

Электронный журнал старосты

Содержание

- **Электронный журнал старосты**
 - **Содержание**
 - **ЛР1**
 - **Название проекта**
 - **Описание идеи проекта**
 - **Описание предметной области**
 - **Краткий анализ аналогичных решени**
 - **Целесообразность и актуальность проблемы**
 - **USE-CASE - диаграмма**
 - **ER-диаграмма**
 - **Архитектурные характеристики**
 - **ЛР2**
 - **Описание типа приложения и выбранного технологического стека**
 - **Верхнеуровневое разбиение на компоненты**
 - **UML диаграмма классов компонента с бизнес-логикой.**
 - **UML диаграмма компонента доступа к данным.**
 - **UML диаграммы «модельных» классов сущностей: сущности базы данных, сущности системы и транспортные сущности.**
 - **ЛР5**
 - **UML-диаграммы классов для компонента/модуля, связывающего GUI и бизнес-логику и компонента/модуля GUI**
 - **UML-диаграмма классов для технологического UI ко всем функциям системы (консольное приложение)**

ЛР1

Название проекта

Электронный журнал старосты

Описание идеи проекта

Заполнение информации о посещениях студентами, формирование отчетов о посещаемости, ведение личных карточек студентов с пользовательскими полями, предоставление студенту интерфейсу о его посещаемости и расписании занятий, возможность отметить на онлайн занятии.

Описание предметной области

Журнал старосты необходим для отметки посещаемости и тем занятий.

Краткий анализ аналогичных решени

Карточка студента - карточка с возможностью добавления произвольных полей

Календарь - календарь со всеми мероприятиями курса

Посещаемость - контроль посещаемости мероприятий

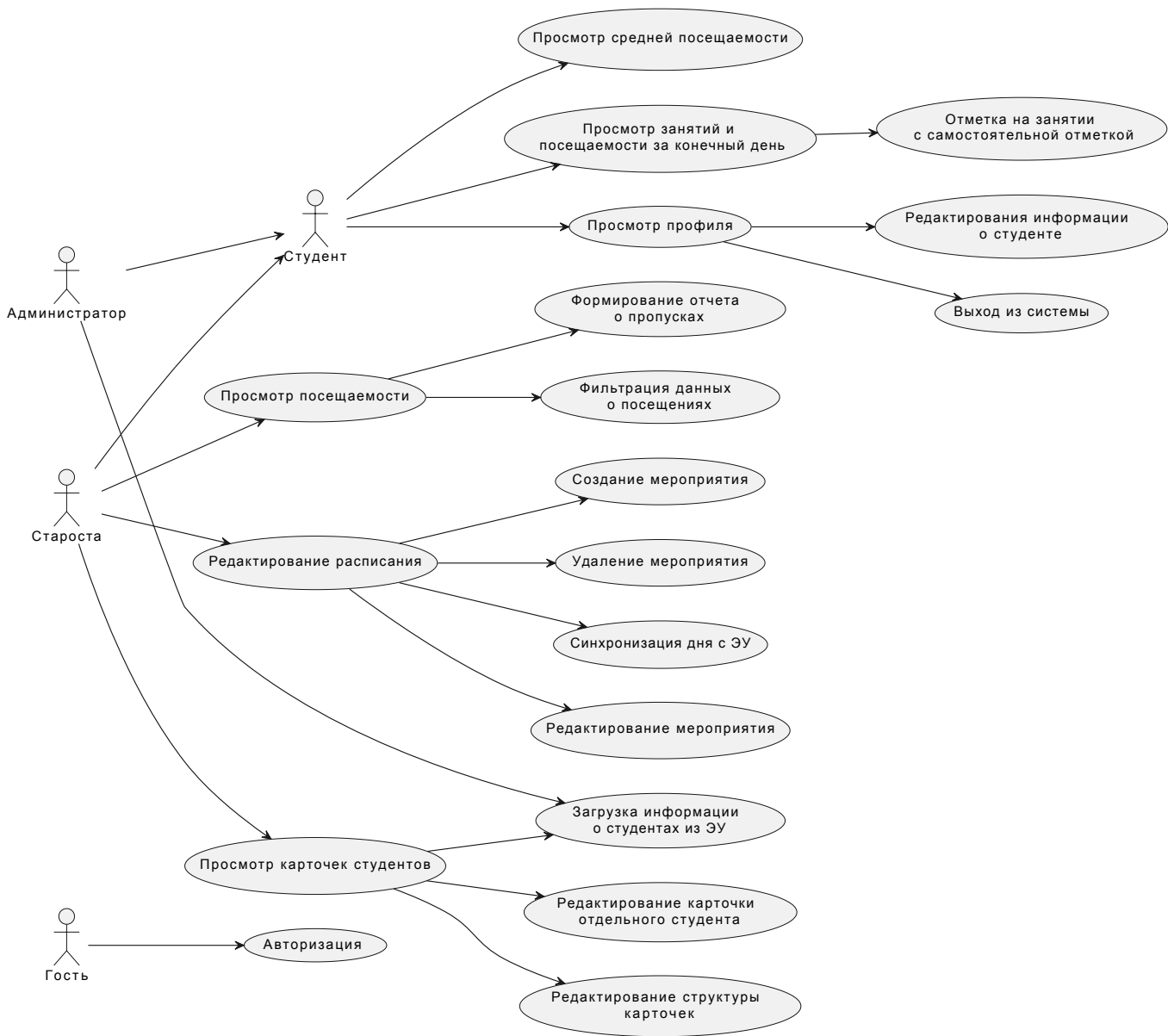
	Карточка студента	Календарь	Посещаемость	Синхронизация с ЭУ
Moodle	+	+	-	Частичная*
ISpring	-	+	-	Частичная*
Miropolis	-	+	+	?

