**КРАТКАЯ**

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

**Программного комплекса «Studium»**

**СОДЕРЖАНИЕ**

[ОСНОВНАЯ ЧАСТЬ 3](#_Toc168739003)

[2 Технические характеристики 3](#_Toc168739004)

[2.2 Описание программного комплекса 3](#_Toc168739005)

[2.2.1 Общие сведения 3](#_Toc168739006)

[2.2.2 Функциональное назначение 4](#_Toc168739007)

[2.2.3 Описание логической структуры 5](#_Toc168739008)

[2.2.5 Вызов и загрузка 9](#_Toc168739009)

[2.2.6 Входные данные 11](#_Toc168739010)

[2.2.7 Выходные данные 11](#_Toc168739011)

[2.2.8 Рекомендации по использованию 12](#_Toc168739012)

[СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ 30](#_Toc168739013)

## **ОСНОВНАЯ ЧАСТЬ**

## **2 Технические характеристики**

## **2.2 Описание программного комплекса**

### 2.2.1 Общие сведения

Программный комплекс разрабатывалось в среде разработки Visual Studio Code, фреймворке ASP NET Core, основанном на языке программирования C# версии от .NET 8.0. Хранение данных в программном средстве основано на ORM Entity Framework Core и базе данных SQLite.

C# (Си шарп) – объектно-ориентированный язык программирования общего назначения. Разработан в 1998-2001 годах группой инженеров компании Microsoft под руководством Андерса Хейлсберга и Скотта Вильтаумота как язык разработки приложений для платформы Microsoft .NET Framework и .NET Core. Впоследствии был стандартизирован как ECMA-334 и ISO/IEC 23270. C# относится к семье языков с C-подобным синтаксисом, из них его синтаксис наиболее близок к C++ и Java. Язык имеет статическую типизацию, поддерживает полиморфизм, перегрузку операторов (в том числе операторов явного и неявного приведения типа), делегаты, атрибуты, события, переменные, свойства, обобщённые типы и методы, итераторы, анонимные функции с поддержкой замыканий, LINQ, исключения, комментарии в формате XML.

ASP.NET – свободно распространяемая платформа для разработки динамических сайтов и веб-приложений, созданная компанией Microsoft и являющаяся частью NET.Framework. Она является дальнейшим развитием более старой технологии Microsoft ASP и сохранила многие ее некоторые внешние признаки и функции, что упрощает переход разработчиков на использование нового инструмента. Разработка ASP.NET началась в 1997 году, когда Microsoft начала поиск новой модели веб-приложения. Первая версия была выпущена одновременно с платформой .NET Framework и позволяет писать веб-приложения и сайты на любом языке, поддерживаемом .NET. С помощью этой технологии были реализованы многие крупные веб-проекты, в том числе основной сайт разработчика, компании Microsoft.

Приставка «Core» в продуктах .NET является знаком более свежей и продвинутой технологии, а самое главное – кроссплатформенной. .NET Core появился на свет в 2016 году, что ознаменовало следующий шаг в развитии платформы для разработки программного обеспечения.

ORM (object-relational mapping – отображения данных на реальные объекты) – технология программирования, которая связывает базы данных с концепциями объектно-ориентированных языков программирования, создавая «виртуальную объектную базу данных».

Entity Framework (EF) представляет собой объектно-ориентированную, легковесную и расширяемую технологию от компании Microsoft для доступа к данным. EF является ORM-инструментом. Это значит, что EF позволяет работать базами данных, но представляет собой более высокий уровень абстракции: EF позволяет абстрагироваться от самой базы данных и ее таблиц и работать с данными независимо от типа хранилища. Если на физическом уровне мы оперируем таблицами, индексами, первичными и внешними ключами, но на концептуальном уровне, который нам предлагает Entity Framework, мы уже работаем с объектами. Entity Framework поддерживает множество различных систем баз данных. Таким образом, мы можем через EF работать с любой СУБД, если для нее имеется нужный провайдер.

LINQ (Language-Integrated Query) представляет простой и удобный язык запросов к источнику данных. В качестве источника данных может выступать объект, реализующий интерфейс IEnumerable (например, стандартные коллекции, массивы), набор данных DataSet, документ XML. Но вне зависимости от типа источника LINQ позволяет применить ко всем один и тот же подход для выборки данных.

SQLite – быстрая и легкая встраиваемая однофайловая СУБД на языке C, которая не имеет сервера и позволяет хранить всю базу локально на одном устройстве. Для работы SQLite не нужны сторонние библиотеки или службы.

В таблице 2.2.1.1 приведены минимальные системные требования к персональному компьютеру для использования серверной части (API) программного комплекса «Studium».

Таблица 2.2.1.1 – Минимальные системные требования

|  |  |
| --- | --- |
| **Параметр** | **Конфигурация** |
| Операционная система | Windows, Linux, MacOS |
| Процессор | 1.5 GHz |
| Оперативная память | 4 GB ОЗУ |
| Видеокарта | None |
| Звуковая карта | None |
| Место на диске | 5-10 GB |
| Docker | версии 24.0.7 |
| DirectX (\*для ОС Windows) | версии 9.0 |

### 2.2.2 Функциональное назначение

Функциональное назначение программного комплекса – предоставление возможности менеджмента данными учащихся и их успеваемости. Программный комплекс разрабатывается по методологии REST API, что предоставляет тесное взаимодействие с клиентами в лице браузеров.

В программном средстве предусмотрено пять ролей:

* администратор;
* секретарь;
* куратор;
* преподаватель;
* студент.

У каждой роли будет свое, личное ограничение в функциональных возможностях. Администратор – имеет возможность управления и настройки любой частью программного комплекса. Секретарь – по своей сути тот же администратор, только изменять сами сущности и поведения не имеет возможности. Куратор – тот же секретарь, только в области своей группы. Преподаватель имеет доступ только к отделу «Электронный журнал» и только к своим предметам. Студент же может просмотреть свою личную информацию и успеваемость, но не более.

### 2.2.3 Описание логической структуры

В соответствии с требованиями пользователь имеет возможность:

* войти в систему;
* переключать разделы;
* фильтровать данные;
* использовать CRUD-операции в базе данных;
* изменять данные аккаунта в профиле;
* завершить работу программного комплекса.

Разработанная ER-диаграмма (Приложение А) отражает отношения между таблицами и описывает систему на концептуальном уровне.

Разработанные диаграмма состояний (Приложение Б) и диаграмма вариантов использования (Приложение В) описывают роли программного комплекса, их взаимодействие с системой и состояние в момент работы программного комплекса.

Структура папок и файлов программного комплекса представлена на рисунке 2.2.3.1.

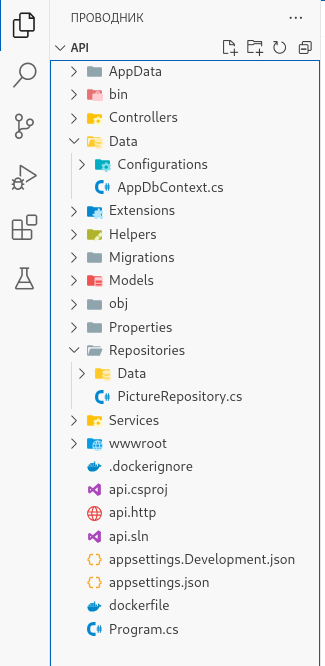


Рисунок 2.2.3.1 – Структура папок и файлов программного комплекса

Основополагающим классом является класс «Program», в нем задаются настройки проекта, необходимые для работы оного – добавление технологий и сущностей в конвейер обработки «Middleware».

Папка «AppData» содержит в себе базу данных проекта под названием «DataBase.db», а также папку «Pictures», где хранятся базовые фотографии для профилей учащихся.

Папка «Controllers» содержит в себе основную часть API для клиента – конечные точки, к которым делаются запросы. Данная папка содержит контроллер для каждой сущности базы данных:

* класс «GradeModelController» содержит методы для всех CRUD операций над сущностью «GradeModelEntity», добавление сущностей «GradeEntity» в список «GradeEntityList» сущности «GradeModelEntity», а также взятие списков сущностей под различных входящие данные;
* класс «GroupController» содержит методы для всех CRUD операций над сущностью «GroupEntity»;
* класс «PassportController» содержит методы для всех CRUD операций над сущностью «PassportEntity», а также методы для менеджмента сканов паспорта в файловой системе;
* класс «PersonController» содержит методы для всех CRUD операций над сущностью «PersonEntity», а также методы для менеджмента фотографий учащихся в файловой системе;
* класс «StudentController» содержит методы для всех CRUD операций над сущностью «StudentEntity». Также содержит вспомогательный метод для обновления дат зачисления/поступления.
* класс «SubjectController» содержит методы для всех CRUD операций над сущностью «SubjectEntity»;
* класс «UserController» содержит методы для некоторых CRUD операций над сущностью «UserEntity». Также данный класс содержит методы для получения сессии текущего пользователя, регистрации и логинации в системе.

Папка «Data» содержит конфигурационные файлы для базы данных, путем связи с ORM Entity Framework Core. Класс «AppDbContext» задаёт основную структуру таблиц и некоторых конфигураций базы данных. Папка «Configurations» содержит отдельные конфигурации для сущностей «UserEntity», «RoleEntity», «UserRoleEntity», «PermissionEntity», «RolePermissionEntity» и «GradeTypeEntity».

Папка «Extensions» содержит классы расширения для компонентов конвейера обработки «Middleware» проекта, а именно: класс «AppExtensions» расширяент функционал аутентификации и авторизации базовых классов системы ASP NET Core. Классы «RequirePermissionsAttribute» и «RequireRolesAttribute» добавляют атрибуты для использования на конечных точках – запроса списка ролей и разрешений для текущего вошедшего пользователя.

Папка «Helpers» содержит вспомогательные методы, перечисления и константы для осуществления работы некоторых частей проекта. Папка «Constants» содержит константы для куки, клэймов и папок с картинками. Папка «Enums» задаёт перечисления для ролей, разрешений и типов отметок, «RoleEntity», «PermissionEntity» и «GradeTypeEntity» соответственно. Класс «PersonHelper» добавляет метод для преобразования числового типа в пол студента.

Папка «Migrations» необходима для хранения миграций базы данных при изменениях сущностей в самой базе данных.

Папка «Models» содержит в себе:

* классы-модели для сущностей (entity) «GradeModel», «Group», «Passport», «Person», «Student», «Subject» и «User» описывают таблицы с полями в базе данных;
* классы-модели DTO для сущностей «GradeModel», «Group», «Passport», «Person», «Student», «Subject», «LoginUser», «RegisterUser», а также другие классы, которые описывают объекты с необходимыми полями для работы сервера с клиентом при запросах.

Папка «Properties» содержит все необходимые настройки для запуска проекта в различных состояниях: «Release», «Debug» и другие.

Папки «Repositories» и «Data» содержат основную составляющую часть проекта – бизнес логику, классы для работы с базой данных с помощью технологий EF Core и LINQ и файловой системой:

* класс «PictureRepository» содержит методы для загрузки, удаления и взятия картинок проекта в файловую систему и наоборот;
* класс «DataRepositoryBase» является абстрактным классом для описания классов при работе с сущностями базы данных с помощью обобщений, наследования, полиморфизма и других приемов объектно-ориентированного программирования;
* класс «GradeModelRepository» содержит методы для всех CRUD операций над таблицей в базе данных сущности «GradeModelEntity», а также для добавления сущностей «GradeEntity»;
* класс «GroupRepository» содержит методы для всех CRUD операций над таблицей в базе данных сущности «GroupEntity»;
* класс «PassportRepository» содержит методы для всех CRUD операций над таблицей в базе данных сущности «PassportEntity»;
* класс «PersonRepository» содержит методы для всех CRUD операций над таблицей в базе данных сущности «PersonEntity»;
* класс «StudentRepository» содержит методы для всех CRUD операций над таблицей в базе данных сущности «StudentEntity»;
* класс «SubjectRepository» содержит методы для всех CRUD операций над таблицей в базе данных сущности «SubjectEntity», а также для получения средних отметок за семестр по предметам, общую среднюю отметку по датам и другим входящим данным;
* класс «UserRepository» содержит методы для некоторых CRUD операций над таблицей в базе данных сущности «UserEntity», а также других вспомогательные методов для преобразования ролей и разрешений из сущностей в перечисления.

Папка «Services» содержит в себе методы для работы контроллеров. По своей сути данные методы сопровождают некоторые сущности, преобразуют из одного формата в другой:

* класс «Authorizing» предоставляет методы для авторизации пользователей в системе по заданным специальным критериям;
* класс «AesWorker» предоставляет методы для шифрования и дешифрования информации при помощи алгоритма Aes;
* класс «HttpContextService» предоставляет методы для работы с контекстом Http – получение пользователя, токенов и клэймов из сессии;
* класс «JwtProvider» служит для генерации и получения клэймов токена JWT. Использует файлы конфигурации;
* классы «PermissionService» и «RoleService» необходимы для взятия соответствующих сущностей для пользователя;
* класс «StringHasher» предоставляет методы для хэширования паролей и генерации GUID;
* класс «UserService» служит для предоставления вспомогательных методов при работе с пользователями.

Папка «wwwroot» содержит картинки для учащихся и паспортов.

Файл «dockerfile» содержит конфигурационные настройки для запуска docker-контейнера.

Файл «appsettings.json» содержит в себе конфигурационные настройки проекта: описание ролей и разрешений, данные для JWT токена, строку подключения к базе данных и другие.

### 2.2.5 Вызов и загрузка

Запуск программного комплекса может быть осуществлен двумя способами, каждый из них требует предустановленные в системе .NET SDK или Docker соответственно.

Для запуска программного комплекса нативно, при помощи .NET SDK необходимо открыть терминал (консоль) и перейти в папку самого программного комплекса и ввести команду «dotnet run» (рисунок 2.2.5.1).

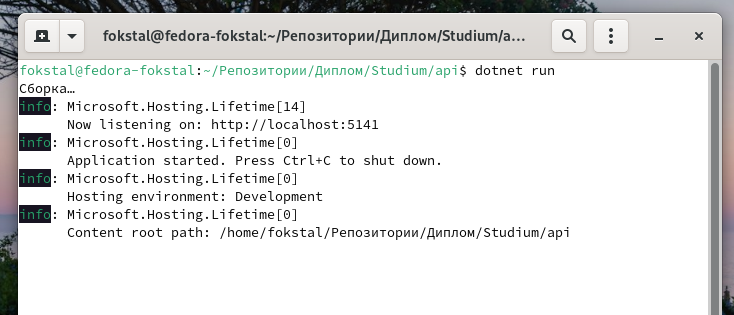


Рисунок 2.2.5.1 – Запуск программного комплекса с помощью команды «dotnet run»

После запуска отобразится дополнительная информация по управлению запущенным сервером. При помощи 1 строки Info можно нажать на адрес и перейти в браузер, где уже будет настроено подключение к серверу с помощью технологии Swagger UI (рисунки 2.2.5.2-2.2.5.3).

