



**Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)**

**Факультет «Информатика и системы управления»
Кафедра ИУ5 «Системы обработки информации и управления»**

**Отчет по лабораторной работе №5
по дисциплине «Технологии разработки программного обеспечения»
по теме «АИС учета успеваемости учеников средней школы»**

**Выполнил:
студент группы № ИУ5-15М
Андреев А.В.
подпись, дата**

**Проверила:
Виноградова М.В.
подпись, дата**

2022 г.

Цель работы.

- Изучить унифицированный процесс разработки (RUP);
- Приобрести умения построения модели анализа;
- Получить навыки построения модели требований в среде Sparx Enterprise Architect.

Задачи:

- I. Описать модель требований – диаграмма прецедентов;
- II. Описание модели анализа требований:
 - 1) Создать трассировку прецедентов в кооперации;
 - 2) Построить обзорные диаграммы классов сущностей, управляющих и граничных;
 - 3) Добавить к диаграмме классов участников коопераций;
 - 4) Построить диаграмму последовательности;
 - 5) Описать обзорную диаграмму классов;
 - 6) Построить диаграмму пакетов (с классами).

Тема: Автоматизированная информационная система учета успеваемости учеников средней школы.

1. Определение коопераций

В начале анализа следует добавить кооперации для реализации основных прецедентов. Для каждого прецедента одну кооперацию. Название кооперации совпадает с названием прецедента. Составляем «обзорную» диаграмму коопераций, содержащую прецеденты и реализующие их кооперации, связь между ними типа «реализации» со стереотипом трассировки (отображения) «trace».

Трассировка (Trace) — это специализация зависимости, связывающая элементы модели или наборы элементов, представляющих одну и ту же идею в разных моделях. Трассировки часто используются для отслеживания требований и изменений модели. Поскольку изменения могут происходить в обоих направлениях, порядок этой зависимости обычно игнорируется.

В результате определения коопераций АИС учета успеваемости учеников средней школы была сформирована обзорная диаграмма коопераций и прецедентов.