- * интеграция с CAS

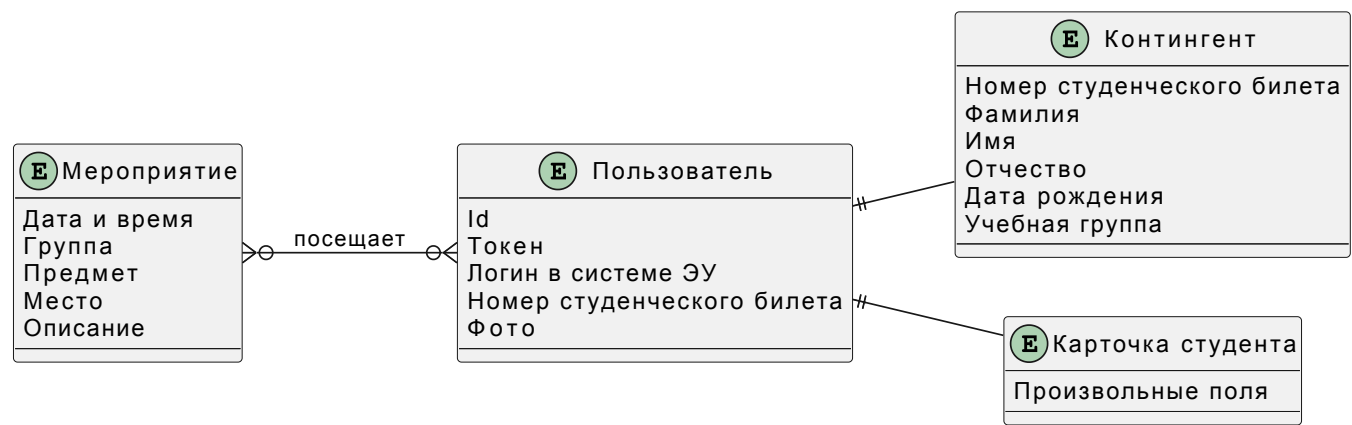
Целесообразность и актуальность проблемы

автоматизации процессов формирования отчетности о посещаемости сокращает время старосты, во время подготовки отчетов для смотров

USE-CASE - диаграмма



ER-диаграмма



Архитектурные характеристики

- предоставление REST API
- авторизация через CAS МГТУ
- поддержка интеграции с ЭУ