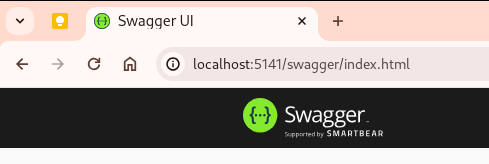


Рисунок 2.2.5.2 – Адресная строка подключения через браузер к серверу

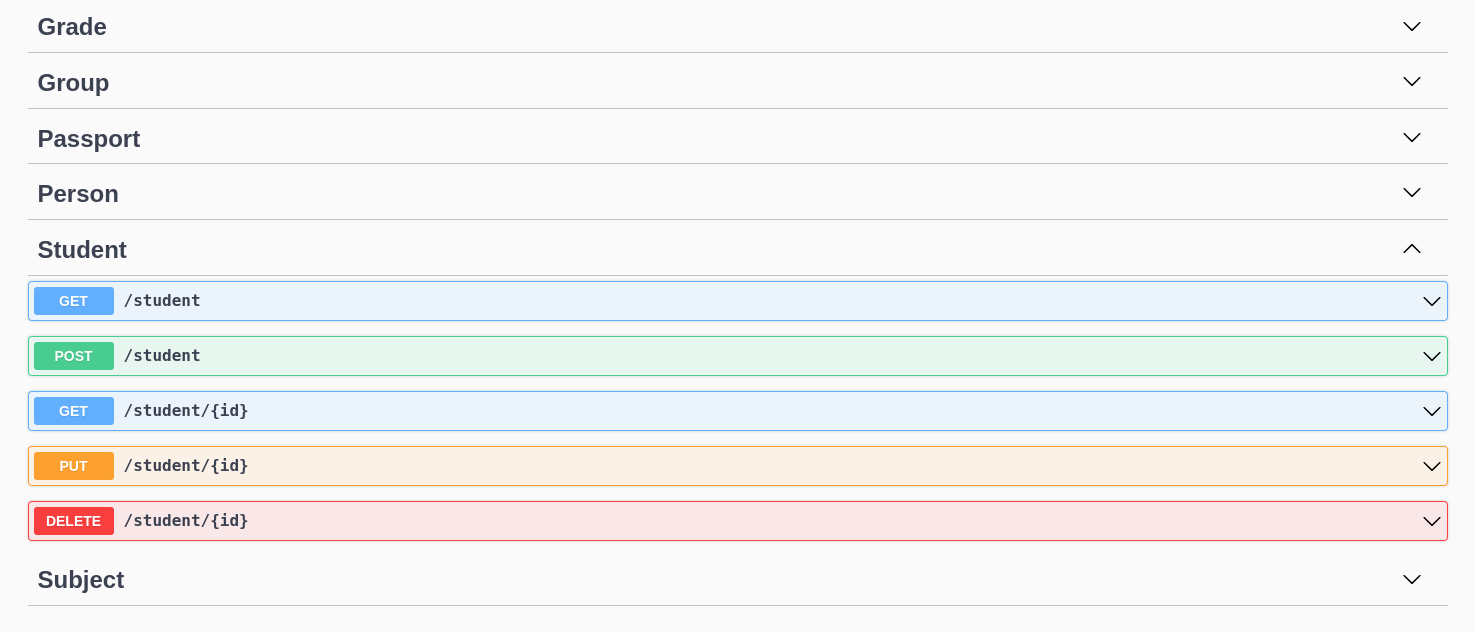


Рисунок 2.2.5.3 – Общий вид страницы подключения

Для завершения работы достаточно вернуться в терминал (консоль) с запущенным сервером и нажать сочетание клавиш «Ctrl+C» или закрыть окно терминала (консоли).

При запуске при помощи Docker подключение к серверу остаётся точно таким же, только не будет связи с Swagger UI, только уже сама связь с клиентом Studium. Для начала необходимо создать новый образ проекта на основе образа .NET SDK от Microsoft, для этого необходимо открыть папку с проектом в терминале (консоли) и вписать команду «docker build -t api-studium .» (рисунок 2.2.5.4).

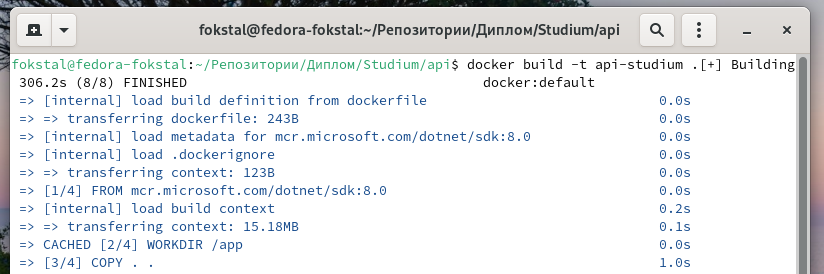


Рисунок 2.2.5.4 – Создание образа программного комплекса

Создание образа занимает некоторое время. После создания образа необходимо запустить контейнер из этого образа при помощи команды «docker run -p 5141:8080 --name api-studium-cont --rm api-studium» (рисунок 2.2.5.5).

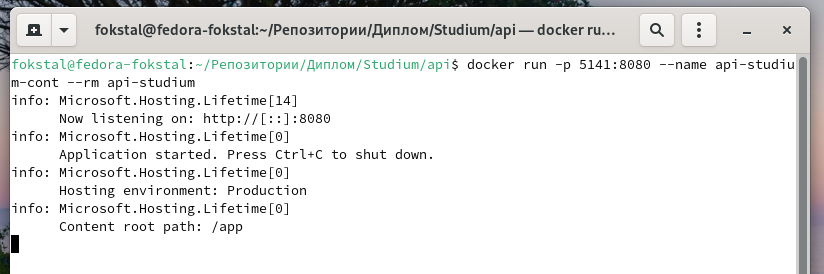


Рисунок 2.2.5.5 – Запуск контейнера программного комплекса

Для завершения работы можно воспользоваться методом, описанным в запуске при помощи .NET SDK или же открыть новое окно в терминале (консоли) и ввести команду «docker stop api-studium-cont» (рисунок 2.2.5.6).

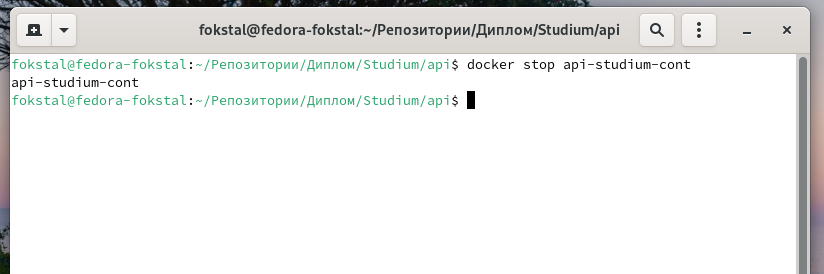


Рисунок 2.2.5.6 – Завершение работы контейнера

### 2.2.6 Входные данные

Входными данными программного комплекса являются:

изображения аватаров пользователей и сканов паспортов;

личные данные пользователей;

данные для описания групп, предметов и отметок;

отметки учащихся по различным предметам;

учетные данные пользователей.

### 2.2.7 Выходные данные

Выходными данными программного комплекса являются:

информирующие сообщение об статусе запросов – ошибки, перенаправления, дополнительные данные;

результаты запросов на ресурс, такие как создание, обновление и удаление объектов;

данные, запрашиваемые с ресурсов.

### 2.2.8 Рекомендации по использованию

Информация по запуску программного комплекса представлена в пункте «2.2.5 Вызов и загрузка» данного подраздела. В данном пункте будут рассмотрены использование всего программного комплекса «Studium», а также отдельное использование серверной части вместе с основными рекомендациями по работе с технологией Swagger UI.

Для начала работы с программным комплексом «Studium», необходима регистрация пользователя в системе, которую может осуществить секретарь или администратор. После чего пользователь может пройти аутентификацию и авторизацию в системе (рисунок 2.2.8.1).

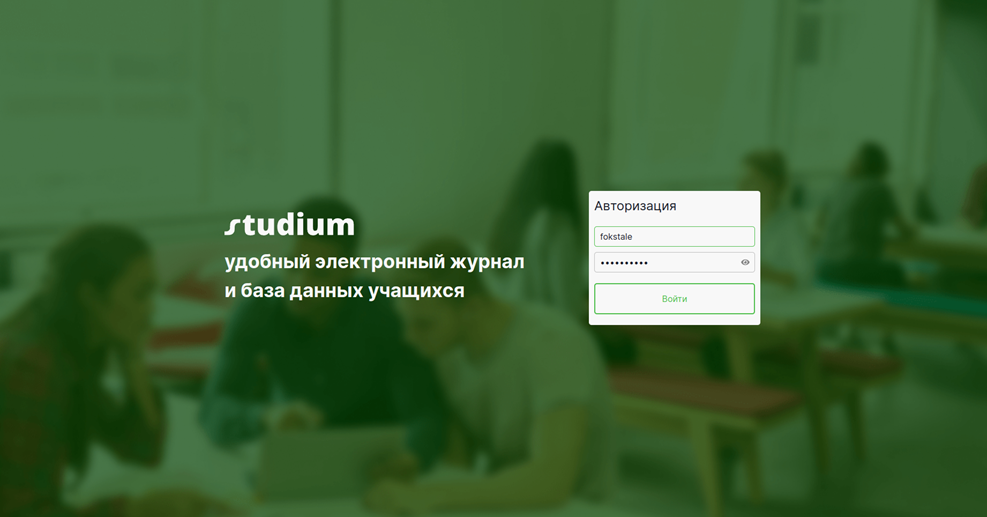


Рисунок 2.2.8.1 – Страница входа в систему

После успешного входа и, если пользователю позволяет доступ – пользователь автоматически перейдет на страницу «Список учащихся», иначе переход выполнится на страницу «Личный кабинет». В шапке сайта появится навигация и несколько функциональных кнопок. На странице «Список учащихся» представлена таблица со списком доступных для текущего пользователя учащихся, поиск, а также кнопка «Фильтры» (рисунок 2.2.8.2).

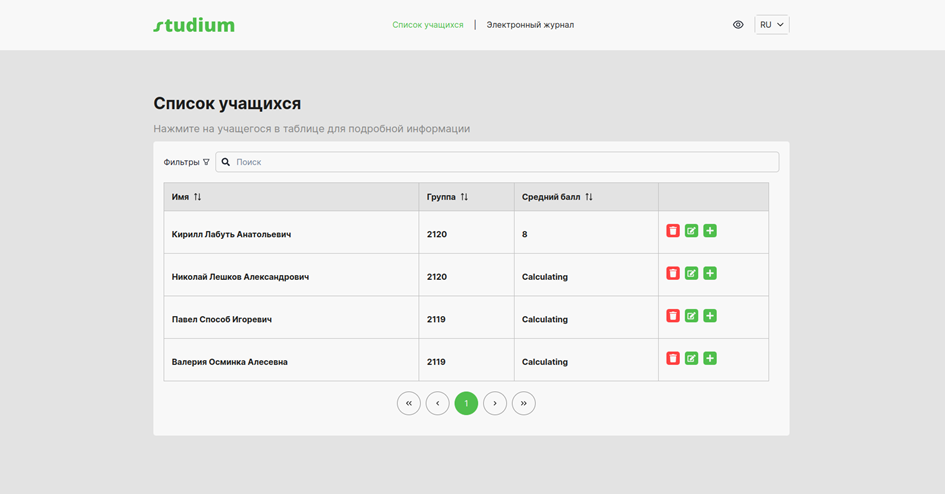


Рисунок 2.2.8.2 – Страница «Список учащихся»

На странице «Личный кабинет» пользователю доступны вкладки данных об аккаунте, успеваемости, если у пользователя есть роль студента, а также кнопка выхода из системы (рисунок 2.2.8.3).

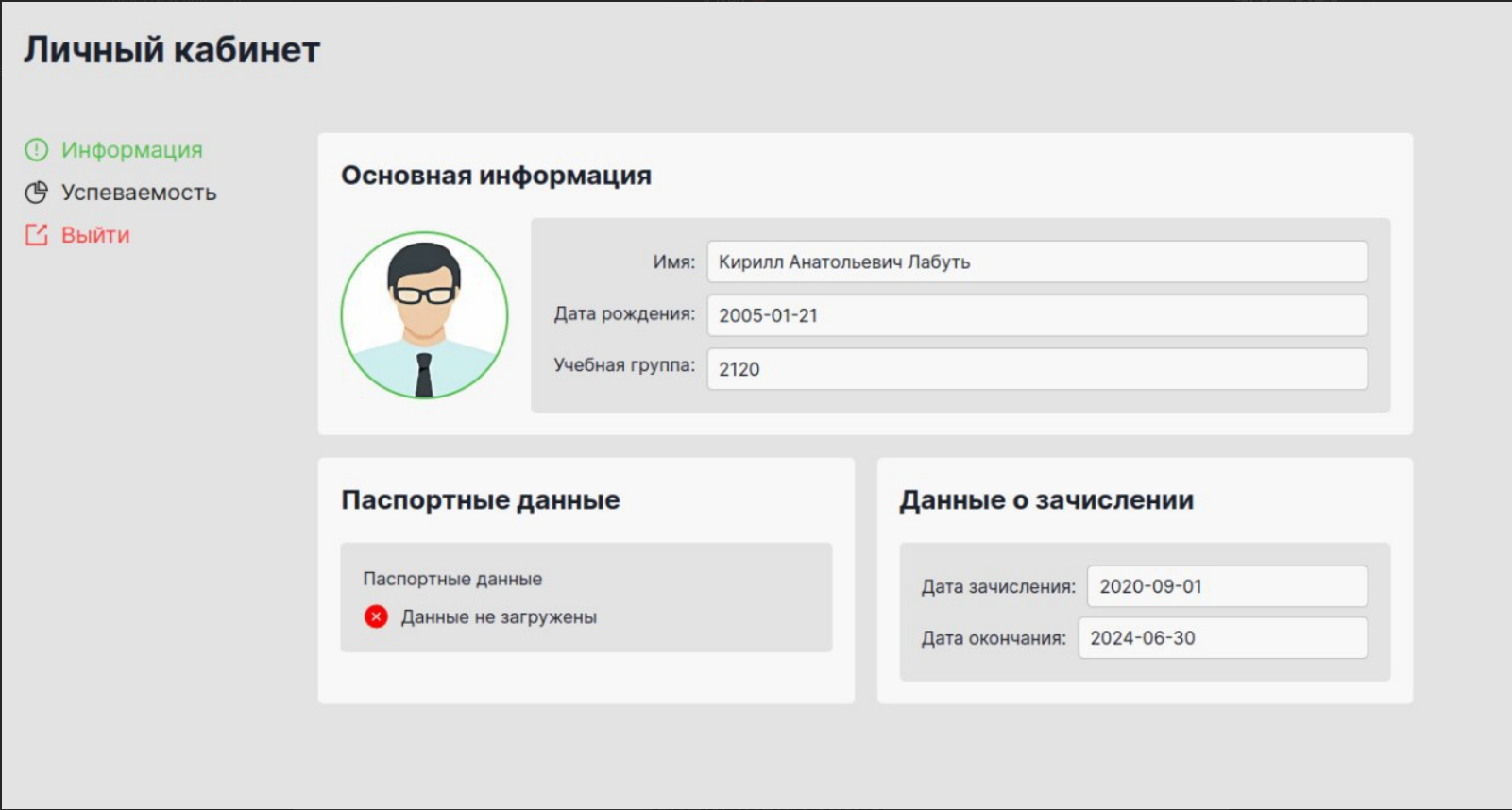


Рисунок 2.2.8.3 – Страница «Личный кабинет»

В правом верхнем углу шапки сайта находятся: кнопка для переключения режима для слабовидящих, кнопка смены языка и картинка аватара пользователя (рисунок 2.2.8.4).

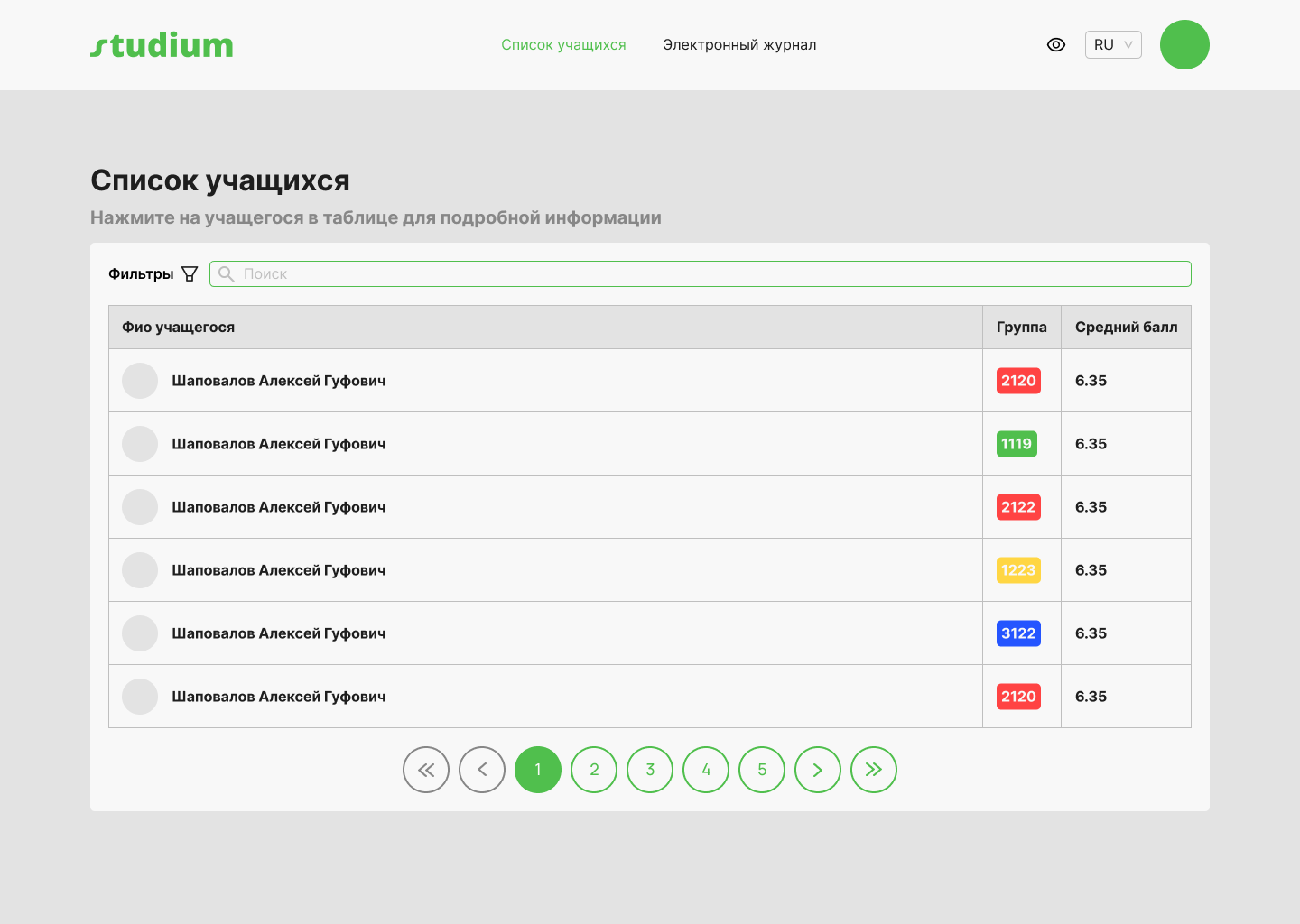


Рисунок 2.2.8.4 – Дополнительные управляющие кнопки в шапке сайта

При нажатии на кнопку переключения режима для слабовидящих, сайт переходит в соответствующий режим, появляются функциональные кнопки и внешний вид сайта изменяется (рисунок 2.2.8.5).