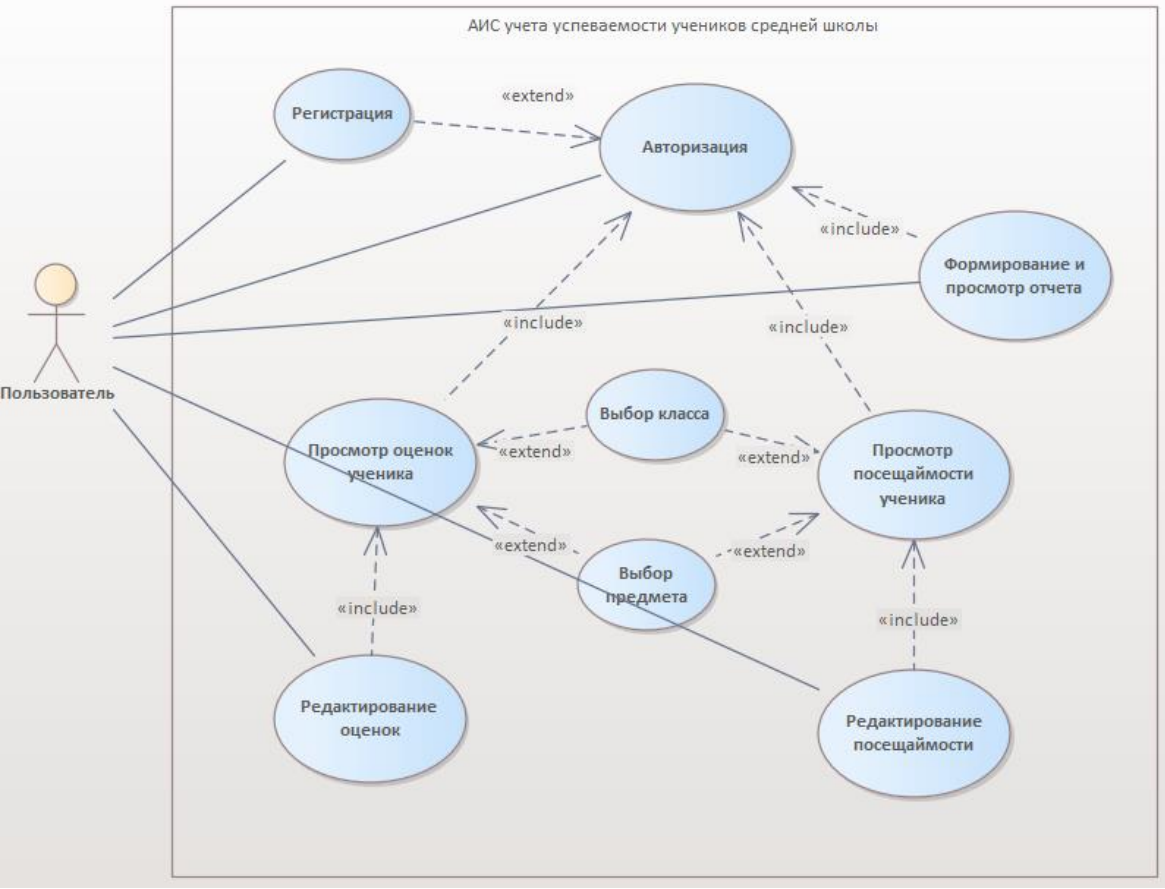


Рисунок 1.1. Диаграмма прецедентов

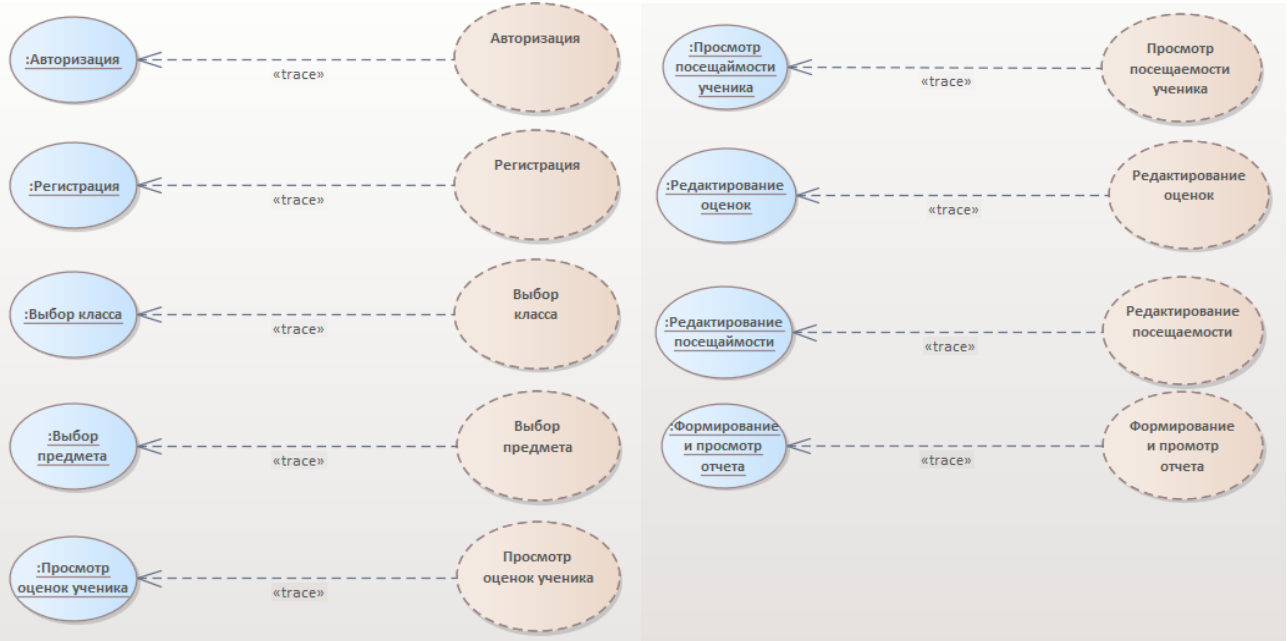


Рисунок 1.2. Диаграмма трассировки коопераций

2. Определение классов

Классы сущностей показывают логическую структуру данных. Они используются для моделирования долгоживущей, часто сохраняемой информации о человеке, объекте или событии реального мира и могут иметь сложное поведение. Классы сущностей определяют на основе модели предметной области или бизнес-модели. В результате определения классов сущностей АИС учета успеваемости учеников средней школы была сформирована обзорная диаграмма классов сущностей.