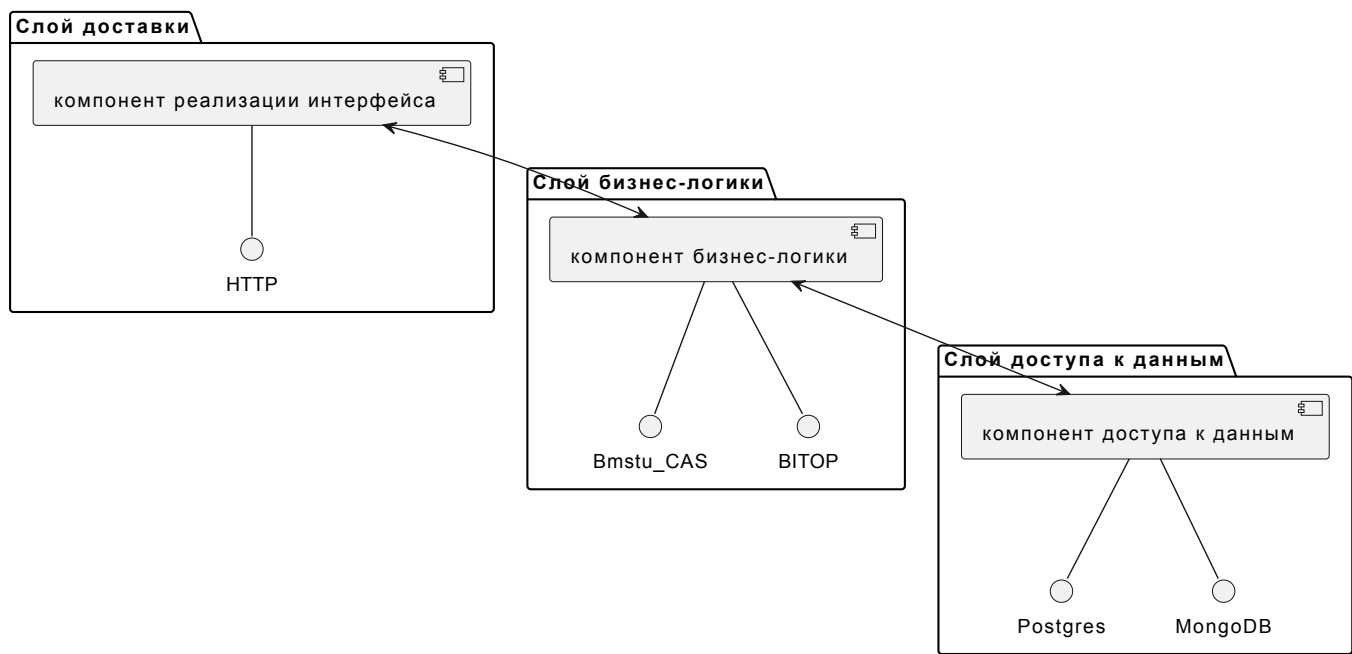
ЛР2

Описание типа приложения и выбранного технологического стека

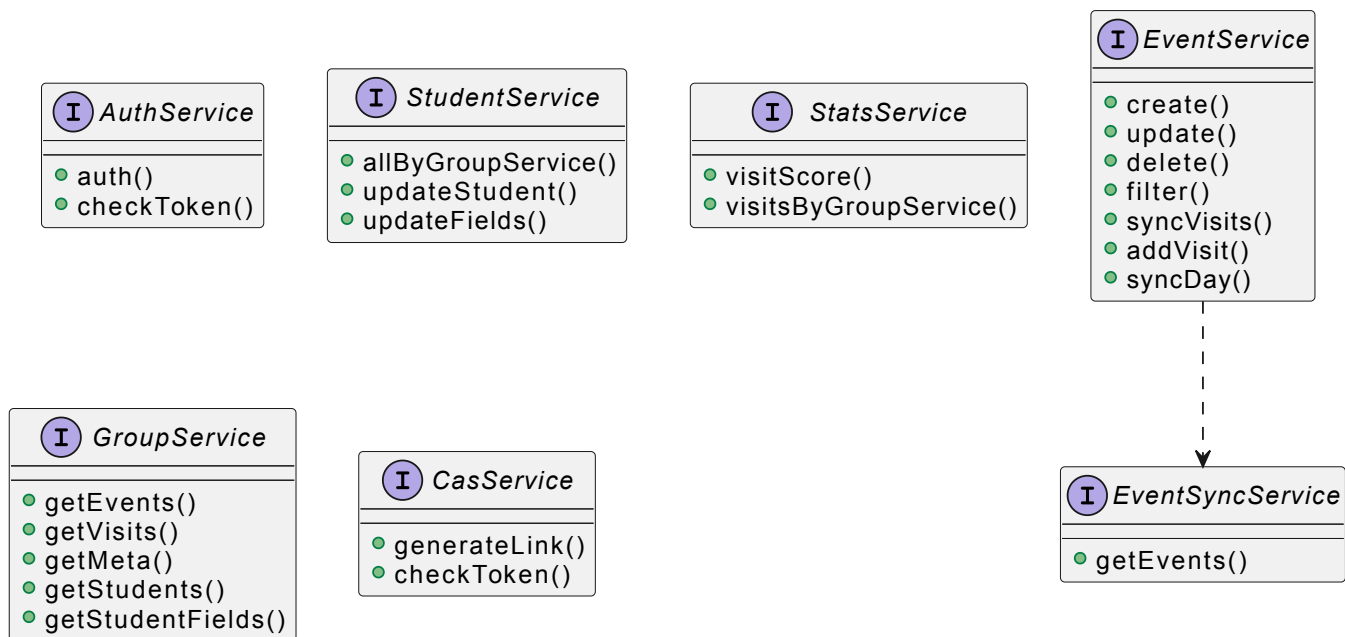