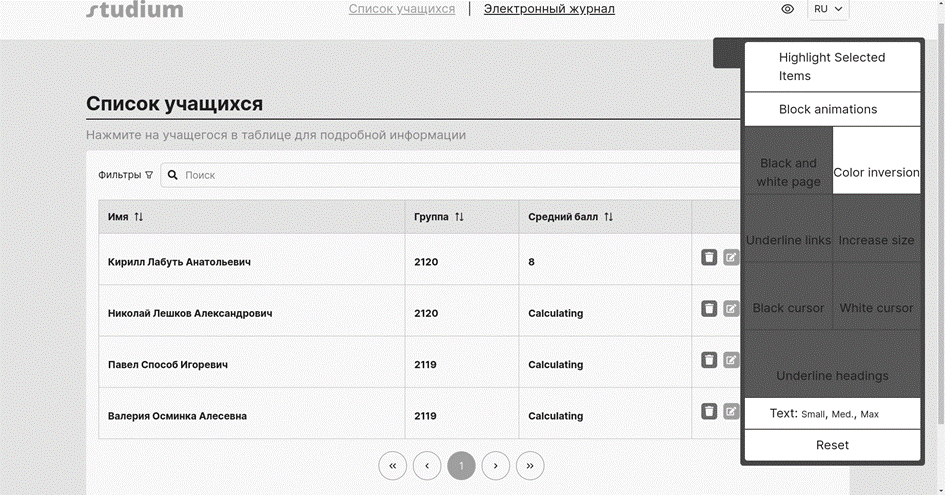


Рисунок 2.2.8.5 – Сайт в режиме для слабовидящих

При нажатии на кнопку смены языка раскрывается список выбора языка, на который будет переведён весь текст сайта (рисунок 2.2.8.6).



Рисунок 2.2.8.6 – Меню выбора языка

При нажатии на кнопку «Фильтры» раскрывается меню фильтрации данных в таблице. Можно выбрать учебную группу, по которой будет производится поиск, выбрать промежуток дат зачисления, а также показать выпуск и зачисленных этого учебного года. По нажатию на кнопку «Поиск» в таблице отображаются данные с учётом фильтров, а при нажатии кнопки «Очистить фильтры» все фильтры поиска сбрасываются (рисунок 2.2.8.7).

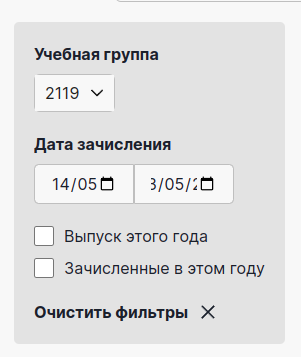


Рисунок 2.2.8.7 – Окно выбора фильтров

При нажатии на учащегося в таблице раскрывается меню дополнительной информации, в котором показаны некоторые дополнительные данные (рисунок 2.2.8.8).

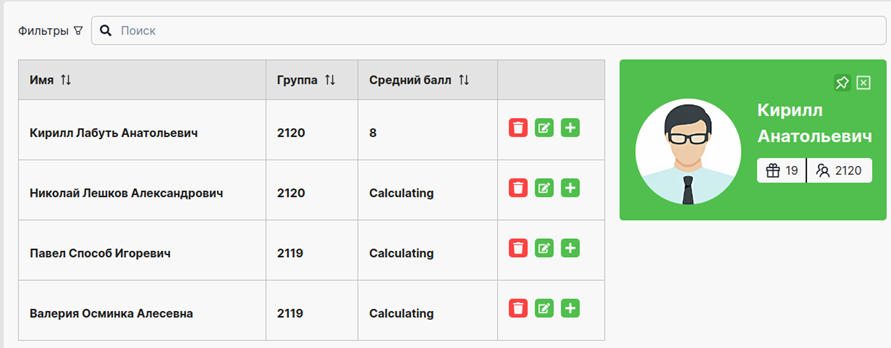


Рисунок 2.2.8.8 – Меню дополнительной информации

Данный компонент имеет кнопки взаимодействия, а именно – ссылку на страницу учащегося (рисунок 2.2.8.9) и кнопка для закрытия самого меню. Для перехода на следующую страницу можно воспользоватся навигационным меню в шапке сайта (рисунок 2.2.8.10), а для перехода обратно на главную страницу, нужно нажать на ссылку-логотип в левом верхнем углу (рисунок 2.2.8.11).

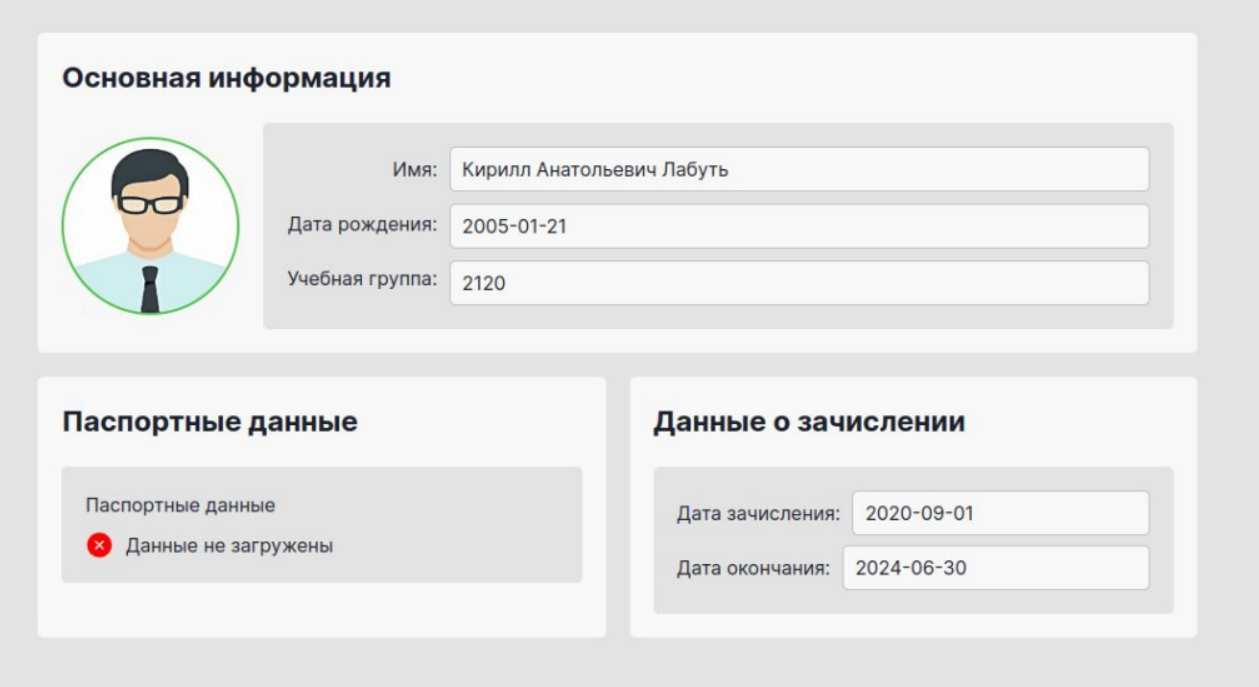
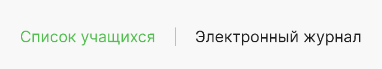
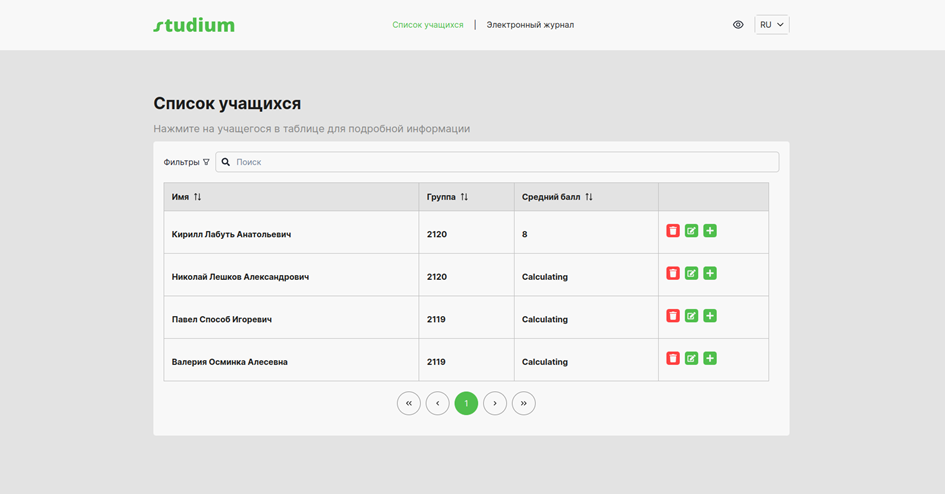


Рисунок 2.2.8.9 ­– Страница учащегося

****

**Рисунок 2.2.8.10 – Навигационное меню**



**Рисунок 2.2.8.11 – Ссылка-логотип для перехода на главную страницу**

Следующая страница – «Электронный журнал», которая содержит меню для фильтрации данных – 3 выпадающих списка для выбора группы, учащегося и предмета, а также кнопку добавления оценки (рисунок 2.2.8.12).



Рисунок 2.2.8.12 – Фильтры для электронного журнала

При нажатии на кнопку «Добавить оценку» открывается модальное окно (рисунок 2.2.8.13), которое нужно заполнить данными и нажать на кнопку «Добавить отметку».

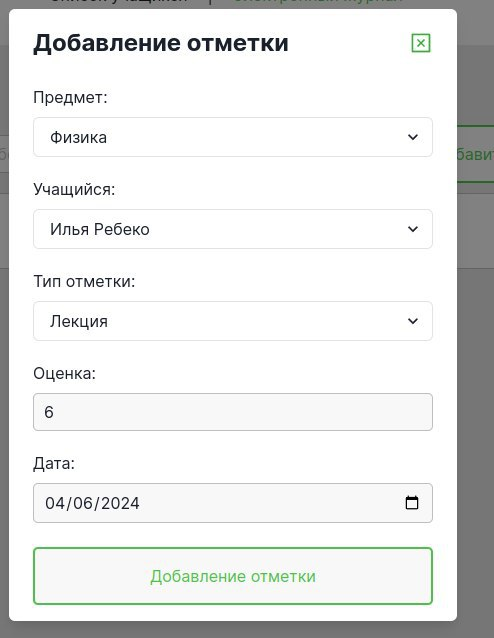


Рисунок 2.2.8.13 – Модальное окно добавления оценки

Сам журнал представляет собой таблицу, где столбцами выступают имя учащегося и даты с отметками (рисунок 2.2.8.14), а строками выступают имена учащихся со значениями отметок на определенную дату (рисунок 2.2.8.15).

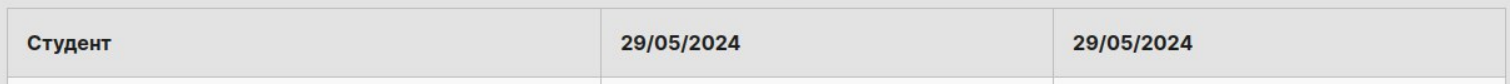


Рисунок 2.2.8.14 – Столбцы таблицы электронного журнала

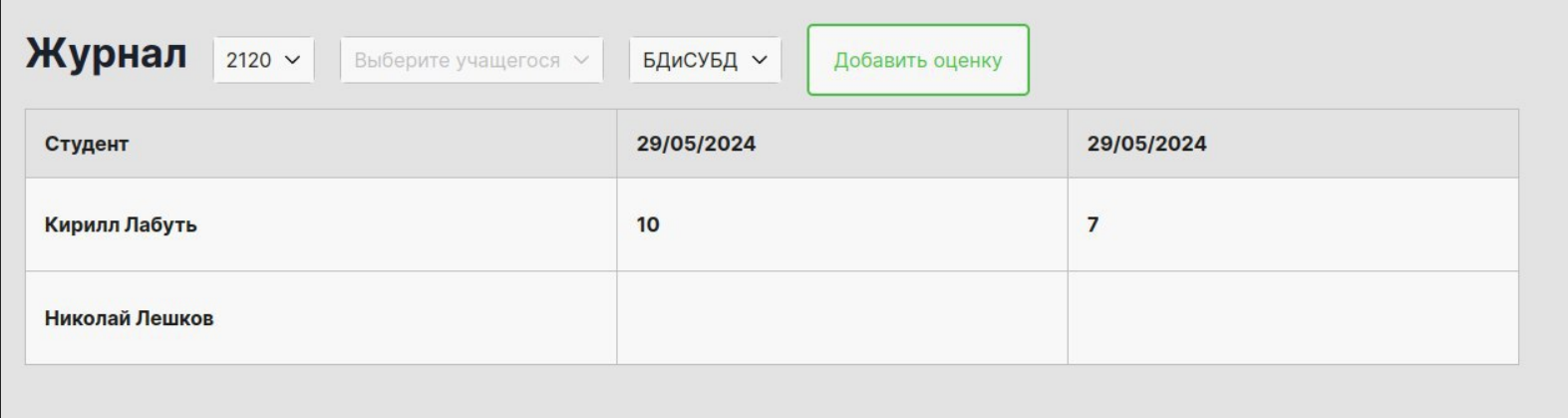


Рисунок 2.2.8.15 – Общий вид электронного журнала

При нажатии на картинку аватара учащийся попадает на страницу «Личный кабинет» (рисунок 2.2.8.16).