Entity

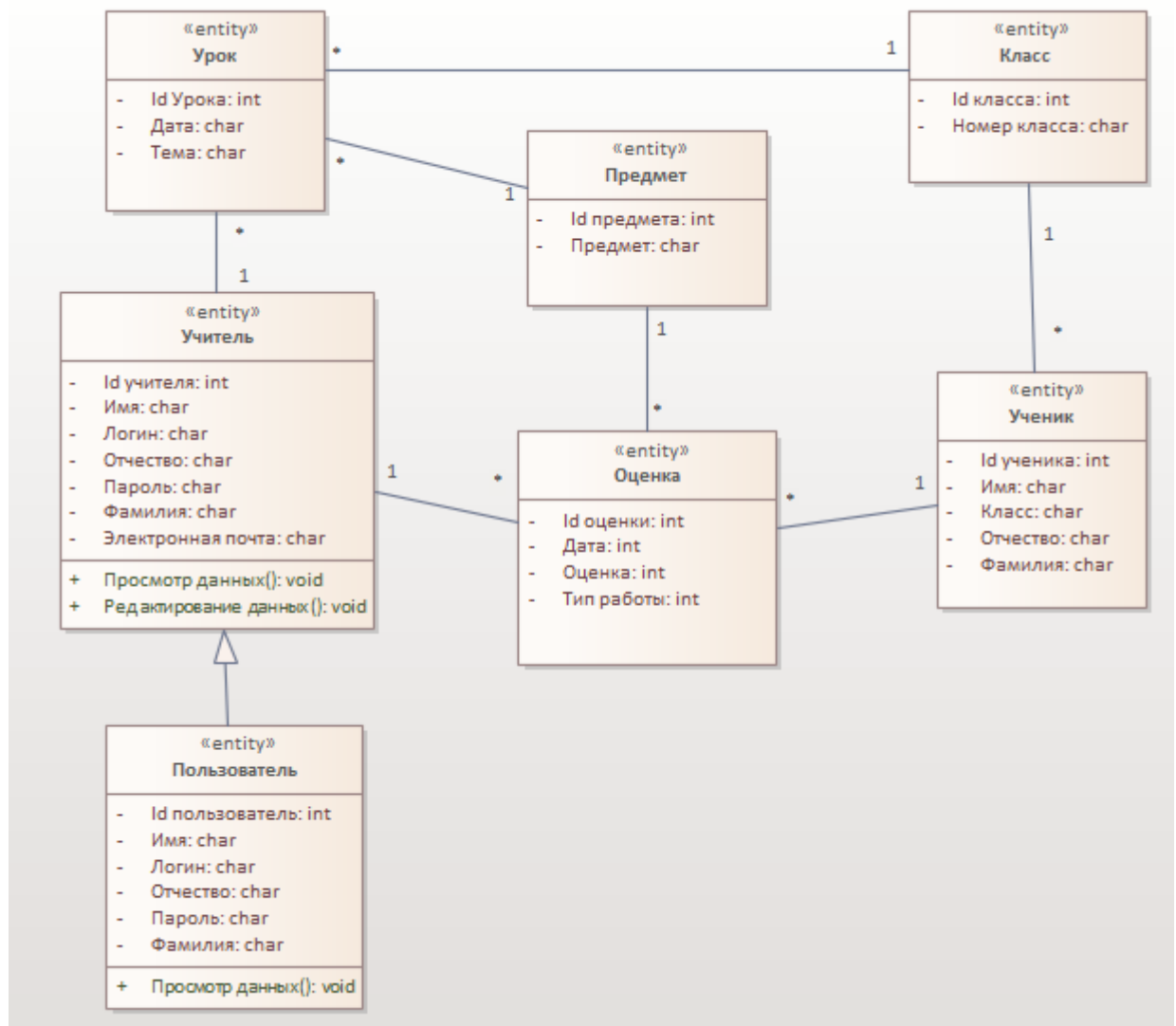


Рисунок 2.1. Диаграмма классов сущностей

Boundary

Граничный класс – выделяют для моделирования взаимодействия между системой и актерами: получение и передача информации или запроса. Граничный класс соответствует пользовательскому интерфейсу, интерфейсу внешних устройств или коммуникационному протоколу, библиотеке API на высоком уровне абстракции без физической реализации. Граничный класс обозначается стереотипом «boundary». В результате определения граничных классов АИС учета успеваемости учеников средней школы была сформирована обзорная диаграмма граничных классов.

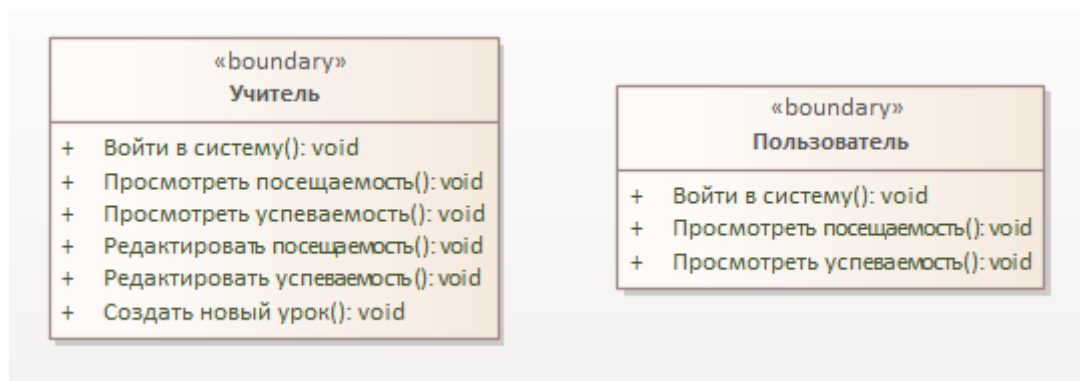


Рисунок 2.2. Диаграмма граничных классов

Control

Управляющий класс – выполняет координацию, последовательность, взаимодействие и управление другими объектами для прецедента. Обрабатывает и координирует действия и потоки управления, реализует бизнес-логику. Управляющий класс обозначается стереотипом «control». В результате определения управляющих классов АИС учета успеваемости учеников средней школы была сформирована обзорная диаграмма управляющих классов