Тип приложения: Backend REST JSON

Стек: Node.js + Express.js + Postgres + MongoDB

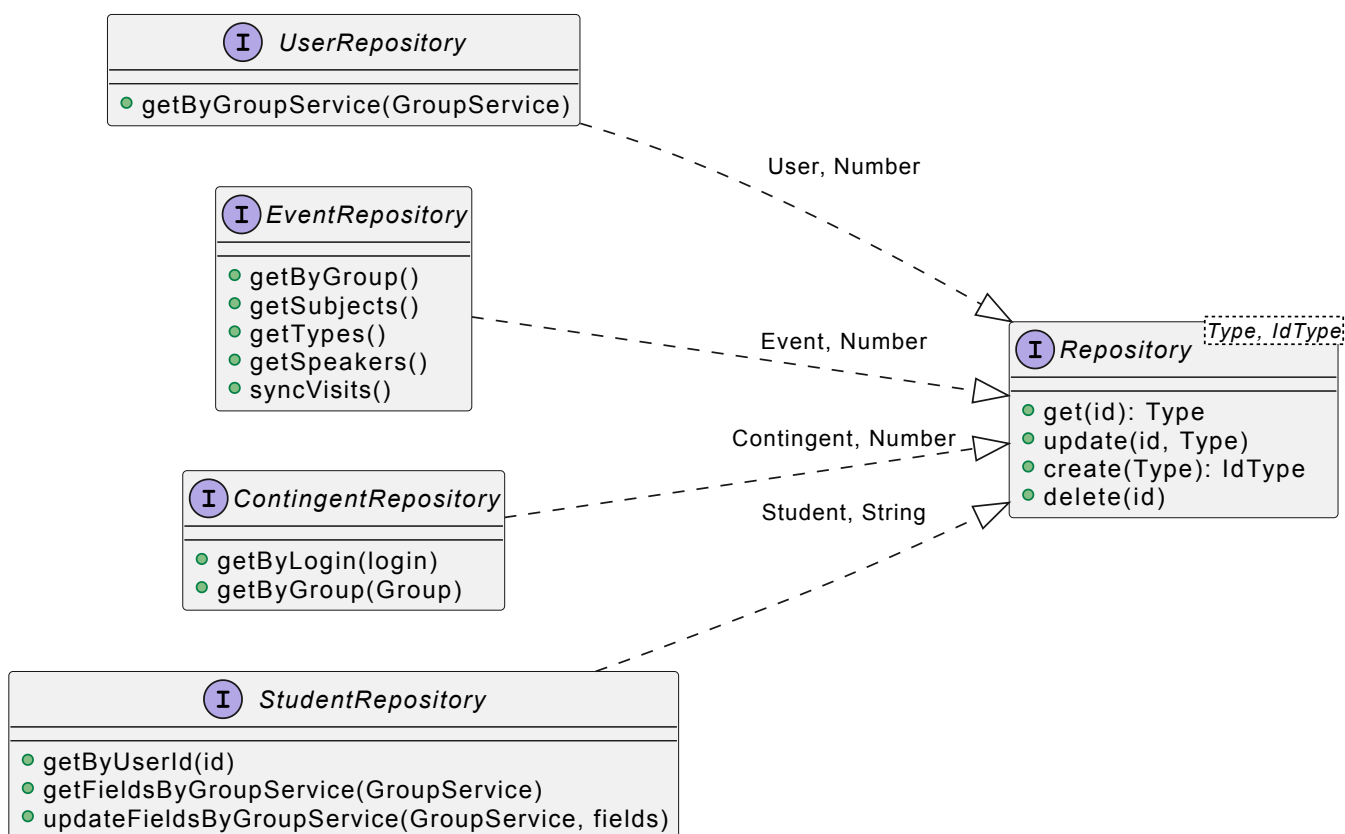