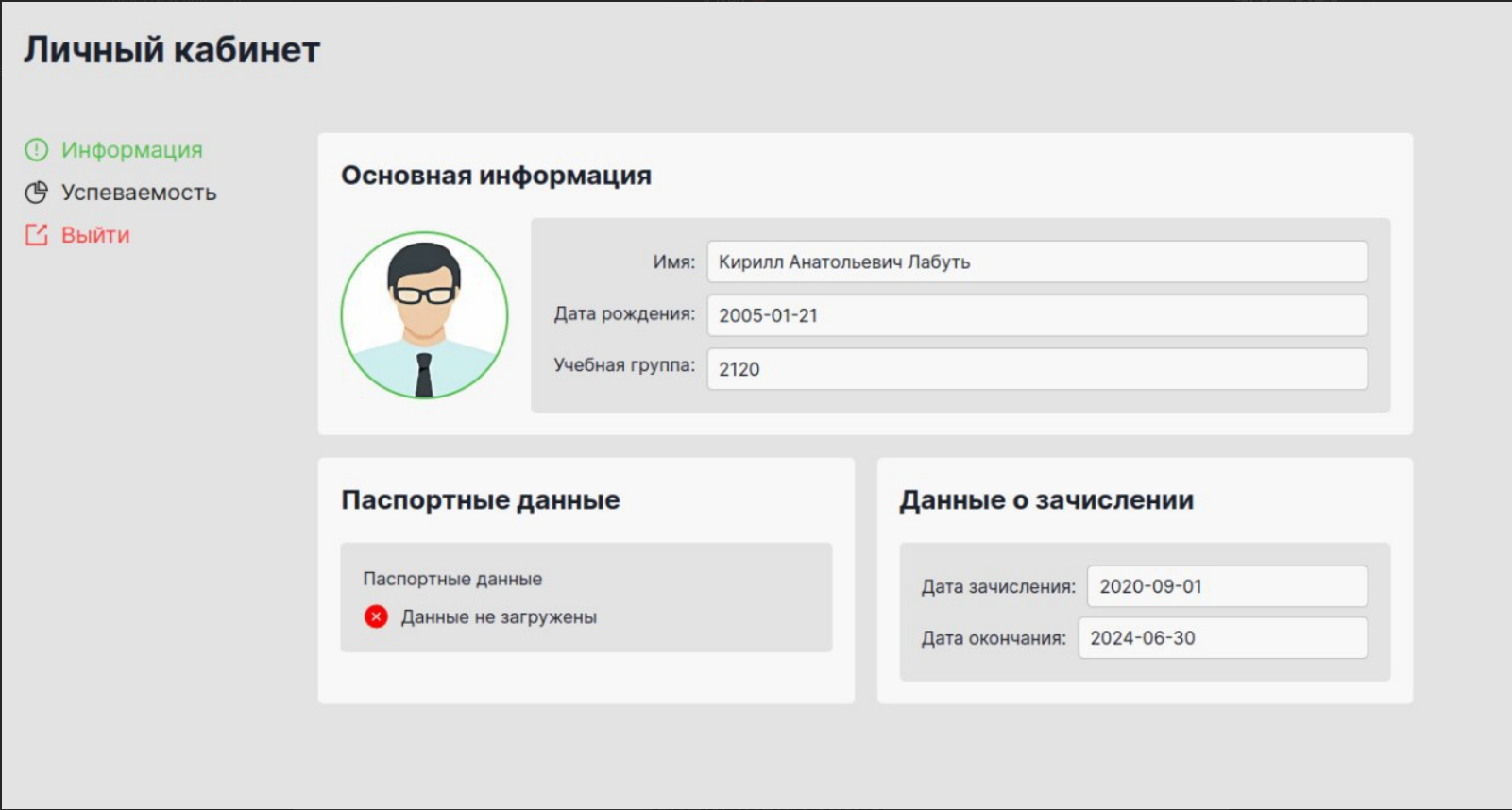


Рисунок 2.2.8.16 – Личный кабинет учащегося

По умолчанию на данной странице будет открыта вкладка «Информация». Она разделена на 3 блока:

* блок «Основная информация» с аватаром, именем, датой рождения и группой учащегося, а также с его личным ключом;
* блок «Паспортные данные» с информацией о загрузке паспортных данных и возможностью просмотра;
* блок «Данные о движении» с информацией о зачислении, отчислении, академических отпусках и прочее.

На левой стороне страницы расположено меню (рисунок 2.2.8.17) с 3 кнопками:

* «Информация»;
* «Успеваемость»;
* «Выйти».

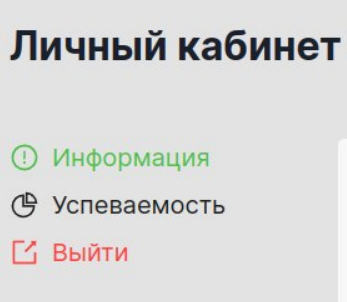


Рисунок 2.2.8.17 – Меню страницы «Личный кабинет»

Вкладка «Успеваемость» представлена на рисунке 2.2.8.18.

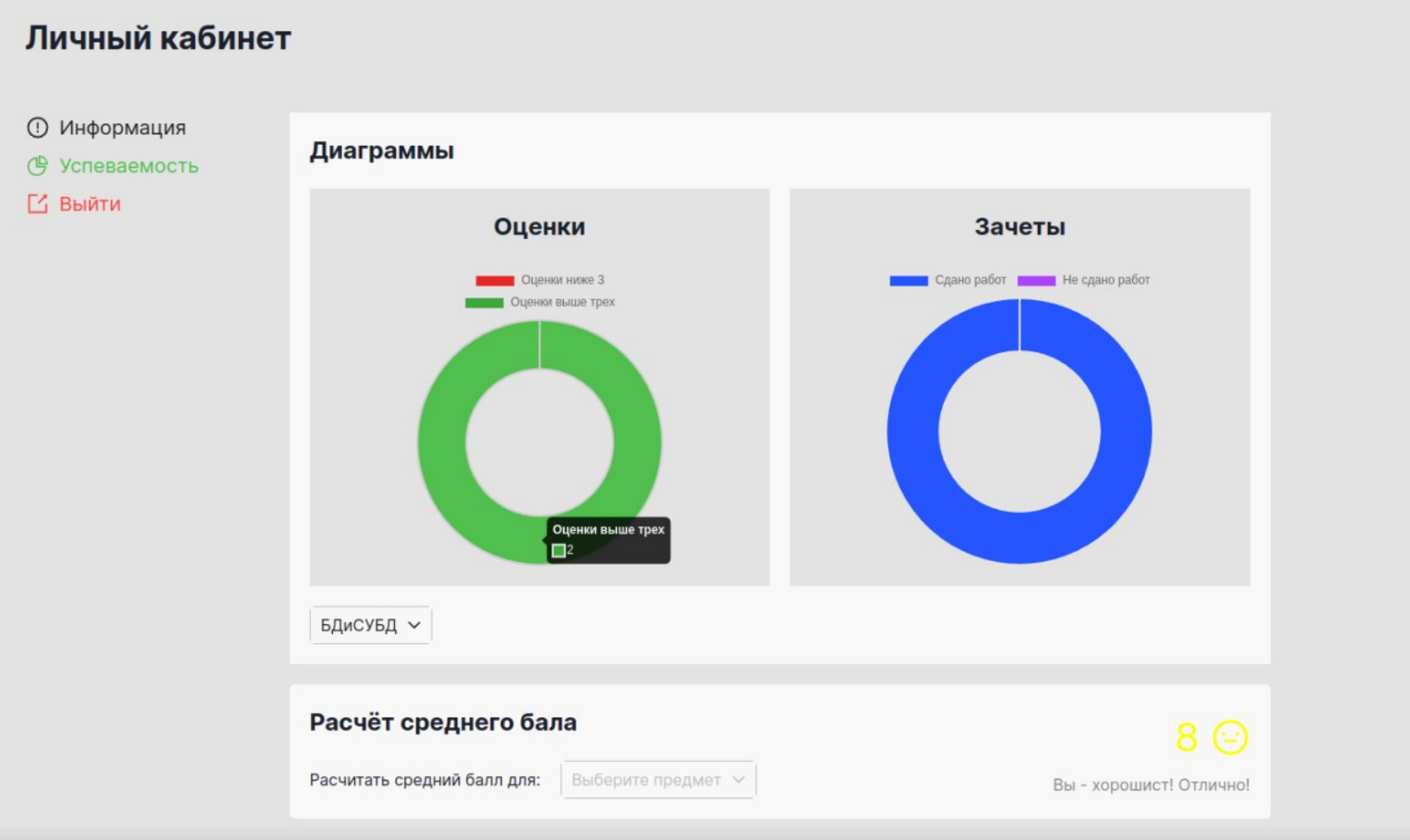


Рисунок 2.2.8.18 – Вкладка «Успеваемость»

Данная вкладка разделена на 2 блока:

* блок «Диаграммы», в котором можно выбрать предмет (рисунок 2.2.8.19) и получить готовые круговые диаграммы со статистикой успеваемости (рисунок 2.2.8.20);
* блок «Расчёт среднего балла», в которой можно выбрать предмет (рисунок 2.2.8.21) и рассчитать средний балл (рисунок 2.2.8.22).

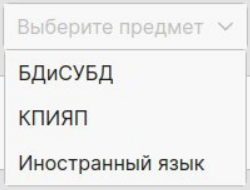


Рисунок 2.2.8.19 – Меню выбора предмета для отображения диаграмм

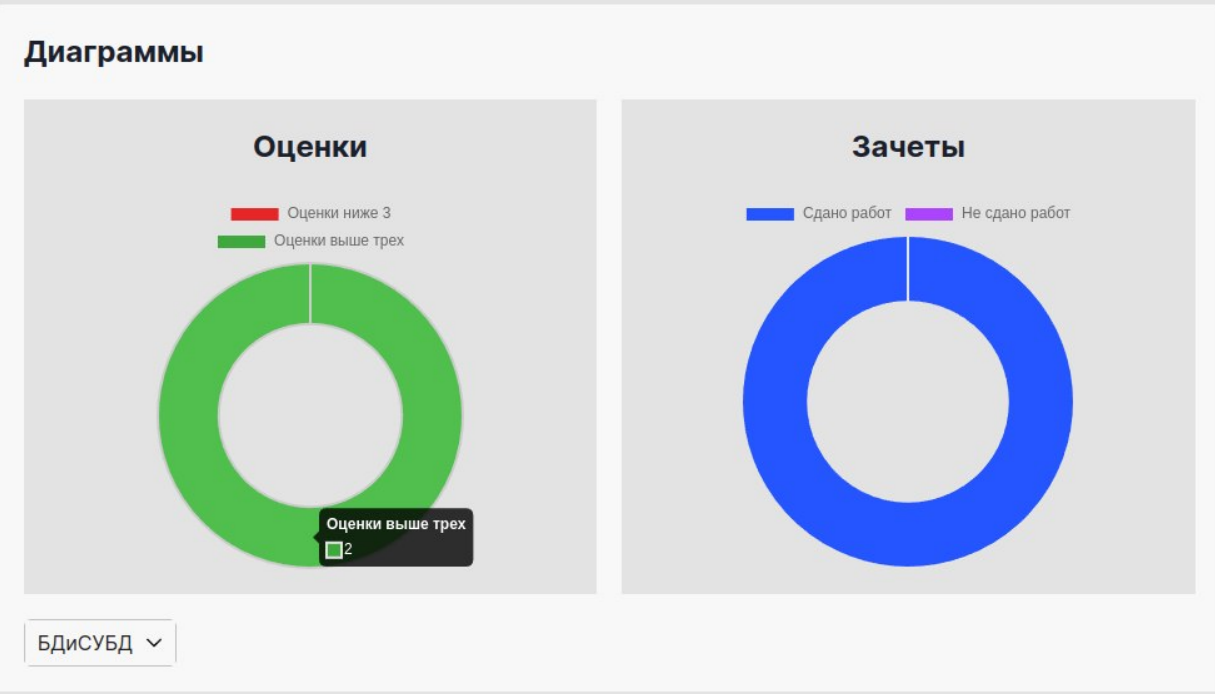


Рисунок 2.2.8.20 – Диаграммы по предмету

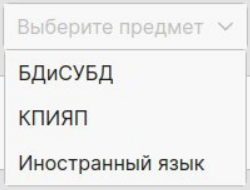


Рисунок 2.2.8.21 – Меню выбора предмета для отображения среднего балла

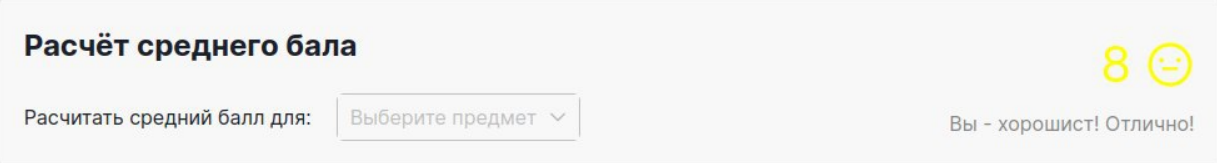


Рисунок 2.2.8.22 – Средний балл по всем предметам

Для секретаря доступна формы создания и редактирования учащегося (рисунки 2.2.8.23-2.2.8.24).

