**ОТЧЁТ**

*ОКЭИ 09.02.07 9022 02 У*

*01.01. Курсовая работа*

*МДК 01.01*

*«Разработка программных модулей»*

Количество листов: *36*

Дата готовности: *27 декабря 2023*

Разработал: *Азизов Амир Наилевич*

Руководитель *Егурнова Елена Николаева*

Оренбург 2023

# Аннотация

Курсовой проект на тему: «Создания сайта общения на основе интересов».

Ключевые слова: «Visual Studio Code», «PostgreSQL», «NodeJS», «HTML», «CSS», фреймворк.

Работа представляет из себя описание архитектуры и логику работы сайта.

Отчет включает в себя 36 страницы, 27 рисунков, 3 таблицы и 3 диаграммы.

**Содержание**

[Аннотация 2](#_Toc154558506)

[Задание 4](#_Toc154558507)

[Введение 5](#_Toc154558508)

[1. Анализ предметной области 7](#_Toc154558509)

[2. Проектирование приложения 8](#_Toc154558510)

[3. Разработка программного обеспечения 9](#_Toc154558511)

[3.1 Описание технологического стека разработки 9](#_Toc154558512)

[3.2 Описание алгоритма работы 10](#_Toc154558513)

[3.3 Описание интерфейса пользователя 20](#_Toc154558514)

[4. Тестирование приложения. 25](#_Toc154558515)

[4.1 План тестирования 25](#_Toc154558516)

[4.2 Оценка результатов проведения тестирования 29](#_Toc154558517)

[Заключение 32](#_Toc154558518)

[Список литературы 33](#_Toc154558519)

[Приложения 34](#_Toc154558520)

# Задание

# Введение

В наше время коммуникация играет ключевую роль в формировании личности, социализации. Особенно актуальным становится вопрос общения с людьми имеющие одинаковые интересы в процессе обучения, где учащиеся ищут возможности для дополнительных занятий и развития своих интересов или изучения новых. Исходя из этой потребности, представленная курсовая работа нацелена на создание платформы для общения на основе интересов для учащихся школ.

Актуальность выбранной темы состоит в том, что современных условиях учащиеся сталкиваются с необходимостью обмена опытом, поиска друзей схожими интересами и расширения своего круга общения. Такая платформа может стать мостом между учащимися, облегчая процесс социализации и формирования социальных связей.

Цель курсового проекта — это создание удобной и безопасной платформы для общения на основе интересов, предоставляющей учащимся возможность находить кружки с такими же учащимися.

Задачи, решаемые в проекте:

* анализ предметной области;
* разработка интерфейса платформы для поиска и регистрации в кружках;
* обеспечение безопасности пользователей и их личных данных.

Практическая значимость полученных результатов – это разработанная платформа может стать инструментом для образовательных учреждений в организации дополнительного образования и стимулирования интереса учащихся к различным направлениям.

Требования к компьютеру:

− операционная система: Windows (7, 8 и 10), macOS X 10.11 или Linux (Ubuntu, Debian и Redhat);

− оперативная память: 1 ГБ. (рекомендуется 4 ГБ);

− процессор: 1,6 ГГц или выше;

− место на жестком диске: 500 МБ (рекомендуется 2 ГБ);

− Microsoft .NET Framework: 4.5.2 и выше.

Устройства ввода:

− клавиатура;

− мышь.

Устройства вывода:

− монитор;

− наушники.

Программной части:

* среда разработки: «Visual Studio Code»;
* языки программирования: «HTML», «СSS», «NodeJS», «SQL»;
* базы данных: «PostgreSQL»;
* фреймворки для веб-разработки: «Express»;
* фреймворки для «unit-тестов»: «Jest».

В будущем, расширяя функционал, платформа будет акцентировать внимание на создании интерактивных возможностей для пользователей. Исходя из текущих потребностей учащихся, платформа уже может предлагать онлайн-мероприятия, обеспечивая пространство для взаимодействия и обмена опытом между кружками. Этот шаг представляет собой естественное развитие, которое может быть усилено интеграцией с другими образовательными платформами, создавая единую среду для обучения и развития. Такое интегрированное решение может значительно усилить учебный процесс и предложить пользователям более глубокий и многогранный опыт.

# 1. Анализ предметной области

Общение между учащимися на основе их интересов в школьной среде играет ключевую роль в адаптация к социуму, формировании личностных качеств и стимулировании активности учащихся сталкивая к необходимости налаживания межличностных связей, основанных на общих интересах. Групповые интересы и увлечения могут стать полезными и интересными связами для формирования дружеских отношений и коллективных инициатив. Однако, параллельно с позитивными моментами, существуют и потенциальные конфликты.

В время технологического развития учащиеся активно используют различные платформы и приложения для общения. Эти инструменты предоставляют уникальные возможности для организации общения на основе интересов, создания сообществ и проведения совместных мероприятий.

# 2. Проектирование приложения

При проектировании приложения для образовательной платформы, нацеленной на обмен и общение учащихся, необходимо поставить ряд задач, чтобы обеспечить максимальную эффективность и удобство использования.

Функциональные требования:

регистрация: для использования всех функциональных возможностей платформы пользователь должен пройти процесс регистрации, включающий в себя отправку необходимых данных, таких как ФИО, электронная почта, номер телефона, класс учащегося и пароль;

авторизация: после успешной регистрации пользователь должен иметь возможность авторизироваться на платформе, предоставляя свои учетные данные для доступа к личному кабинету;

профиль пользователя: в личном кабинете каждый пользователь имеет возможность указать свои интересы, которые будут использоваться для предложения соответствующих кружков;

* поисковая строка: пользователи имеет возможность вводить ключевые слова или фразы в специальное поле поиска для быстрого доступа к нужному кружку.
* фильтры: Предоставление различных фильтров позволяет уточнить результаты поиска. Эти фильтры могут включать в себя категорию класса, максимальное количество человек и интересы.

Нефункциональные требования:

производительность: быстрый и отзывчивый интерфейс приложения без задержек;

безопасность: защита личных данных пользователей, шифрование передачи информации;

масштабируемость: гибкая архитектура приложения, способная адаптироваться к росту числа пользователей;

интуитивный интерфейс: легкость в использовании, понятные и интуитивные элементы управления.

Модели будущего приложения:

У каждого пользователя есть свой профиль, где может указать свои интересы. После входа в систему пользователь попадает на главную страницу с рекомендациями, основанными интересами из профиля. Учащийся может искать кружки, по ключевым словам, или категориям используя фильтры. При нахождении интересующего кружка пользователь может вступить в выбранный кружок.

# 3. Разработка программного обеспечения

# 3.1 Описание технологического стека разработки

Технологический стек для создания данного приложения включает: «HTML», «CSS», «Node.js», «PostgreSQL» и «фреймворк» «express» для «Node.js».

«HTML» и «CSS» основа веб-страниц:

«HTML» отвечает за структурирование содержимого, данный язык предоставляет «скелет» страницы, определяя разметку, заголовки, абзацы, изображения и другие элементы;

«CSS» предоставляет средства для стилизации и представления этого содержимого. Благодаря CSS, разработчики могут контролировать внешний вид элементов, устанавливая цвета, шрифты, отступы и многие другие атрибуты. Оптимизированный и структурированный код на «HTML» и «CSS» обеспечивает быструю загрузку страницы и корректное отображение на различных устройствах и браузерах.

Преимущества использования «HTML» и «CSS»:

Структурирование и стилизация: «HTML» обеспечивает четкую иерархию контента, упрощая его понимание и обработку, в то время как «CSS» предоставляет гибкость в стилизации, позволяя создавать эстетически привлекательные и функциональные интерфейсы;

«Кроссбраузерность»: стандартизированный код на «HTML» и «CSS» гарантирует, что веб-страницы корректно отображаются на различных браузерах и устройствах.

«Node.js» — это среда выполнения «JavaScript», которая позволяет серверной части приложения работать на этом языке. Асинхронная природа обеспечивает высокую производительность и возможность одновременного обслуживания большого числа пользователей.

Преимущества использования «Node.js»:

высокая производительность: благодаря асинхронной архитектуре, Node.js может эффективно обрабатывать множество одновременных запросов, минимизируя задержки и улучшая отзывчивость приложения;

единый язык: Использование одного языка «JavaScript» на клиентской и серверной стороне обеспечивает упрощенное взаимодействие между «фронтендом» и «бэкендом», сокращая время разработки и упрощая поддержку.

«PostgreSQL» — это мощная реляционная база данных с открытым исходным кодом. Она предлагает надежное хранение данных, а также поддерживает расширенные функциональные возможности, такие как триггеры, хранимые процедуры и географические запросы.

Преимущества использования «PostgreSQL»:

надежность и целостность данных: «PostgreSQL» предлагает транзакционную безопасность, обеспечивая сохранность данных при любых операциях и предотвращая их потерю или повреждение;

гибкость и масштабируемость: благодаря поддержке расширенных типов данных и возможности создания пользовательских функций, «PostgreSQL» может быть легко адаптирован к различным требованиям и масштабирован в зависимости от потребностей проекта.

«Express» — это минималистичный и гибкий фреймворк для создания веб-приложений на «Node.js». Предоставляет множество встроенных функций и инструментов, которые упрощают процесс разработки серверной части приложения.

Преимущества использования «Express»:

быстрота разработки: благодаря своей простой и интуитивно понятной структуре, «Express» позволяет разработчикам быстро создавать мощные веб-приложения с минимальными усилиями;

маршрутизация и контроль: «Express» предоставляет гибкий механизм маршрутизации, который позволяет легко управлять запросами и определять, какие действия должны быть выполнены при различных «URL-адресах».

«Jest» — это быстрый фреймворк для тестирования, который легко настроить. Он активно разрабатывается и используется Facebook для тестирования всех своих приложении и сайтов, а также многими другими разработчиками и компаниями.

Преимущества использования «Jest»:

«мокирование» и замены: «Jest» предоставляет мощные инструменты для изоляции тестируемого кода от внешних зависимостей, обеспечивая эффективное тестирование взаимодействий;

автоматическое сопоставление снимков: функциональность снимков в «Jest» упрощает обнаружение изменений в «UI» или логике, автоматически сравнивая текущие и ожидаемые результаты.

# 3.2 Описание алгоритма работы

В основе функционирования данного веб-приложения лежит слаженное взаимодействие клиентской и серверной частей, обеспечивающее пользователю гладкое и интуитивно понятное взаимодействие с ресурсом.

Первым шагом в взаимодействии с сайтом это регистрация пользователя. Регистрация – каждый новый пользователь, желающий воспользоваться преимуществами ресурса, проходит процесс регистрации. Введенные пользователем данные, такие как ФИО, электронная почта, номер телефона, класс и пароль, передаются на серверную сторону, где, через безопасное соединение, сохраняются в базе данных «PostgreSQL» как показано на «Рисунок 1».

