**Федеральное государственное автономное образовательное учреждение**

**высшего образования**

**РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ**

**Факультет физико-математических и естественных наук**

ЛАБОРАТОРНАЯ РАБОТА

Дисциплина: Программная инженерия

Студент: Шатохина В.С.

Группа: НФИбд-02-21

Студенческий билет № 1032217046

Москва

2023

Сервис бронирования жилья AirBnB

* **Airbnb** — это сервис, где путешественники арендуют места для проживания в 191 стране мира.

Пользователи могут арендовать или сдавать в аренду жилые помещения.

Диаграмма Use Case

Диаграмма вариантов использования является исходным концептуальным представлением или концептуальной моделью системы в процессе ее проектирования и разработки.

Изображение выглядит как текст, диаграмма, круг, снимок экрана

Автоматически созданное описание

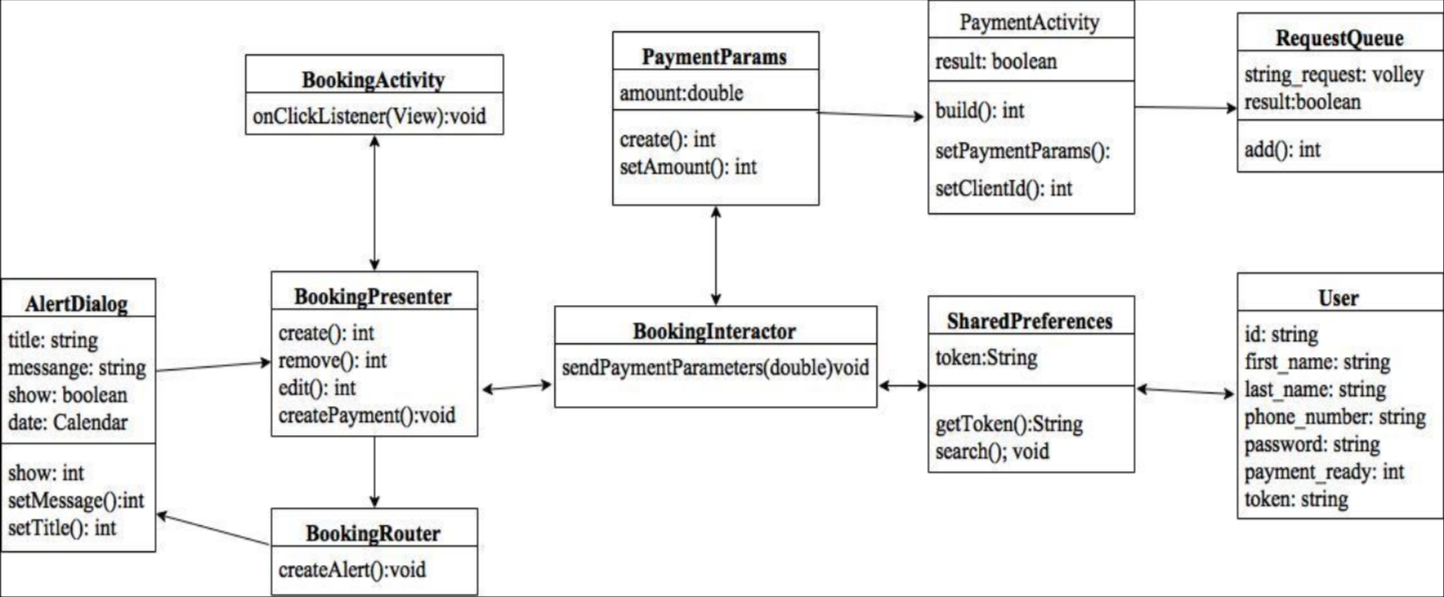
* Пользователь
* Агент недвижимости
* Администратор

Uses:

1. Поиск жилья: Пользователь выбирает критерии поиска (тип жилья, цена, кол-во комнат, город и т.д.). Система отображает список доступных вариантов жилья, соответствующих критериям. Пользователь выбирает желаемый вариант жилья. Система отображает подробную информацию о выбранном варианте жилья.
2. Бронирование жилья: Пользователь выбирает вариант жилья для бронирования. Пользователь вводит детали бронирования (даты, количество гостей и т.д.). Система проверяет доступность жилья в указанные даты. Если жилье доступно, система запрашивает подтверждение бронирования у пользователя. Пользователь подтверждает бронирование. Система создает бронь для выбранного жилья.
3. Управление объявлениями: Агент недвижимости создает новое объявление о жилье. Агент недвижимости вводит информацию о жилье (описание, фотографии, цена и т.д.). Система проверяет введенные данные на корректность. Если данные корректны, система сохраняет объявление и делает его доступным для поиска.
4. Управление агентами: Администратор создает нового агента недвижимости. Администратор вводит информацию о новом агенте (имя, контактная информация и т.д.). Система проверяет введенные данные на корректность. Если данные корректны, система сохраняет информацию агента.
5. Управление бронями: Агент недвижимости просматривает список активных броней для его объявлений. Агент недвижимости проверяет детали бронирования и связывается с клиентом для подтверждения бронирования. Агент недвижимости обновляет статус брони (подтверждена, отклонена и т.д.). Система обновляет статус брони и уведомляет пользователя о результате.

