Учреждение образования

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Лабораторная работа № 5. Объектно-ориентированное моделирование. Физические диаграммы UML

Студент: Валдайцев А. Д.

ФИТ 4 курс 5 группа

Преподаватель: Панченко О. Л.

Минск 2023

# Тема и цель работы

Темой данной лабораторной работы является построение UML-диаграммы классов, необходимое для графического представления работ, объектов и функциональных требований системы, представленной в первой лабораторной работе – сервисе бронирования номеров в отеле «StaySpot».

Целью лабораторной работы является изучение методологии объектно-ориентированного моделирования средствами UML, ознакомление с основными принципами объектно-ориентированного проектирования программного обеспечения и получение навыков проектирования структуры информационной системы с применением UML.

# Описание функциональных требований

Функциональны требования к системе можно разделить на требования к функционалу для различных ролей приложения – пользователя, гостя, администратора и владельца отеля.

Функционал для пользователя:

* регистрация и авторизация;
* поиск отеля на выбранную дату;
* поиск и фильтрация отелей;
* просмотр правил отеля;
* бронирование номера в отеле;
* добавление отзыва на отель;
* просмотр и обновление личных данных;
* отмена бронирования комнаты.

Функционал для владельца отеля:

* добавление, изменение и удаление информации об отеле;
* добавление, изменение и удаление информации о комнатах;
* добавление, изменение и удаление информации о правилах отеля;
* добавление фотографий комнат.

Функционал для администратора:

* просмотр списка всех пользователей;
* просмотр списка всех владельцев отелей;
* добавление, изменение и удаление информации об отелях и комнатах;
* подтверждение регистрации владельца отеля.

Функционал для гостя:

* регистрация;
* авторизация;
* просмотр списка отелей.

# Описание программных средств

Для построения UML-диаграмм классов использовался веб-ресурс Draw.io, разрабатываемый компанией JGraph Ltd. и направленный на построение диаграмм. Адрес веб-ресурса – <https://www.drawio.com>. Данный ресурс доступен на всех платформах, имеющих веб-браузер и доступ в Интернет.

В функционал веб-ресурса Draw.io входит построение графиков, смысловых карт, UML-диаграмм, диаграммы Венна, Agile и Kanban досок, графиков мозговых штурмов, диаграмм архитектур технических систем и многие другие.

# Описание практического задания

В ходе выполнения практического задания необходимо построить диаграмму классов. Данная диаграмма описывает типы объектов системы и различного рода статические отношения, которые существуют между ними.

В каждом прямоугольнике, отражающем класс, содержатся имя класса, его атрибуты (поля и свойства) и операции (методы).

Связи разделяются на зависимость (если изменения одного вызовут изменения другого), ассоциацию (в случае, когда один класс пользуется функционалом другого), обобщения (наследование всех открытых полей, свойств и методов), агрегации (включение в другой класс в виде составной части, при которой зависимый класс не может существовать без основного) и композиции (также включение в виде составной части другого класса, но составной класс не может существовать обособленно от основного).

Диаграмма классов для сервиса «StaySpot» представлена на рисунке 1.1.