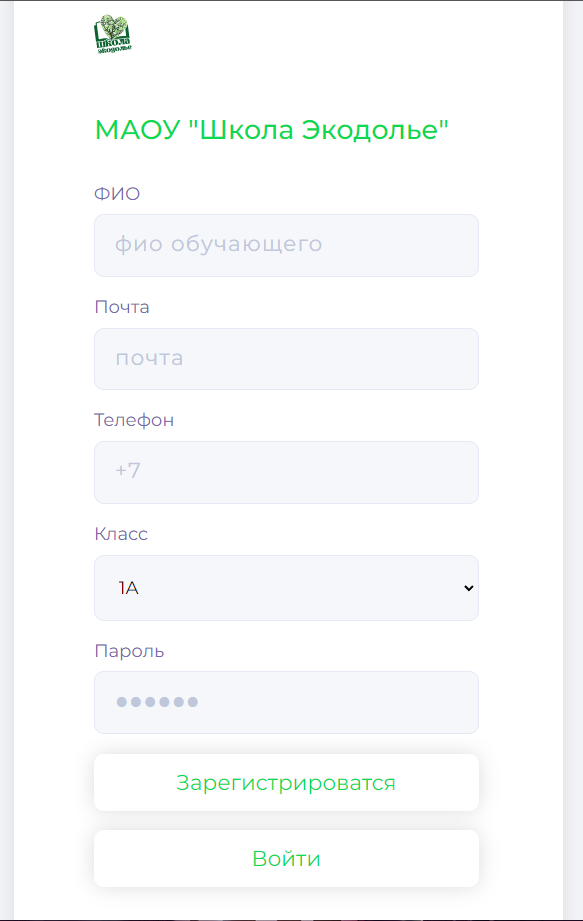


Рисунок 1 – Окно регистрации

После нажатия на кнопку «зарегистрироваться» данные отправляются на сервер, где метод «app.post» получающий форму берет данные из полей что позволяет дальше проверить на наличие имеющегося пользователя с таким же ФИО. При удачной проверке данные заполняются в базу данных и после чего пользователя перенаправляют на страницу авторизации. При не удачной регистрации у пользователя появляется надпись что «Такой пользователь уже существует», код предоставлен на «Рисунок 2».

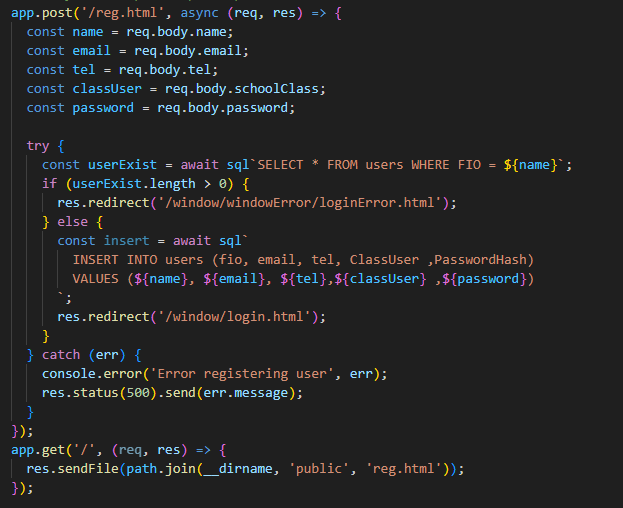


Рисунок 2 – Код окна регистрации

Вход в систему – после успешной регистрации пользователь получает возможность входа в систему, указывая электронный адрес или ФИО или номер телефона и пароль. Благодаря паттерну «стратегии» пользователь имеет возможность «гибкой» авторизации через данные указанные при регистрации. После авторизации данные также проходят через сервер и сверяются с сохраненными в базе данных как показано на «Рисунок 3».

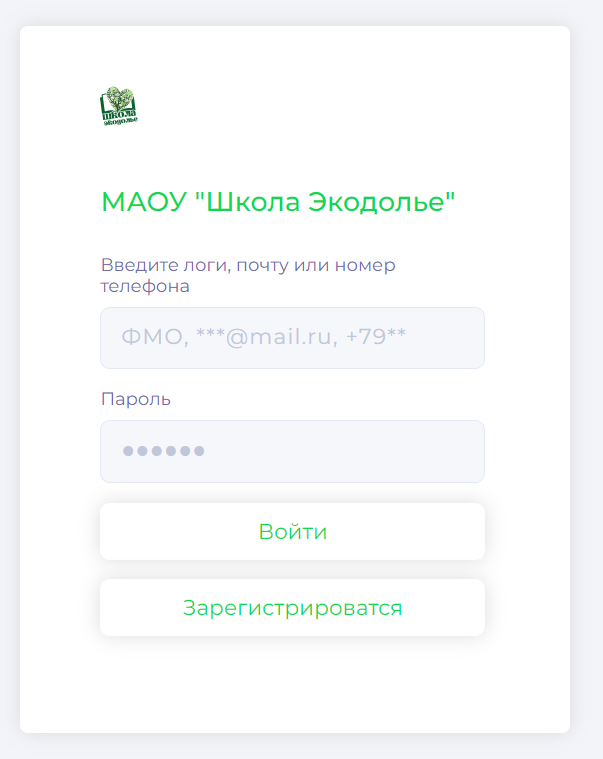


Рисунок 3 – Окно авторизации

После отправки формы на сервер «POST–запрос» запрос принимает данные вводимые пользователем после чего создается переменная «result» вызывая метод «authenticate» из стратегии «multiAuthStrategy» с передачей имени пользователя и пароля для аутентификации. Если результат аутентификации является строкой, пользователь перенаправляется на страницу ошибки аутентификации, иначе аутентификация успешна, устанавливаются куки с информацией о пользователе, и пользователь перенаправляется на главную страницу как показано на «Рисунок 4».

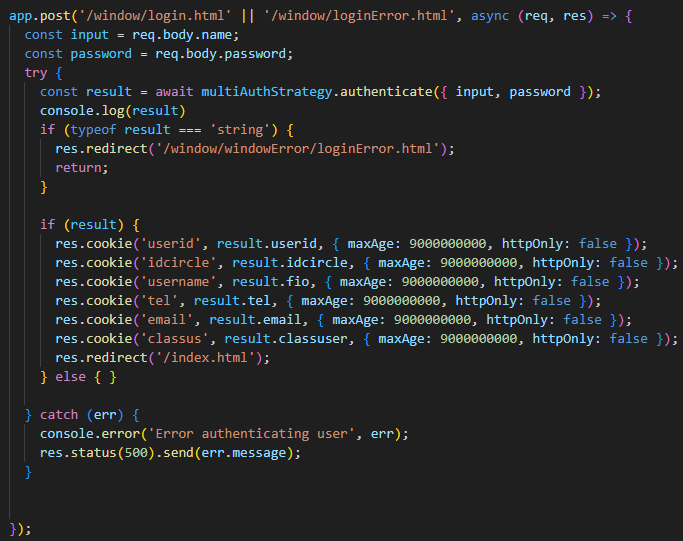


Рисунок 4 – Код авторизации

Получая доступ к сайту пользователь может прейти в свой профиль где показано все данные пользователя, а так же интересы которые можно выбрать и убрать при необходимости как показано на «Рисунок 5».

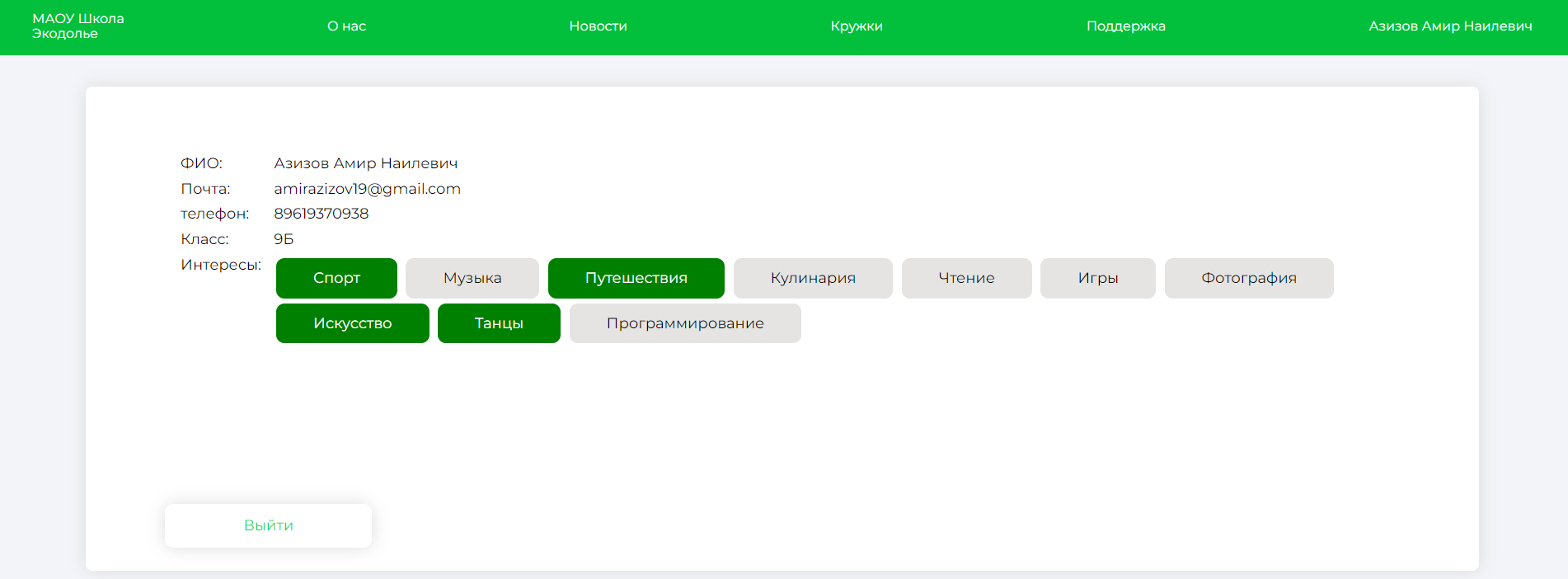


Рисунок 5 – Профиль пользователя

Вывод данных и генерация кнопок реализовано через скрипт и «Html» и запросом «.get» на сервере. На сервере в «.get» запросе создаются две переменные «interes» который получает полностью все данные по таблице «Interes» и «userinteres» получает в свою очередь данные из таблицы «InteresUsers» c фильтрами по полю «UserID» получаемые из «куки». Из массива interes извлекаются элементы. Для каждого элемента из «interes «проверяется, существует ли элемент в массиве «userinteres» с таким же «interestsid» и если такой элемент найден в «userinteres», у соответствующего элемента из «interes» устанавливается новое свойство «joined» в значение «true» что означает что выбранный интерес связывается с пользователем по «ID». Если элемент не найден, свойство «joined» устанавливается в значение «false», обновленные элементы возвращаются в новом массиве как показано на «Рисунок 6».

