СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 4](#_Toc105530015)

[1. ОБЩАЯ ЧАСТЬ 7](#_Toc105530016)

[1.1. Цель разработки 7](#_Toc105530017)

[1.2. Средства разработки 7](#_Toc105530018)

[2. СПЕЦИАЛЬНАЯ ЧАСТЬ 9](#_Toc105530019)

[2.1. Постановка задачи 9](#_Toc105530020)

[2.1.1. Входные данные предметной области 9](#_Toc105530021)

[2.1.2. Выходные данные предметной области 9](#_Toc105530022)

[2.1.3. Требования к проекту 10](#_Toc105530023)

[2.2. Внешняя спецификация 12](#_Toc105530024)

[2.2.1. Описание задачи 12](#_Toc105530025)

[2.2.2. Сценарий работы программы 17](#_Toc105530026)

[2.2.3. Требование к дизайну 19](#_Toc105530027)

[2.2.4. Требования к работе пользователя 20](#_Toc105530028)

[2.2.5. Входные и выходные данные 21](#_Toc105530029)

[2.2.6. Метод 22](#_Toc105530030)

[2.2.7. Тесты 26](#_Toc105530031)

[2.2.8. Контроль целостности данных 27](#_Toc105530032)

[2.3. Проектирование 29](#_Toc105530033)

[2.3.1. Схема баз данных 29](#_Toc105530034)

[2.3.2. Диаграмма классов 31](#_Toc105530035)

[2.3.3. Функциональная схема задачи 34](#_Toc105530036)

[2.3.4. Структурная схема программы 37](#_Toc105530037)

[2.3.5. Схема архитектуры программы 38](#_Toc105530038)

[2.3.6. Схема пользовательского интерфейса 38](#_Toc105530039)

[2.4. Результаты работы программы 40](#_Toc105530040)

[3. ТЕХНОЛОГИЧЕСКАЯ ЧАСТЬ 42](#_Toc105530041)

[3.1. Инструментальные средства разработки 42](#_Toc105530042)

[3.2. Отладка программы 43](#_Toc105530043)

[3.3. Защитное программирование 44](#_Toc105530044)

[3.4. Характеристики программы 44](#_Toc105530045)

[3.4.1. Список подключенных библиотек 44](#_Toc105530046)

[3.4.2. Список модулей 45](#_Toc105530047)

[ЗАКЛЮЧЕНИЕ 47](#_Toc105530048)

[СПИСОК ИСПОЛЬЗОВАННЫХ МАТЕРИАЛОВ 48](#_Toc105530049)

# ВВЕДЕНИЕ

Автоматизация — одно из направлений научно-технического прогресса, использующее саморегулирующие технические средства и математические методы с целью освобождения человека от участия в процессах получения, преобразования. [[21](https://ru.wikipedia.org/wiki/%D0%90%D0%B2%D1%82%D0%BE%D0%BC%D0%B0%D1%82%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D1%8F)] Автоматизация позволяет повысить производительность труда, оптимизировать процессы управления, сократить временные траты человека.

Разрабатываемый программный продукт будет представлять комплекс по автоматизации составления графика учебного процесса для учебного отдела ФГБОУ ВО «Российский экономический университет им. Г.В. Плеханова» (далее – РЭУ им. Г.В. Плеханова) Московский приборостроительный техникум.

Московский приборостроительный техникум – структурное подразделение РЭУ им. Г.В. Плеханова, реализующее программы среднего профессионального образования IT-специальностей.

Отдел производственного обучения является постоянным структурным подразделением Московского приборостроительного техникума, основным органом управления отделом производственного обучения, и подчиняется заместителю директора по производственному обучению.

