Грачев Даниил ПРИ-120

**Цель**: В ходе выполнения курсового проекта требуется спроектировать и разработать программную систему «Университет» для автоматизации взаимодействия университета с пользователями.

**Описание предметной области**: В университете, обучающем студентов и осуществляющем набор абитуриентов, важно иметь программную систему, обеспечивающую:

* автоматизацию рейтинговых списков абитуриентов
* сдачу лабораторных работ в электронной форме и контрольных работ в онлайн формате с автоматическим указанием баллов
* новостную ленту с возможностью написания статей любым пользователем программной системы

**Функциональные и не функциональные требования**

Функциональные требования:

* Зарегистрировать аккаунт
* Войти в аккаунт
* Изменить данные о пользователе
* Пройти тест
* Выйти из аккаунта
* Присвоить пользователю роль
* Оставить заявку на обратную связь
* Отображение информации о ближайшем дне открытых дверей

Нефункциональные требования:

* Для доступа к настройкам пользовательских настроек не должно затрачиваться больше трех нажатий
* Для более легкого понимания роли пользователя, имя пользователя должно отображаться разными цветами
* Реализация серверной части на Java с использованием технологии Spring
* Взаимодействие интерфейса и сервера с помощью API
* Реализация интерфейса с помощью библиотеки React.js
* Использование СУБД MySQL

