

## ЭТАП 2. ПРОЕКТИРОВАНИЕ ПРОГРАММНОГО РЕШЕНИЯ

Заказчик:	НИУ ВШЭ, Международная лаборатория интеллектуальных систем и структурного анализа, Паринов Андрей Андреевич
Название проекта:	Клиент-серверное iOS приложение Ассистента Студента
Исполнители	Зубарева Наталия Дмитриевна, Мостачев Андрей Олегович, Поволоцкий Виктор Александрович, Сальникова Алиса Дмитриевна

### 1) Проектирование основных составляющих (данные, классы, клиентские компоненты/серверные компоненты): статические диаграммы

#### Модель данных

Эта диаграмма показывает структуру данных, с которой работает серверная база данных. Здесь отражены знания, которые приложение хранит о проектах и студентах и связи между ними. Основными сущностями являются студент и проект, однако для нормализации данных и разбиения связей многие ко многим вводятся дополнительные таблицы, отражающие эти связи. Также отдельные таблицы существуют для терминов технических средств проектов и областей учебных интересов студентов. Таблица, содержащая основную информацию о студентах, разделена на две - одна с информацией ID и корпоративного email для авторизации в ЕЛК ВШЭ и одна с прочими личными данными, которые могут быть пустыми и заполняться по мере пользования студента приложением.

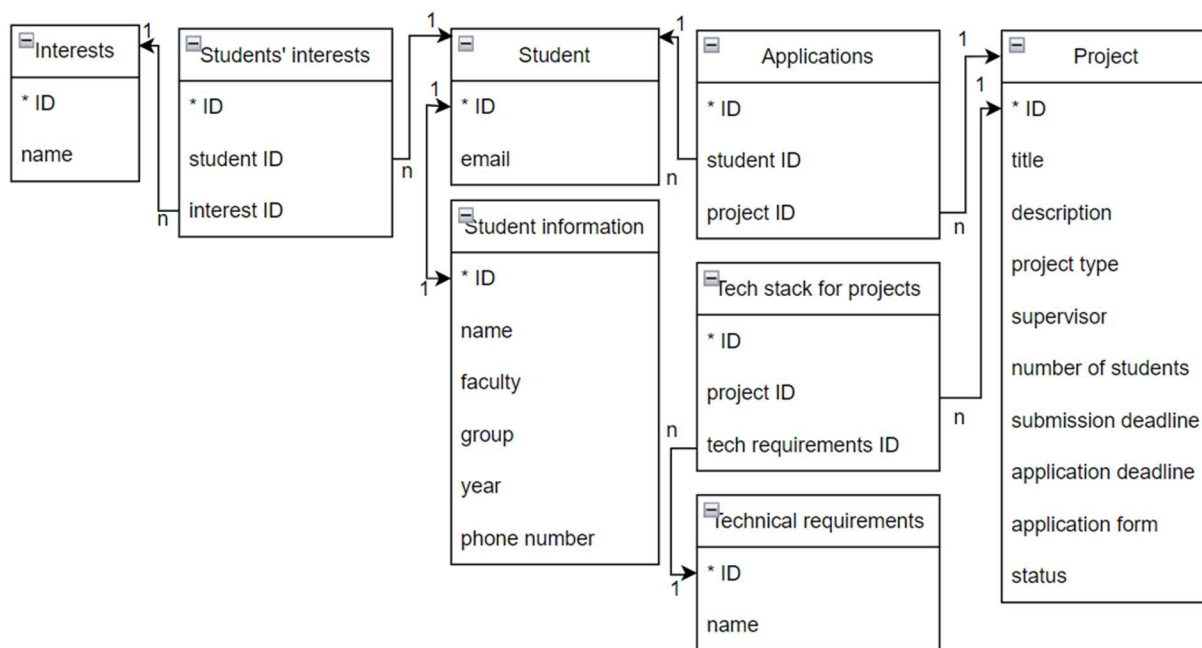


Рис. 1 - модель данных приложения

#### Диаграмма классов

Диаграмма классов иллюстрирует классы, с которыми приложение работает в клиентской части: класс проекта, участника команды (студента или преподавателя), и т.д. Также здесь показаны агрегационные связи между классами и их зависимости друг от друга.

