**КУРСОВАЯ РАБОТА**

Анализ основных этапов работы над проектом

по дисциплине «Конструирование ПО»

Выполнили:

Московская П. А. гр. 3530904/80102

Шредер А. Д. гр. 3530904/80102

Лукина В. А. гр. 3530904/80101

Принял:

Иванов А. С.

Оглавление

[Проблема 3](#_Toc59050311)

[Требования 4](#_Toc59050312)

[Разработка архитектуры и детальное проектирование 5](#_Toc59050313)

[1. System Context diagram 5](#_Toc59050314)

[2. Container diagram 6](#_Toc59050315)

[Кодирование и отладка 7](#_Toc59050316)

[1. Серверная часть 7](#_Toc59050317)

[2. Клиентская часть 8](#_Toc59050318)

[Тестирование 9](#_Toc59050319)

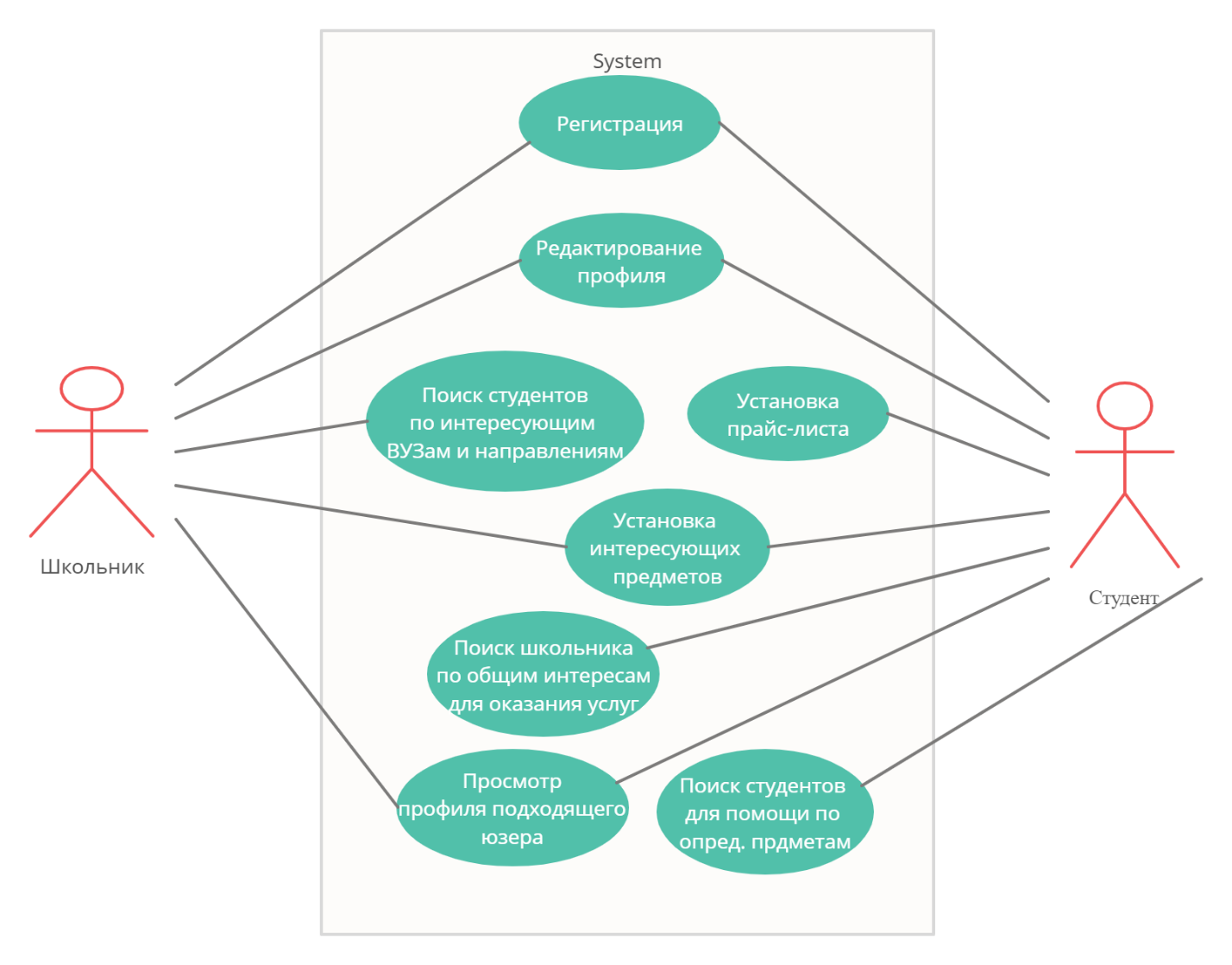
[Команды запуска и тестирования 10](#_Toc59050320)

Проблема

Проект решает проблему связи абитуриентов и студентов университета. В популярных социальных сетях сложно найти человека, готового проконсультировать школьника по поводу интересующего его направления. Проект позволяет школьникам и абитуриентам узнать необходимую информацию от людей из конкретных университетов, обучающихся по конкретным образовательным программам.