Диаграмма Классов

Определение программистом собственных классов объектов для конкретного набора задач должно позволить описывать отдельные задачи в терминах самого класса задач. При разработке программы должны быть определены классы используемых в программе объектов и построены их описания, затем созданы экземпляры этих объектов и определено взаимодействие между ними.



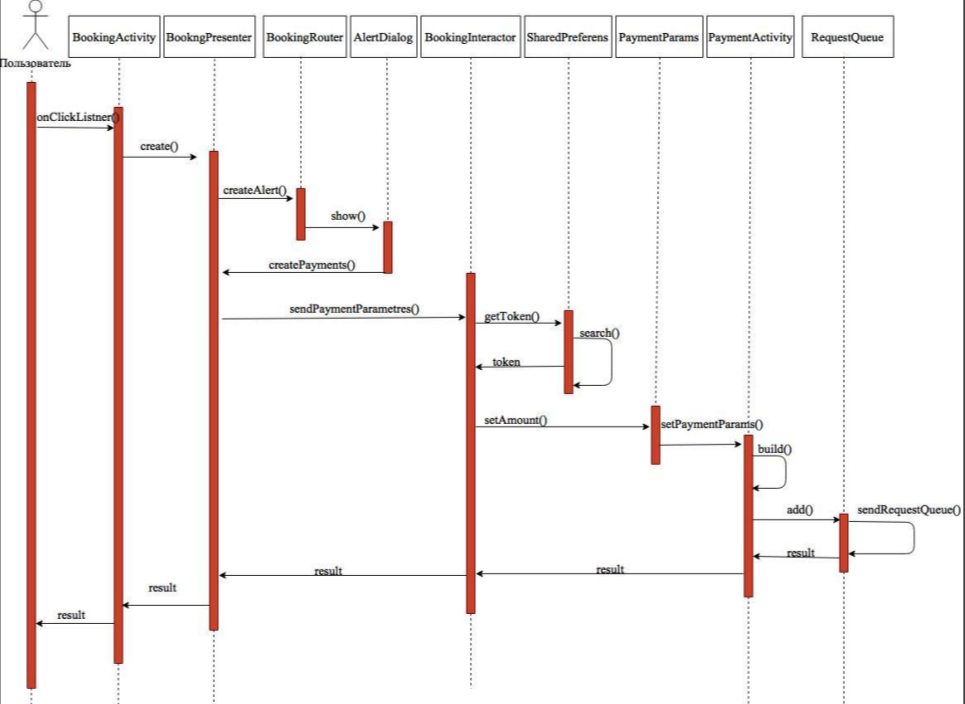
Классы:

* Класс User: представляет пользователя AirBnB, содержит информацию о имени, электронной почте, пароле, идентификаторе пользователя и другой персональной информации.
* Класс AlertDialog: формируется во время процесса выбора жилья для подтверждение бронирования.
* Класс PaymentParams: формируется при подтверждении, желания пользователя оплатить заказ выводит стоимость заказа для оплаты.
* Класс PaymentActivity: формируется для оплаты.
* Класс RequestQueue: создаётся для отправки информации о заказе на сервер.
* Класс SharedPreference: репозиторий приложения.
* Класс BookingInteractor: содержит в себе бизнес-логику процесса бронирования
* Класс BookingPresenter: определяет куда отдать управление.
* Класс BookingRouter: отвечает за переключение между различными экранами.
* Класс BookingActivity: содержит в себе окно выбора приложения.

Диаграмма последовательности действий

Диаграмма последовательности предназначена для представления динамики поведения объектов, отображая передачу сообщений между соответствующими классами.

Прецедент: Бронирование и оплата



Краткое описание:

Действующие лицо этого прецедента пользователь.

Основываясь на диаграмме классов, выберем объекты, задействованные в данной диаграмме:

* User.
* AlertDialog.
* PaymentParams.
* PaymentActivity.
* RequestQueue.
* SharedPreference.
* BookingInteractor.
* BookingPresenter
* BookingRouter.
* BookingActivity.

Прецедент начинается, когда пользователь хочет забронировать жильё:

* Объект «Пользователь», осуществляет регистрацию в приложение.
* «Пользователь» посылает объекту «User» сообщение «create», которое инициирует создание учетной записи пользователя.
* Объект «User» посылает «setMessage» и «setTitle» объекту «AlertDialog», что означает выбор пользователя для бронирования жилья.
* Объект «AlertDialog» посылает «setAmount» объекту «PaymentParams», содержащий в себе стоимость заказа.
* Объект «PaymentParams» посылает «setPaymentParams» и «setClientid» объекту «PaymentActivity» для произведения оплаты.
* Объект «PaymentActivity» объекту «RequestQueue» переменную «result» сообщающую о исполнение оплаты заказа.