Среди классов также можно найти разделение на реализующие вью, модель, координатор и сервисы, с помощью которых клиентская часть получает информацию от серверной.

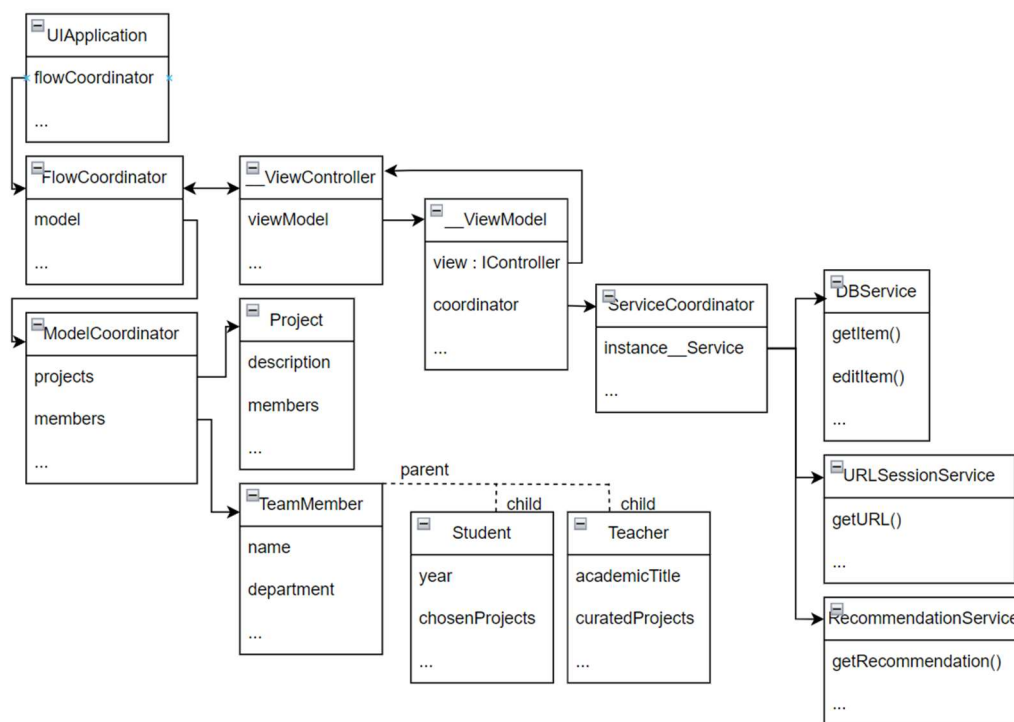


Рис. 2 - диаграмма классов приложения

### Диаграмма компонентов

Диаграмма компонентов отражает смысловые категории элементов в коде приложения и структуры, которые в нем взаимодействуют, такие как модель, отображение, связи между ними и то, как информация проходит между ними при работе пользователя в приложении.

Пользователь взаимодействует с прослойкой View, затем информация оттуда переходит в модель, которая координируется с серверной частью.