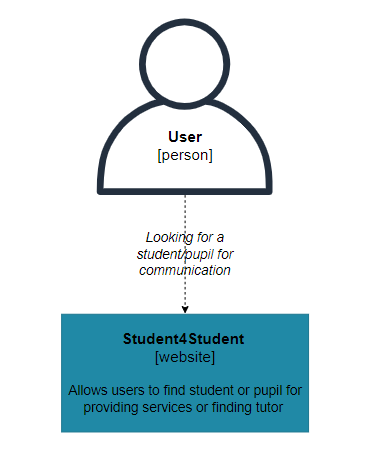
Требования

Требования представлены в формате UML-диаграммы

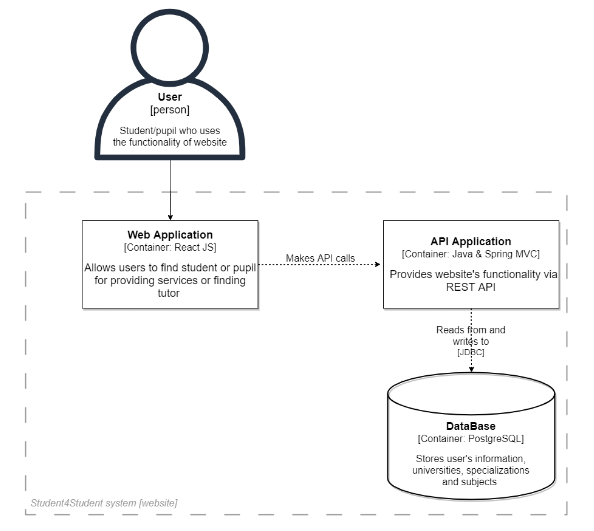


Разработка архитектуры и детальное проектирование

1. System Context diagram



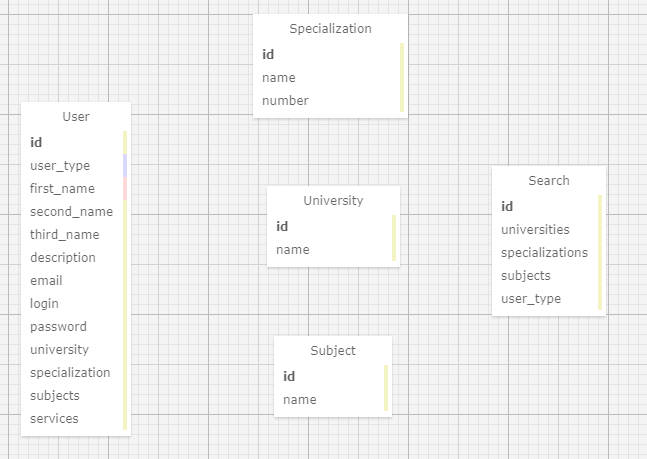
1. Container diagram



Кодирование и отладка

1. Серверная часть

Серверная часть сайта написана на языке Java, был также использован фреймворк Java Spring Framework. В качестве системы управления базами данных был выбран PostgreSQL. Схема базы данных представлена ниже:



Сначала были описаны необходимые сущности, представляющие собой таблицы в базе данных. Далее для доступа к информации(строкам) были описаны репозитории - дополнительный удобный механизм для взаимодействия с сущностями базы данных, извлечение данных, изменение. Затем были реализованы REST контроллеры для обработки запросов.

Для отладки обработки запросов была использована программа Postman.

1. Клиентская часть

Клиентская часть реализована с помощью JavaScript-библиотеки с открытым исходным кодом для разработки пользовательских интерфейсов React. Для настройки окружения разработки проекта и для компиляции JavaScript-модулей используется Webpack. Для управления состоянием приложения используется библиотека Redux. Также используется транспилятор Babel для работы с JSX и для преобразования конструкций, принятых в свежих версиях стандарта ECMAScript, в вид, понятный как современным, так и не самым новым браузерам и другим средам, в которых может выполняться JavaScript. Настройка маршрутизации осуществляется с помощью библиотеки react-router.

Тестирование

Для серверной части были написаны Unit-тесты для тестирования всех эндпоинтов, для чего была использована мок-библиотека Mockito и фреймворк Spring MVC Test.