К основным функциям отдела производственного обучения относятся:

* разработка локальных нормативных актов по своему профилю деятельности (положения, правила, должностные инструкции);
* проведение и организация проведения инструктажей по охране труда с регистрацией в специальных журналах. Осуществление контроля за выполнением правил и норм по охране труда и технике безопасности и правил технической эксплуатации оборудования лабораторий, учебно- производственных мастерских Техникума;
* составление графика практики на учебный год;
* организация разработки и согласование содержания программ учебной и производственной практики обучающихся по образовательным программам СПО;
* организация разработки материалов по составлению смет расходов на содержание учебных лабораторий, учебно-производственных мастерских, на проведение практических занятий и учебной практики;
* расчет педагогической нагрузки по проведению и руководству практикой и ее распределение, контроль выполнения, учет;
* составление статистической отчетности по деятельности отдела производственного обучения;
* оценка качества, эффективности и результативности производственного обучения в Техникуме;
* заключение договоров с предприятиями, учреждениями, организациями по обеспечению ими мест прохождения практики обучающимися Техникума;
* заключение договоров с работодателями обучающихся с целью перевода их на обучение по индивидуальному графику в связи трудоустройством по специальности;
* контроль проведения практических занятий и учебной практики на учебно-производственной базе Техникума, производственной практики - на базе предприятий, организаций и учреждений, учет проведенных занятий на основании журнала учета производственного обучения и планирующей документации по руководству практики обучающихся.

**График учебного процесса – это документ, определяющий последовательность и чередование теоретического обучения, практического обучения, промежуточной аттестации, итоговой аттестации обучающихся всех специальностей и профессий всех курсов в течение учебного года.** [[25]](http://www.пгатк.рф/students/dnevnoe-otdelenie/the-schedule-of-the-educational-process/) **График учебного процесса создается каждый год.**

В графике учебного процесса отражаются особенности каждого учебного года, видно, сколько учебных групп в техникуме и на каждой специальности и профессии. Специальными обозначениями на графике определены виды учебной деятельности студентов, каникулярное время. Объем практического и теоретического обучения, виды практики, зачеты, экзамены, каникулы соответствуют требованиям Федеральных государственных образовательных стандартов. Появилась необходимость в автоматизации процесса формирования графика учебного процесса для экономии времени работников отдела производственного обучения.

Было принято решение разработать программный комплекс по автоматизации графика учебного процесса Московского приборостроительного техникума.

Данная тема является актуальной на сегодняшний момент, работа отдела производственного обучения связана с составлением графика учебного процесса.

# ОБЩАЯ ЧАСТЬ

## Цель разработки

Целью разработки является автоматизация и упрощение процесса составления графика учебного процесса.

## Средства разработки

Для разработки использовался ноутбук Dell G3 3579 характеристики которого указаны в Таблице 1.

Таблица - Техническое средства разработки

| № | Тип оборудования | Наименование оборудования |
| --- | --- | --- |
| 1 | 2 | 3 |
| Ноутбук Dell G3 3579 | | |
| 1 | Размер экрана | 15,6 |
| 2 | Разрешение экрана | 1920х1080 |
| 3 | Линейка процессора | Intel Core i7-8750H |
| 4 | Количество ядер процессора | 6 |
| 5 | Оперативная память | 8 ГБ |
| 6 | Видеокарта | GeForce GTX 1050 Ti |
| 7 | Конфигурация накопителей | SSD |
| 8 | Общий̆ объем всех накопителей | 1128ГБ |
| 9 | Операционная система | Windows 10 Pro 64-bit |

Для разработки использовались программные средства, полная информация о которых указана в Таблице 2.

