Министерство образования Республики Беларусь

Учреждение образования

«Белорусский государственный технологический университет»

Кафедра информационных систем и технологий

**Отчет по лабораторной работе на тему**:

**«Объектно-ориентированное моделирование. UML-диаграммы поведения»**

Выполнил:

студент 4 курса 7 группы ФИТ

Бобрович Г.С.

Минск 2023

# **1. Теоретические вопросы**

## 1.1 Дайте описание понятиям Unified process (UP) и UML.

Unified process – это методология для построения процессов разработки программного обеспечения, позволяющий команде разработки преобразовывать требования заказчика в работоспособный продукт.

UML – это унифицированный язык моделирования (Unified Modeling Language) и система обозначений, которую можно применять для объектно-ориентированного анализа и проектирования. Его можно использовать для визуализации, спецификации, конструирования и документирования программных систем.

## 1.2 Перечислите основные диаграммы UML 2.0.

Диаграммы в UML 2.0 делятся на два вида:

1) Структурные

- диаграммы классов;

- диаграммы компонентов;

- диаграммы объектов и др.

2) Поведенческие

- диаграммы вариантов использования;

- диаграммы активностей;

- диаграммы взаимодействий и др.

## 1.3 Назовите CASE-средства, поддерживающие создание UML диаграмм.

CASE-средства (Computer - Aided Software Engineering) - это методы и технологии, которые позволяют проектировать различные информационные системы (в частности, базы данных) и автоматизировать их создание.

CASE-средства, поддерживающие создание UML диаграмм:

- IBM Rational Rose;

- Microsoft Visio;

- Draw.io;

- Borland Together.

## 1.4 Укажите назначение диаграммы вариантов использования.

Диаграмма вариантов использования описывает, какой функционал разрабатываемой программной системы доступен каждой группе пользователей.

## 1.5 Опишите нотации, которые используются для построения Use-Case диаграммы.

Actor — это типовые пользователи (менеджер, операторы и т. д.). Актер представляет собой некоторую роль, которую играет пользователь (или другая система) по отношению к системе.

Возможные отношения между вариантами использования:

1) Отношения Include (Включения) — используется, когда имеется какой-либо фрагмент поведения системы, который повторяется более чем в одном варианте использования и необходимо, чтобы его описание копировалось в каждом из этих вариантов использования;

2) Отношение Extend (Расширения) — расширяющий вариант использования может дополнять (делать более точным) поведение базового варианта использования, определив в базовом варианте использования точки расширения.

Возможные отношения между актерами (группировка прецедентов):

1) Отношение Generalization (Обобщения) — если есть варианты использования, которые присущи нескольким ролям, то можно их обобщить и выделить общую абстрактную роль.

2) Отношение Uses (Использование) – основное отношение на диаграмме, которое отражает взаимодействие между актерами и прецедентами.

Примечания (Комментарии) предназначены для включения в модель произвольной текстовой информации, имеющей непосредственное отношение к контексту разрабатываемого проекта.

# **2. Описываемые функциональные требования**

## 2.1 Функциональные требования

Пользователю доступны функции, такие как:

* Регистрация/авторизация;
* Бронирование времени репетиции/записи;
* Система обратной связи (отзывы);
* Просмотр истории забронированного времени;
* Просмотр информации о студии;
* Редактирование профиля пользователя.

Модератору, помимо вышеперечисленных функций, предоставляется:

* Блокировка пользователей и ее отмена;
* Редактирование информации о студии;
* Отмена и восстановление забронированного времени;
* Просмотр списка пользователей;
* Просмотр актуального бронирования.

Администратор имеет весь описанный выше функционал, с добавлением:

* Выдача/снятие полномочий модератора;
* Удаление отзывов.

## 2.2 Основные системные требования

Приложение состоит из 3 основных компонентов:

* Сервер, необходимый для хранения информации в базе данных и для взаимодействия с пользователями интерфейса;
* Мобильное приложение;
* Веб приложение.

База данных сервера хранит забронированное время пользователей, отзывы, отмененное время. Основана на СУБД MySQL. Так же, присутствует локальная база данных SQLite, необходимая для работы приложения без подключения к сети.

Взаимодействие пользователей с интерфейсом происходит с помощью REST API. Используется технология Node.js, а передача данных выполняется в формате JSON.