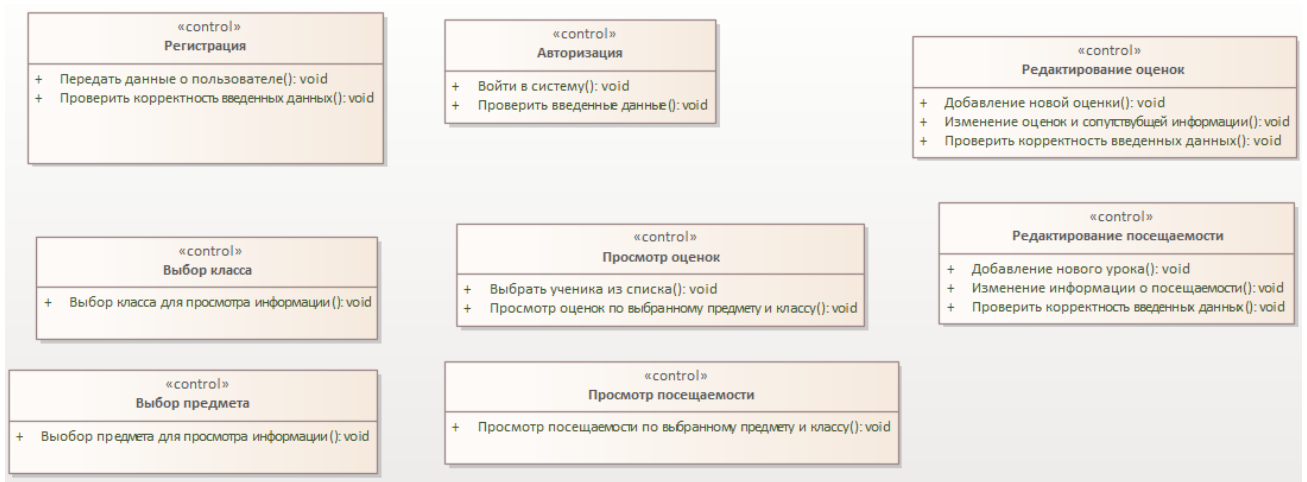


Рисунок 2.3. Диаграмма классов управления

3. Диаграмма классов с участниками кооперации

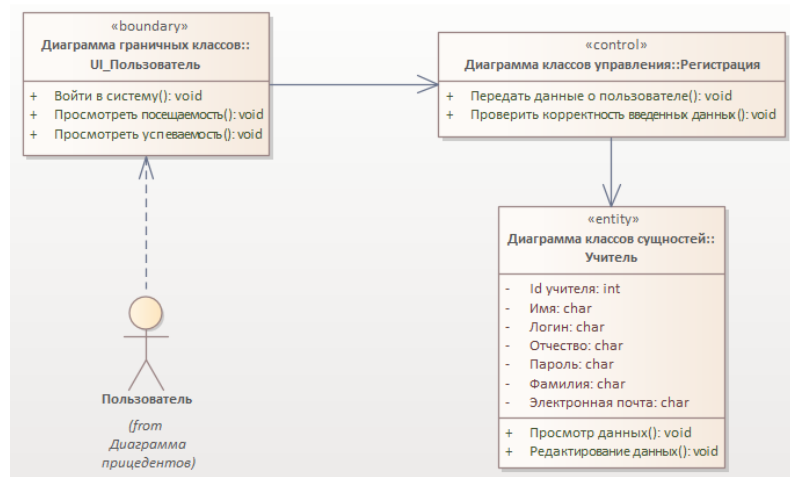


Рисунок 3.1. Диаграмма классов участников кооперации «Регистрация»