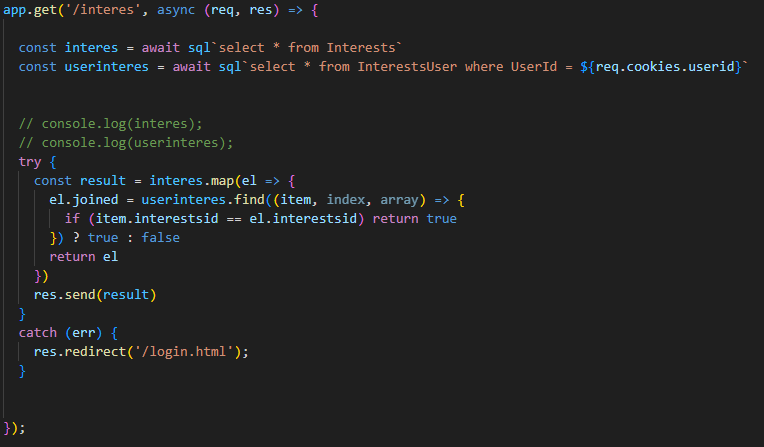


Рисунок 6 – Логика привязки интереса и пользователя

«POST–запрос» отвечает за добавление новой связи между пользователем и интересом. Получая идентификатор интереса «id», который передается в теле «POST-запроса» и идентификатор пользователя «req.cookies.userid», который извлекается из «cookies». Используя «SQL-запрос» вставки «INSERT», создается новая запись в таблице «InterestsUser», где «UserId» равен идентификатору пользователя из «cookies», а «InterestsId» равен идентификатору интереса из тела запроса.

«DELETE–запрос» отвечает за удаление связи между пользователем и интересом. Алгоритм работы происходит такой же, как и при «POST-запрос» за исключение действия запроса не добавление, а удаление записи из таблицы как показано на «Рисунок 7».

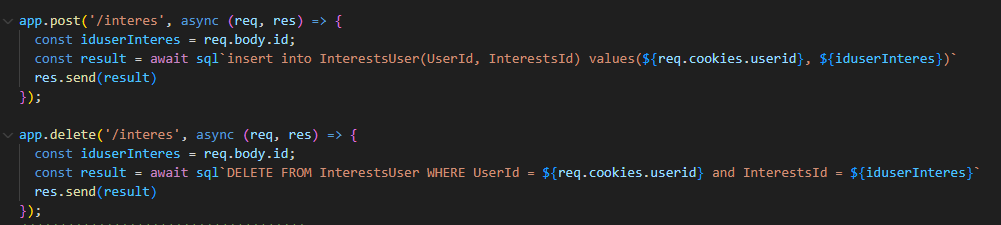


Рисунок 7 – Запросы на удаление или добавления записи

Для создания своего или вступления в кружок необходимо перейти в вкладку «Кружки», где находится доступные кружки, и фильтрация по поискам необходимых кружков.

На данной форме можно вступить в выбранный кружок. Для нахождения конкретного кружка необходимо выставить фильтры с права или же просто ввести название интересующего кружка и для вступления в силу введенных фильтров нажать кнопку «Найти» как показано на «Рисунок 8».

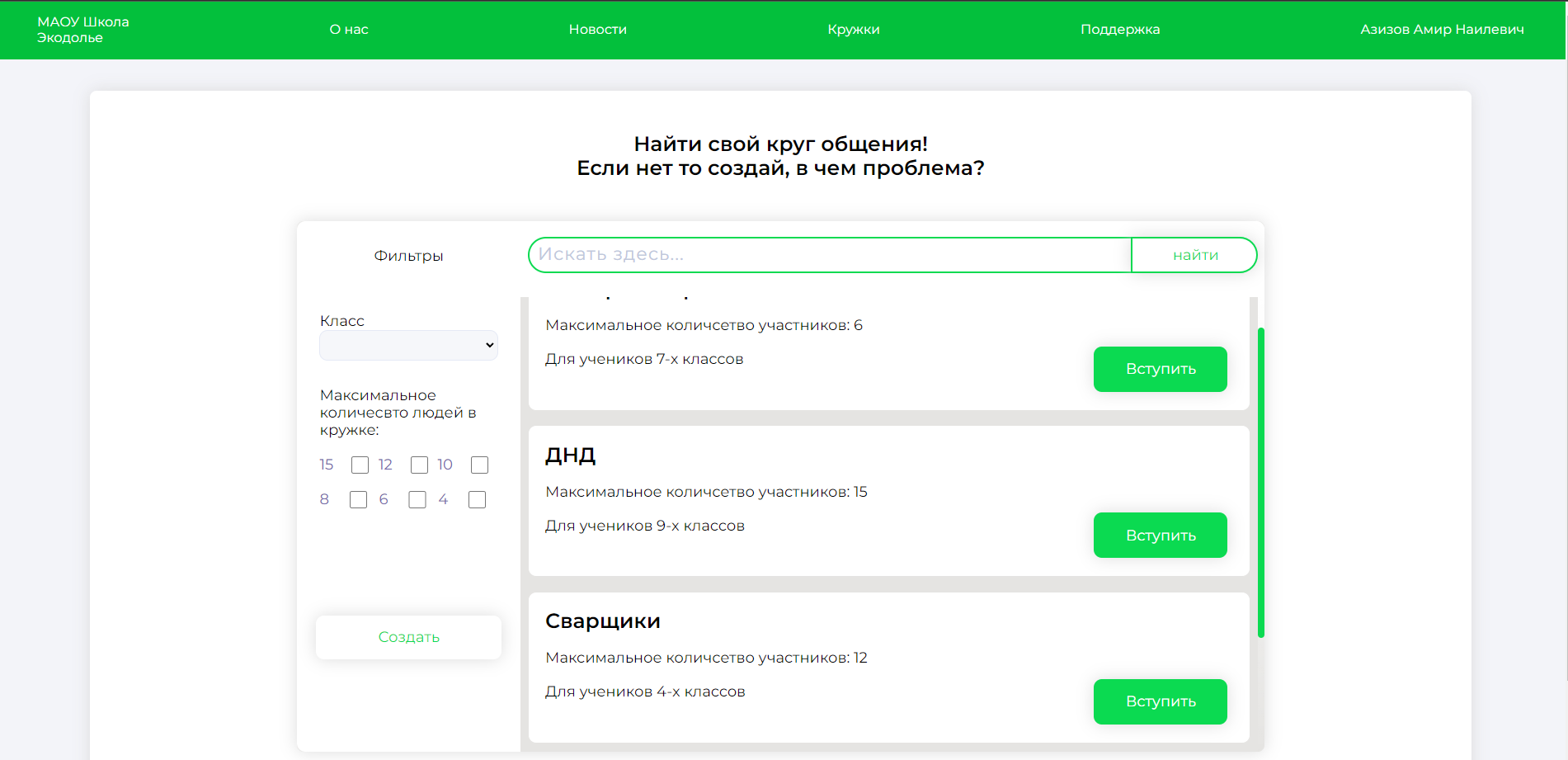


Рисунок 8 – Форма поиска кружков

Для получения кружков из базы данных используется первый «POST-запрос» который принимает три параметра из тела запроса: «classNum», «serch», и «maxPeople». Затем выполняется запрос к базе данных для извлечения всех записей из таблицы «Сircle». Эти данные фильтруются на основе предоставленных параметров. Например, если «classNum» не пуст, то из результатов выборки удаляются записи, которые не соответствуют указанному классу. Аналогичным образом происходит фильтрация по «maxPeople» и «serch» после чего результаты фильтрации отправляются клиенту.

Дальше «GET–запрос» который просто возвращает сохраненный в «куки» результат предыдущего запроса. Второй «POST – запрос» который обновляет информацию о пользователе в базе данных, присоединяя его к определенному кружку как показано на «Рисунок 9».

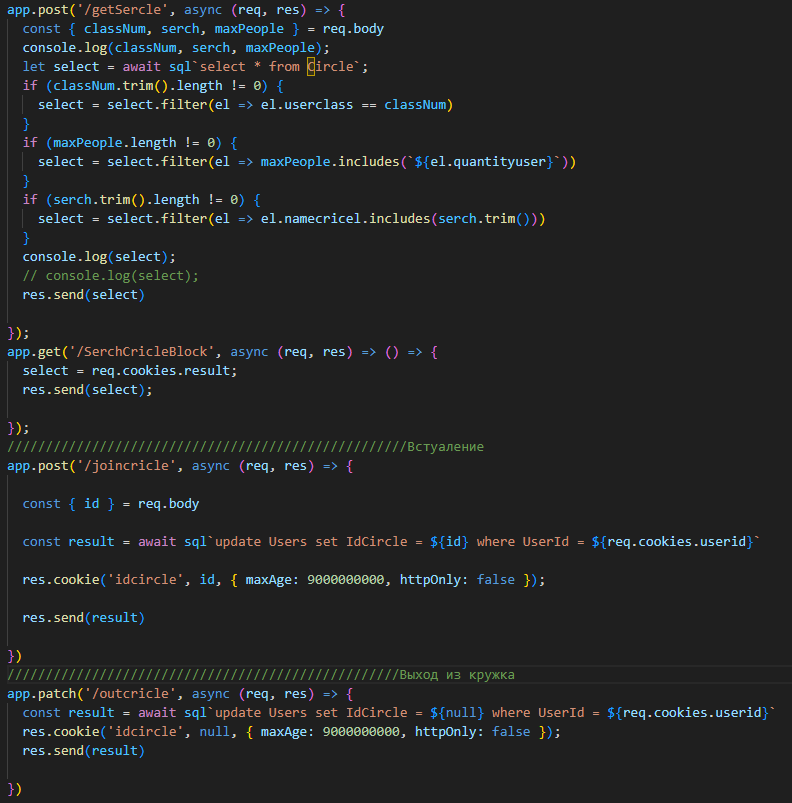


Рисунок 9 – Код для кружков

Для создания кружков необходимо нажать на кнопку «создать» с лево с низу под фильтрами, после чего пользователь переходит на форму создания кружка где нужно заполнить соответствующие данные как показано на «Рисунок 10».