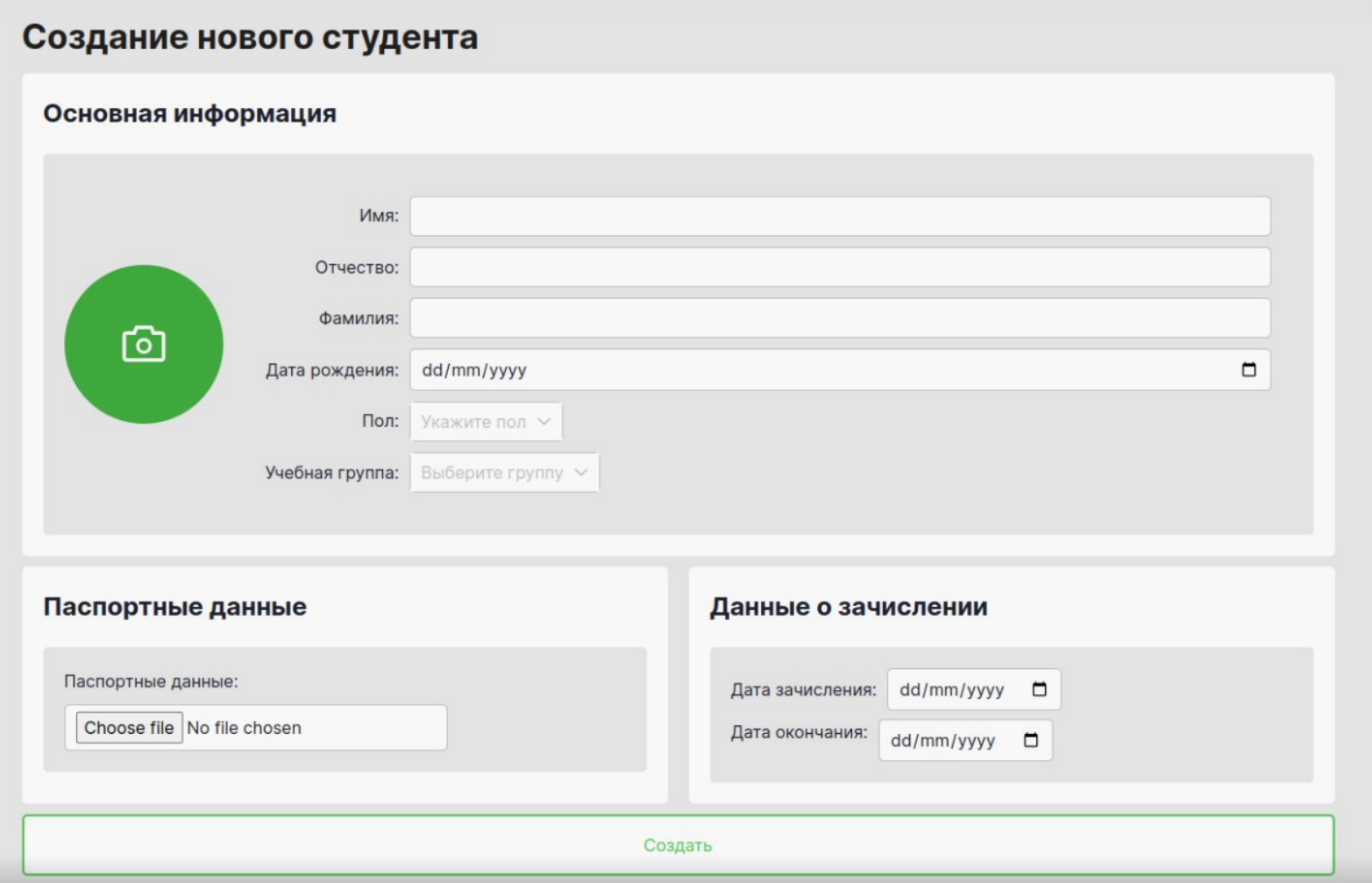


Рисунок 2.2.8.23 – Форма создания учащегося

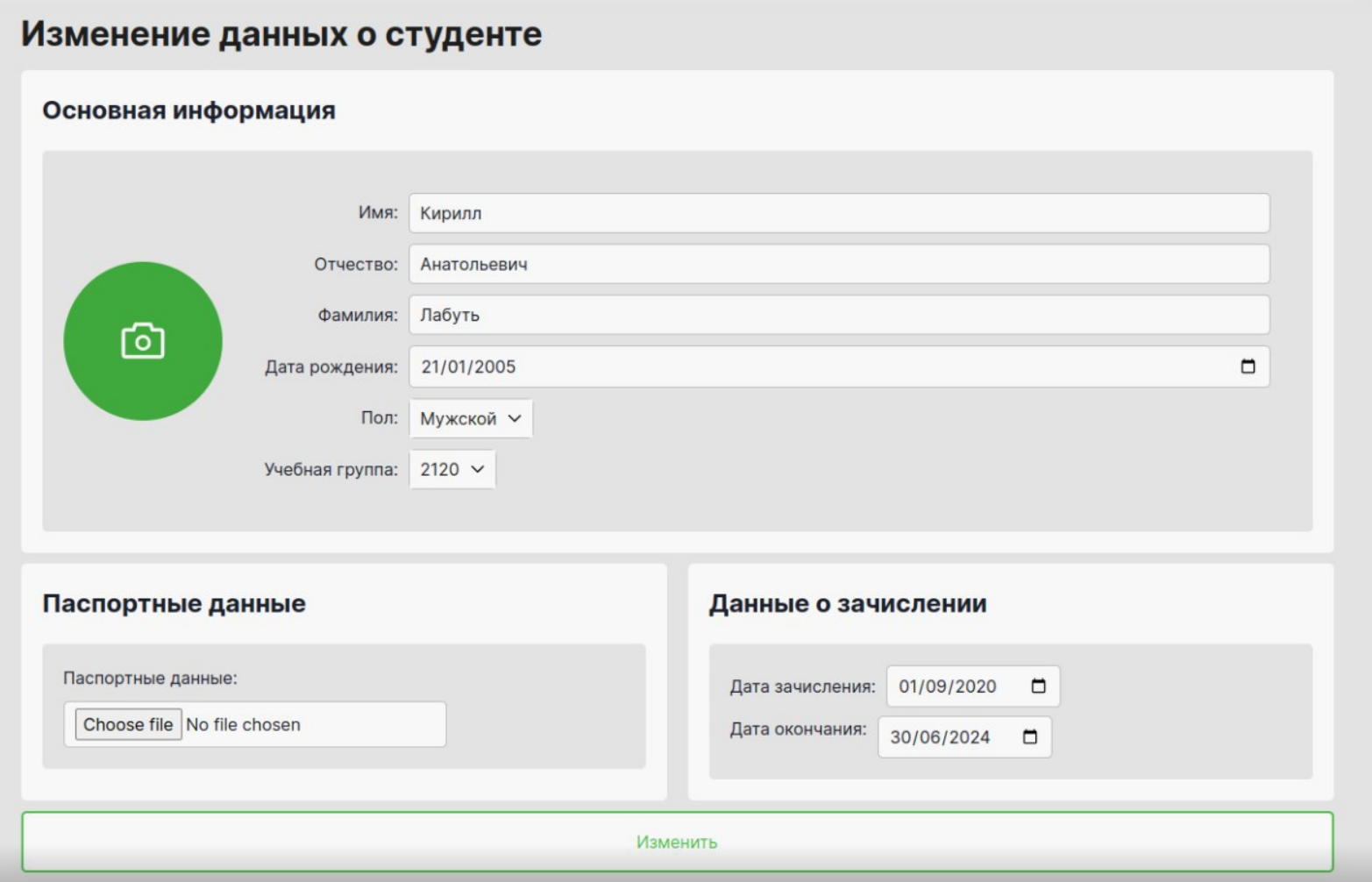


Рисунок 2.2.8.24 – Форма изменения данных учащегося

В данных формах можно заполнить все те данные, что отображаются в личном кабинете и в списке учащихся.

Так же можно перейти на страницу с группами (рисунок 2.2.8.25). Если пользователь является секретарем, перейти на страницу создания или редактирования (рисунки 2.2.8.26-2.2.8.27) посредством нажатия одной из трех кнопок (рисунок 2.2.8.28).

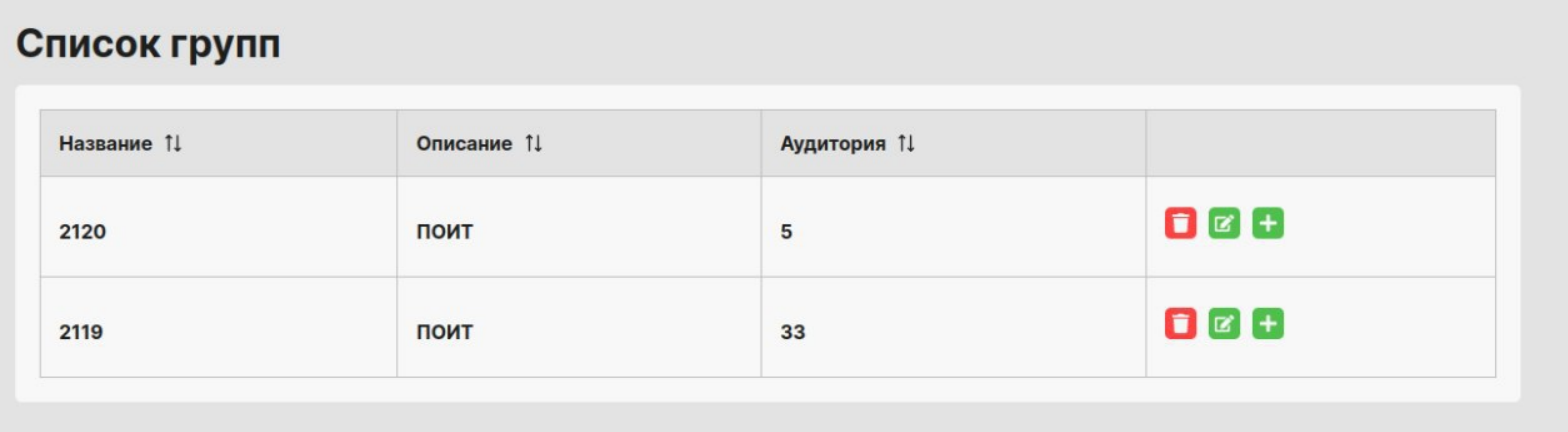


Рисунок 2.2.8.25 – Страница с списком групп

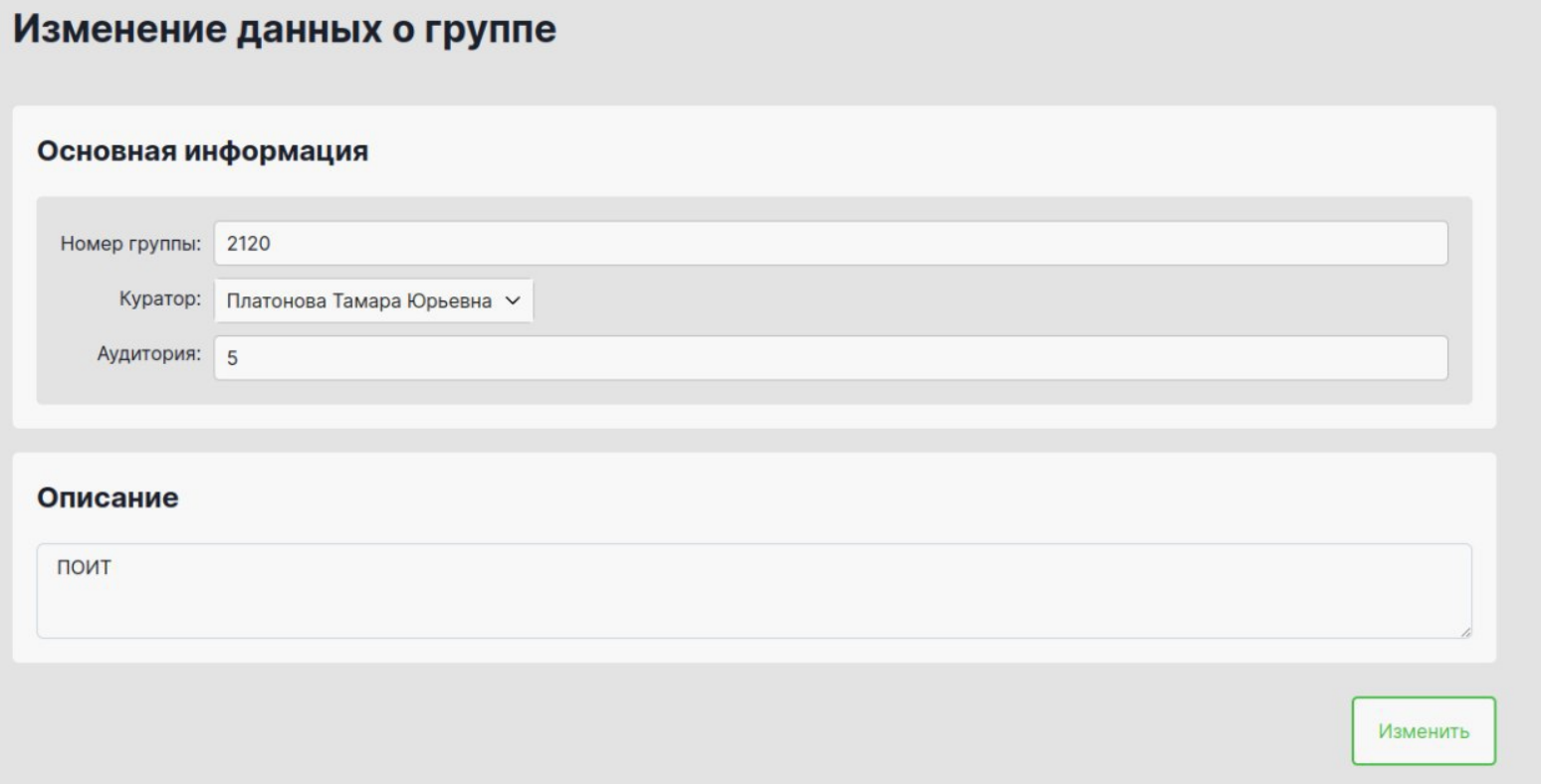


Рисунок 2.2.8.26 – Страница редактирования группы

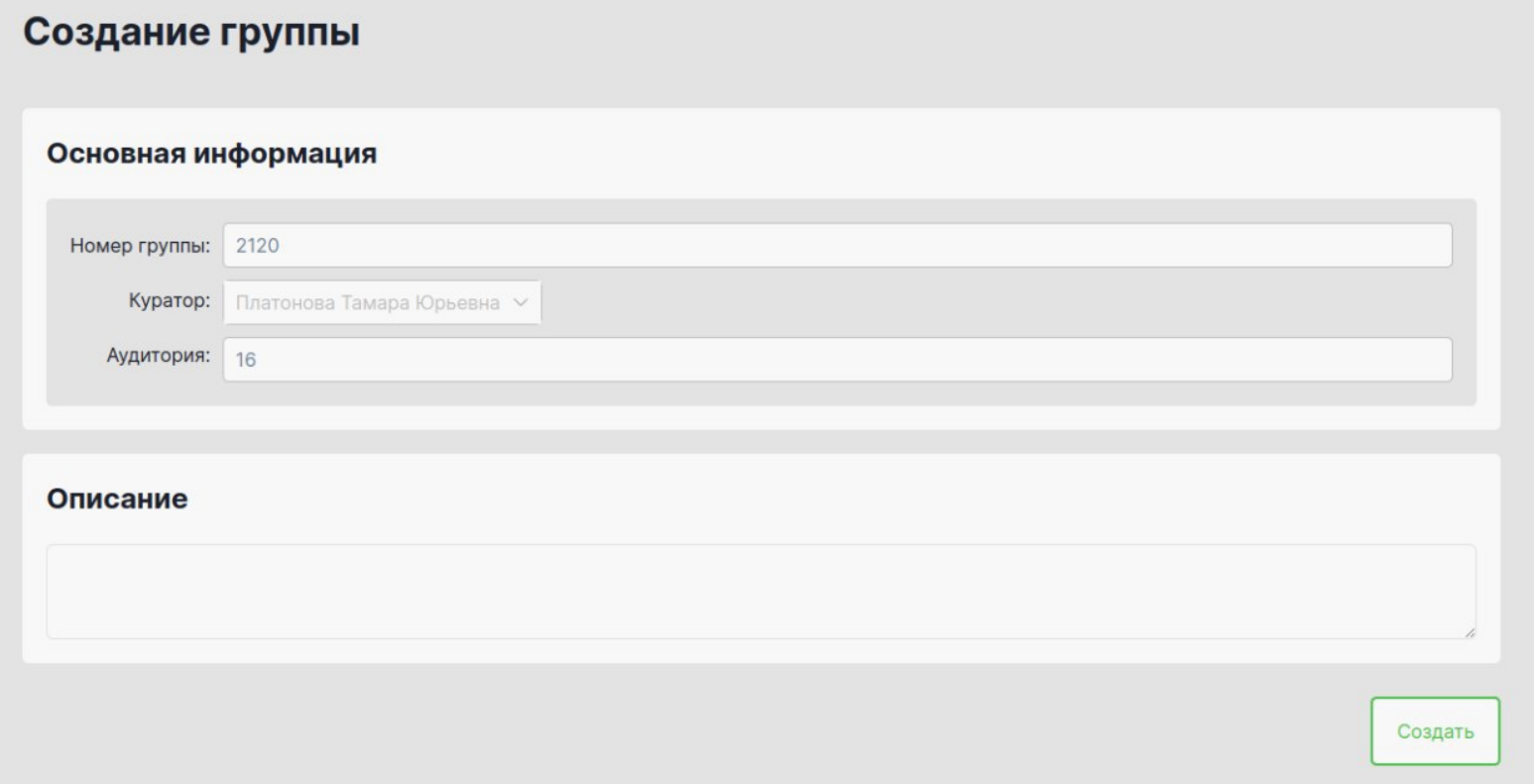


Рисунок 2.2.8.27 – Страница создания группы



Рисунок 2.2.8.28 – Кнопки изменения, создания и удаленния

Также есть страница с предметами (рисунок 2.2.8.29), на которой расположены кнопки создания, редактирования и удаления, посредством которых можно открыть страницу создания (рисунок 2.2.8.30) и страницу изменения (рисунок 2.2.8.31).