Таблица - Программные средства

| № | Тип средства | Название средства | Назначение |
| --- | --- | --- | --- |
| 1 | 2 | 3 | 4 |
| 1 | Инструментальное средство разработки программных решений | Microsoft Visual Studio 2022 | Разработка клиентского приложения |
| 2 | Операционная система | Microsoft Windows 10 Pro | Организация взаимодействия программ и пользователя |
| 3 | Текстовый редактор | Microsoft Word 2019 | Разработка документации, формирование отчётных документов по шаблонам |
| 4 | Графическая оболочка | SQL Server Management  Studio 17 | Разработка базы данных |
| 5 | Средство проектирования | Веб-сервис  Draw.io 2.0.9 | Разработка схем для проектирования приложения |
| 6 | Табличный редактор | Microsoft Excel 2019 | Программа использовалась для вывода информации из программы |
| 7 | Браузер | Opera GX | Средство для отладки системы |
| 8 | Средство для создания и редактирования презентаций | Microsoft PowerPoint 2019 | Разработка презентаций |

Для разработки автоматизированной системы использовалась среда разработки Microsoft Visual Studio 2022.

Для запуска комплекса приложений требуется подключение к интернету и любой браузер.

Для хранения данных используется графическая оболочка SQL Server Management.

Для вывода данных использовался табличный редактор Microsoft Excel 2019.

Для написания технической документации использовался текстовый редактор Microsoft Office Word 2019.

Для проектирования диаграмм использовался веб-сервис Draw.io.

Операционная система, на которой разрабатывался комплекс приложений Microsoft Windows 10 Pro, организовывалось взаимодействие программ и пользователя.

Для создания презентаций используется средство для создания и редактирования презентация Microsoft PowerPoint 2019.

Таблица 3 представляет минимальный набор требований для стабильного использования приложения.

Таблица - Минимальный набор требований для технических средств

| № | Тип оборудования | Наименование оборудования |
| --- | --- | --- |
| 1 | 2 | 3 |
| 2 | Процессор | Intel Core i7 |
| 3 | Размер экрана | 15” |
| 4 | Графический процессор | Intel |
| 5 | Оперативная память | 4 ГБ |
| Рекомендованные характеристики | | |
| 1 | Процессор | Intel Core i5 |
| 2 | Размер экрана | 15” |
| 3 | Графический процессор | GeForce MX150 2ГБ |
| 4 | Оперативная память | 8 ГБ |

# СПЕЦИАЛЬНАЯ ЧАСТЬ

## Постановка задачи

Необходимо разработать программный комплекс под операционную систему Windows, обеспечивающий автоматизацию по составлению графика учебного процесса для ускорения процесса создания документа.

### Входные данные предметной области

Входными данными являются:

* Авторизация: логин (никнейм пользователя: не менее 8 и не более 50 символов) и пароль (не менее 8 и не более 50 символов; только латинские и кириллические буквы; минимум одна цифра; без пробелов; допустимые символы: + = - \_ ( ) \* ! ? @);
* Информация о группах (наименование группы);
* Информация о преподавателях (фамилия, имя, отчество и номер телефона преподавателя);
* Информация о предметах (учебных практик, производственных практик, каникул и сессий);
* шаблон графика учебного процесса (формат .xlsx).

### Выходные данные предметной области

Выходные данные:

* Сообщение об ошибке при авторизации;
* Список групп (код группы, наименование группы);
* Список предметов (код предмета, наименование предмета);
* Список преподавателей (код преподавателя; фамилия, имя и отчество преподавателя; номер телефона преподавателя);
* График учебного процесса для групп по специальностям (формат .xlsx, в уже готовый шаблон выводятся наименования групп и с соответствием какая группа проставляется наименование практики в определенную неделю).

### Требования к проекту

Для разработки программного комплекса по автоматизации графика учебного процесса были выявлены требования, необходимые к реализации:

* + Разработать API с использованием REST-архитектуры для взаимодействия с сервером, содержащим методы авторизации, работы с данными;
  + Приложение должно иметь следующие окна: авторизация, основное рабочие окно приложения;
  + Вход в приложение должен осуществляться по логину (никнейм пользователя: не менее 8 и не более 50 символов) и паролю (не менее 8 и не более 50 символов; только латинские и кириллические буквы; минимум одна цифра; без пробелов; допустимые символы: + = - \_ ( ) \* ! ? @), хранимых в БД через API;
  + Окно приложения должно иметь вкладки: «Добавить группу», «Добавить предмет», «Добавить преподавателя»;
  + Авторизация должна происходить с использованием запроса GET на API для сравнения данных пользователя;
* Вкладка «Добавить группу» должна содержать список групп;
* Вкладка «Добавить предмет» должна содержать список предметов;
* Вкладка «Добавить преподавателя» должна содержать список преподавателей.
* Добавление данных необходимо выполнять вручную, данные записываются по средствам запроса POST на API;
* Данные необходимо выводить по средствам запроса GET на API;
* Для удаления данных необходимо нажать на кнопку удалить, данные удаляются по средствам запроса DELETE на API;
* Изменение данных должно происходит путем ввода новых данных и нажатием на кнопку изменить, данные изменяются по средствам запроса PUT на API.
* Создание графика (формата .xlsx) производится автоматически при нажатии на кнопку «Сформировать Excel файл» на главной странице;
* Вся информация в программном комплексе должна выводиться на страницах веб-приложения и окон приложения, отображена читабельным текстом;
* Должно быть распределение ролей: администратор (может вводить, изменять и удалять все данные по группам (наименование группы), преподавателям (фамилия, имя, отчество и номер телефона преподавателя), предметам (наименование предмета) и создавать график учебного процесса (формат .xlsx)) и пользователь (просматривает график учебного процесса (просмотр Excel файла с последними изменениями));
* Должно быть предусмотрено составление графика учебного процесса (формат .xlsx). Шаблон графика должен храниться в системе. Для создания графика необходимо нажать на кнопку «Создание Excel файла», производится автоматическое создание графика учебного процесса. Заполнение данных графика происходит из базы данных (заполнение групп (наименование группы) и предметов (наименование предмета));
* Должна быть предусмотрена проверка по полям на уже созданных группу (наименование группы), преподавателя (фамилия, имя, отчество и номер телефона) и предмета (наименование предмета).

## Внешняя спецификация

### Описание задачи

Основной задачей является создание программного комплекса, который автоматизирует и упростит процесс создания учебного процесса.

Данное приложение состоит из трех частей: база данных и график учебного процесса. Первое это реляционная база данных, с которой взаимодействует приложение. База данных должна спроектирована с помощью модели, которые создаются при использовании технологий миграции, для отслеживания и управления версиями и объектами базы данных.

Вторя часть — это приложение, которым является график учебного процесса. Приложение должно иметь возможность отображать определённую информацию, которая хранится в базе данных, через технологию API (Application programming interface). Помимо отображения приложение должно предоставлять функциональные возможности пользователю по созданию и управлению данными.

Такое решение должно решить проблему в процессе формирования графика учебного процесса в формате .xlsx.

На Рисунке 1 представлена диаграмма прецендентов, которая отображает возможности пользователя в рамках графика учебного процесса. Также на данном изображении представлены все возможные роли, которые учитываются в приложении.