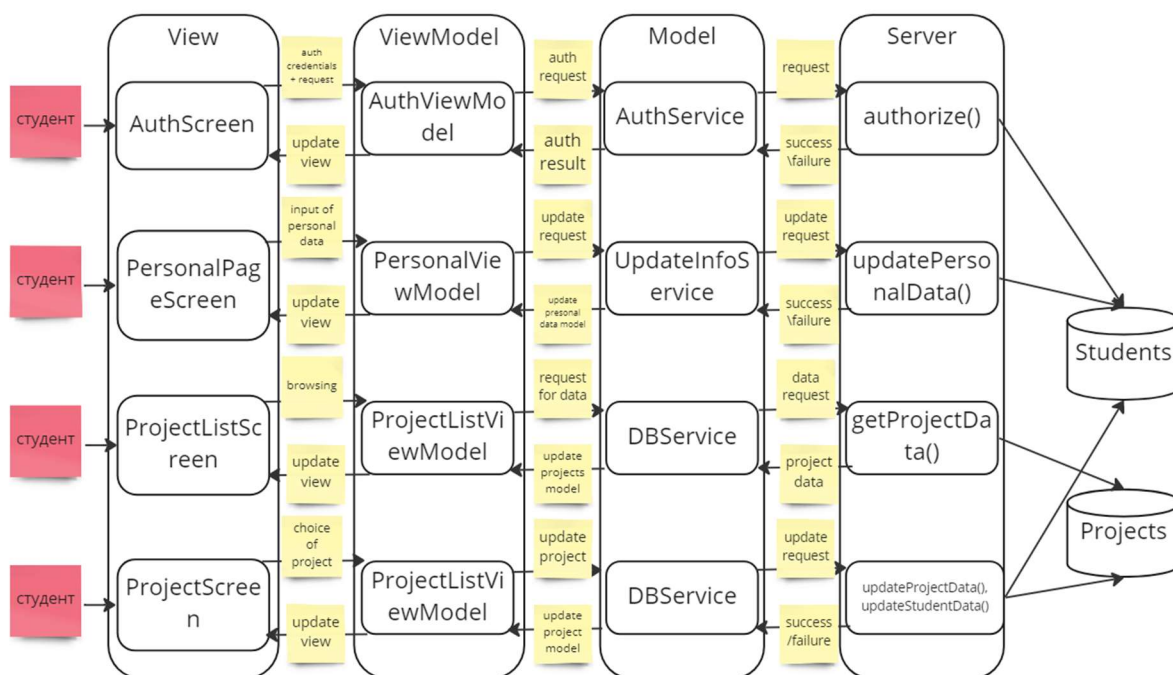


Рис. 3 - диаграмма компонентов приложения

## 2) Описание динамических процессов: диаграммы использования системы, показывающие взаимодействие между воздействующими объектами и функциями

### Таблица прецедентов

Диаграмма прецедентов описывает состояния и события в системе в связке с условиями их возникновения и данными, которые при этом задействованы. Отражает смысловые категории элементов в коде приложения и структуры, которые в нем взаимодействуют, такие как модель, отображение, связи между ними и то, как информация проходит между ними при работе пользователя в приложении.

Возможные направления действия пользователя (прецеденты):

- авторизация;
- заполнение личных данных;
- просмотр проектов;
- просмотр конкретного проекта;
- регистрация на проект;
- запрос рекомендаций;

Номер требования	Требование	Действующие лица	Прецеденты использования
1	Программа должна позволять пользователю авторизоваться в системе через ЕЛК ВШЭ	Student Data, Auth Service	авторизация
2	Программа должна выводить сообщение о неверных данных при неудачной авторизации	View, Auth Service, View Controller	авторизация
3	Программа должна отображать данные пользователя в личном кабинете	View, DB Service, Student Data	заполнение личных данных
4	Программа должна позволять пользователю заполнять и редактировать персональные данные	View, DB Service, Student Data	заполнение личных данных
5	Программа должна сохранять и обновлять измененные данные пользователя	View, DB Service, Student Data	заполнение личных данных
6	Программа должна узнавать у пользователя его научные интересы	View, Student Data	заполнение личных данных
7	Программа должна создавать рекомендации проектов на основе научных интересов пользователя	Student Data, Recommendation Model	запрос рекомендаций
8	Программа должна предоставлять рекомендации проектов пользователю по запросу	View, Recommendation Service, Student Data	запрос рекомендаций