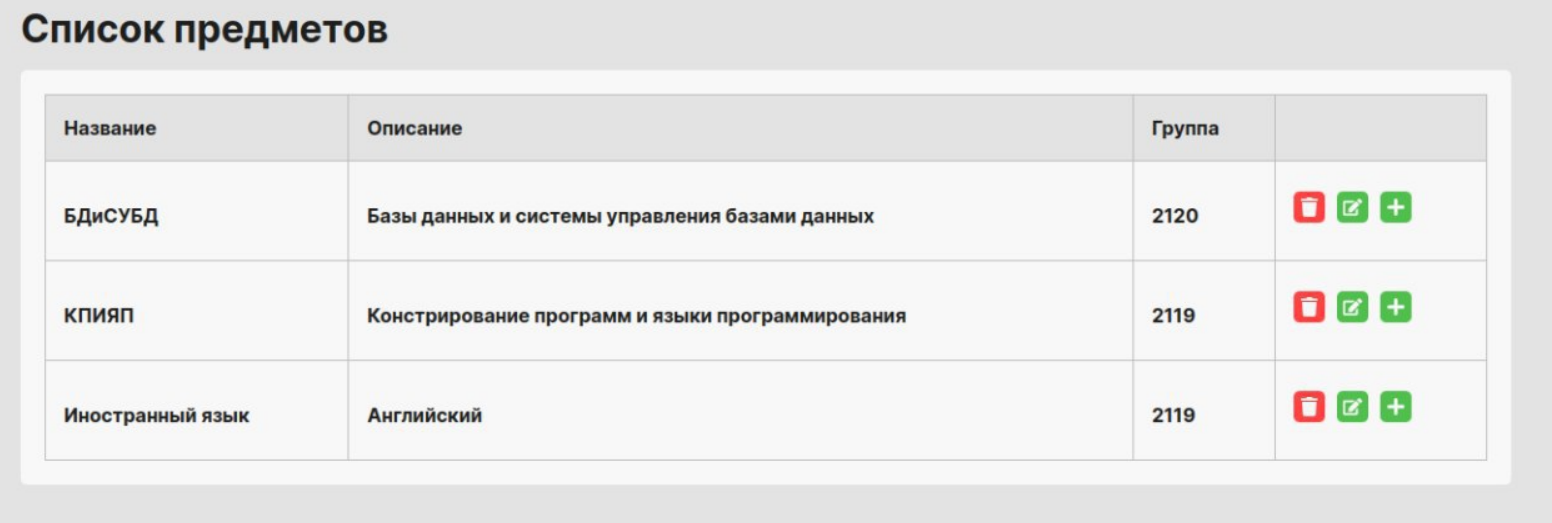


Рисунок 2.2.8.29 – Страница со списком предметов

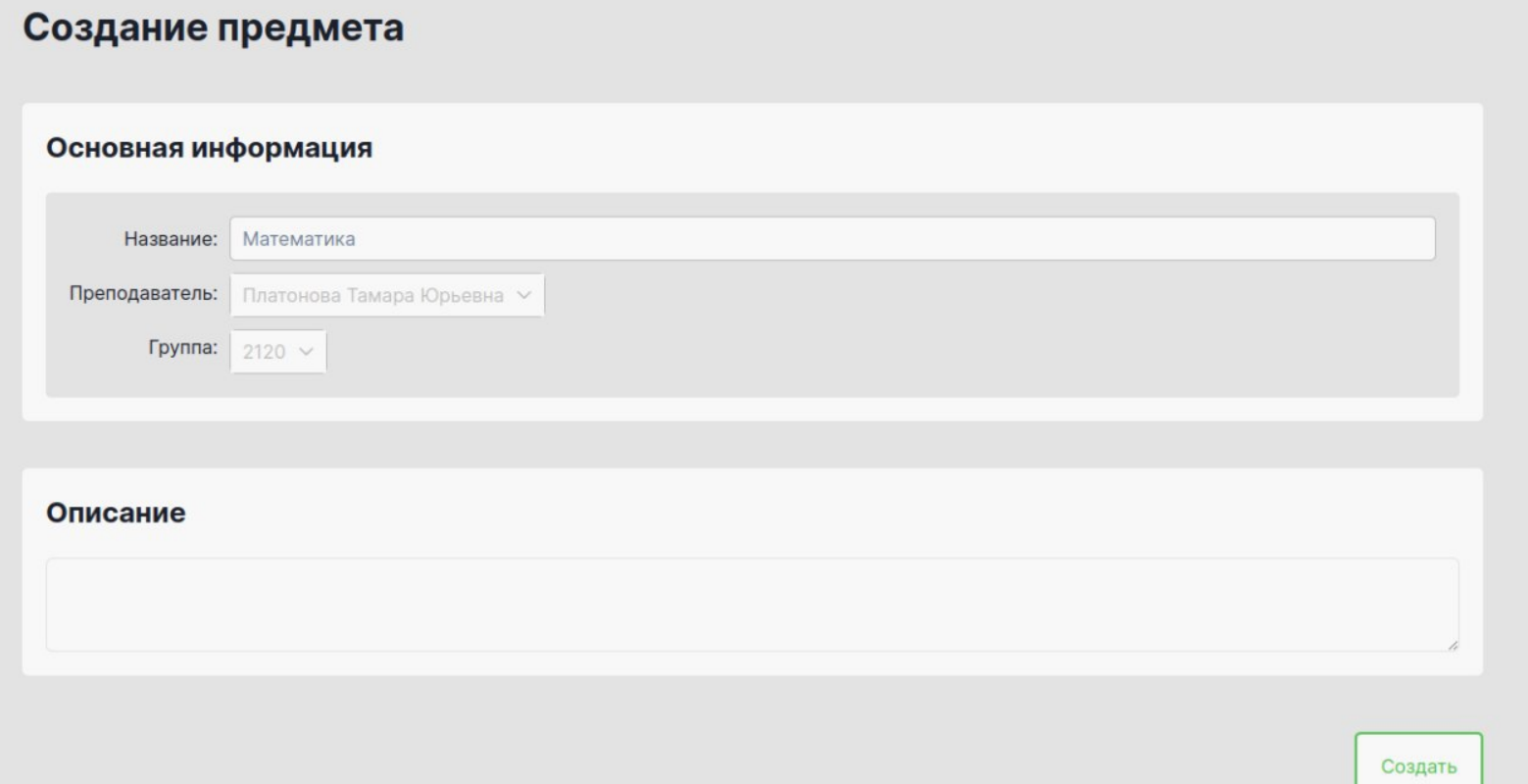


Рисунок 2.2.8.30 – Страница создания предмета

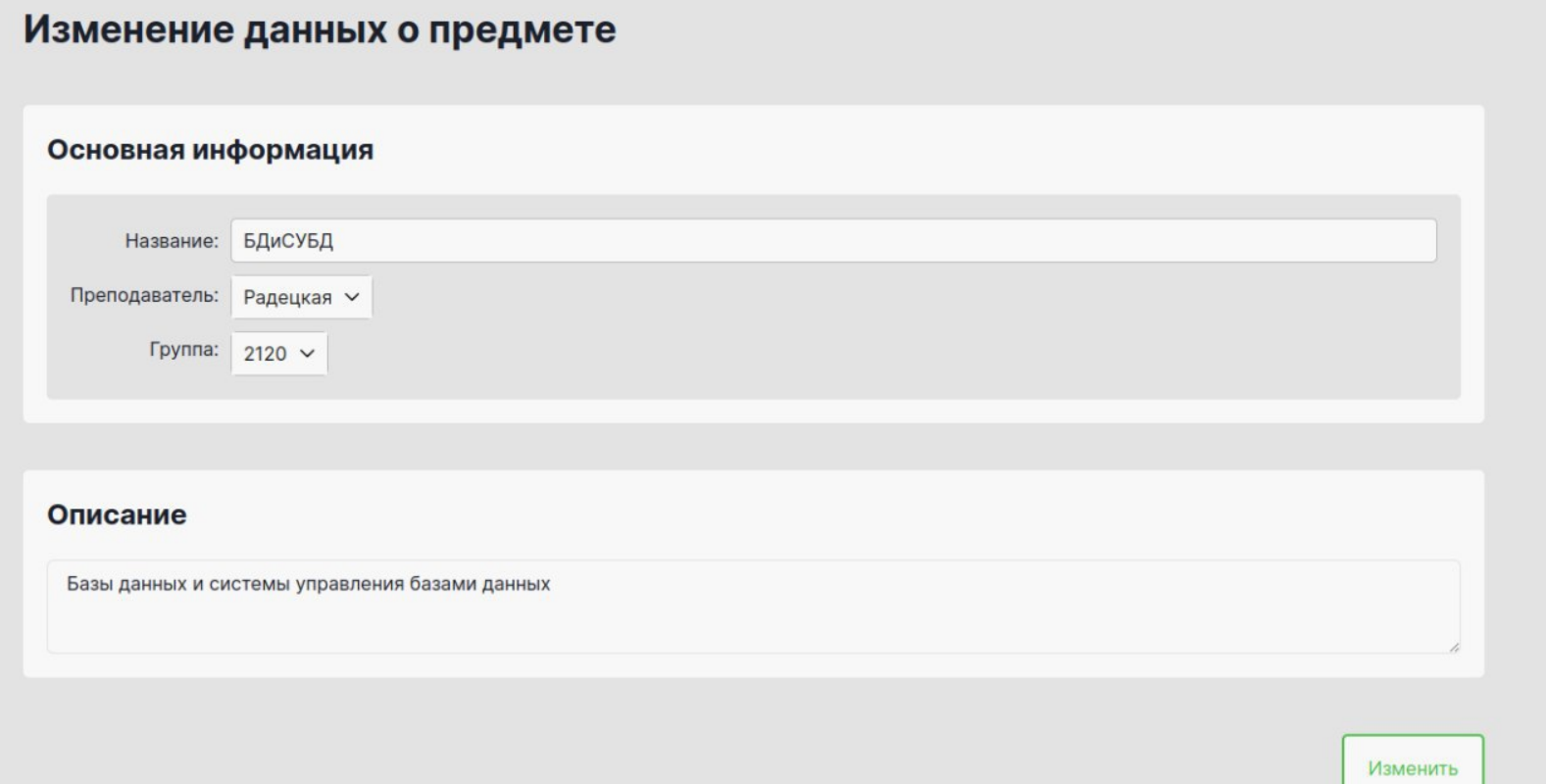


Рисунок 2.2.8.31 – Страница редактирования предмета

Также для администратора есть страница для просмотра всех пользователей зарегистрированных в системе (рисунок 2.2.8.32) и страница для их создания (рисунок 2.2.8.33).

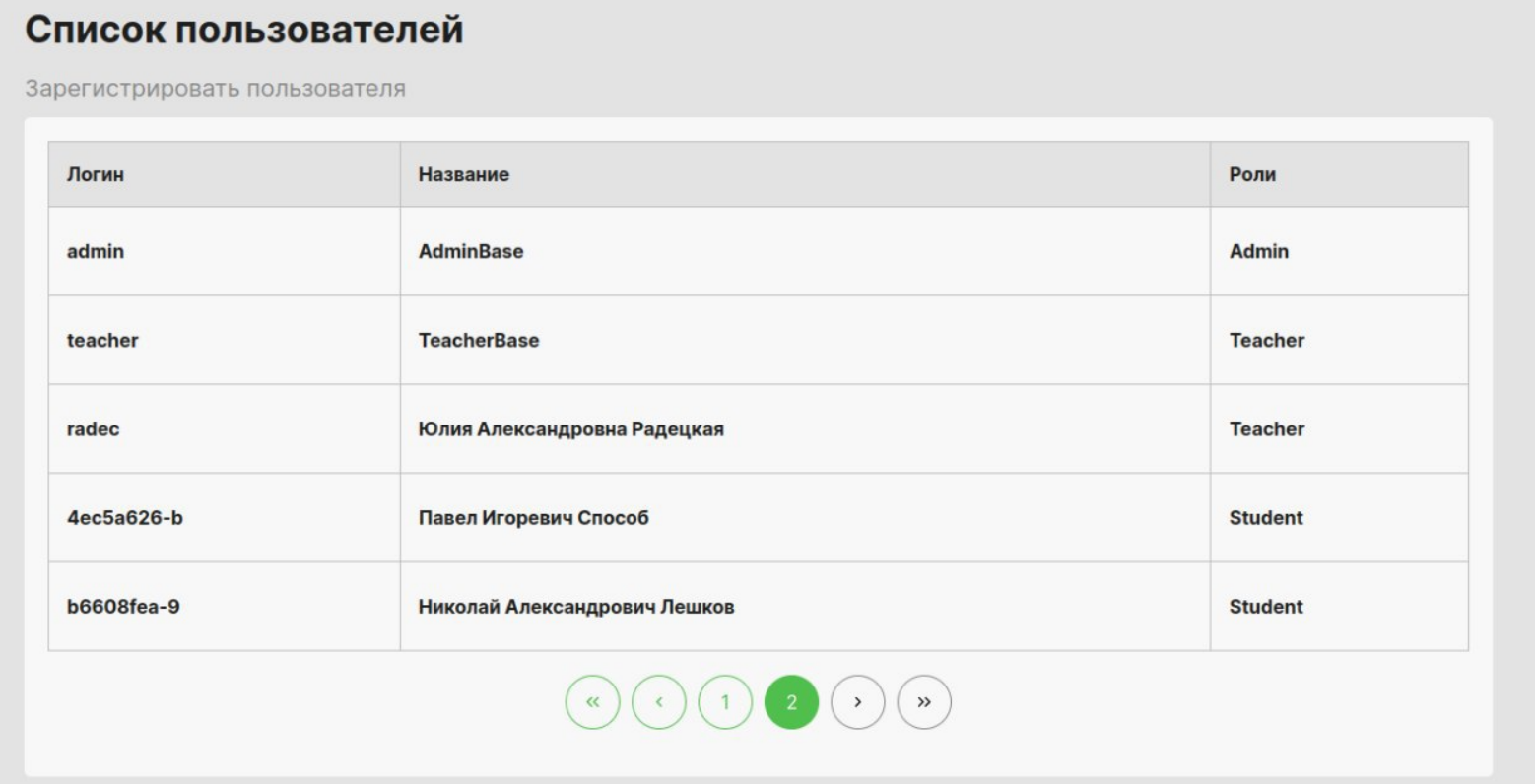


Рисунок 2.2.8.32 – Страница со списком пользователей

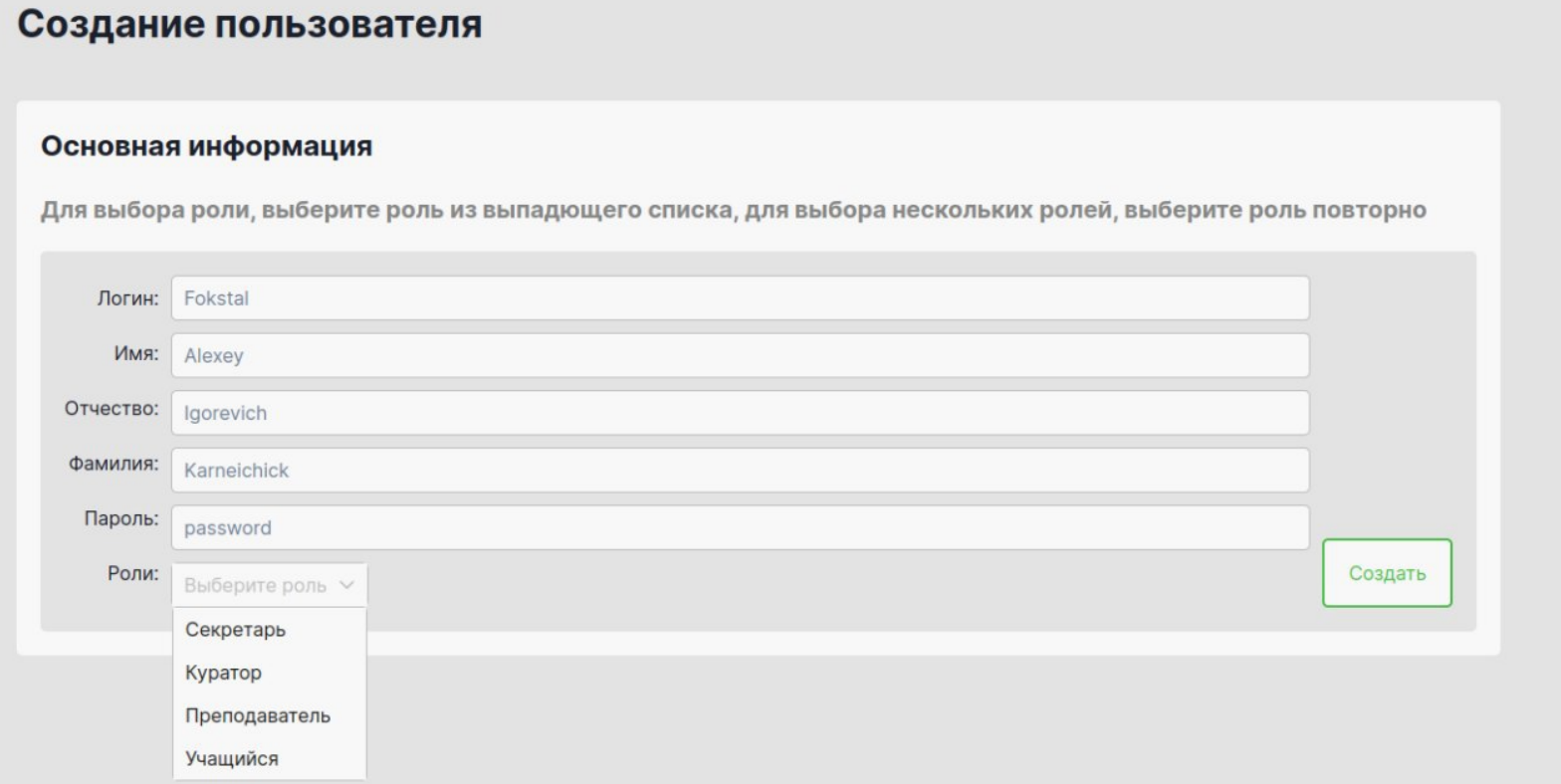


Рисунок 2.2.8.33 – Страница создания пользователя

При переходе на несуществующую страницу предусмотрена страница «Ошибка 404» (рисунок 2.2.8.34). При открытии страниц, будучи неавторизованным, предусмотрена страница «Ошибка 401» (рисунок 2.2.8.35).