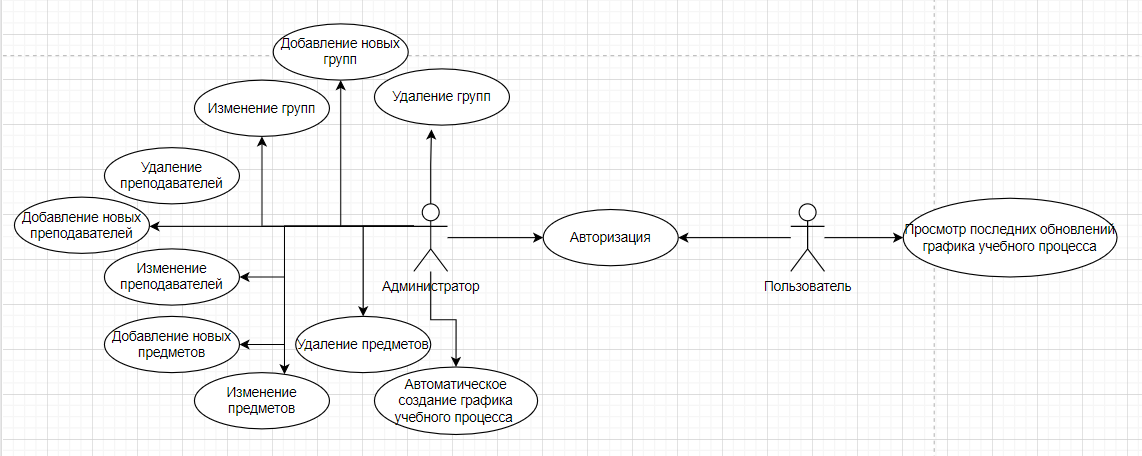


Рисунок - Диаграмма прецендентов

На данном Рисунке 1 представлены 2 роли пользователей, которые будут в системе:

Администратор:

* авторизация;
* добавление новых групп;
* изменение групп;
* удаление групп;
* добавление новых преподавателей;
* изменение преподавателей;
* удаление преподавателей;
* добавление новых предметов;
* изменение предметов;
* удаление предметов;
* автоматическое создание графика учебного процесса.

Пользователь:

* авторизация;
* просмотр последних обновлений графика учебного процесса.

На Рисунке 2 продемонстрирована диаграмма бизнес-процессов (IDEF0), а именно уровень А-0, на котором графически изображены механизмы, различные алгоритмы или инструкции для управления процессом, а также входные и выходные данные.

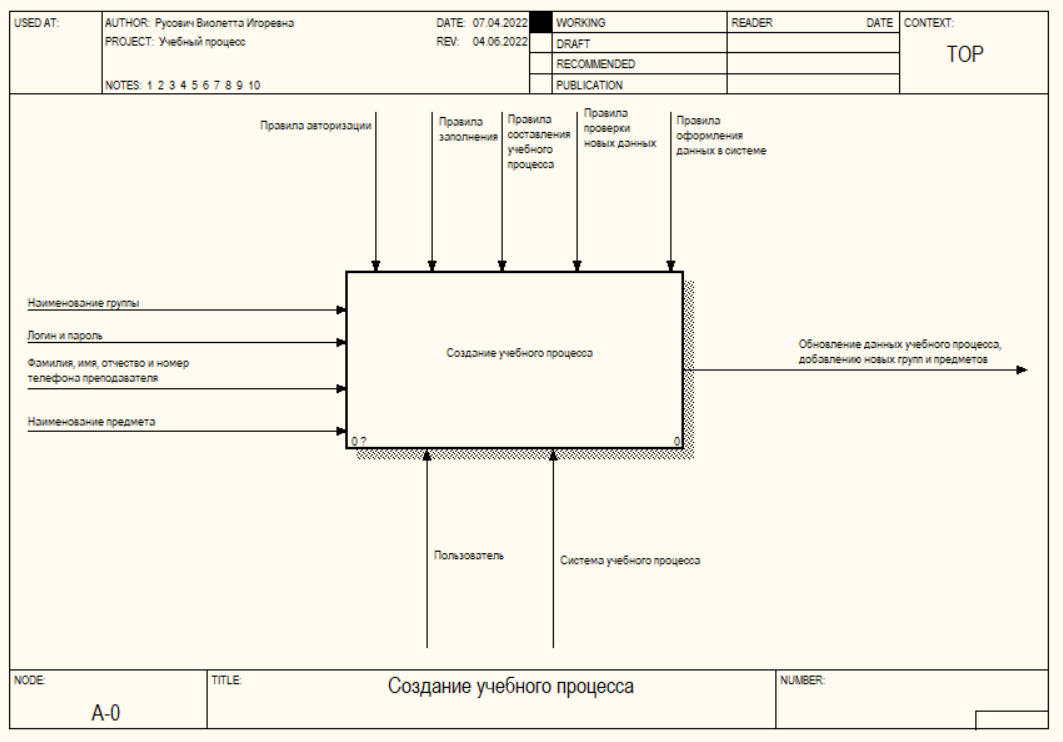


Рисунок - Диаграмма бизнес-процессов (уровень А-0)