9	Программа должна позволять пользователю просматривать все доступные в системе проекты	View, DB Service, Project Data	просмотр проектов
10	Программа должна позволять пользователю настраивать отображение проектов с помощью фильтров	View, DB Service, Project Data	просмотр проектов
11	Программа должна периодически (раз в 1 час или по запросу пользователя) синхронизировать информацию о проектах и участников с базой данных	DB Service, Project Data, Student Data	просмотр проектов
12	Программа должна предоставлять пользователю подробную информацию о каждом проекте по запросу пользователя	View, DB Service, Project Data	просмотр конкретного проекта
13	Программа должна позволять пользователю зарегистрироваться на проект и отменить существующую регистрацию	View, DB Service, Project Data	регистрация на проект

### Data-flow-диаграмма

Data-flow-диаграмма показывает маршрут данных при различных действиях пользователя и информацию, которой при этом обмениваются системы. Пользователь начинает свой путь в приложении с авторизации и проходит несколько стадий работы перед регистрацией на проект, что является финальной целью приложения.

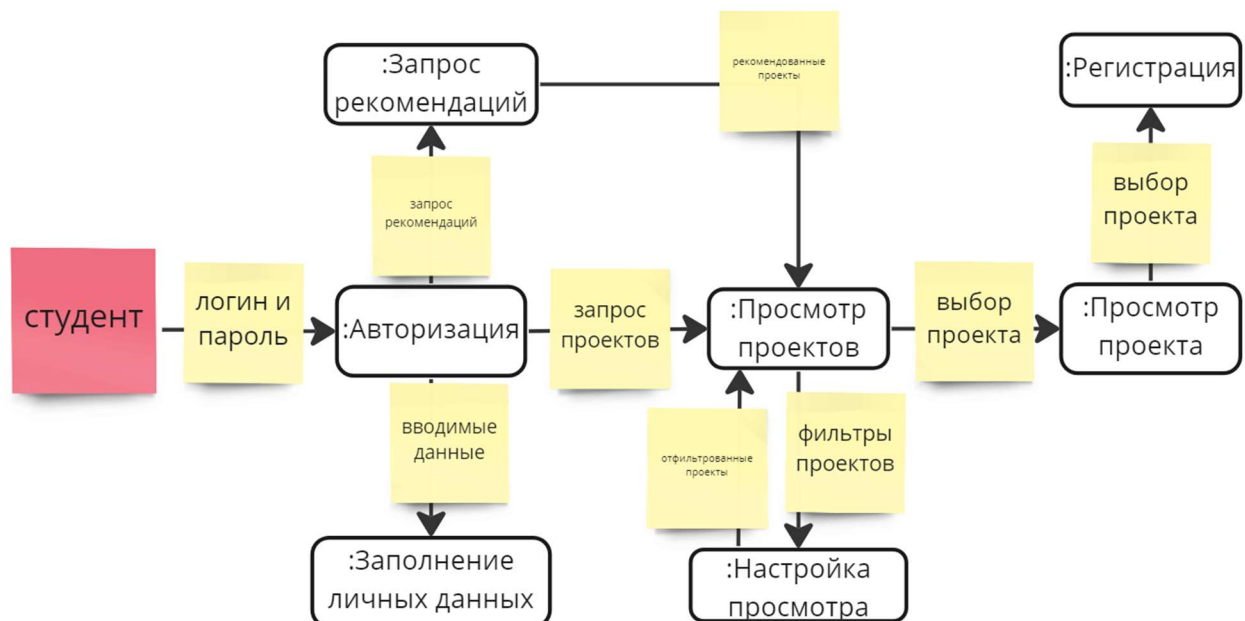


Рис. 4 - data-flow диаграмма

## Sequence-диаграмма

Диаграмма последовательности описывает то, как динамически при работе пользователя в приложении осуществляются его функции. Как можно видеть, информация переходит от пользователя через интерфейс в координирующие компоненты, откуда синхронизируется с базой данных и другими элементами серверной части (например, рекомендательной модели, которая создает рекомендации проектов для пользователя по запросу).