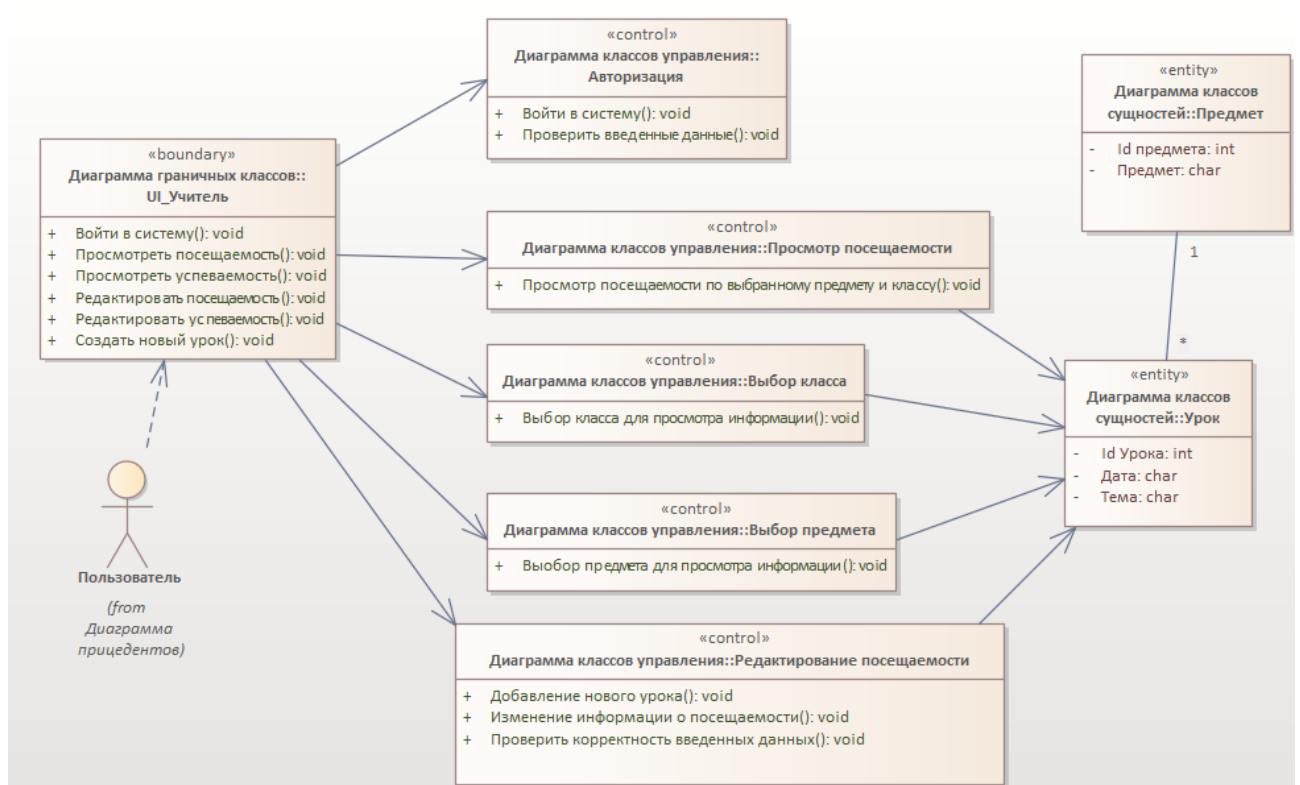


Рисунок 3.2. Диаграмма классов участников кооперации «Просмотр и редактирование посещаемости»