Верхнеуровневое разбиение на компоненты



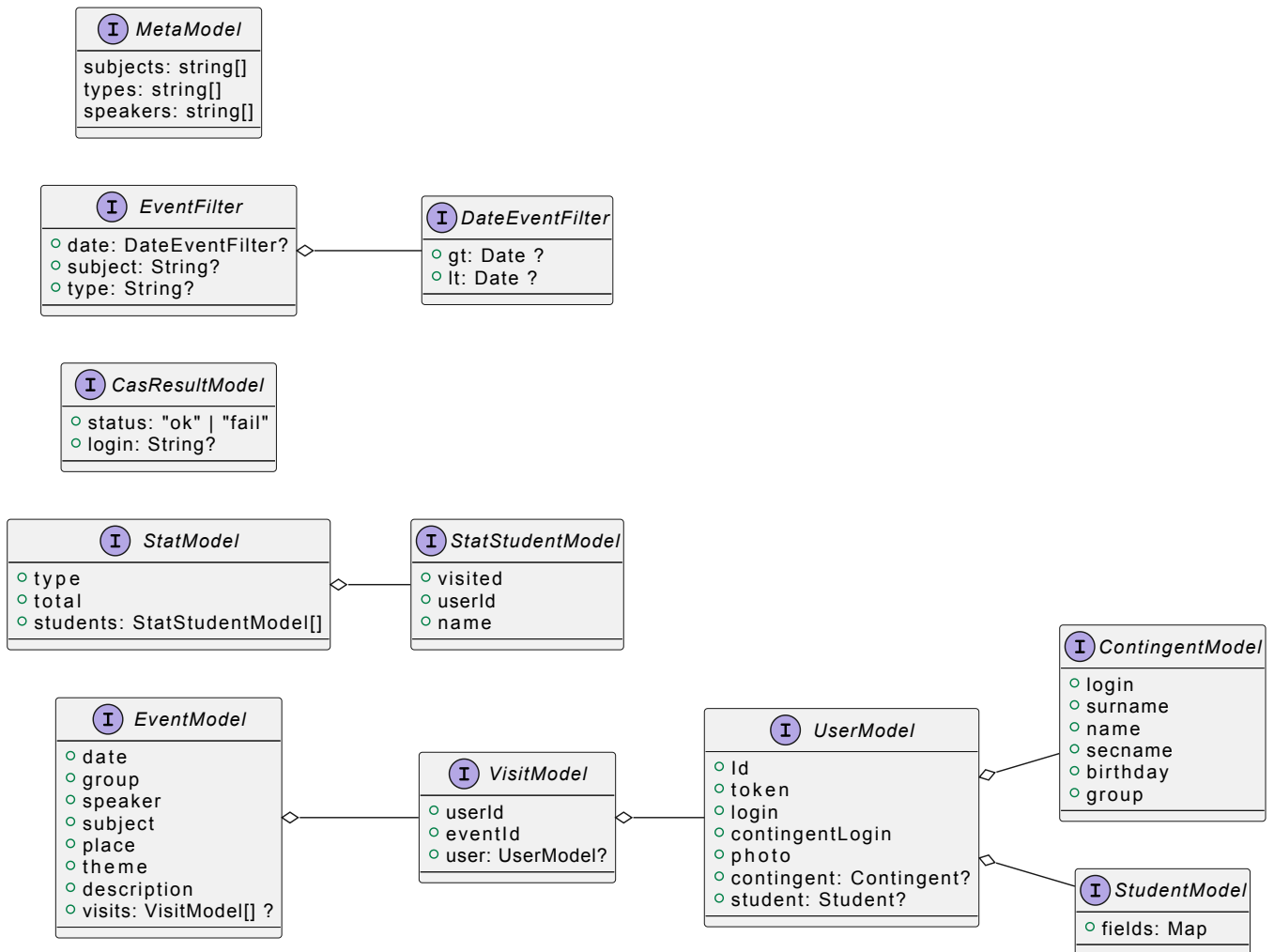
UML диаграмма классов компонента с бизнес-логикой.