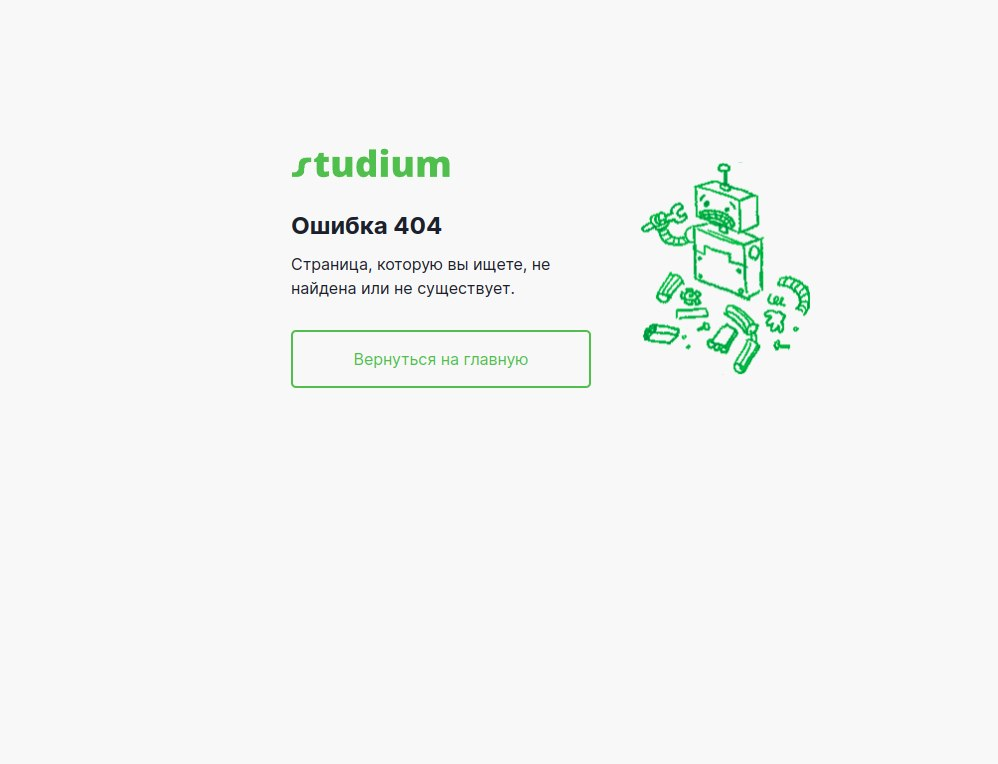


Рисунок 2.2.8.34 – Страница «Ошибка 404»

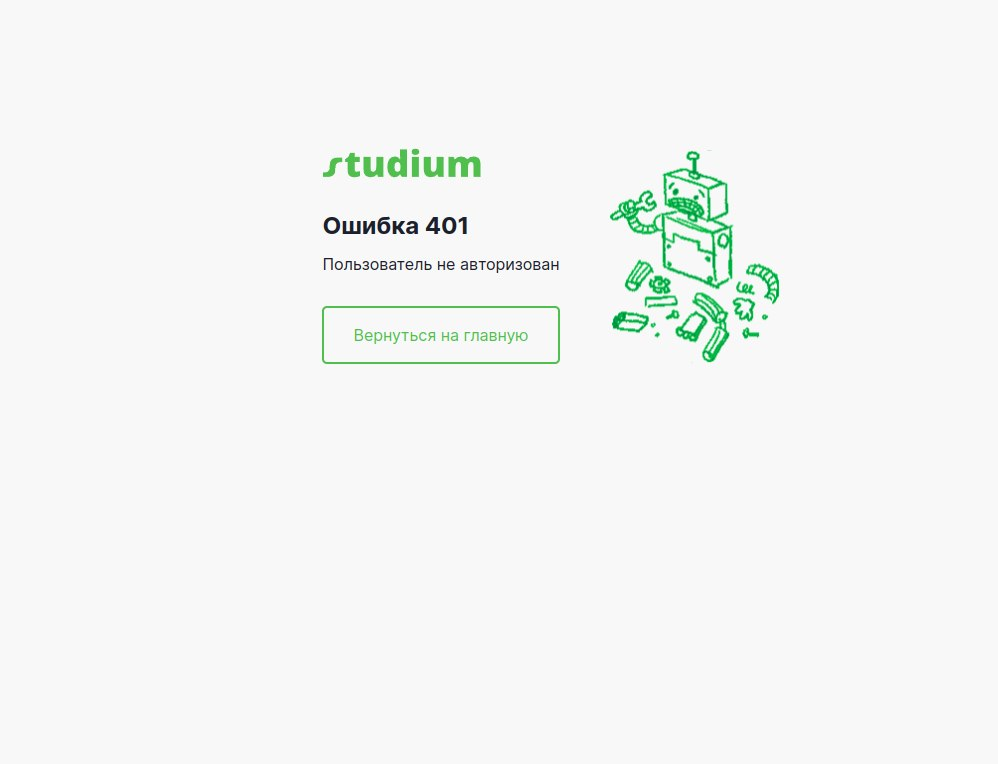


Рисунок 2.2.8.35 – Страница «Ошибка 401»

Для входа в запущенную серверную часть программного комплекса «Studium» необходимо перейти по адресу, представленном на рисунке 2.2.8.36. Примечание: технология Swagger UI не будет доступна при запуске программного комплекса с помощью контейнера Docker.

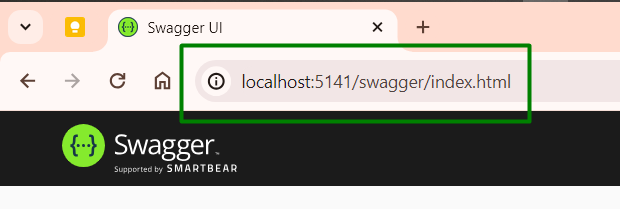


Рисунок 2.2.8.36 – Корректный адрес для взаимодействия программного комплекса с Swagger UI

Пролистав чуть ниже, пользователь сталкивается со всеми контроллерами, представленными в программном комплексе и методами для работы с данными, которыми они управляют (рисунки 2.2.8.37-2.2.8.38).



Рисунок 2.2.8.37 – Общий вид контроллера «Grade»

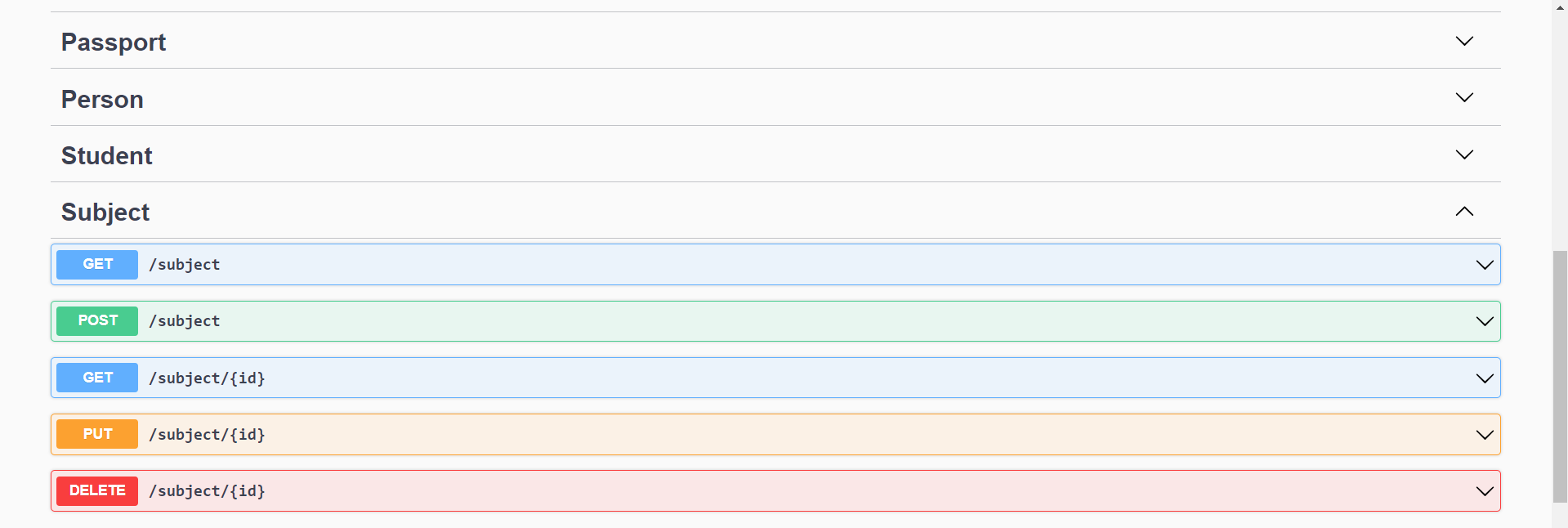


Рисунок 2.2.8.38 – Общий вид некоторых контроллеров

Для работы непосредственно с запросами пользователю необходимо выбрать сущность и сам запрос, развернув при этом нужные списки, изначально они все раскрыты (рисунок 2.2.8.39).

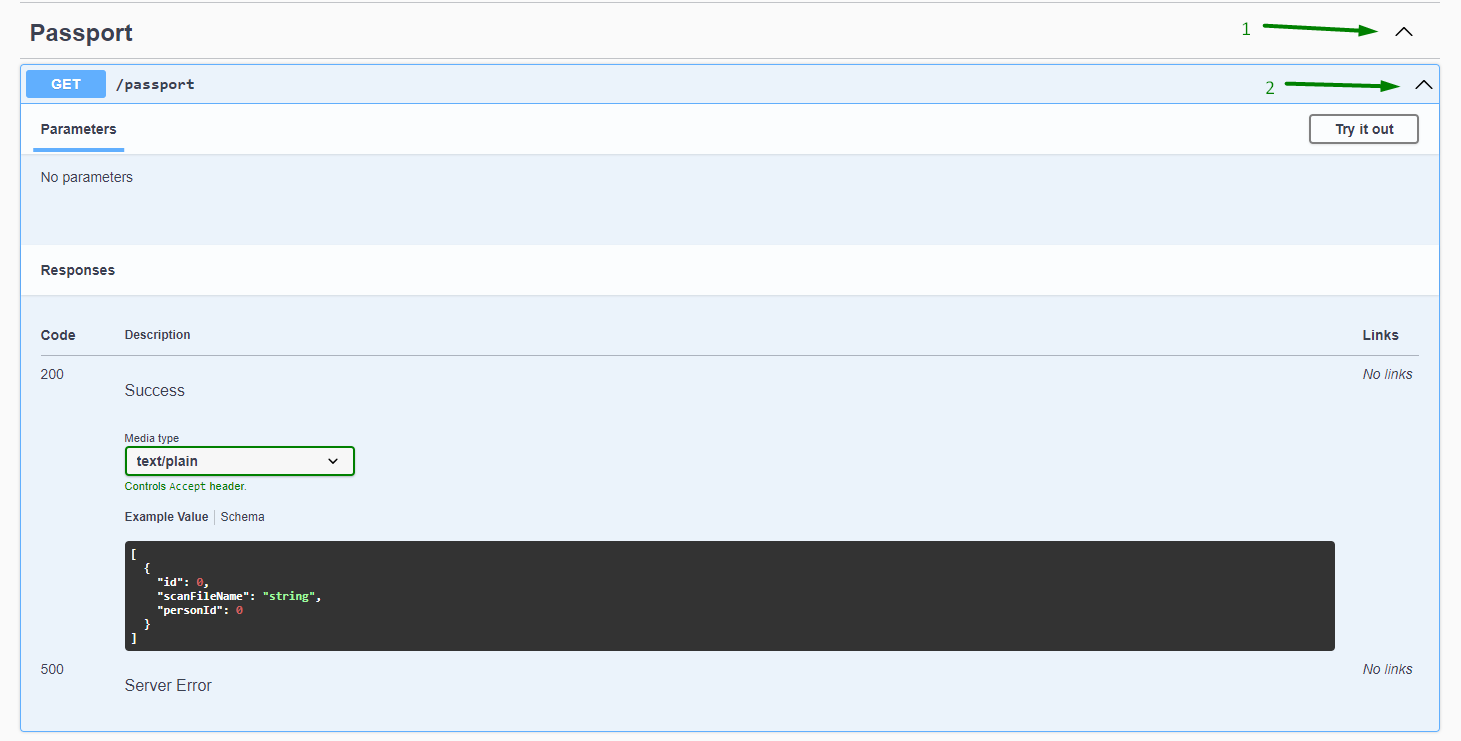


Рисунок 2.2.8.39 – Открытый контроллер «Passport» и запрос на взятие списка сущностей

Далее, для отправки запроса на сервер необходимо нажать кнопку «Try it out», после чего надпись данной кнопки изменится на «Cancel» и откроется еще одно подменю и уже там нужно нажать кнопку «Execute» для запуска процесса (рисунки 2.2.8.40-2.2.8.41).

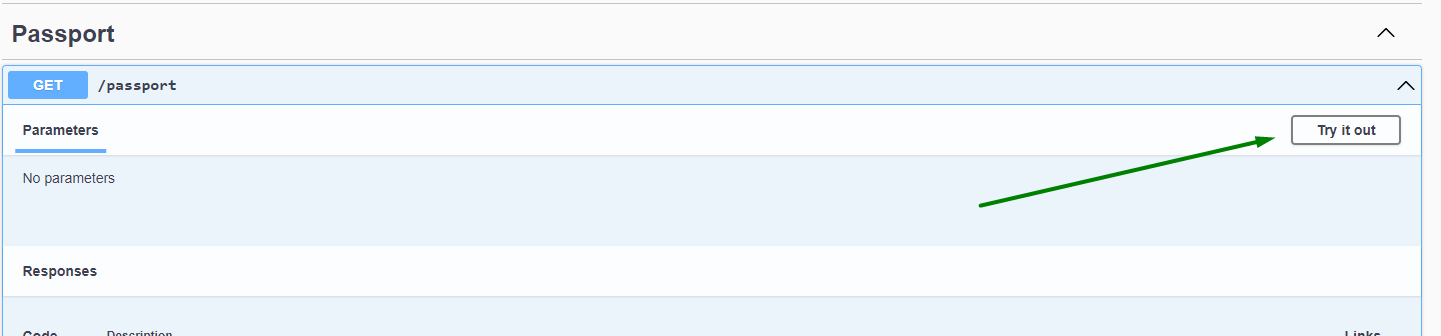


Рисунок 2.2.8.40 – Нажатие кнопки «Try it out»

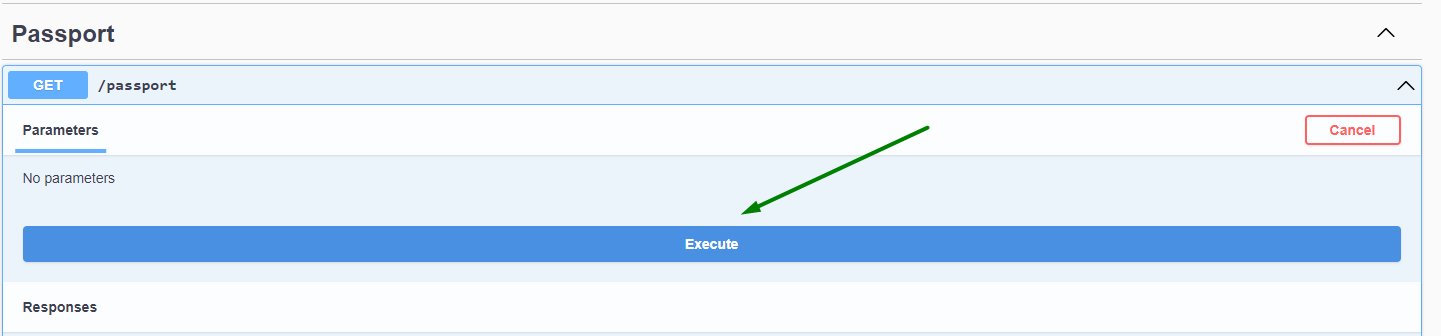


Рисунок 2.2.8.41 – Нажатие кнопки «Execute»

После всех действий запрос будет отправлен на сервер и через некоторое время будет возвращен ответ о завершении действия: с ошибкой, с результатом или с другой информацией ответа сервера. В данном случае будет выведен пустой список сущностей «Passport» и дополнительная информация ответа (рисунок 2.2.8.42).