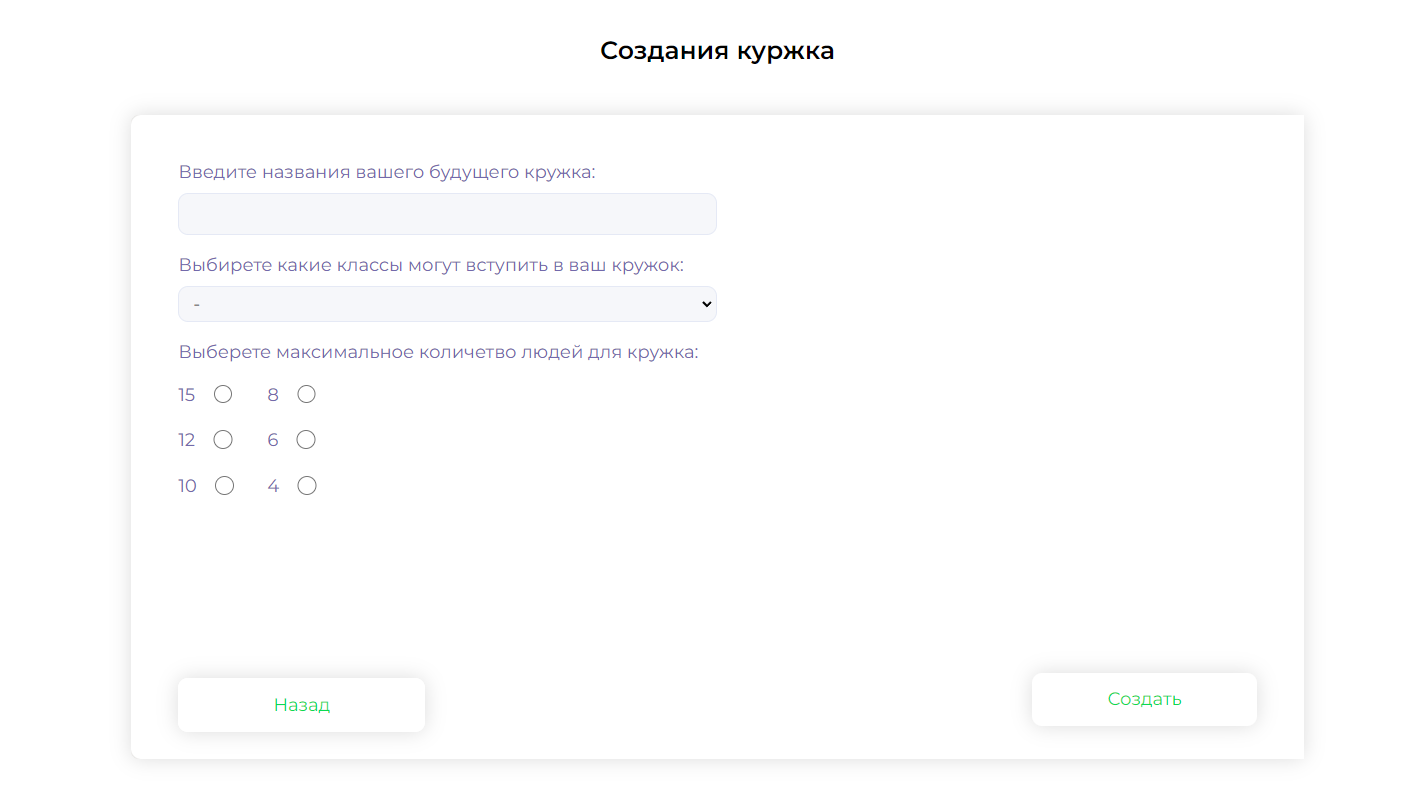


Рисунок 10 – Форма создания кружка

«POST–запрос» принимает различные параметры из тела запроса, такие как имя кружка «name», класс пользователей «ClassUsers», и максимальное количество пользователей «fifteenValue», «twelveValue» и так далее.

Создается переменная «quantity» на основе условных проверок для нескольких переменных: «fifteenValue», «twelveValue», «tenValue», «eightValue», «sixValue» и «fourValue». Если одна из этих переменных не определена, значение «quantitu» устанавливается равным значению этой переменной.

Затем выполняется проверка наличия кружка с заданным именем в базе данных. Если такой кружок уже существует, происходит перенаправление на страницу ошибки. В противном случае происходит вставка новой записи в таблицу «Сircle» с указанными параметрами, после чего пользователь перенаправляется на страницу успешного создания кружка.

Если в процессе выполнения запросов возникает ошибка, это также обрабатывается, и клиенту отправляется соответствующий статус ошибки с сообщением об ошибке как предоставлено на «Рисунок 11».

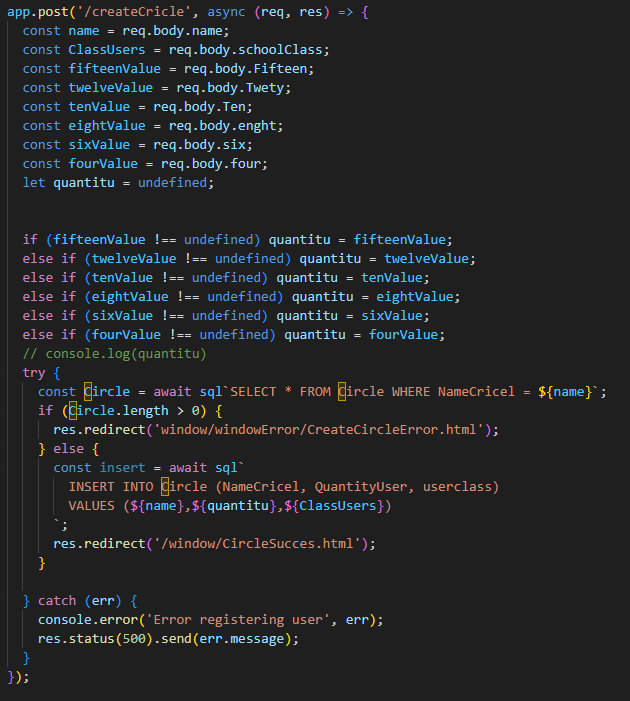


Рисунок 11 – Код создания кружка

# 3.3 Описание интерфейса пользователя

Интерфейс пользователя — это совокупность графических и текстовых элементов, предназначенных для взаимодействия пользователя с программным продуктом или устройством.

При переходе на сайт пользователь попадает в окно авторизации где интуитивно понятно что нужно вводить, если у пользователя не получается авторизироваться то кнопка с низу позволяет пользователю перейти на форму регистрации как показано на «Рисунок 12».

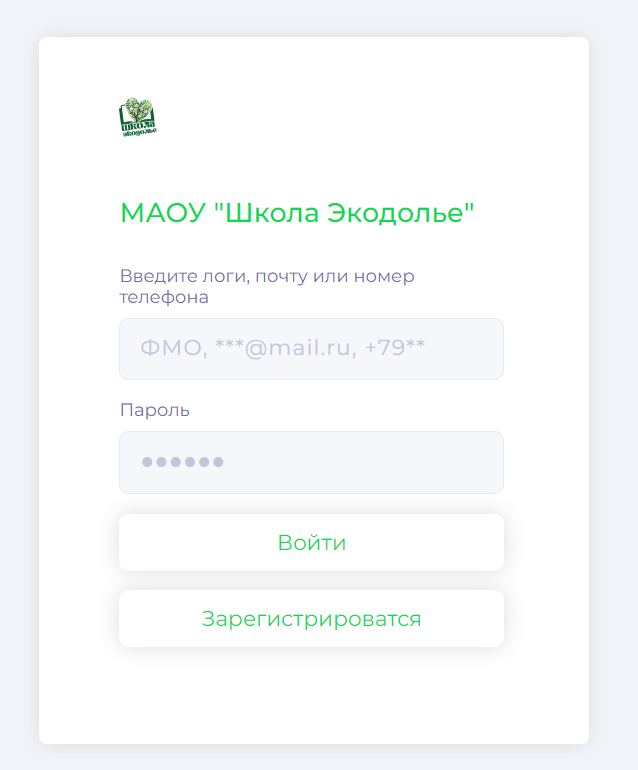


Рисунок 12 – Авторизация

На форме регистрации пользователь куда интуитивно понятно может ввести необходимые данные для успешной регистрации, а в дальнейшем для авторизации, как показано на «Рисунок 13».

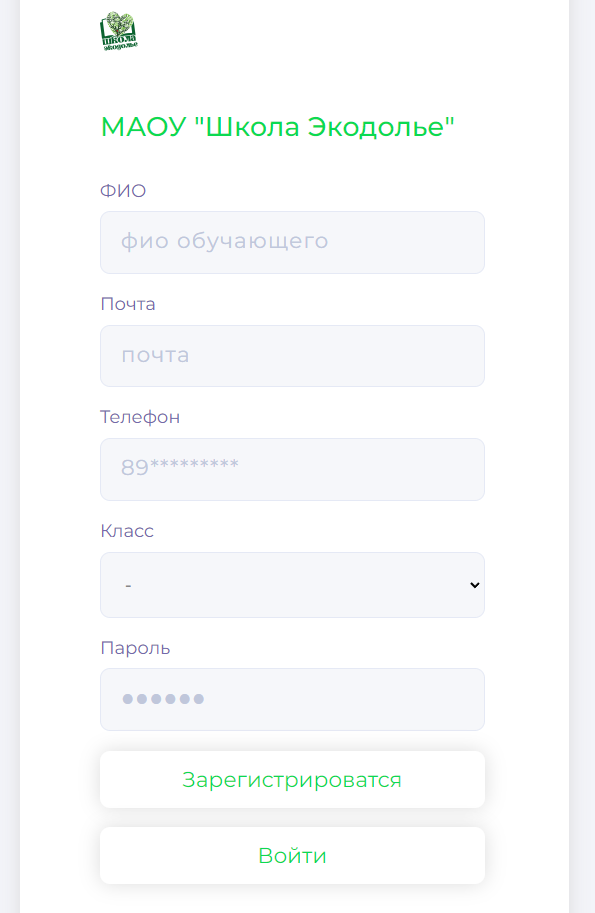


Рисунок 13 – Регистрация

После успешной авторизации пользователь попадает на главную страницу сайта с приветствующим сообщения для пользователя, как показано на «Рисунок 14».

