МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждение образования «БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ

ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

**Проектирование информационных систем**

**Тема:** «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.

Выполнил:

Тышкевич Р.А.,

3 курс 7 группа

Проверил:

Якунович А. В.

Минск 2023

**Описание программных средств**

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.

Инструмент работает с Google Диск, Google Workspace и Dropbox, глубоко интегрирован и удобен для работы с продуктами Confluence и Jira от Atlassian. Пользователи также могут работать с диаграммами в автономном режиме и сохранять их локально, используя настольное приложение для персональных компьютеров.

Инструмент позволяет создавать: графики, диаграммы, таблицы, презентации, блок-схемы, планы помещений, воронки продаж, ментальные карты, карты сайтов.

**Описание практического задания**

**Список пользователей и их роли:**

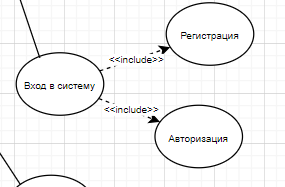
1. Пользователь: может создавать сцены, работать с движком, его версией и компонентами.
2. Администратор: имеет возможности, внедрение новых компонентов в движок, изменение движка под нужды разработчиков и обеспечения безопасности данных.
3. Разработчик: независимая роль, обладает полным доступом к движку и контролем над его версиями. Может управлять пользователями и их доступом к движку.

**Иерархия актеров подсистемы:**

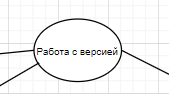
1. «Разработчик» – независимая роль.
2. Роль «Администратор» является «расширением» роли «Пользователь».

**Выделенные прецеденты подсистемы и схемы их взаимодействия:**

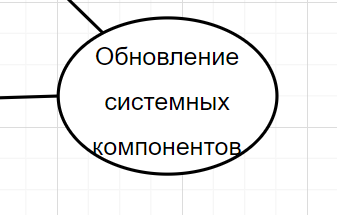
1. Регистрация
2. Вход в систему.
3. Авторизация.



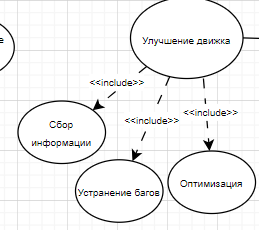
1. Работа с версией.



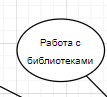
1. Обновление системных компонентов.



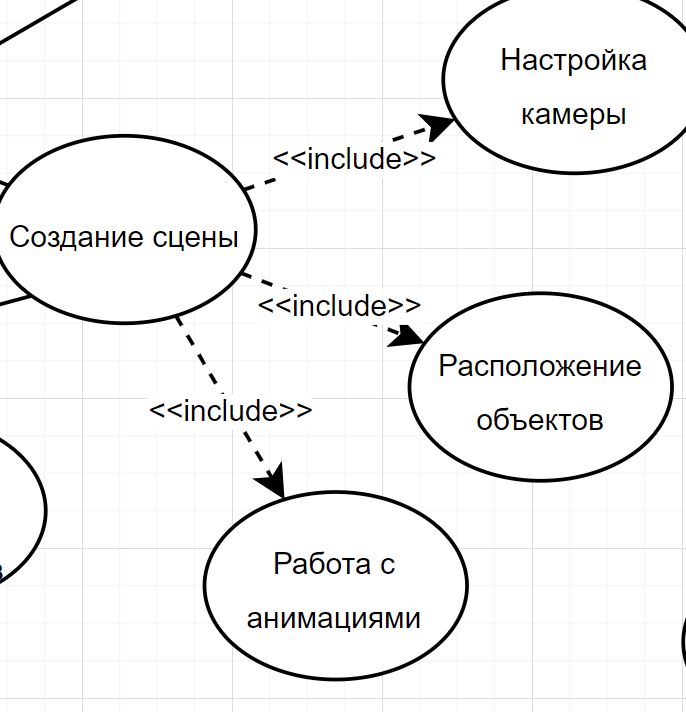
1. Улучшение движка.
2. Сбор информации.
3. Устранение багов.
4. Оптимизация.



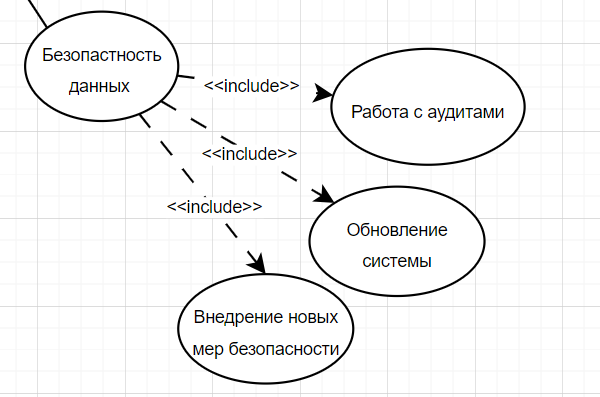
1. Работа с библиотеками.



1. Создание сцены.
2. Настройка камеры.
3. Расположение объектов.
4. Работа с анимациями.



1. Безопасность данных.
2. Работа с аудитами.
3. Обновление системы.
4. Внедрение новых мер безопасности.



Построенные диаграммы представлено на рисунке 1.

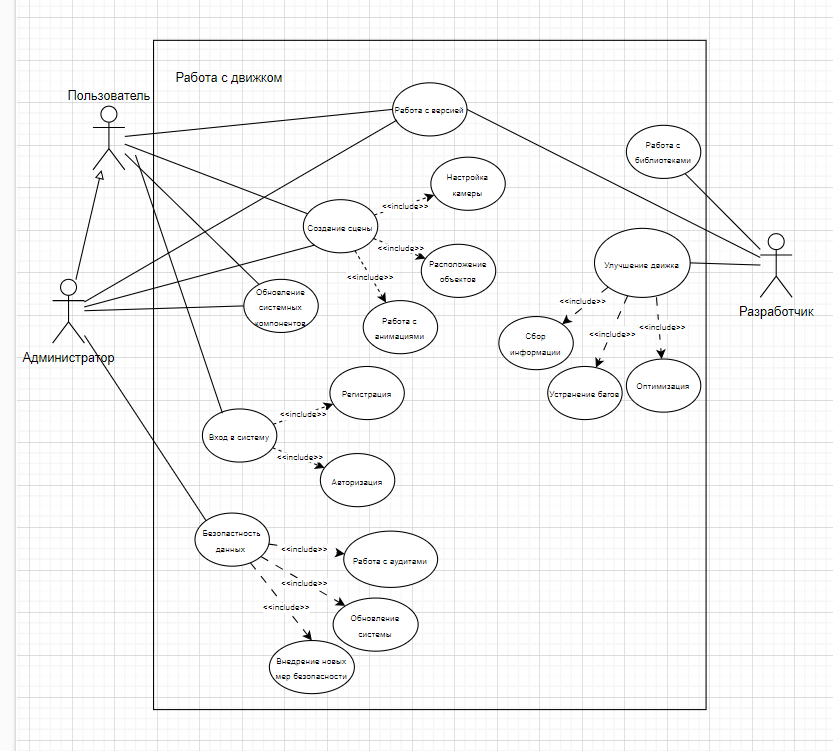


Рисунок 1 – UML-диаграмма поведения

**Вывод:** Я изучил методологию объектно-ориентированного моделирования средствами UML. Ознакомился с основными принципами объектно-ориентированного проектирования программного обеспечения, получил навыки проектирования функциональности информационной системы с применением UML.