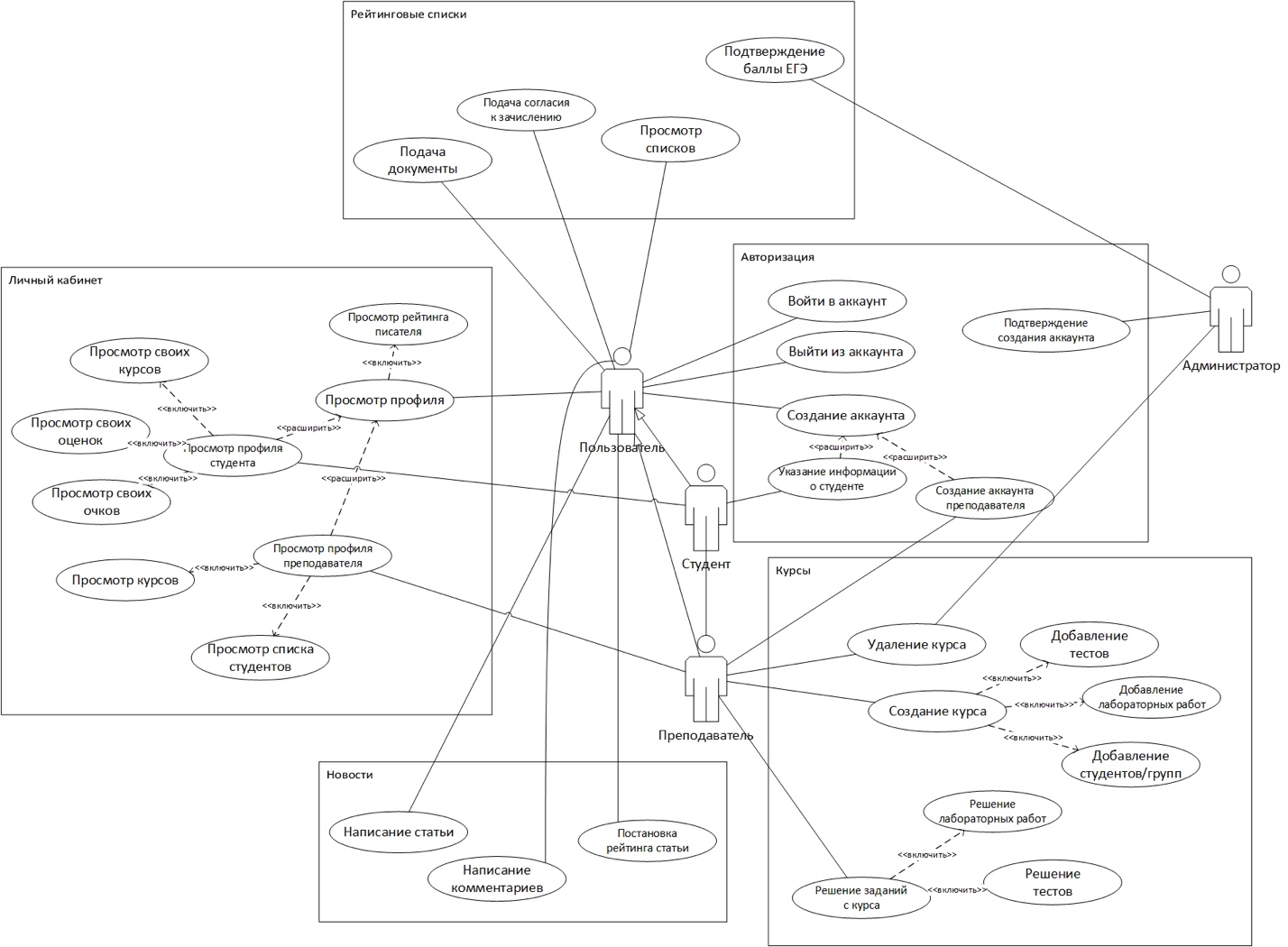
**Словарь предметной области:**

1. *Курс* – часть программной системы, с помощью которой «Преподаватель» может создавать задания для «Студента», такие как контрольные работы
2. *Новостная лента* – часть программной системы, обеспечивающая возможность написания новостей любым пользователем, так же любой прочитавший новость может поставить оценку прочитанному материалу и оставить комментарий
3. *Рейтинговые списки* – часть программной системы, обеспечивающая автоматизацию работы с абитуриентами
4. *Личный кабинет* – часть программной системы, внутри которой пользователь может увидеть данные о своем аккаунте, такие как количество очков, курсы, если это аккаунт студента или преподавателя
5. *Студент* – роль пользователя в системе, являющегося студентом в университете, может выполнять тесты и лабораторные работы с курса
6. *Преподаватель* – роль пользователя в системе, являющегося преподавателем в университете, может создавать и редактировать курсы
7. *Тест* – часть курса, задание, в ходе которого студенты должны отвечать на поставленные вопросы с данными вариантами ответов
8. *Группа* – выделенная часть студентов, объединённая одной специальностью
9. *Обращение* – заполненная пользователем форма обратной связи

**Пользователи системы:**

1. *Администратор* – сотрудник университета, который администрирует работу программной системы, подтверждает некоторые действия пользователей
2. *Пользователь* – зарегистрированный участник системы, который может полностью взаимодействовать с подсистемой новостей и подавать документ на поступление в университет
3. *Студент* – зарегистрированный участник системы, который владеет всеми возможностями обычного пользователя, так же может взаимодействовать с подсистемой курсов
4. *Преподаватель* – зарегистрированный участник системы, который обладает всеми возможностями обычного пользователя, так же может взаимодействовать с подсистемой курсов