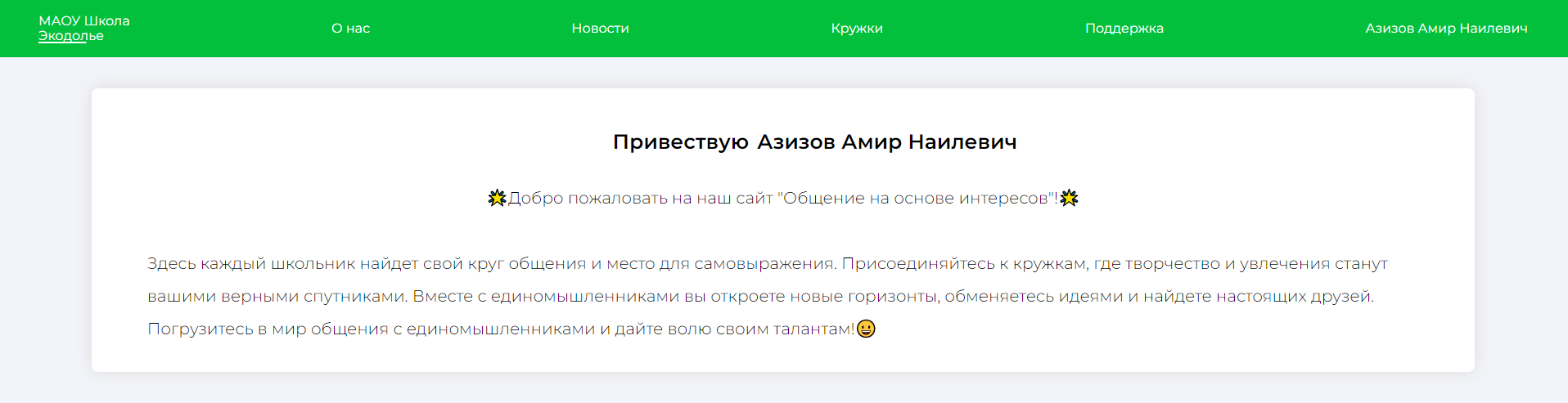


Рисунок 14 – Главная страница

Дальше переходя по вкладкам сверху на сайте, пользователь, переходя на вкладку «О нас» открывается страница с информацией о школе как показано на «Рисунок 15».

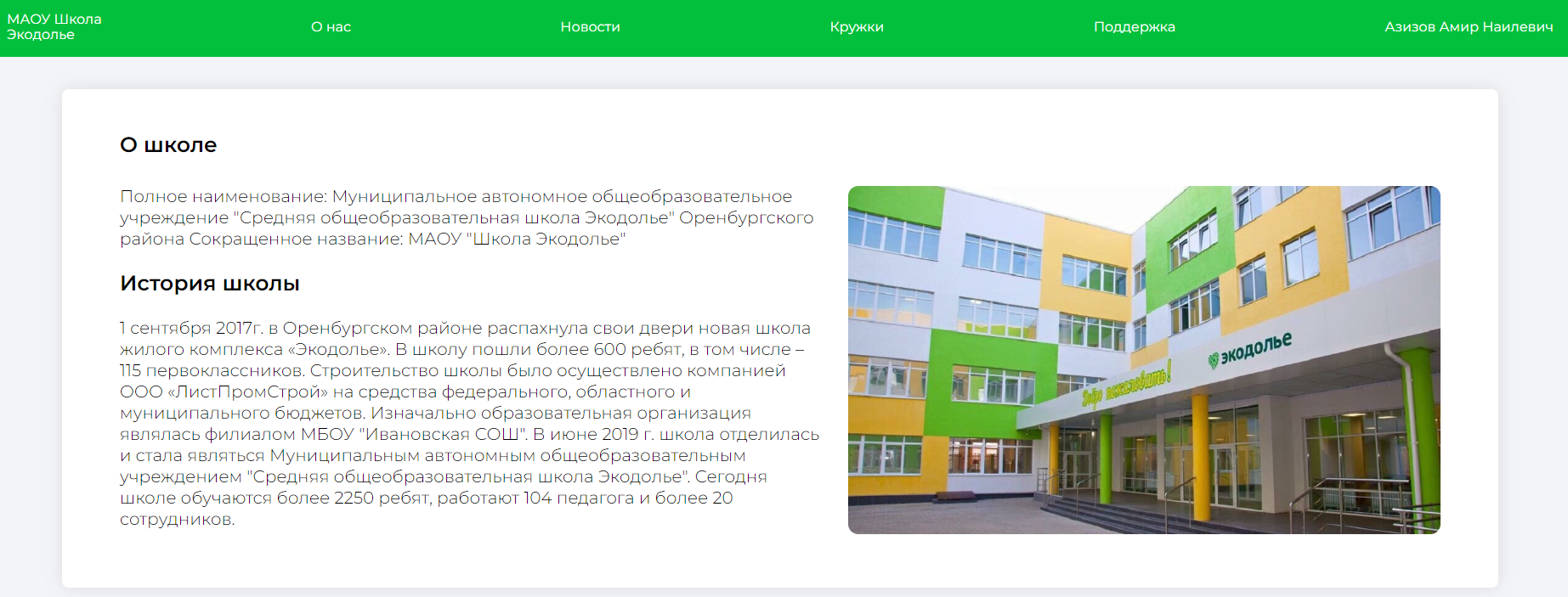


Рисунок 15 – О нас

Вкладка новости в разработке и поэтому для пользователя является бесполезной. Следующая вкладка «Кружки», в которой пользователь может найти интересующий кружок и вступить как показано на «Рисунок 16».

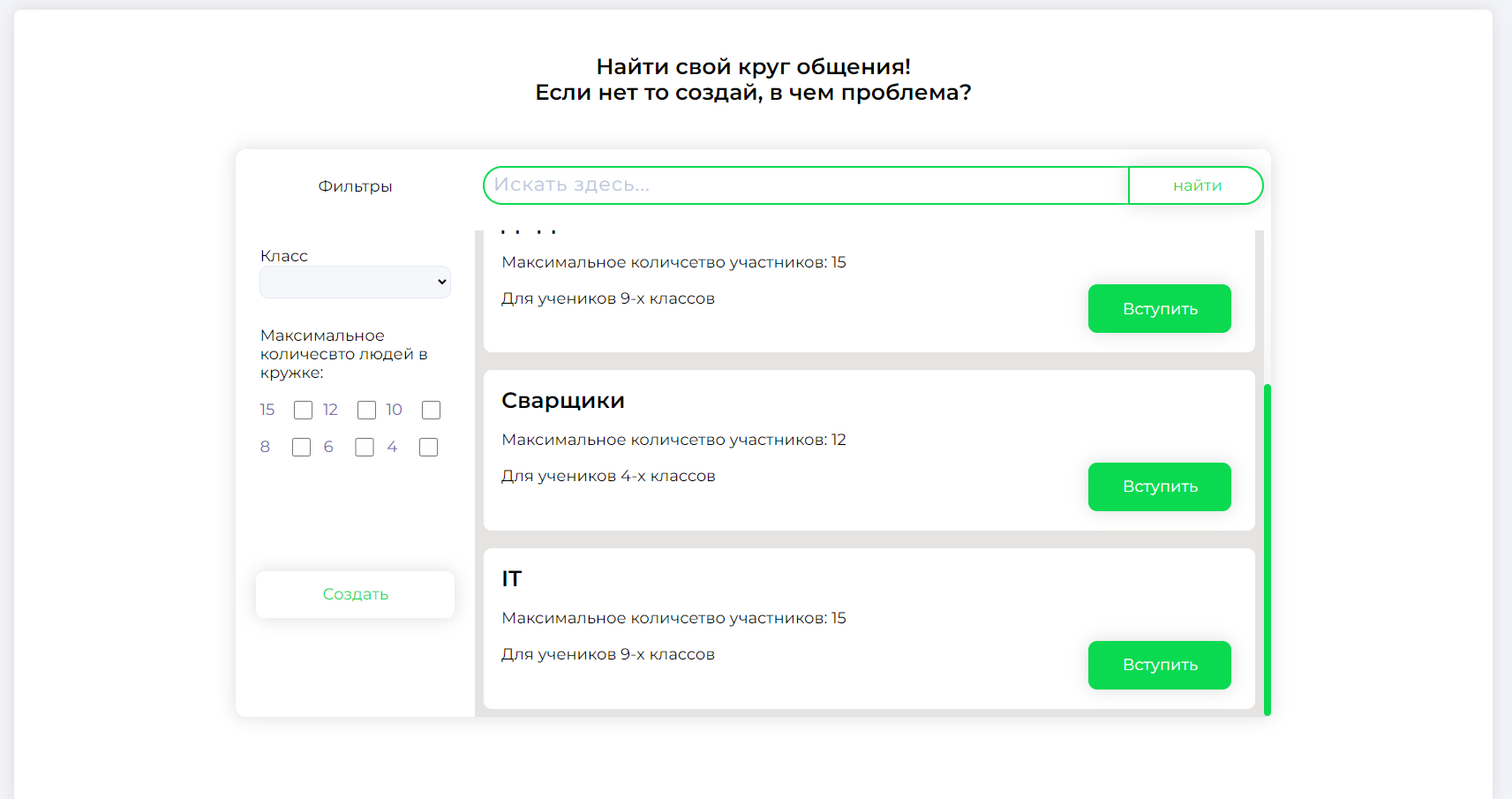


Рисунок 16 – Кружки

Если пользователь не может найти интересующий кружок, то есть возможность создать свой при нажатии на кнопку «создать», после нажатия пользователь переходит на форму создания кружка как показано на «Рисунок 17».