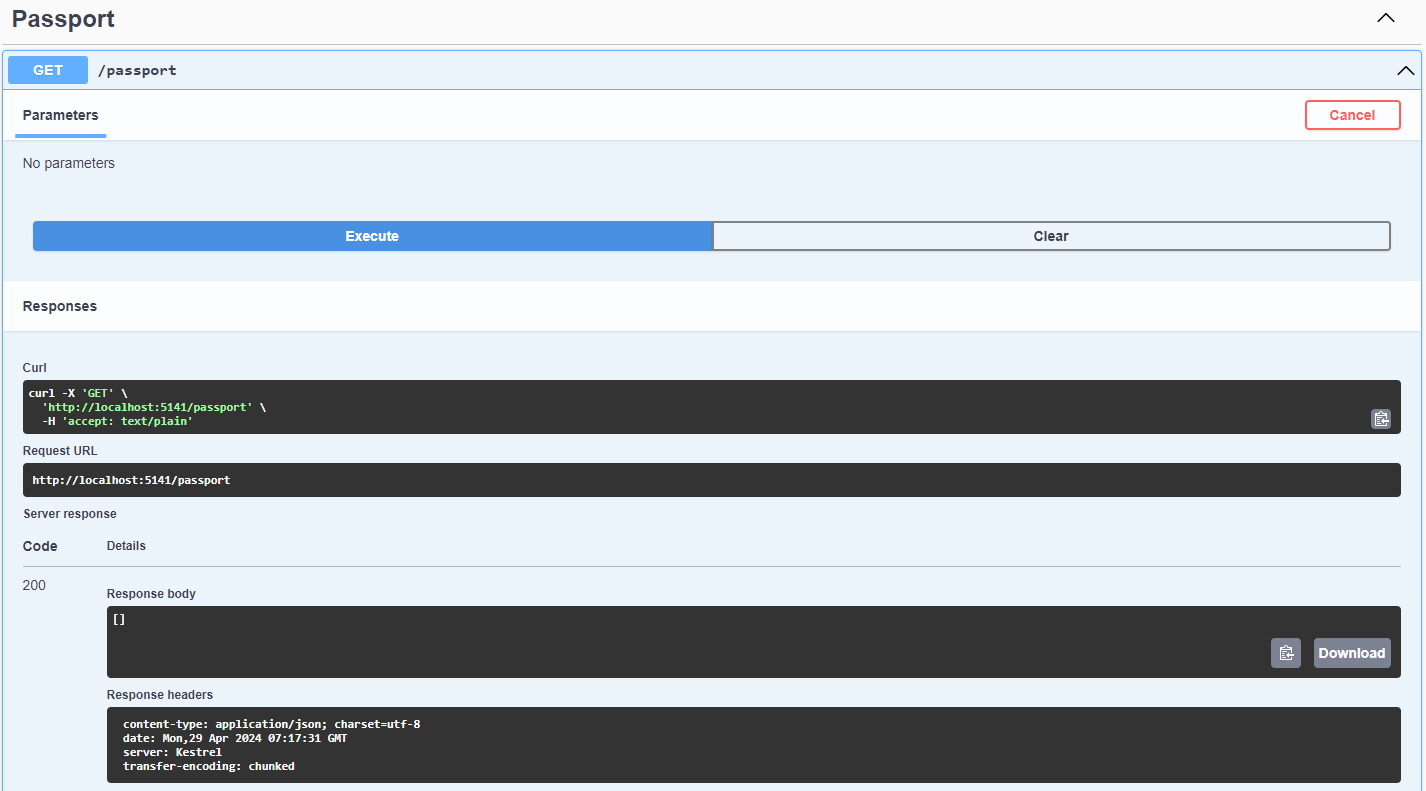


Рисунок 2.2.8.42 – Ответ от сервера на получение списка сущностей «Passport»

Все остальные запросы работают аналогичным способом, только при в некоторых необходимо указывать дополнительную информацию с необходимыми параметрами: для создания – данные сущности, для обновления – новые данные сущности и идентификатор обновляемой сущности, для удаления – идентификатор удаляемой сущности и так далее (рисунок 2.2.8.43).

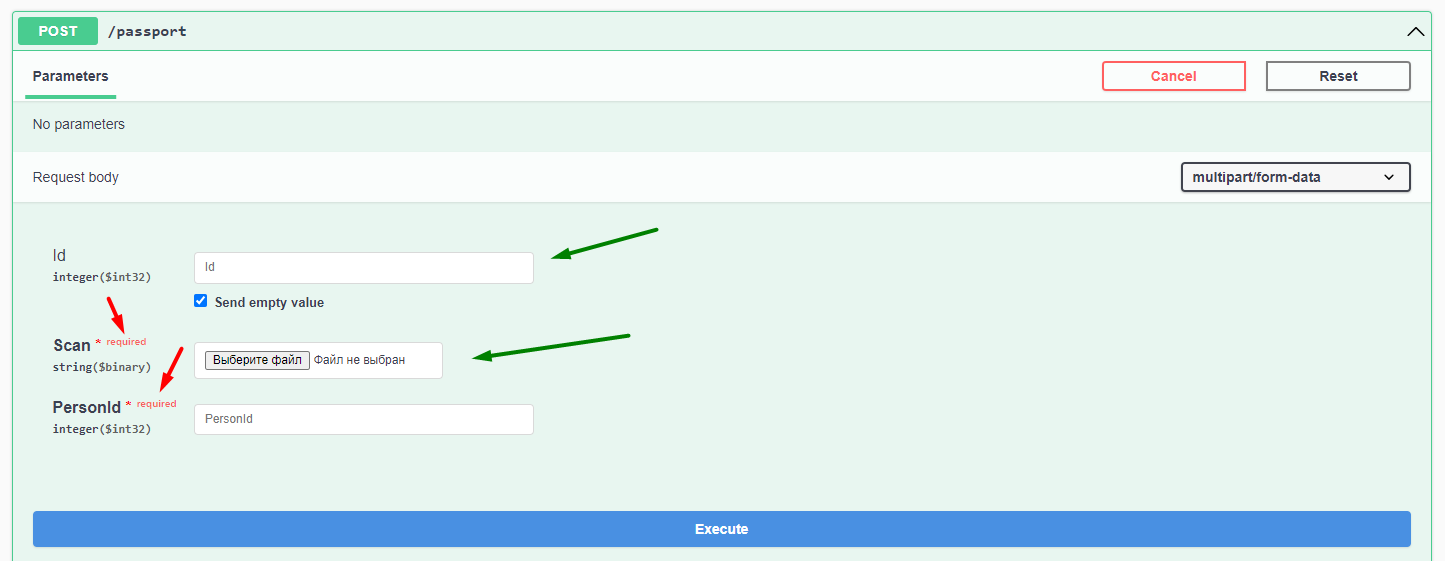


Рисунок 2.2.8.43 – Общий вид запроса на создание сущности «Passport»

При работе с контроллерами могут возникать ошибки при запросах и непосредственно ошибки работы самого сервера. Для данных моментов есть описание всех возможных ответов и ряд дополнительной информации (рисунки 2.2.8.44-2.2.8.45).

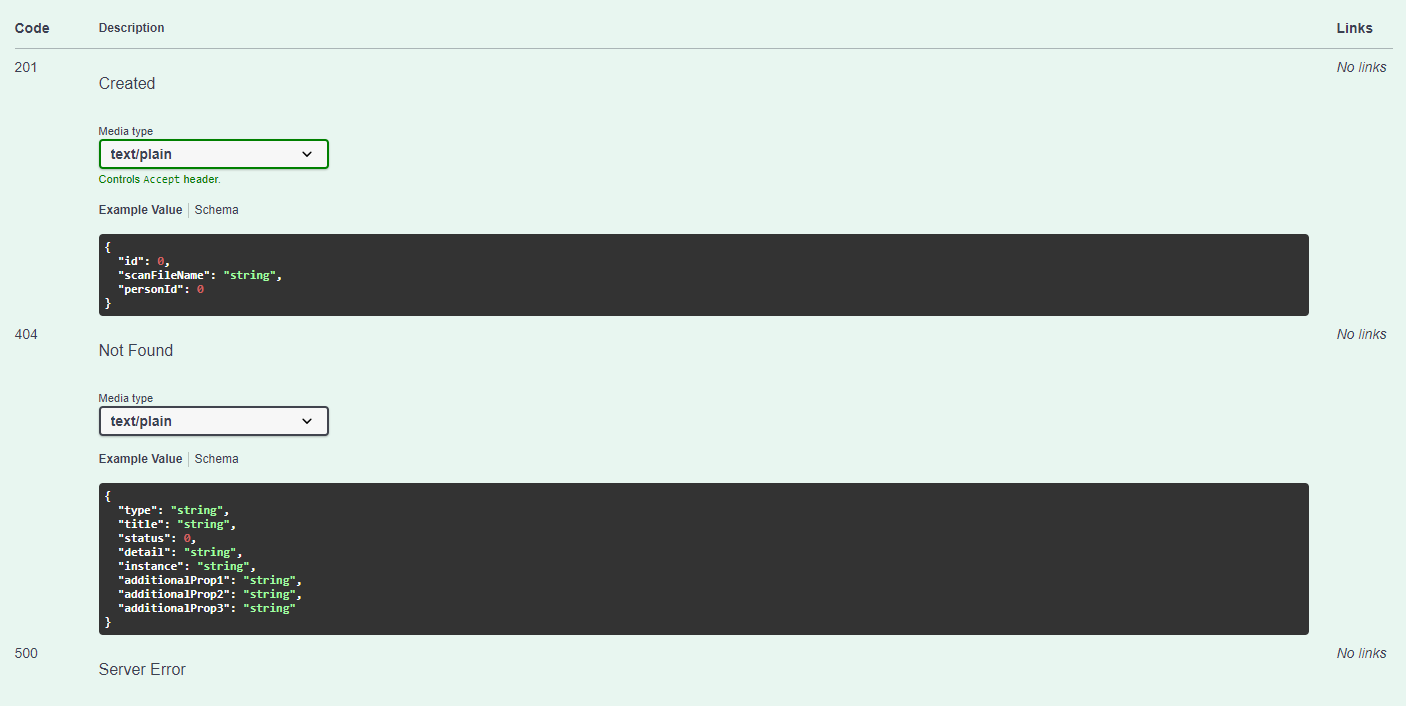


Рисунок 2.2.8.44 – Ответы для запроса на создание сущности «Passport»



Рисунок 2.2.8.45 – Описание сущностей во вкладке «Schemas»

На данном этапе основные моменты для работы с технологией Swagger UI в серверной части программного комплекса рассмотрены. Дополнительную информацию и документацию по Swagger UI в частности можно найти к открытом доступе.

## **СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ**

1. Албахари, Д. C# 7.0. Справочник. Полное описание языка / Д. Албахари, Б. Албахари – Москва: Вильямс, 2018. – 1024 с.
2. Ахметова, Н.Д. Электронные журналы и дневники как одно из средств повышения качества образования в школе / Н.Д. Ахметова – Ялга: МОУ «Ялгинская СОШ», 2010. – 7 с.
3. Бусел, И.П. Организация сельскохозяйственного производств / И.П. Бусел, П.И. Малихтарович – Минск: Лiтаратура i Мастацтва, 2021. – 251 с.
4. Бусел, И.П. Экономика сельского хозяйства: учеб. пособие / И.П. Бусел, П.И. Малихтарович – Минск: РИПО, 2020. – 367 с.
5. Бусел, И.П. Экономика сельскохозяйственного предприятия с основами менеджмента / И.П. Бусел, П.И. Малихтарович. – Минск: Лiтаратура i Мастацтва, 2020. – 341 с.
6. Веб-сайт «Laboratoria.by» [Электронный ресурс] / Ответственность за нарушение требований охраны труда. – 2022. – Режим доступа: https://laboratoria.by/stati/otvetstvennost-za-narusheniye-ot. – Дата доступа: 15.05.2024.
7. Онлайн-платформа обучения «Selectel» [Электронный ресурс] / Введение в Git: настройка и основные команды. – 2024. – Режим доступа: https://selectel.ru/blog/tutorials/git-setup-and-common-commands/. – Дата доступа: 09.05.2024.
8. Онлайн-платформа обучения «Skillbox» [Электронный ресурс] / Что такое образовательный процесс и из чего он состоит. – 2024. – Режим доступа: https://skillbox.ru/media/education/chto-takoe-obrazovatelnyy-protsess-i-iz-chego-on-sostoit/. – Дата доступа: 04.05.2024.
9. Онлайн-платформа обучения «Skillbox» [Электронный ресурс] / Что такое API и как он работает. – 2022. – Режим доступа: https://skillbox.ru/media/code/chto\_takoe\_api/. – Дата доступа: 05.05.2024.
10. Онлайн-платформа обучения «Udemy» [Электронный ресурс] / Курс «ASP.NET Core MVC для .NET 5. Первая часть». – 2018. – Режим доступа: https://www.udemy.com/course/aspnetcoremvc1net5/learn/lecture/23820774?start=45#overview – Дата доступа: 10.05.2024.
11. Онлайн-платформа по работе с заметками «Notion» [Электронный ресурс] / Владилен Минин – Docker для начинающих (2021). – 2021. – Режим доступа: https://vladilen.notion.site/Docker-2021-a72201ec8573461c8a 2e62e2fcf33aa3. – Дата доступа: 25.05.2024.
12. Портал «Охрана труда. Портал для инженеров по охране труда Беларуси» [Электронный ресурс] / Какие государственные органы надзора и контроля за соблюдением законодательства о труде и правил по охране труда существуют в республике? – 2024. – Режим доступа: https://ohranatruda.of.by/kakie-gosudarstvennye-organy-nadzora-i-kontrolya-za-soblyudeniem-zakonodatelstva-o-trude-i-pravil-po-oxrane-truda-sushhestvuyut-v-respublike.html. – Дата доступа: 13.05.2024.
13. Портал «Охрана труда. Портал для инженеров по охране труда Беларуси» [Электронный ресурс] / Какие требования к организации режима труда и отдыха должны обеспечиваться при работе на персональных ЭВМ? – 2024. – Режим доступа: https://ohranatruda.of.by/kakie-trebovaniya-k-organizacii-rezhima-truda-i-otdyxa-dolzhny-obespechivatsya-pri-rabote-na-personalnyx-evm.html. – Дата доступа: 16.05.2024.
14. Портал «Espot.by» [Электронный ресурс] / Производственный контроль: необходимость или возможность?! – 2015. – Режим доступа: https://www.espot.by/izdaniya/espot/proizvodstvennyi-kontrol-neobkhodimost-i\_0000000. – Дата доступа: 26.05.2024.
15. Сборник статей «Научные Статьи.Ру» [Электронный ресурс] / Организация процесса обучения: ключевые принципы и методы. – 2023. – Режим доступа: https://nauchniestati.ru/spravka/organizacziya-proczessa-obucheniya/. – Дата доступа: 04.05.2024.
16. Троелсен, Э. Язык программирования C# 5.0 и платформа .NET 4.5 / Э.  Троелсен. – М.: Вильямс, 2015. – 486 c.
17. Фаронов, В.В. C#. Программирование на языке высокого уровня: Учебник для вузов / В.В. Фаронов. – СПб.:Питер, 2008. – 640 с.
18. Энциклопедия «Microsoft Learn» [Электронный ресурс] / Документация по C#. – 2024. – Режим доступа: https://learn.microsoft.com/ru-ru/dotnet/csharp. – Дата доступа: 10.04.2024.
19. Энциклопедия «Microsoft Learn» [Электронный ресурс] / Сборник курсов и документации по ASP.NET Core. – 2024. – Режим доступа: https://dotnet.microsoft.com/ru-ru/learn/aspnet. – Дата доступа: 11.05.2024.
20. Энциклопедия «Microsoft Learn» [Электронный ресурс] / Руководство по созданию веб-API с помощью ASP.NET Core. – 2024. – Режим доступа: https://learn.microsoft.com/ru-ru/aspnet/core/tutorials/first-web-api?view=aspnetcore-8.0&tabs=visual-studio. – Дата доступа: 11.05.2024
21. Энциклопедия «Metanit» [Электронный ресурс] / Руководство по Entity Framework Core 8 и C#. – 2023. – Режим доступа: https://metanit.com/sharp/efcore/. – Дата доступа: 12.05.2024.
22. Энциклопедия «Metanit» [Электронный ресурс] / Руководство по ASP.NET Core 8. – 2023. – Режим доступа: https://dotnet.microsoft.com/ru-ru/learn/aspnet. – Дата доступа: 13.05.2024.