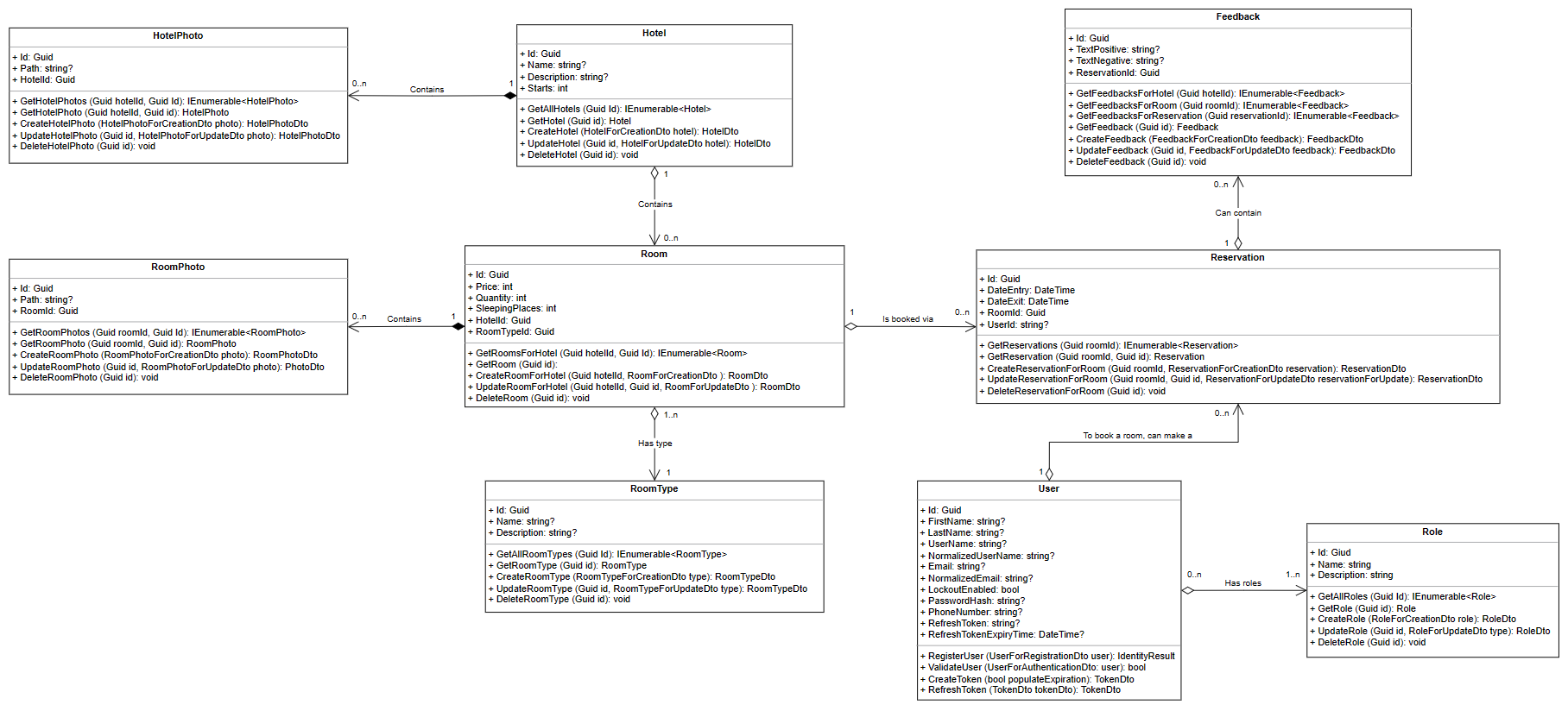


Рисунок 1.1 – Диаграмма классов

Рассмотрим подробнее составные части данной диаграммы. В ней содержится 9 классов со следующими атрибутами и операциями:

* Класс: User.  
  Атрибуты: Id, FirstName, LastName, UserName, NormalizedUserName, Email, NormalizedEmail, LockoutEnabled, PasswordHash, PhoneNumber, RefreshToken, RefreshTokenExpiryTime.  
  Операции: RegisterUser, ValidateUser, CreateToken, RefreshToken.
* Класс: Role.  
  Атрибуты: Id, Name, Description.  
  Операции: GetAllRoles, GetRole, CreateRole, UpdateRole, DeleteRole.
* Класс: RoomType.  
  Атрибуты: Id, Name, Description.  
  Операции: GetAllRoomTypes, GetRoomType, CreateRoomType, UpdateRoomType, DeleteRoomType.
* Класс: Room.  
  Атрибуты: Id, Price, Quantity, SleepingPlaces, HotelId, RoomTypeId.  
  Операции: GetRoomsForHotel, GetRoom, CreateRoomForHotel, UpdateRoomForHotel, DeleteRoom.
* Класс: Hotel.  
  Атрибуты: Id, Name, Description, Stars.  
  Операции: GetAllHotels, GetHotel, CreateHotel, UpdateHotel, DeleteHotel.
* Класс: RoomPhoto.  
  Атрибуты: Id, Path, RoomId.  
  Операции: GetRoomPhotosForRoom, GetRoomPhoto, CreateRoomPhotoForRoom, UpdateRoomPhotoForRoom, DeleteRoomPhoto.
* Класс: HotelPhoto.  
  Атрибуты: Id, Path, HotelId.  
  Операции: GetHotelPhotosForHotel, GetHotelPhoto, CreateHotelPhotoForHotel, UpdateHotelPhotoForHotel, DeleteHotelPhoto.
* Класс: Reservation.  
  Атрибуты: Id, DateEntry, DateExit, RoomId, UserId.  
  Операции: GetReservations, GetReservation, CreateReservationForRoom, UpdateReservationForRoom, DeleteReservationForRoom.
* Класс: Feedback.  
  Атрибуты: Id, TextPositive, TextNegative, ReservationId.  
  Операции: GetFeedbacksForHotel, GetFeedbacksForRoom, GetFeedback, CreateFeedback, UpdateFeedback, DeleteFeedback.

Данные сущности имеют следующие связи:

* Пользователи User связаны с ролями Role агрегационной связью многие-ко-многим, причем у каждого пользователя может быть 1 или несколько ролей, но каждая роль принадлежит 0 или нескольким пользователям.
* Пользователи User бронируют номера через класс Reservation, связанный агрегацией. У одного пользователя может быть 0 или несколько резерваций – экземпляров класса Reservation.
* Комнаты Room связаны с типами комнат RoomType агрегационной связью один-ко-многим, то есть у каждой из комнат есть один из многи типов.
* Отели Hotel связаны с комнатами Room агрегацией один-ко-многим, то есть в одном отеле может быть несколько комнат, но каждая комната может принадлежать только одному отелю.
* Отзывы Feedback связаны с бронями Reservation, то есть если пользователь пользовался услугами отеля посредством резервации, то он может оставить отзыв, причем на каждую бронь может не быть ни одного или быть несколько отзывов.
* Фото комнат RoomPhoto связаны с комнатами Room композицией, так как фотографии комнат могут существовать только в контексте существования самих комнат, и у каждой комнаты может быть 0 или несколько фотографий, а каждая фотография относится только к одной из комнат.
* Фото отелей HotelPhoto связаны с отелями Hotel также композицией. У каждого отеля может быть 0 или несколько фотографий, а каждая фотография относится только к одному конкретному отелю.