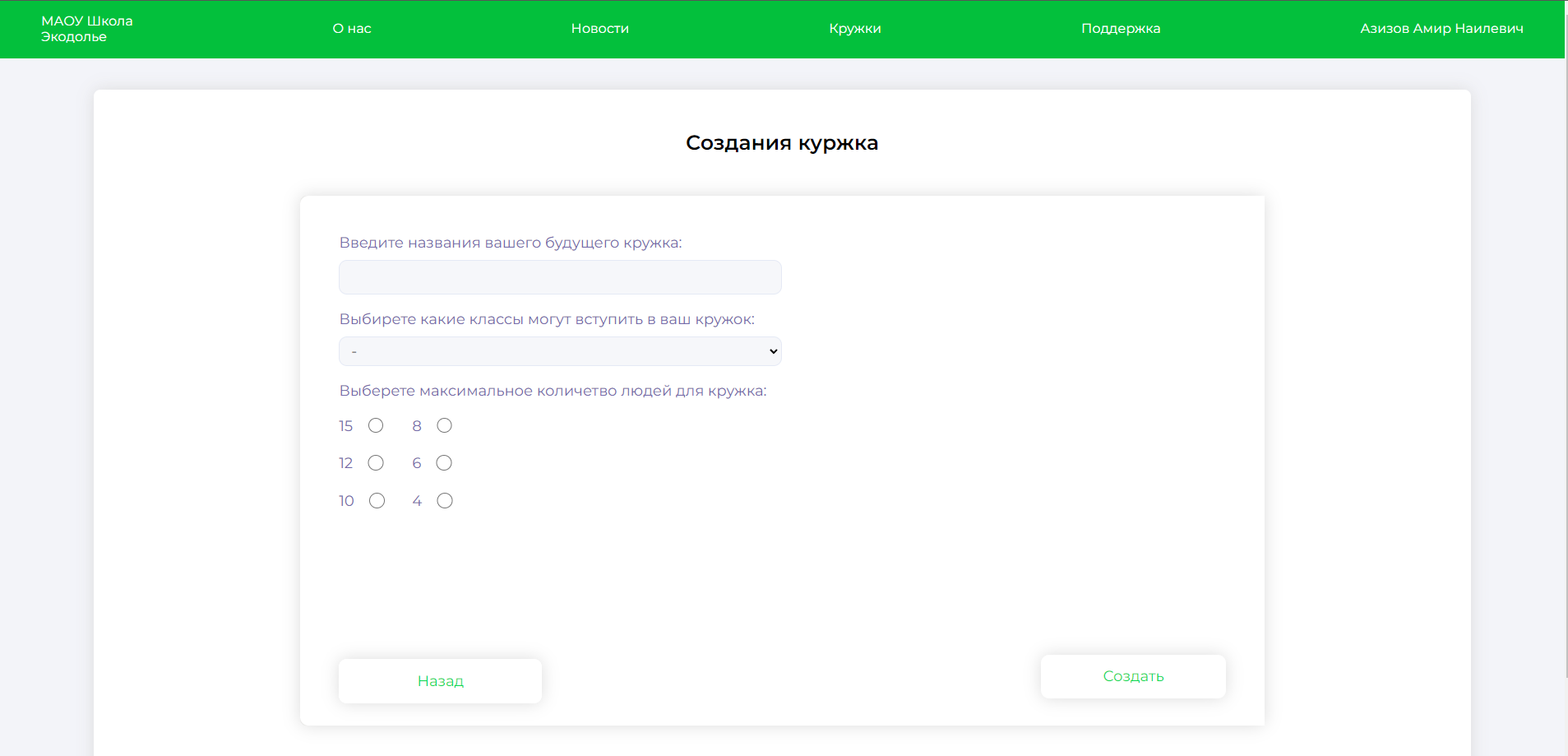


Рисунок 17 – Создание кружка

После ввода всех данных пользователя оповещает о том что кружок успешно создан и необходимо перейти в «Кружки» где будет созданный кружок как показано на «Рисунок 18».

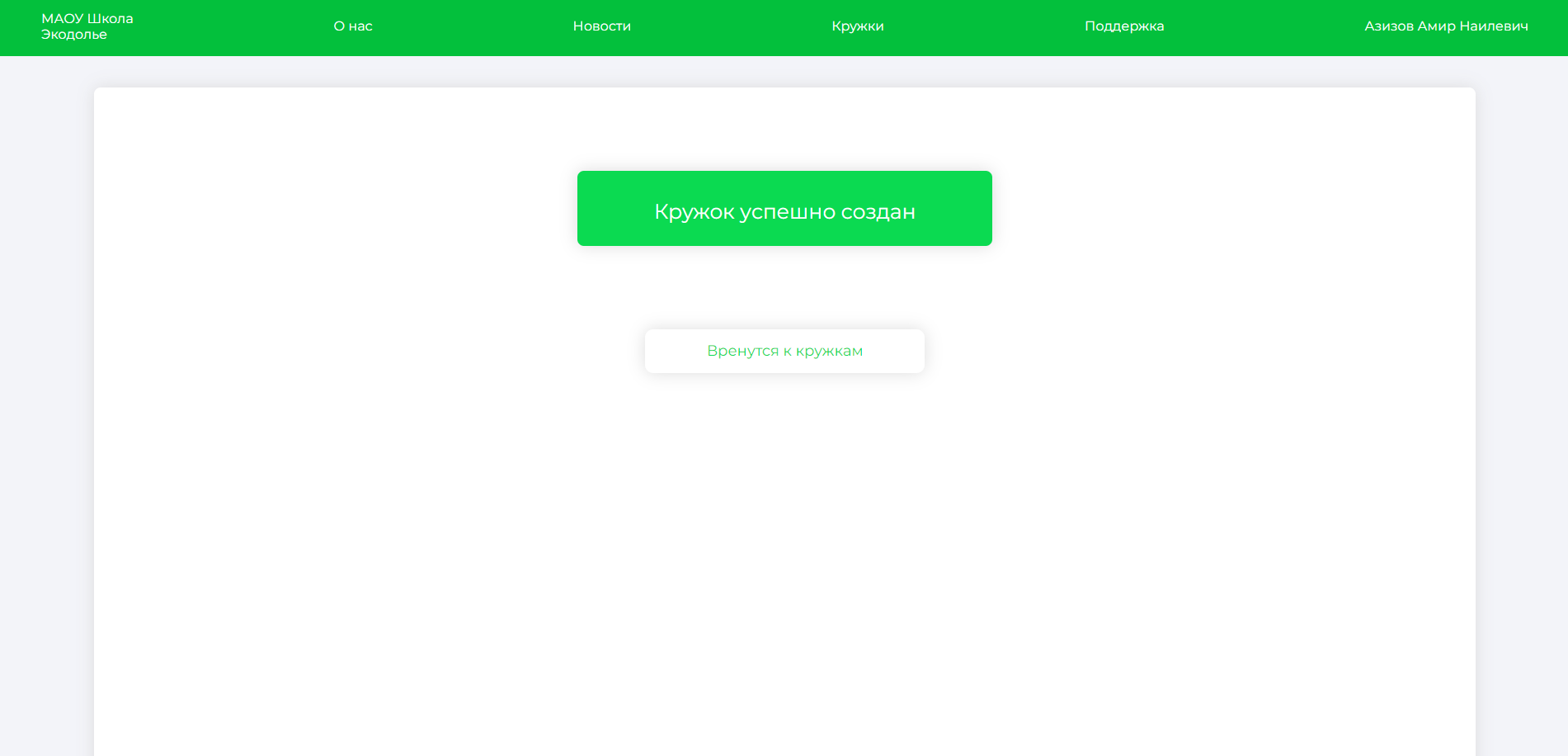


Рисунок 18 – Успешное создания кружка

Следующая вкладка «Поддержка» где находятся часто задаваемые вопросы пользователей как показано на «Рисунок 19».

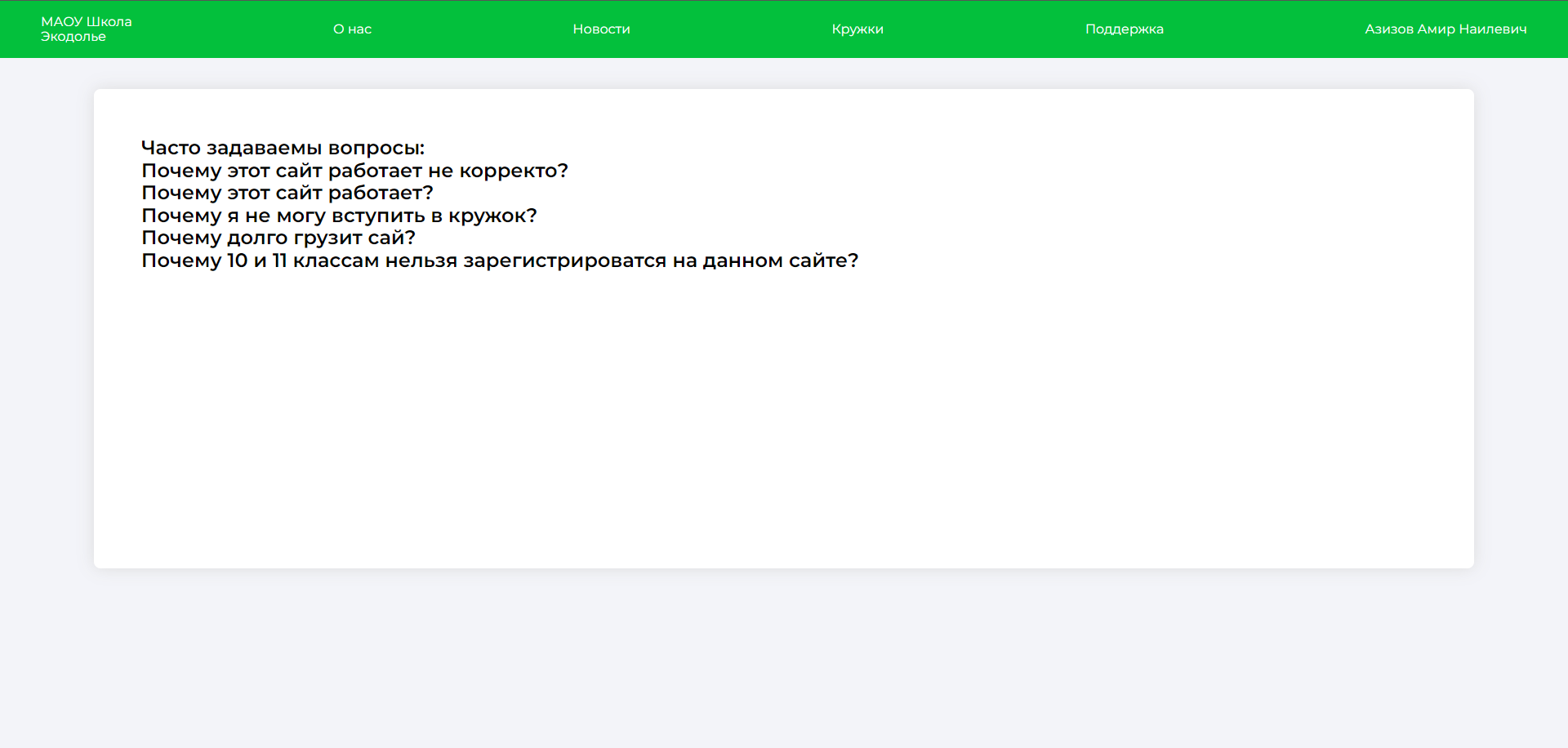


Рисунок 19 – Поддержка

Последней вкладкой является профиль пользователя, где показываются все данные, вводимые при регистрации и настройки интересов из предложенных на выбор при необходимости какие-то интересы можно убрать, какие-то добавить, а также если пользователь захочет выйти из профиля, то необходимо нажать кнопку «Выйти» находящеюся снизу слева как показано на «Рисунок 20».

