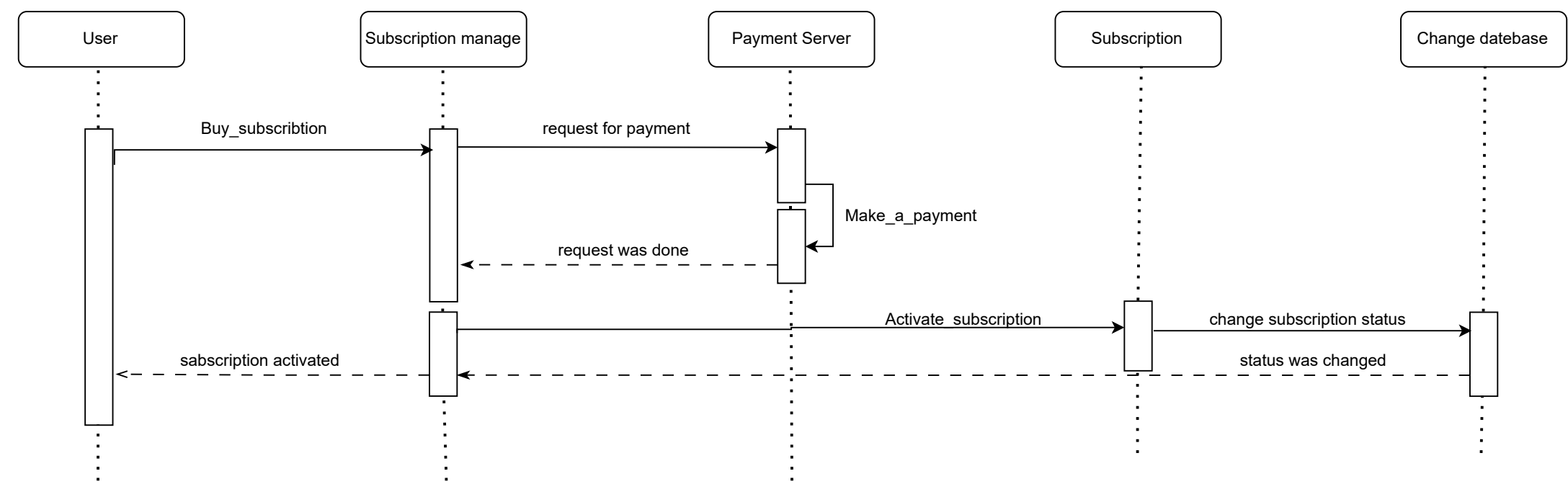


Диаграмма последовательности действий

Сценарий 1: Бронирование и аренда самоката



Сценарий 2: Покупка подписки



Сценарий 3: Запрос в тех-поддержку