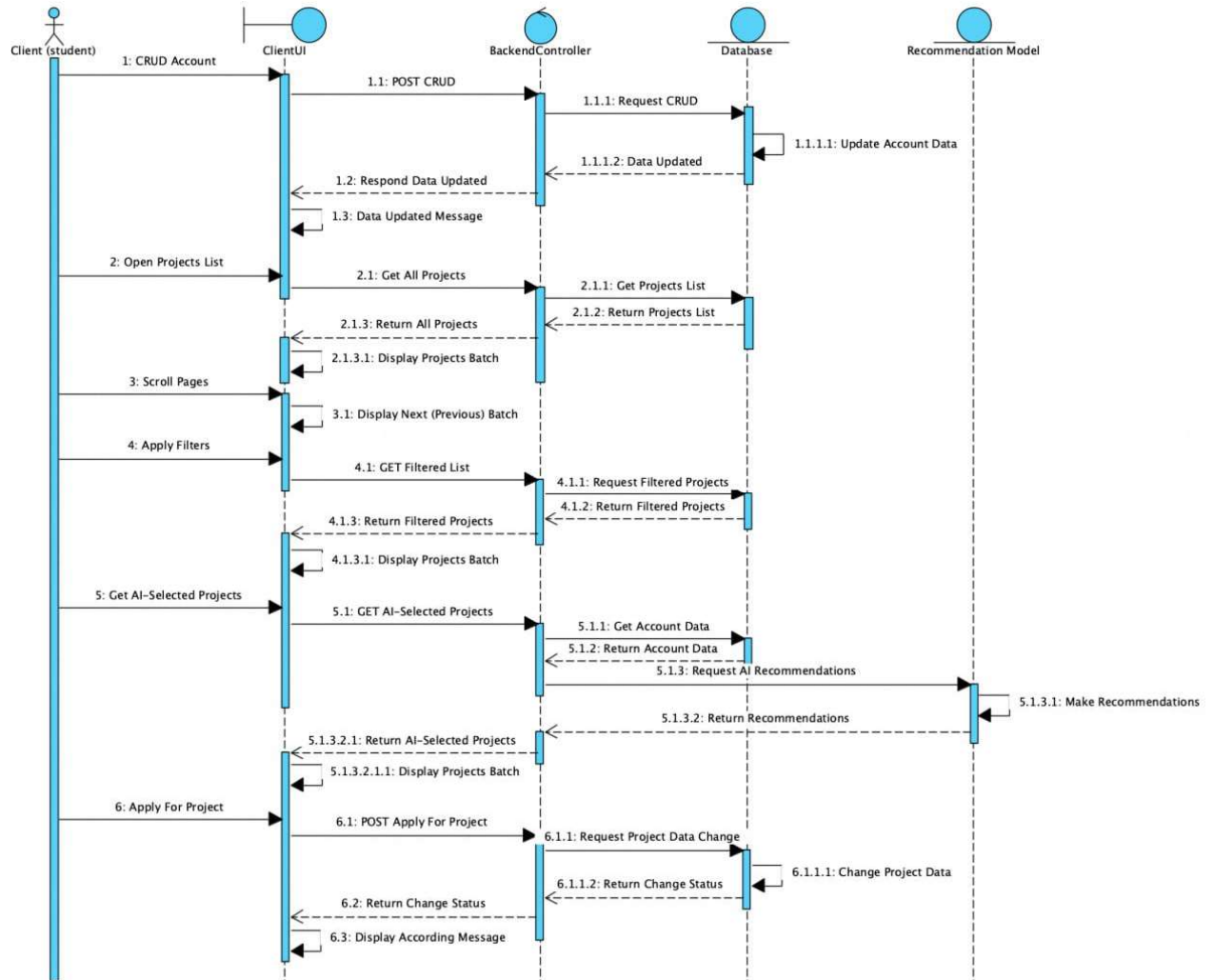


Рис. 5 - sequence-диаграмма

Заказчик

Личная подпись

/Паринов А.А

Ответственный по проекту

Личная подпись

/Зубарева Н.Д.

Дата: 15.01.2023