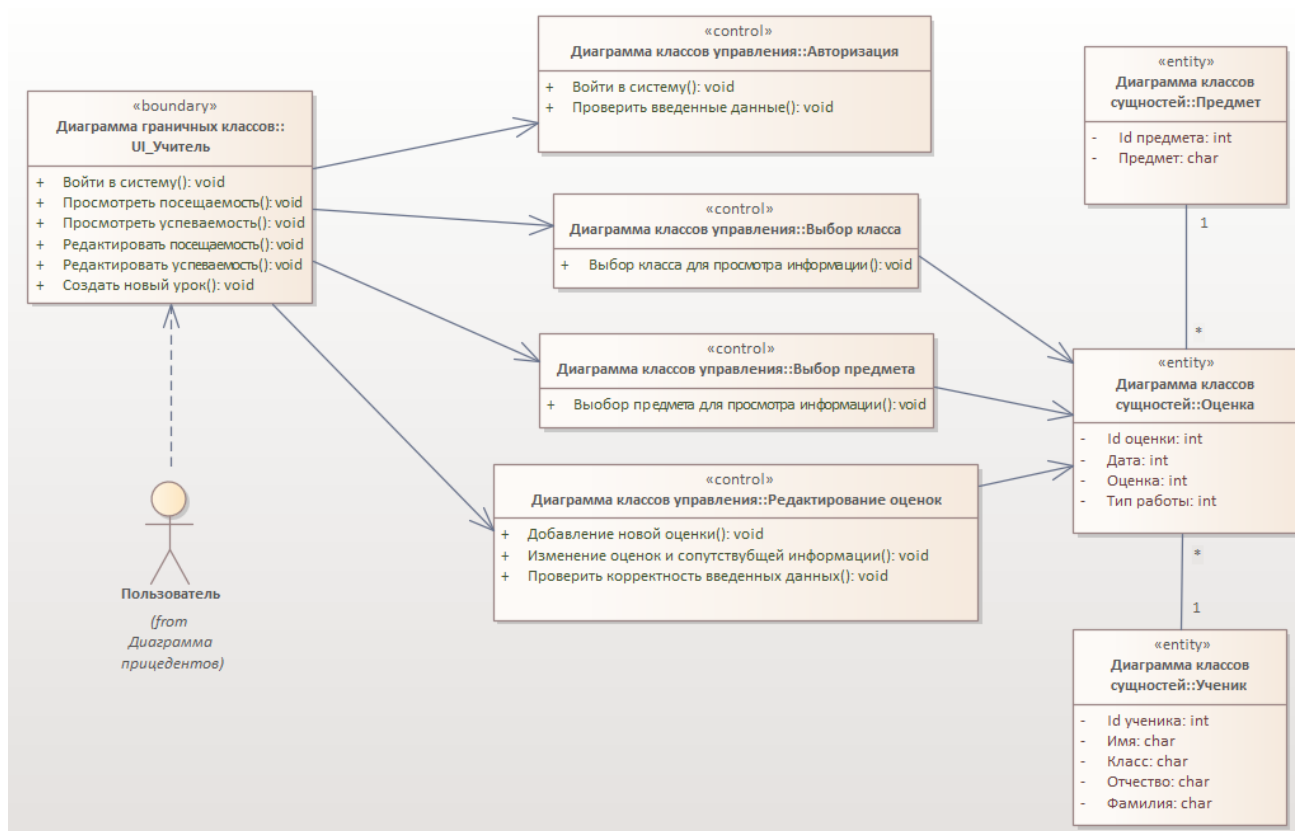


Рисунок 3.3. Диаграмма классов участников кооперации «Просмотр и редактирование оценок»