UML диаграмма компонента доступа к данным.

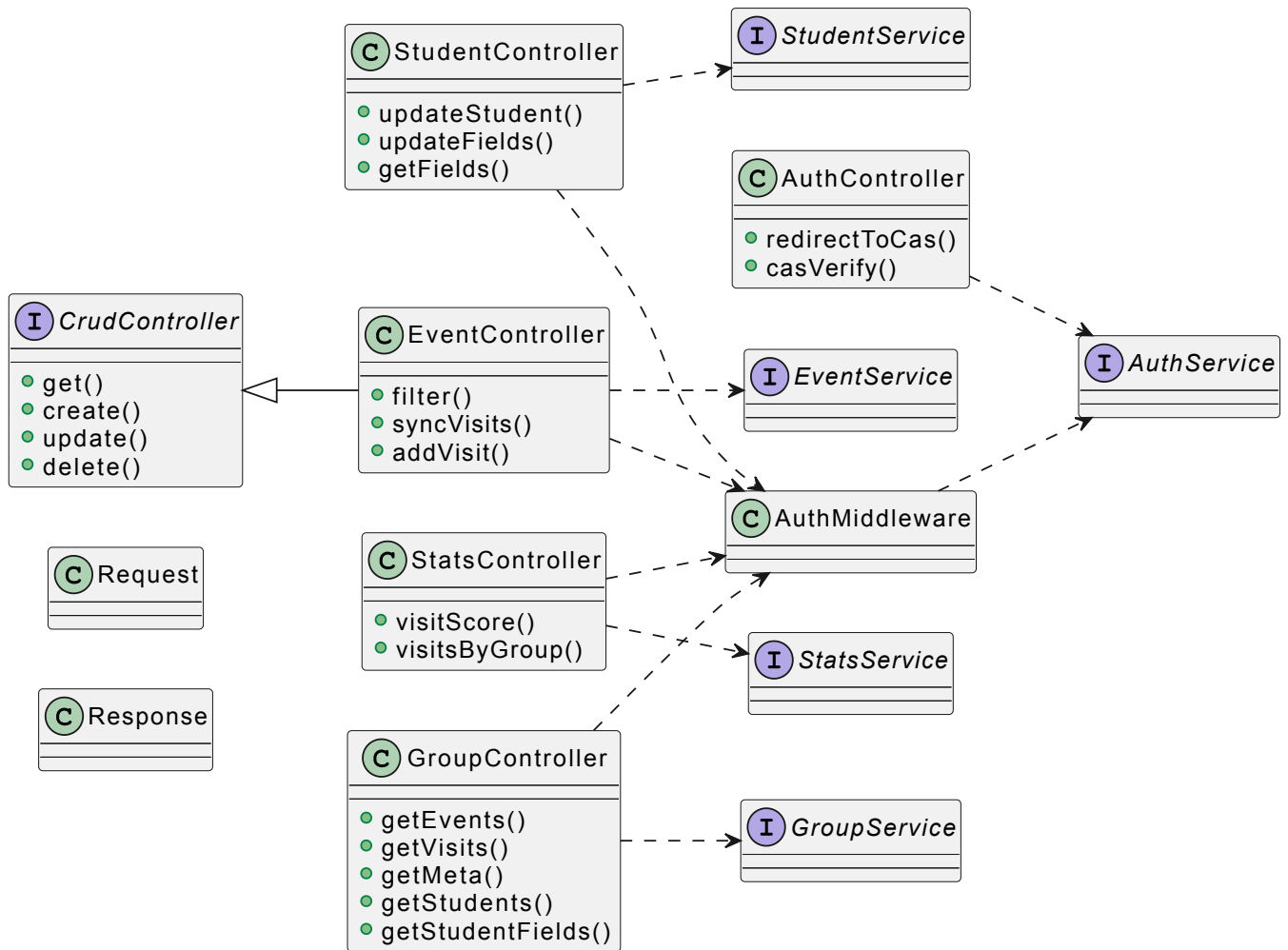


UML диаграммы «модельных» классов сущностей: сущности базы данных, сущности системы и транспортные сущности.



ЛР5

UML-диаграммы классов для компонента/модуля, связывающего GUI и бизнес-логику и компонента/модуля GUI



UML-диаграмма классов для технологического UI ко всем функциям системы
(консольное приложение)