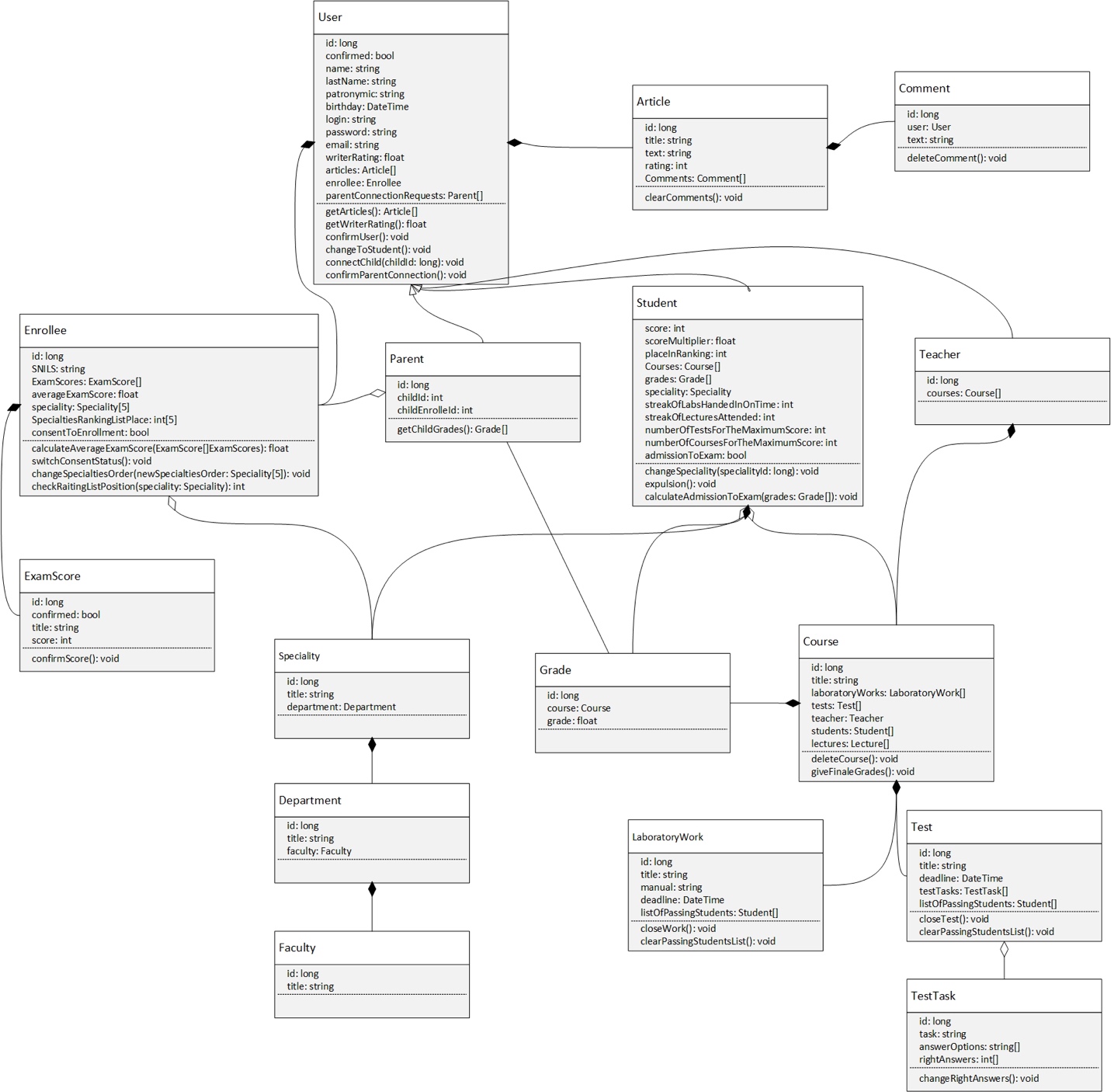
**Диаграмма прецендентов**



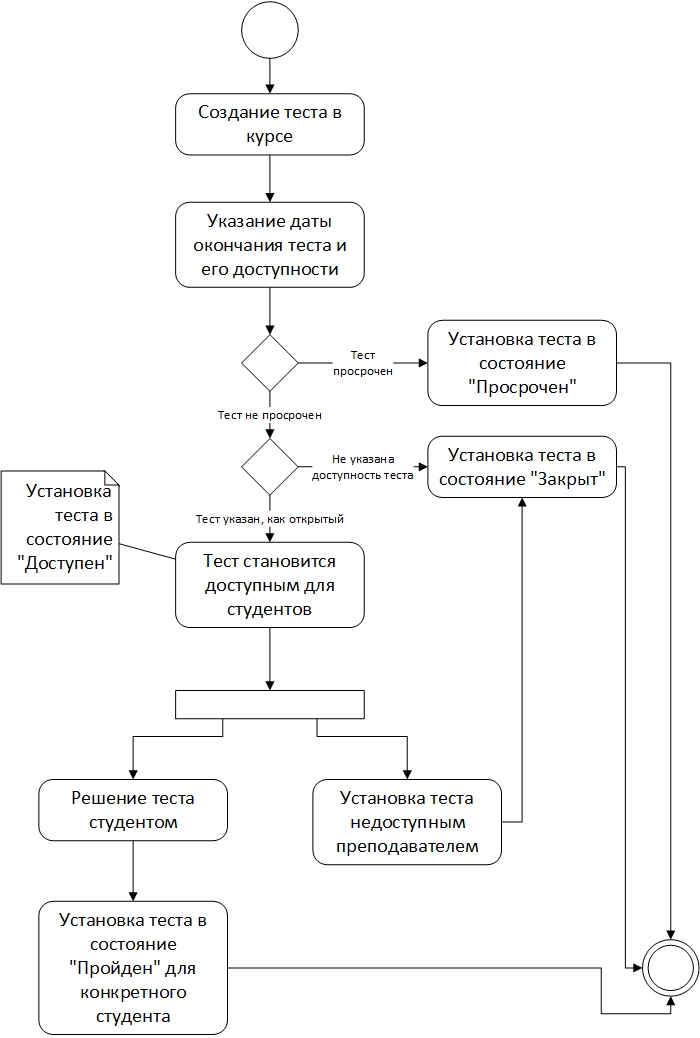
Описание прецендентов:

|  |  |
| --- | --- |
| Раздела спецификации прецендента | Описание |
| Название | Создание аккаунта |
| Актер | Не зарегистрированный пользователь |
| Предусловия | Пользователь заходит на сайт, нажимает кнопку войти и переходит на вкладку регистрации |
| Основной поток | Пользователь заполняет форму регистрации, состоящую из шести полей: логин, пароль, электронная почта, имя, фамилия, отчество. При нажатии на кнопку «Зарегистрироваться» из браузера отправляется два запроса на наличие в системе пользователя с таким логином и такой почтой, если оба запроса возвращают отрицательный ответ, значит такого пользователя нет в системе, значит можно создать, отправляется еще один запрос на создание пользователя, ему присваивается дата регистрации равная сегодняшней дате и времени, а также в поле роли устанавливается пустая строка, после этого новый пользователь может войти в аккаунт и пользоваться веб-приложением |
| Альтернативный поток | В случае, если в системе уже оказался пользователь с таким логином или почтой, незарегистрированному пользователю будет предложено сменить логин или почту, чтобы пройти регистрацию |
| Постусловие | После регистрации нового пользователя его данные, такие как логин и дата регистрации отправляются в панель администратора, чтобы тот мог присвоить аккаунту нужную роль, после чего в профиле пользователя будет отображаться новая информация в зависимости от роли |
| Раздел спецификации прецендента | Описание |
|  |  |
| Название | Написание статьи |
| Актер | Пользователь |
| Предусловие | Пользователь вошел в свой аккаунт, перешел на страницу с новостями и нажал кнопку «Написать статью» |
| Основной поток | Пользователь заполняет форму, состоящую из пяти полей: Название статьи, фон статьи, заглавное изображение, текст статьи, дополнительные файл. При нажатии на кнопку «Опубликовать», происходит валидация данных формы и отправка статьи |
| Альтернативный поток | В случае возникновения какой-либо ошибки во время валидации формы или при попытке ее отправить, система уведомляет о ней пользователя, с просьбой исправить проблемы(если это возможно) |
| Постусловие | После отправки, статья попадает на рассмотрение администратора, который проверяет статью, после чего либо опубликовывает ее, либо направляет сообщение пользователю с просьбой исправить недочеты |
| Раздел спецификации прецендента | Описание |
|  |  |
| Название | Решение теста с курса |
| Актер | Студент |
| Предусловие | Преподаватель во время создания курса, либо позже добавил студента в курс, чтобы тот мог видеть задания в нем. Студент, войдя в свой аккаунт и перейдя в профиле во вкладку «Курсы» открывает нужный курс и открывает нужный тест |
| Основной поток | Перед студентом высвечивается тема теста, а также вопросы по тесту с вариантами ответов в виде чекбоксов, после того как студент выберет ответы, которые считает нужными во всех вопросах, он нажимает кнопку «Сдать», далее отправляется запрос на проверку теста |
| Альтернативный поток | В случае закрытия пользователем страницы с тестом, все данные об ответах сбрасываются и студенту придется проходить тест заново |
| Постусловие | После отправки запроса на проверку теста, высчитываются баллы студента по стобалльной системе, а также этот студент добавляется в список сдавших данный тест |