Интеграционные тесты были написаны для основных пунктов из требований: регистрация пользователя, редактирование профиля, добавление интересующих предметов, поиск подходящих студентов или старшеклассников. Для тестирования API был использован TestRestTemplate, а также дополнительная база данных PostgreSQL.

Для клиентской части были реализованы тесты с использованием фреймворка Jest

Команды запуска и тестирования

1. Серверная часть

Сборка: gradle build

Запуск: gradle bootRun

Запуск тестов: gradlew test (интеграционных – gradle test - -tests IntegrationTesting)

1. Клиентская часть

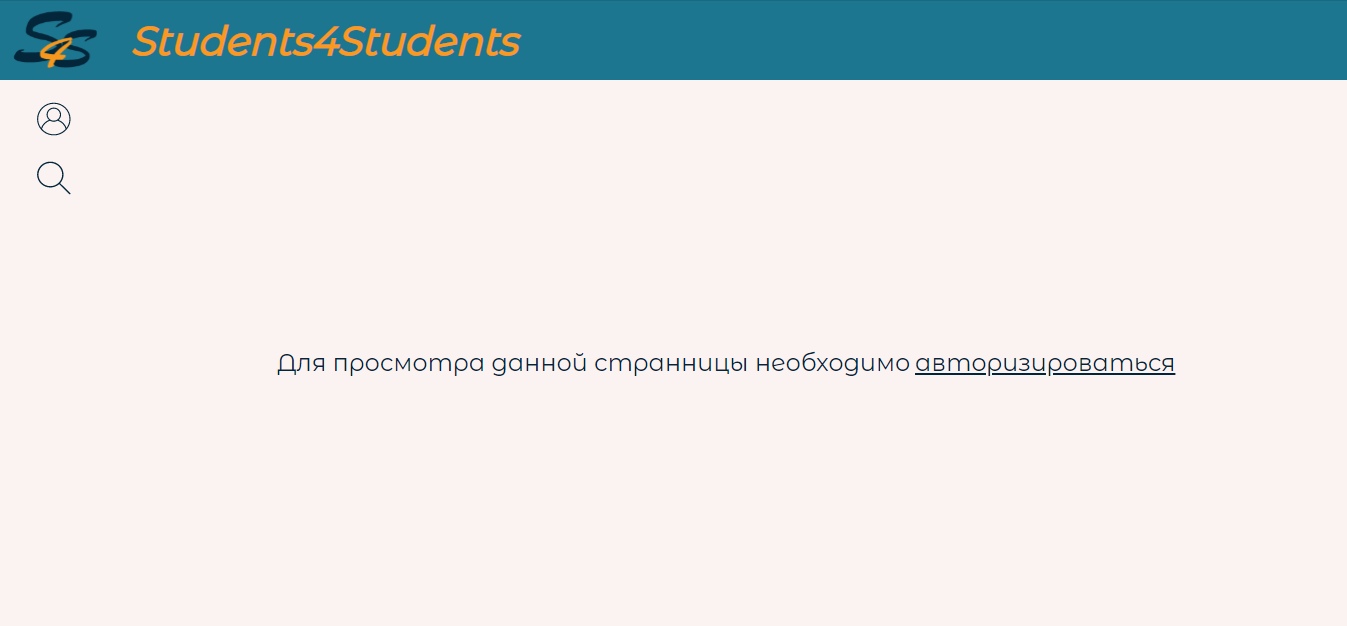
Сборка: npm build

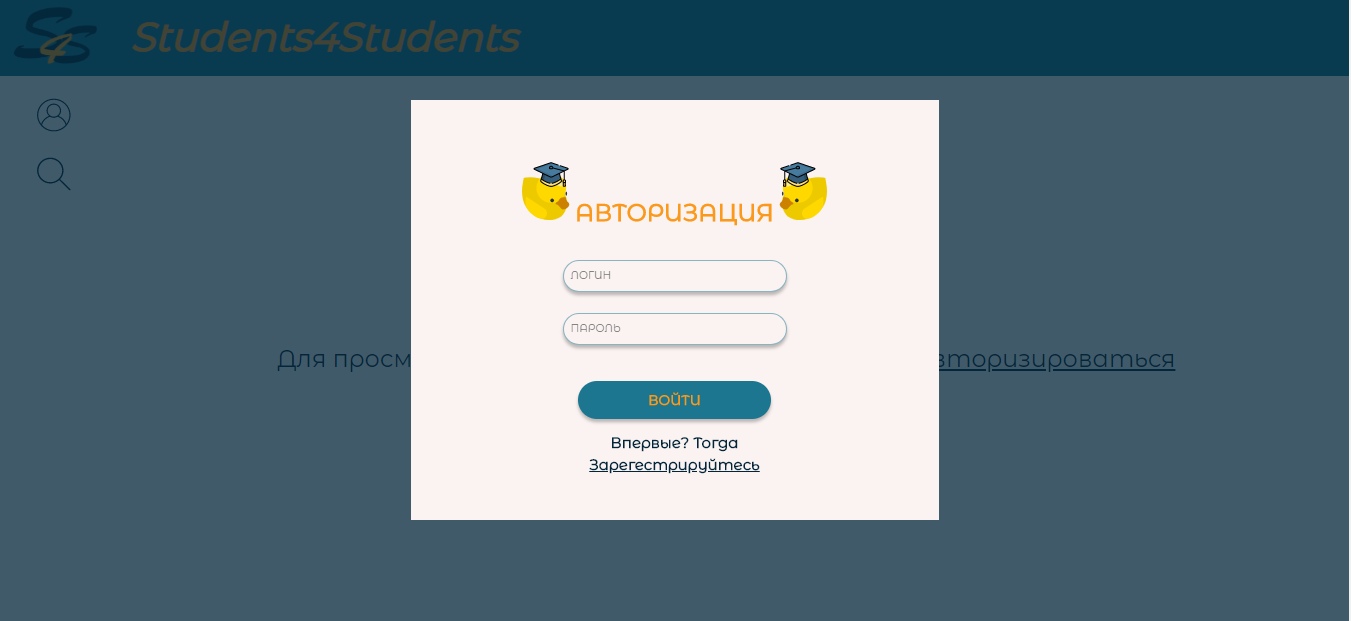
Запуск: npm start

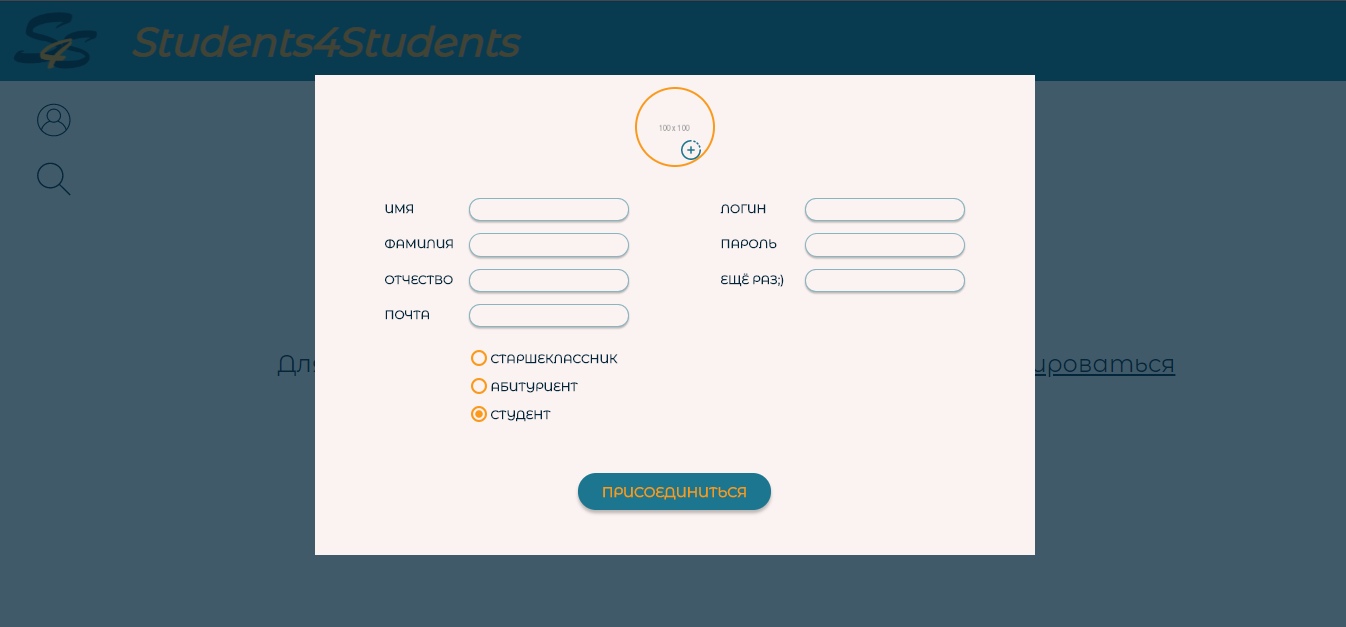
Запуск тестов: npm test

Скриншоты









Изображение выглядит как текст

Автоматически созданное описание