На данном уровне отображен основной процесс «Создания учебного процесса», который обеспечивает приложение. В качестве входных данных используется информация, которую ввел пользователь или уже хранятся в системе и после определённых действий выгружаются и отображаются пользователю из базы данных.

Механизмами управления всегда будет выступать пользователь, как субъект управляющий программой, так и сама система учебного процесса, которая является объектом.

Выходными данными является новые данные учебного процесса, которые были сформирована в процессе работы приложения и загружена в Excel файл с последующим редактированием.

Управляющими механизмами выступает алгоритмы системы и различные правила, с помощью которых введенная информация обрабатывается и записывается в базу данных с последующим отображением пользователю в Excel файле.

На Рисунке 3 представлена декомпозиция основного уровня (А0).

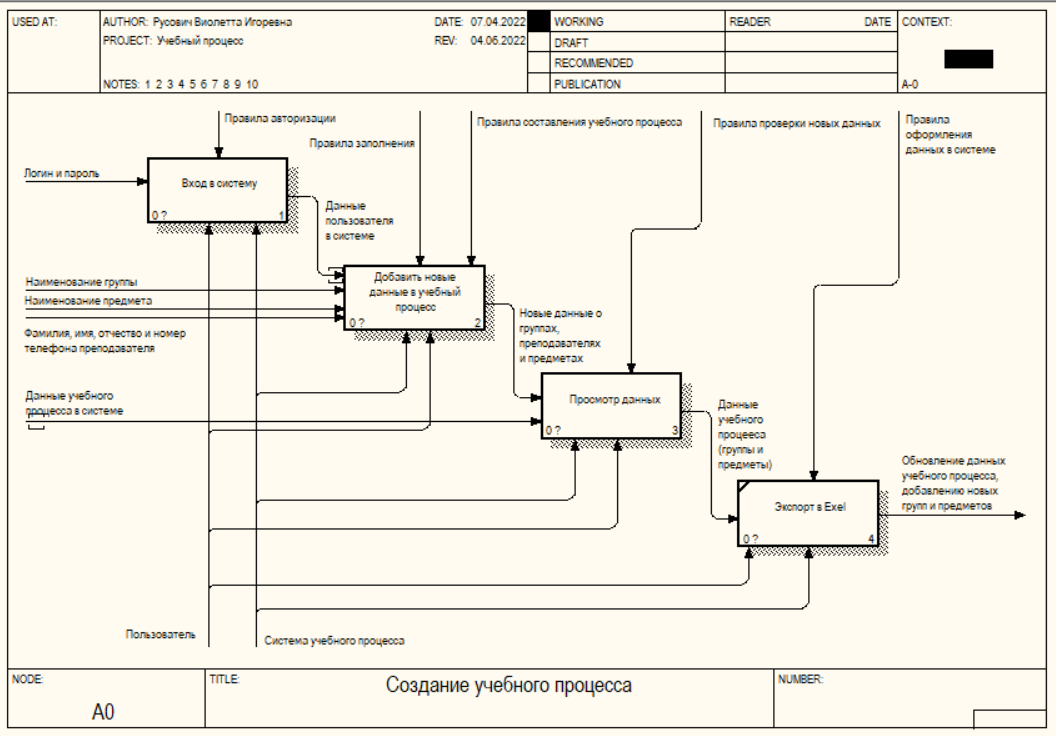


Рисунок - Декомпозиция диаграммы бизнес-процессов (уровень А0)

На данной декомпозиции отображены 4 основных подпроцесса, из которых состоит основной процесс.