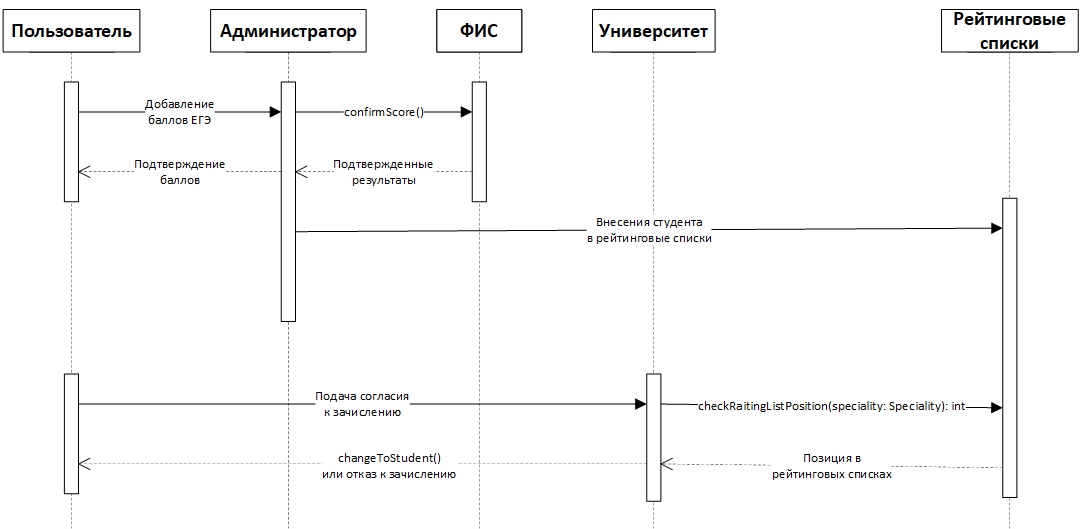
**Диаграмма классов**



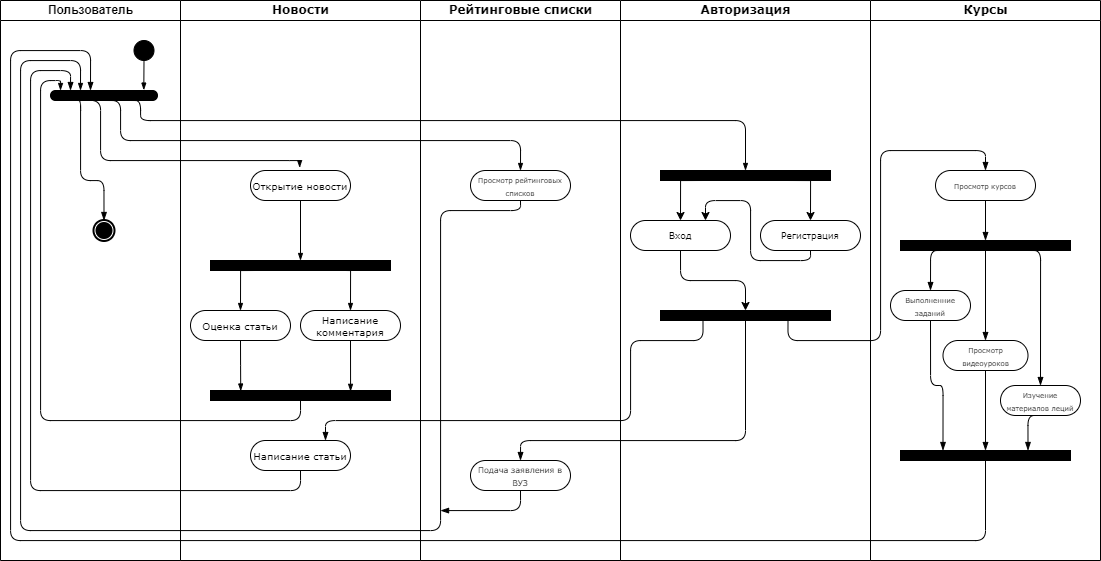
**Диаграмма состояний**:



**Диаграмма последовательностей:**

****

**Диаграмма видов деятельности**

****

**Макет:**

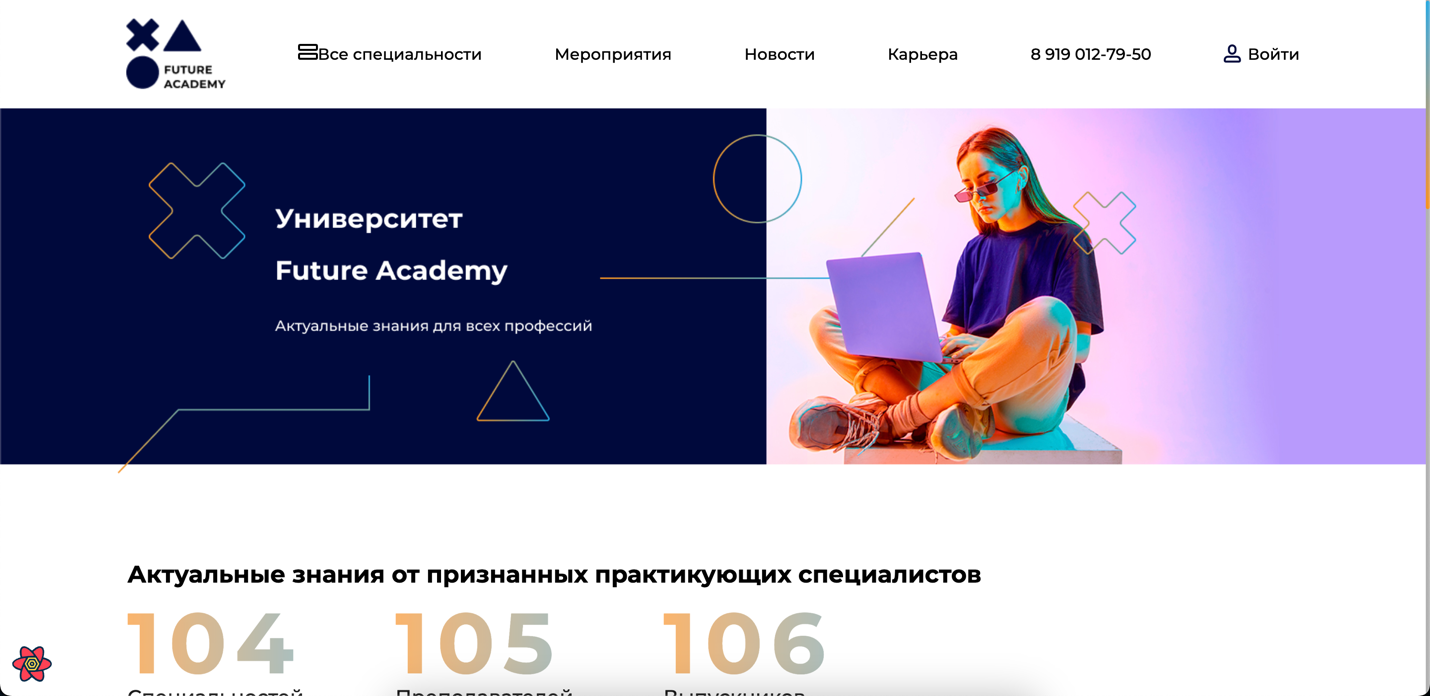