4. Диаграмма последовательности

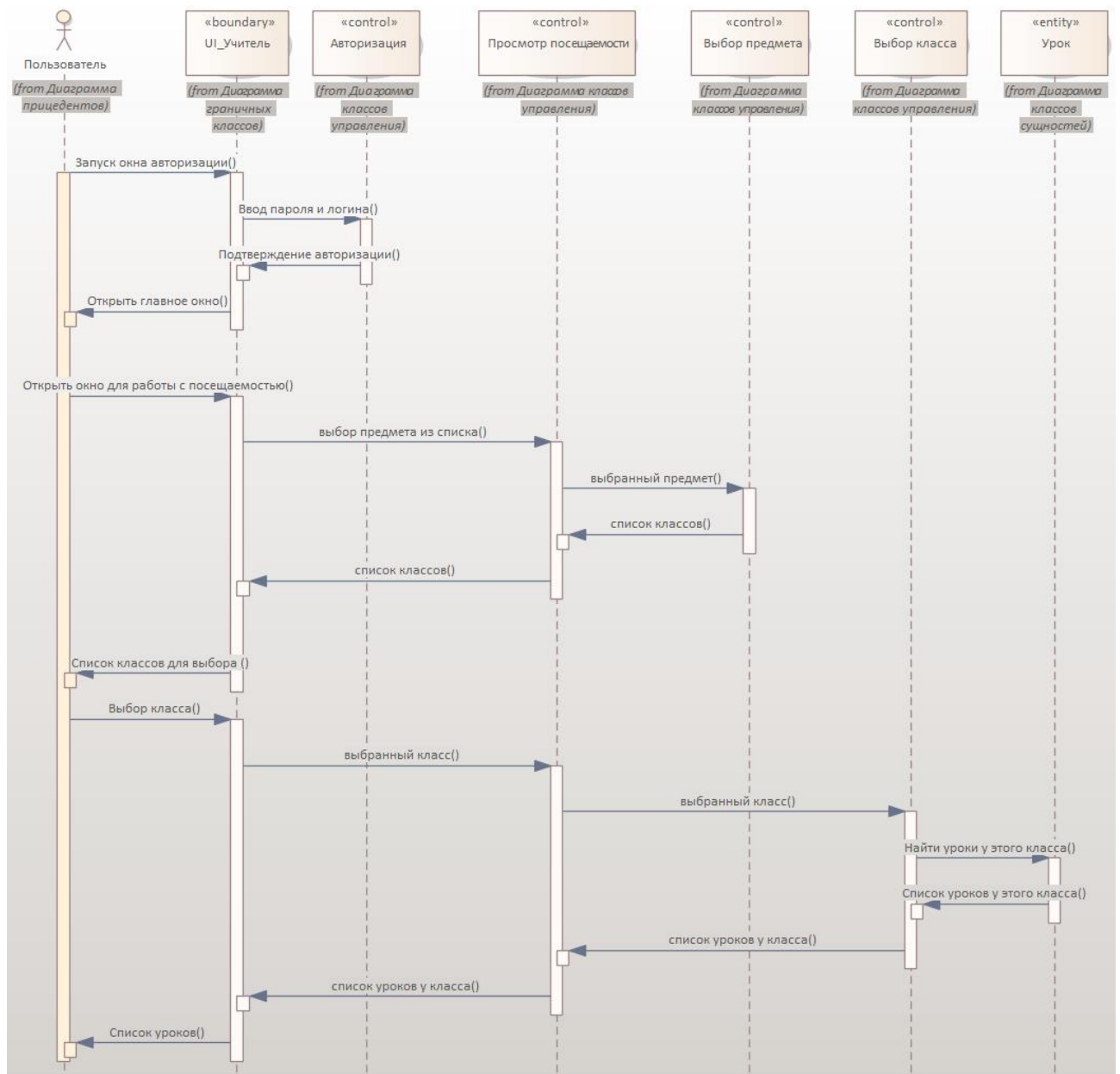


Рисунок 4.1. Обзорная диаграмма последовательности (Просмотр посещаемости)

5. Диаграмма пакетов

Библиографический список

1. Sparx Systems – Текст. Изображение.: электронные // Sparx Systems : [сайт]. – URL: <https://sparxsystems.com/> (дата обращения 15.06.2022)
2. Виноградова, М. В. Унифицированный процесс разработки программного обеспечения: учебное пособие / М. В. Виноградова, В. И. Белоусова. — Москва: Издательство МГТУ им. Н.Э. Баумана, 2015. — 80, [2] с.: ил. ISBN 978-5-7038-4265-2
3. Якобсон А., Дуч Г., Рамбо Дж. Унифицированный процесс разработки программного обеспечения. / А. Якобсон, Г. Дуч, Дж. Рамбо. – Спб.: Питер. – 2002.
4. Арлоу Д., Нейштадт И. UML 2 и Унифицированный процесс. Практический объектноориентированный анализ и проектирование, 2е издание. – Пер. с англ. – СПб: Символ Плюс, 2007. – 624 с., ил. ISBN13: 9785932860946 ISBN10: 5932860944
5. Руководство пользователя Enterprise Architect 15.1 – Текст. Изображение.: электронные // Sparx Systems : [сайт]. – URL: https://sparxsystems.com/enterprise_architect_user_guide/15.1/index/index.html (дата обращения 15.06.2022)
6. Технологии разработки программного обеспечения: Учебник/ С. Орлов. — СПб.: Питер, 2002. — 464 с.: ил. ISBN 5-94723-145-X