Учреждение образования

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

«Проектирование программного обеспечения»

Отчёт по лабораторной работе №4

ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ МОДЕЛИРОВАНИЕ.

UML – ДИАГРАММЫ ПОВЕДЕНИЯ

Цель: Изучение методологии объектно-ориентированного моделирования средствами UML. Ознакомление с основными принципами объектно-ориентированного проектирования программного обеспечения, получение навыков проектирования функциональности информационной системы с применением UML.

Студент: Коренчук А.В.

ФИТ 3 курс 2 группа

Преподаватель: Якунович А.В.

Минск 2023

1. **Описание программных средств**

Как и в прошлой лабораторной работе, для построения диаграмм будем использовать Draw.io –бесплатное онлайн-приложение для создания диаграмм и схем.

Название: Draw.io

Версия: Веб-приложение

Разработчик: JGraph Ltd.

Адрес : https://www.draw.io/

Режим использования: Онлайн

Доступность на платформах: любой веб-браузер.

Draw.io позволяет создавать различные типы диаграмм, такие как блок-схемы, организационные диаграммы, UML-диаграммы, сетевые диаграммы и многое другое.

1. **Описание практического задания**

Глобально выделим две группы пользователей: слушатели (обычные пользователи), а также контент-менеджеры (управляющие площадкой). Контент-менеджеры обеспечивают целостность данных системы, чтобы слушатели взаимодействовали только с актуальной музыкальной информацией. Слушатели будут выполнять основные функции – просмотр доступных треков, их детализация, создание и управление плейлистами, оценка и комментирование треков, а также подписка на интересных исполнителей.

Анализ решений конкурентов показал, что эти действия могут выполнить пользователи без учётной записи. Следовательно, возник вопрос о её наличии в данном ресурсе и преимуществах, которые она может предоставить. Было принято решение о предоставлении пользователю с учётной записью возможности создания собственной музыкальной коллекции, оценки и комментирования контента Здесь же и получится иерархическое разделение роли «слушатель» на двух актёров – гость и авторизированный пользователь. Так как пользователь, имеющий учётную запись, получает расширенные возможности, он стоит выше в иерархии, а гость – только базовый и, следовательно, ниже по иерархии. Роль «Авторизированный пользователь» является расширением роли «Гость». Готовая диаграмма представлена на рисунке 1.

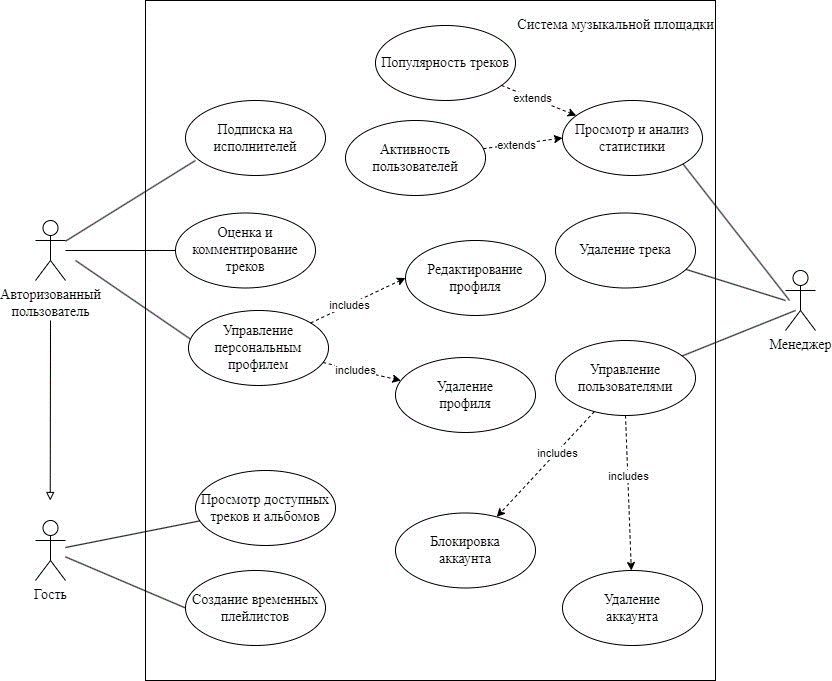


Рисунок 1 – Use-case диаграмма для музыкальной площадки

**Вывод:** В результате лабораторной работы изучены основные принципы объектно-ориентированного проектирования программного обеспечения, получены навыки проектирования функциональности системы с применением UML, а также составлена Use-case диаграмма авиакомпании, которая демонстрирует функционал и выделяет группы пользователей системы.

**Ответы на теоретические вопросы**

1. Дайте описание понятиям Unified process (UP) и UML.

Унифицированный процесс (Unified Process) – процесс разработки программного обеспечения, который обеспечивает упорядоченный подход к распределению задач и обязанностей в организации-разработчике.

Унифицированный язык визуального моделирования - Unified Modeling Language (UML) — это стандартная нотация визуального моделирования программных систем, принятая консорциумом Object Managing Group осенью 1997г., и на сегодняшний день поддерживаемая многими объектно-ориентированными CASE-продуктами.

1. Перечислите основные диаграммы UML 2.0.

Диаграммы деятельности, классов, взаимодействия, компонентов, составных структур, развертывания, обзора взаимодействия, объектов, пакетов, последовательности, конечных автоматов, временная, прецедентная.

1. Назовите CASE-средства, поддерживающие создание UML диаграмм.

Draw.io, Microsoft Visio, Software Ideals Modeler, Creately.

1. Укажите назначение диаграммы вариантов использования.

Отображает функции информационной системы, взаимодействие между актерами и функциями.

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

*Актёр* – представляет лицо или систему, которое взаимодействует с системой или подсистемой. Актёры могут быть как внешними (например, пользователи, другие системы), так и внутренними (например, другие подсистемы или компоненты).

*Прецеденты* – описывают функциональность или поведение системы с точки зрения пользователя. Они представляют цели или задачи, которые пользователь или другой актёр может выполнить с помощью системы.

*Подсистема* – является частью системы и содержит группу связанных функций (прецедентов). Подсистемы используются для структурирования системы на более управляемые части.