Первый подпроцесс «Вход в систему», обеспечивает процесс авторизации пользователя в системе. После авторизации система дает доступ к возможностям создания графика учебного процесса.

Второй подпроцесс «Добавить новые данные в учебный процесс», позволяет на основе полученных (введенной или выгруженной из базы данных) данных выбрать данные для добавления.

Третий подпроцесс «Просмотр данных», просмотр данных еще раз для проверки их точности.

Четвертый подпроцесс «Экспорт Excel» осуществляет проверку системой, формирование и запись новых данных, с выгрузкой их в Excel файл.

На Рисунке 4 представлена декомпозиция уровня А1.

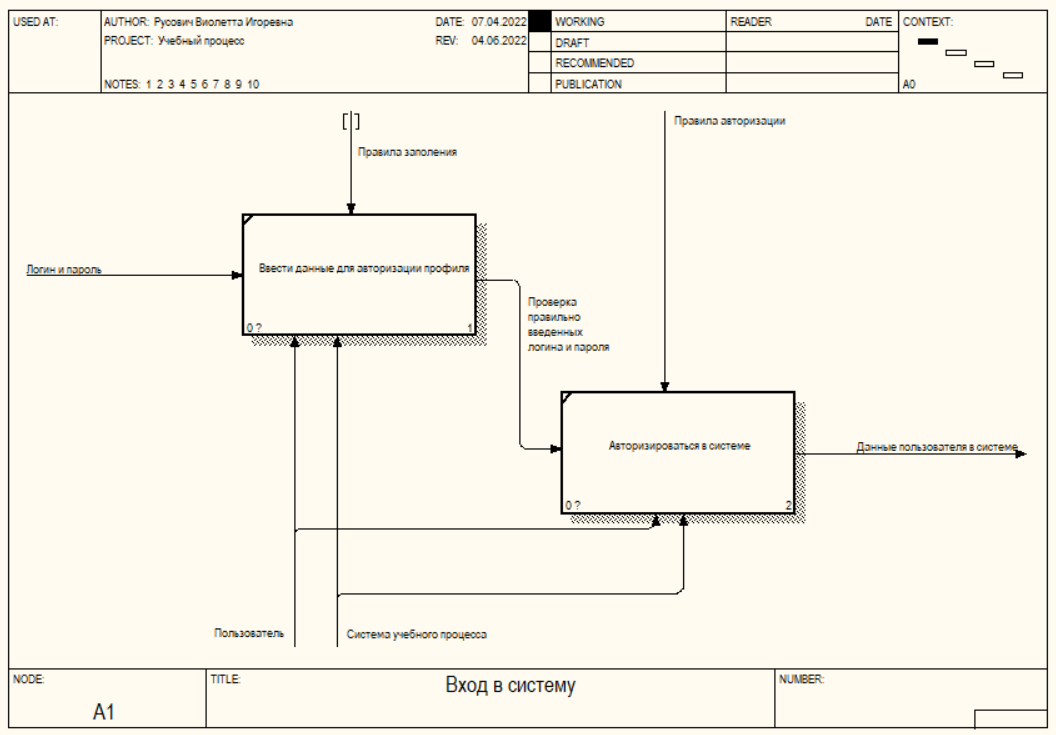


Рисунок - Декомпозиция диаграммы бизнес-процессов (уровень А1)

На данной декомпозиции графически продемонстрирован состав подпроцесса «Войти в систему», который состоит из двух частей. Первая часть – это ввод данных, которые ввел пользователь при помощи правил авторизации. Вторая часть – это сам процесс авторизации пользователя в системе.

На Рисунке 5 представлена декомпозиция уровня А2.