Рисунок 1. Главная страница

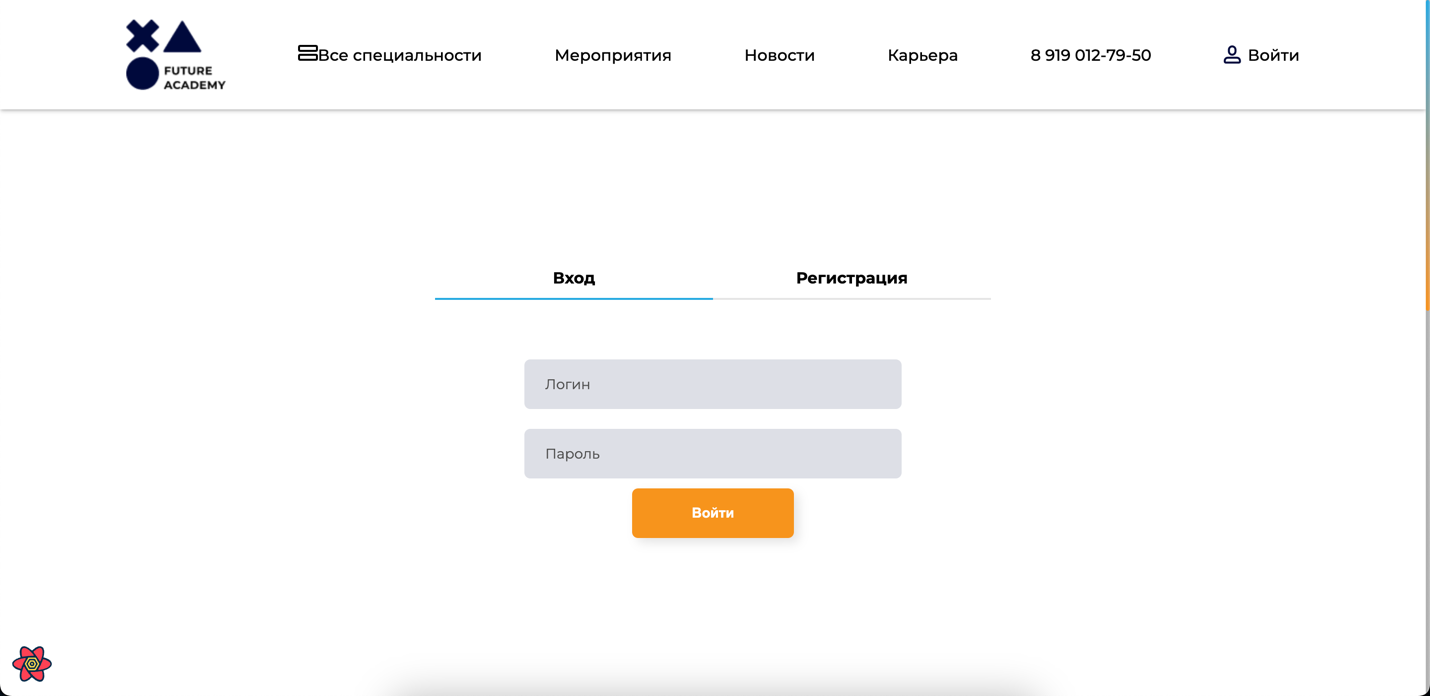


Рисунок 2. Страница входа/регистрации