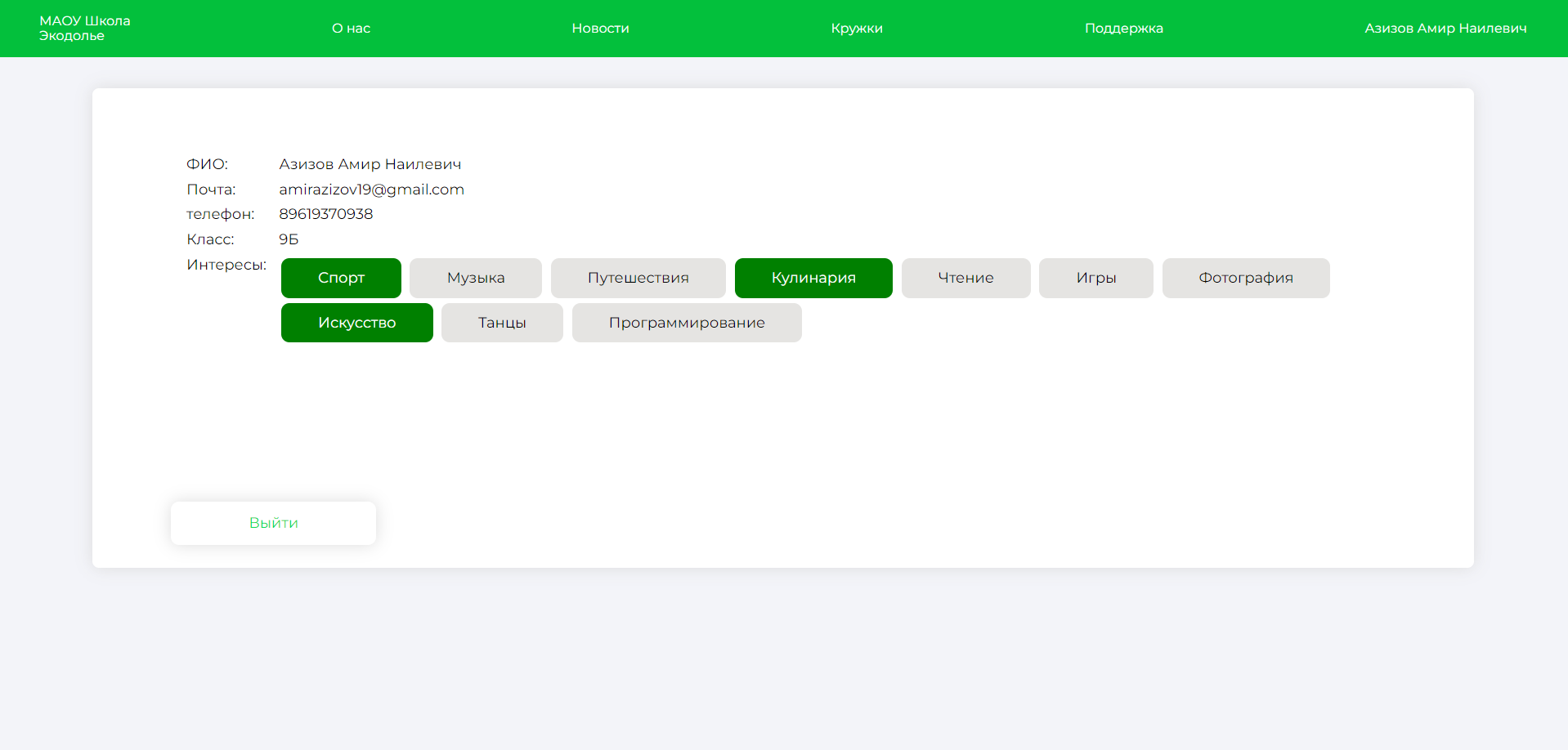


Рисунок 20 – Профиль

# 4. Тестирование приложения.

# 4.1 План тестирования

Тестирование программного обеспечения — это процесс проверки и валидации программных компонентов или системы в целом с целью обнаружения дефектов, ошибок или несоответствий заявленным требованиям. Оно направлено на обеспечение качества продукта и уверенности в его корректной работе в различных условиях эксплуатации.

«Unit-тестирование» — это метод тестирования программного обеспечения, при котором отдельные компоненты (или "юниты") программы тестируются независимо от других частей. Целью «unit-тестирования» является проверка корректности работы отдельных модулей или функций в изоляции, без учета их взаимодействия с другими компонентами.

«Unit-тестирования» — будут проводится с помощью фреймворка «jest», а также для работы тестов необходимо создать файл с тестами, фалы для настройки «jest» и прописать сами тесты.

План тестирования:

* регистрация;
* авторизация;
* интересов пользователя;
* создания и поиска кружков;
* присоединения и выход из кружка.

Для начала тестирования необходимо создать два файла для настройки фреймворка «jest», а именно «jest.config.cjs» и «.babelrc» как показано на «Рисунок 21».

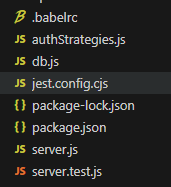


Рисунок 21 – Создания файлов настройки

Тестирования регистрации: необходимо написать код, который будет отправлять данные к запросу и возвращать путь если регистрация успешной регистрации и при не удачной регистрации как представлено на «Рисунок 22».



Рисунок 22 – Тест регистрации

Тестирования авторизации: необходимо написать код, который будет отправлять данные к запросу и возвращать путь если авторизация успешная и при не удачной авторизации как представлено на «Рисунок 23».

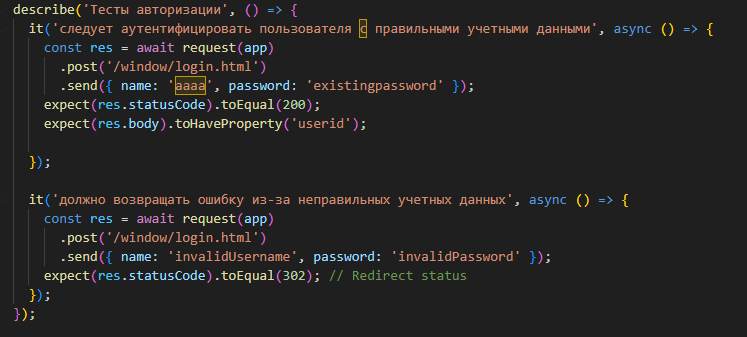


Рисунок 23 – Тест авторизации

Тестирования интересов пользователя: необходимо написать код, который будет отправлять «id» интересов и «userid» к запросу и возвращать результат из запроса как показано на «Рисунок 24».

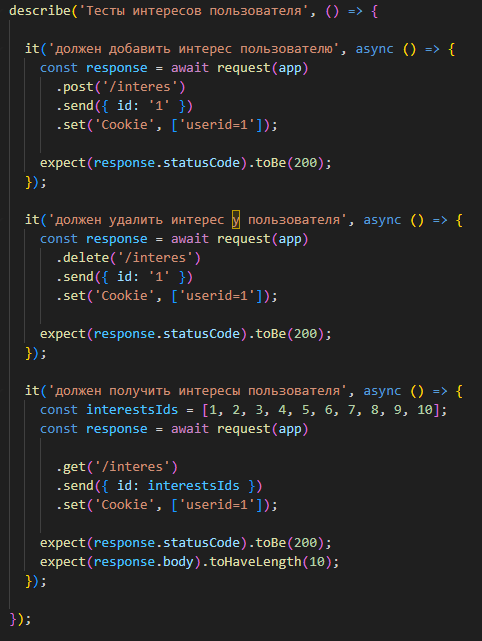


Рисунок 24 – Тест интересов пользователя

Тестирования создания и поиска кружков: необходимо написать код, который будет отправлять данные к запросу и возвращать код статуса если создания кружка успешная и при создания кружка с использованным именем, а так же поиск и фильтры как представлено на «Рисунок 25».



Рисунок 25 – Тестирования создания и поиск кружка

Тестирования вход и выхода из кружка: необходимо написать код, который будет отправлять данные к запросу и возвращать код статуса если связывания кружка и пользователя успешная и выход из кружка как представлено на «Рисунок 26».

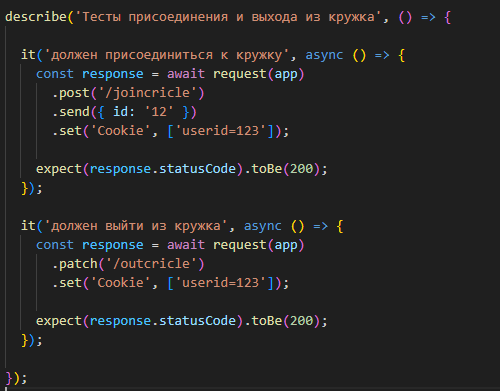


Рисунок 26 – Тестирование присоединению к кружку и выход

# 4.2 Оценка результатов проведения тестирования

В результате тестирования различных функций сайта unit – тестирование прошло не очень успешно как показано на «Рисунок 27».

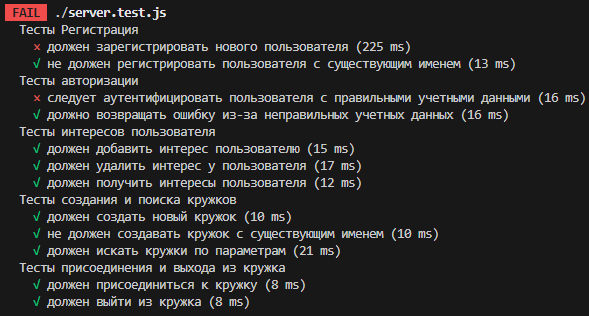


Рисунок 27 – Итоги тестирования

Был проведен «тест-кейс» с регистраций и авторизацией для проверки отправки и сохранности данных, представлено в «Таблица 1».

Таблица 1 – Тест корректную работу авторизации и регистрации

|  |  |  |
| --- | --- | --- |
| Название: | Тест авторизации регистрации | |
| Действие | Ожидаемый результат | Результат теста |
| Предусловие: | | |
| Открытия окна регистрации | Погрузка окна | Пройден |
| Шаги теста: | | |
| Ввод данных | Корректное отображения данных и обязательно к заполнению с форматом | Пройден |
| Отправка данных | Открытие страницы с входом | Пройден |
| Ввод данных | Корректное отображения данных | Пройден |
| Вход | Открытие главной страницы сайта | Пройден |

Был проведен «тест-кейс» на корректное отображение с сохранением интересов и кружков к которым пользователь присоединился для проверки отправки и сохранности данных у конкретного пользователя, представлено в «Таблица 2».