Мобильное и веб приложение написано на языке Dart. Реализованы следующие экраны приложения:

* Регистрация и авторизация;
* Главная (информация о студии);
* Расписание;
* Профиль:
  + Обновление профиля;
  + История бронирования;
* Отзывы;
* Админ-панель;

## 2.3 Аппаратные требования

* ОЗУ 1 GB;
* Частота процессора не меньше 1 GHz
* Операционная система: Android 4.1 и выше, iOS 9.0 и выше;
* Интернет-соединение: 3G, 4G, Wi-Fi;
* Внутренняя память не менее 128 Mb.

# **3. Описание программных средств**

Построение моделей выполнялось в <https://app.diagrams.net/>. Draw.io — инструмент для создания диаграмм, блок-схем, интеллект-карт, бизнес-макетов, отношений сущностей, программных блоков и другого. Сервис распространяется на бесплатной основе с открытым исходным кодом. Draw.io обладает богатым набором функций для визуализации большинства задач пользователя.

*Разработчик*: JGraph Ltd. Сервис распространяется на бесплатной основе с открытым исходным кодом.

# **4. Описание практического задания**

В данной лабораторной были рассмотрены возможности пользователя и администратора при помощи диаграммы вариантов использования. Как следует из названия, в данной диаграмме рассматриваются возможности пользователей системы.

Все возможности, имеющиеся у ролей, представлены на рисунке 4.1.

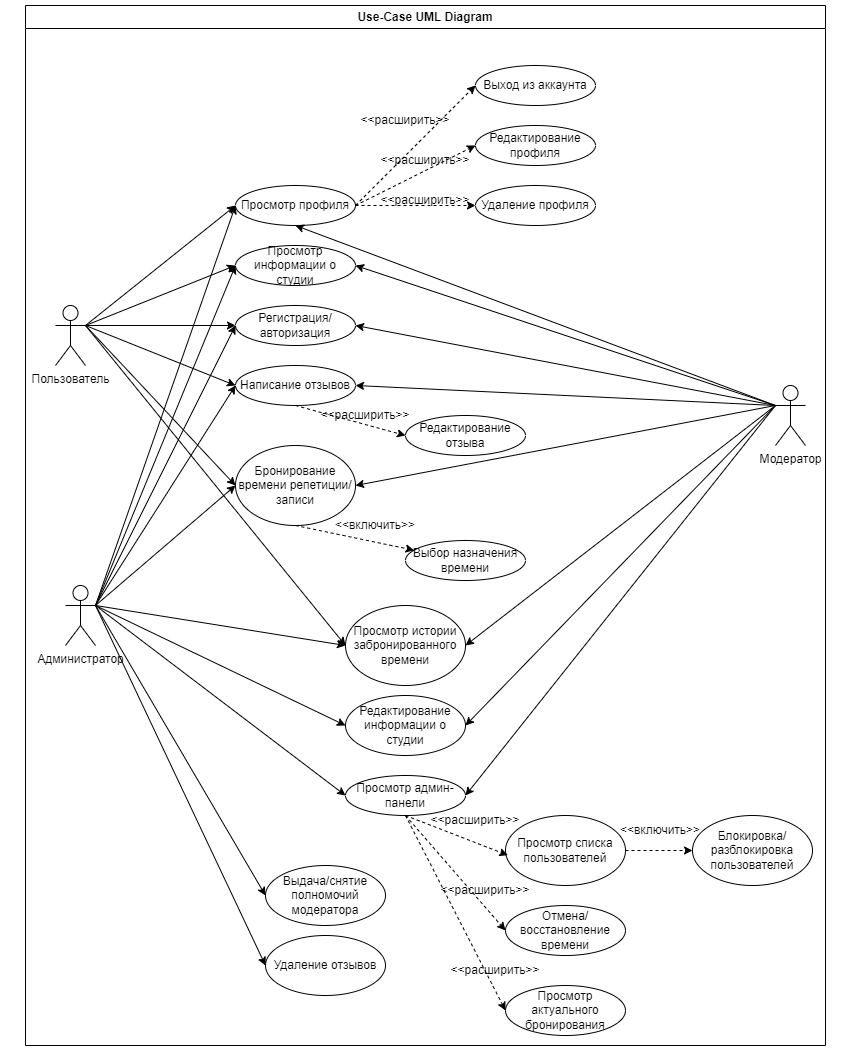


Рисунок 4.1 – Диаграмма вариантов использования

«расширить» означает, что данный прецендент (овал), от которого исходит это стрелка, подразумевает под собой некий набор более специфических возможностей. При этом прецендент, куда приходит стрелка, может существовать независимо. «включить» означает, что прецендент, куда направленна стрелка, зависит от того, откуда стрелка исходит.

Стрелка с пустым наконечником обозначает обобщение, то есть разные преценденты объединяются в один общий, ибо у них схожий функционал.