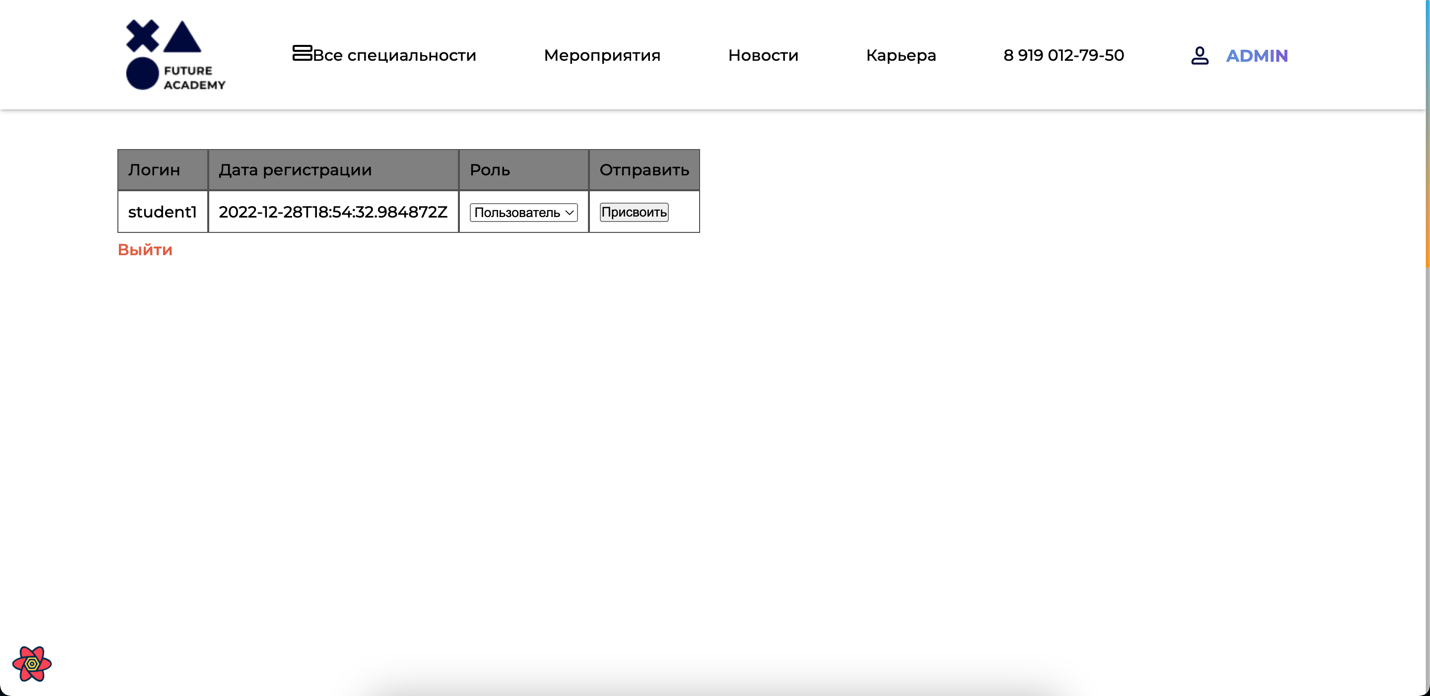


Рисунок 3. Панель админа

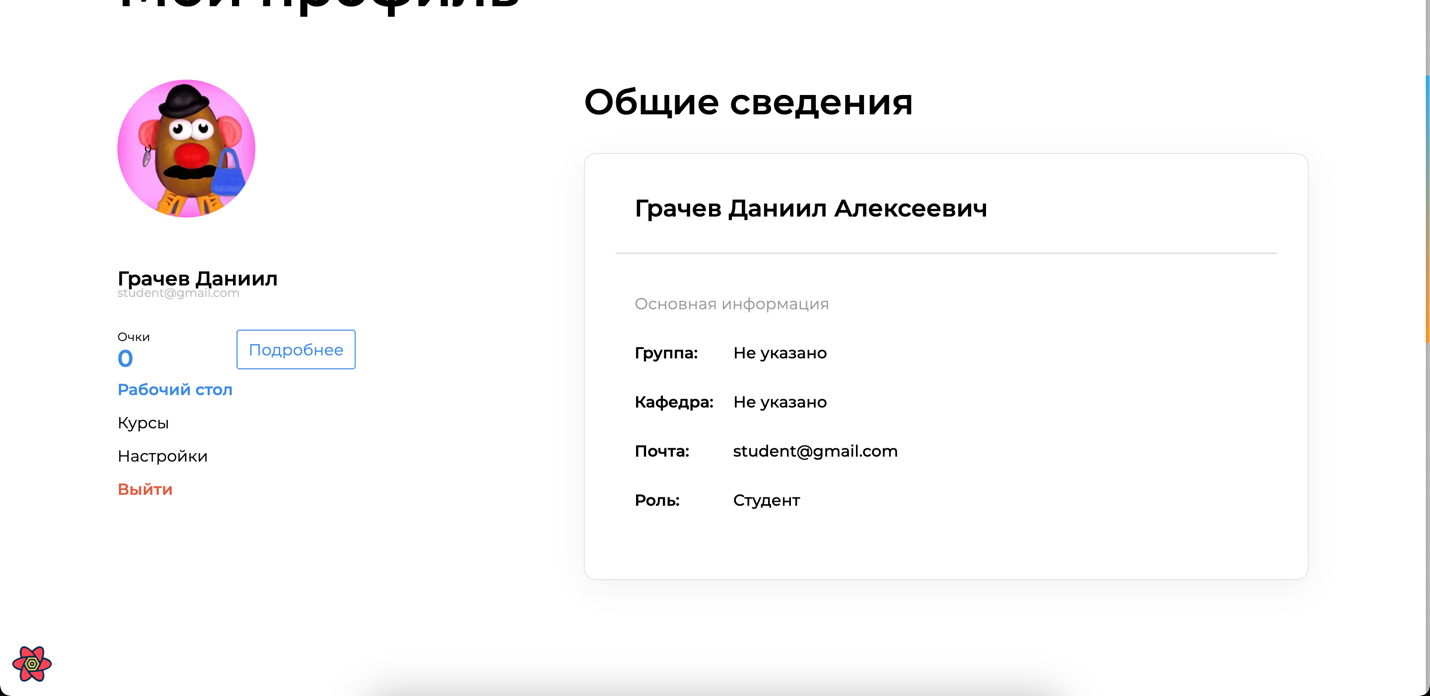


Рисунок 4. Страница профиля