Таблица 2 – Тест сохранения выбранных данных с корректным выводом

|  |  |  |
| --- | --- | --- |
| Название: | Тест авторизации регистрации | |
| Действие | Ожидаемый результат | Результат теста |
| Предусловие: | | |
| Авторизация | Доступ к сайту | Пройден |
| Шаги теста: | | |
| Переход в профиль | Корректное отображения данных и интересов | Пройден |
| Выбор интересов | Смена цвета при выборе и сохранения данных интересов у данного пользователя | Пройден |
| Переход к кружкам | Корректное отображения данных, ввод поиска и изменении фильтрации | Пройден |
| Нажатие на кнопку для присоединения к кружку | Смена цвета при присоединении и сохранения данных выбранного кружка у данного пользователя | Пройден |
| Нажатие на кнопку для присоединения к другому кружку | Смена цвета при присоединении и сменна данных выбранного кружка у данного пользователя | Пройден |
|  | | |
| Выход из аккаунта | Переход на страницу входа и очищения куки | Пройден |
| Авторизация | Переход на главную страницу и корректное отображение данных в профиле | Пройден |
| Переход к кружкам | Отображения кружка к которому было присоединение | Пройден |

Был проведен «тест-кейс» на создание и корректное отображение кружков, к которым пользователь присоединился, представлено в «Таблица 2».

Таблица 3 – Тест создание и корректное отображение кружков

|  |  |  |
| --- | --- | --- |
| Название: | Тест авторизации регистрации | |
| Действие | Ожидаемый результат | Результат теста |
| Предусловие: | | |
| Открытия окна с кружками | Погрузка окна | Пройден |
| Шаги теста: | | |
| Нажатие кнопки создать | Открытия и корректное отображение страницы с формой создания кружка | Пройден |
| Создания кружка | Открытия окна с успешным создания кружка | Пройден |
| Переход к кружкам | Корректное отображения данных и созданного кружка | Пройден |
| Нажатие на кнопку для присоединения к кружку | Смена цвета при присоединении и сохранения данных выбранного кружка у данного пользователя | Пройден |

Все «Тест-Кейсы» успешно пройдены за исключения «unit-тестов» где регистрация и авторизация были не удачными, но при этом сайт работал корректно и сохранял данные.

# Заключение

Целью курсового проекта было создания сайта для общения на основе интересов позволяя учащимся школы найти новых друзей и находить или оттачивать навыки. Используя языки программирования такие как «NodeJS» и «PostgreSQL», а так же языки текстовой разметки «HTML» и язык стилей «СSS». Создание данного сайта направлено на удовлетворение потребностей учащихся в обмене опытом, нахождении единомышленников и расширении своего круга общения в рамках образовательных учреждений.

Анализ предметной области, проведенный в рамках данной работы, выявил актуальность и необходимость такой платформы для современных учащихся. Отмечается, что созданная система не только облегчает процесс поиска кружков по интересам и способствует общению пользователей внутри данных кружков

Практическая значимость предложенной платформы подтверждается её потенциалом в организации дополнительного образования, стимулировании интереса учащихся к различным образовательным направлениям и формировании сильных социальных связей.

Выполнена цель курсового проекта, а именно создание удобной и безопасной платформы для общения на основе интересов, предоставляющей учащимся возможность находить кружки с такими же учащимися.

Так же выполнены задачи, решаемые в проекте:

* анализ предметной области;
* разработка интерфейса платформы для поиска и регистрации в кружках;
* обеспечение безопасности пользователей и их личных данных;

Практическая значимость полученных результатов – это разработанная платформа может стать инструментом для образовательных учреждений в организации дополнительного образования и стимулирования интереса учащихся к различным направлениям.

Основываясь на текущем функционале платформы и планах по дальнейшему развитию, можно утверждать, что данное решение представляет собой удобную платформу для общения учащихся на основе интересов, способный адаптироваться к меняющимся потребностям пользователей и становиться частью интегрированной образовательной среды будущего.

# Список литературы

1. Информация о школе «МАОУ "Школа Экодолье" Оренбургского района». Режим доступа URL: https://ecodolie56.gosuslugi.ru/ - дата обращения: 20.11.2023.
2. Информация о программе «PostgreSQL». Режим доступа URL: https://www.postgresql.org/- дата обращения: 01.12.2023.
3. Построение сайта на «html». Режим доступа URL: https://developer.mozilla.org/ru/docs/Learn/Gettingstartedwiththeweb/HTMLbasics- дата обращения: 15.12.2023.
4. Стилизация сайта через язык «css». Режим доступа URL: https://developer.mozilla.org/ru/docs/Learn/Getting\_started\_with\_the\_web/CSS\_basics - дата обращения: 16.12.2023.
5. Анимация кнопок и плавных переходов на языке «сss». Режим доступа URL: https://webdeasy.de/en/top-css-buttons-en/ - дата обращения: 17.12.2023.
6. Документация «PostgreSQL». Режим доступа URL: https://www.postgresql.org/docs/16/index.html - дата обращения: 18.12.2023.
7. Информация о работе «NodeJS» и «PostgreSQL». Режим доступа URL: https://rukovodstvo.net/posts/id\_836/?ysclid=lql8jiiehk117719315 - дата обращения: 19.12.2023.
8. Информация о отправки формы на языке «NodeJS». Режим доступа URL: https://metanit.com/web/nodejs/4.5.php?ysclid=lql8lwp85p148621976 дата обращения: 19.12.2023.
9. Информации об принятия отправленных данных на языке «NodeJS». Режим доступа URL https://expressjs.com/ru/guide/routing.html дата обращения: 20.12.2023.
10. Информация об отправки данных на языке «JavaScript» URL: https://learn.javascript.ru/fetch дата обращения: 21.12.2022.

# Приложения

**Приложение A**

**(*обязательно*)**

**Диаграмма прецедентов**

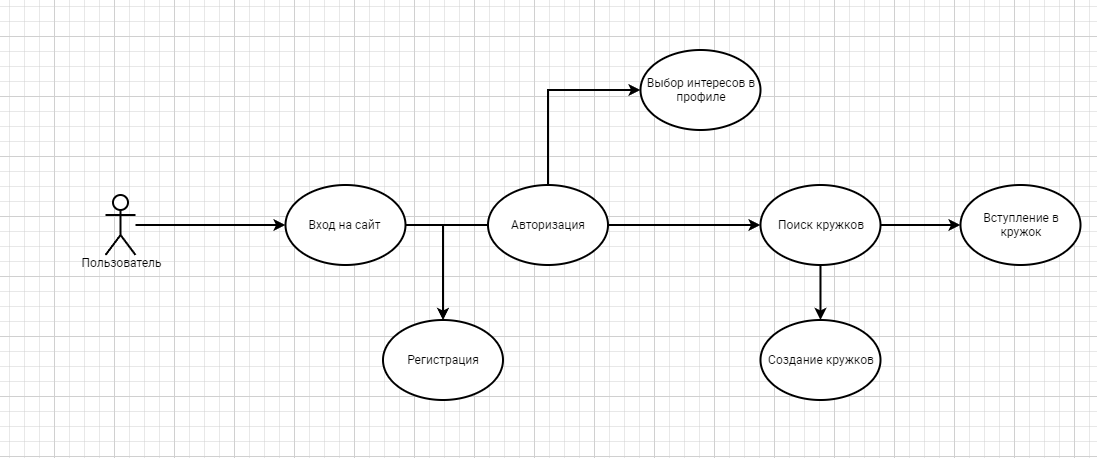
****

Рисунок А.1 - Диаграмма прецедентов

**Приложение Б**

**(*обязательно*)**

**Диаграмма последовательности**

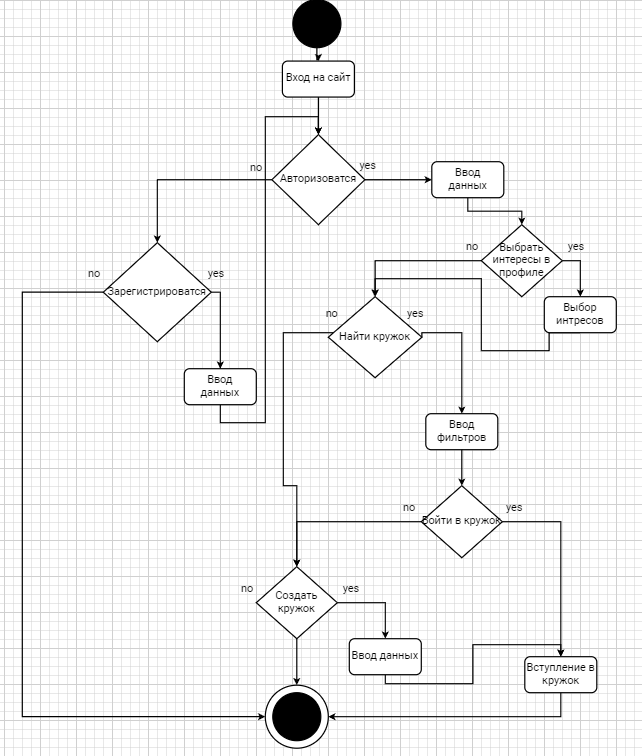
****

Рисунок Б.1 - Диаграмма последовательности

**Приложение В**

**(обязательно)**

**Диаграмма классов**

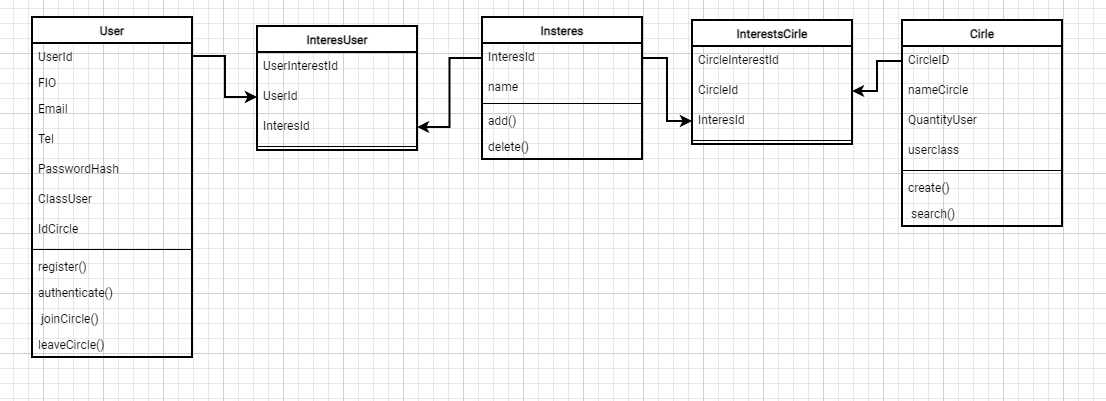
****

Рисунок В.1 – Диаграмма классов