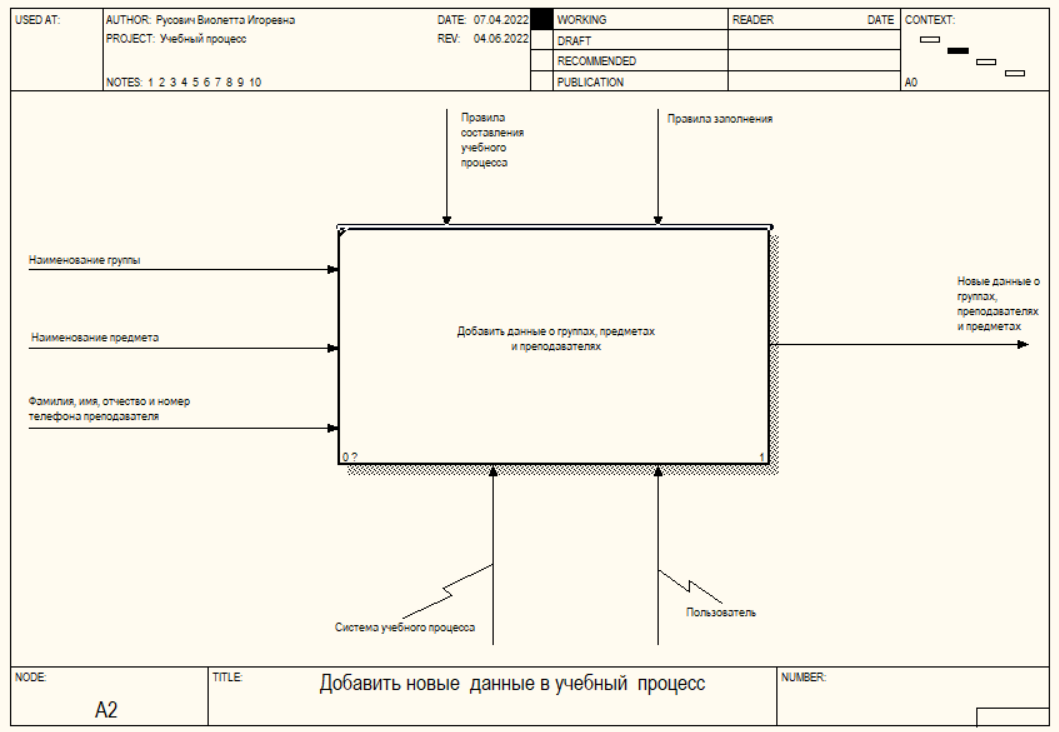


Рисунок - Декомпозиция диаграммы бизнес-процессов (уровень А2)

На данном Рисунке 5 изображен состав третьего подпроцесса, который состоит из одного функционального модуля. Модуль отвечает за выбор данных, которые выбрал пользователь при помощи списка с данными.

### Сценарий работы программы

Пользователь открывает систему и перед ним возникает окно авторизации. Чтобы авторизоваться, пользователю необходимо ввести логин (никнейм: не менее 8 и не более 50 символов) и пароль (не менее 8 и не более 50 символов; только латинские и кириллические буквы; минимум одна цифра; без пробелов; допустимые символы: + = - \_ ( ) \* ! ? @).

Если роль - администратор, которая принадлежит пользователю, после авторизации ему откроется панель управления:

* Добавить группу
* Добавить предмет
* Добавить преподавателя
* Создать Excel файл

Если роль – пользователь, которая принадлежит пользователю, после авторизации ему откроется обновленный график учебного процесса (Excel файл).

Переходя на страницу «Добавить группу», администратор может увидеть поле для записи «Наименование группы», кнопку «Добавить» и лист, где выведен список групп из базы данных и кнопки «Изменить» и «Удалить». При заполнении поля «Наименование группы», нажатие на кнопку «Добавить» производит добавление группы в базу данных и обновление списка групп, если такая группа уже существует в базе данных, то появится сообщение «Такая группа уже существует». При заполнении поля «Наименование группы», нажатие на кнопку «Изменить», в списке групп под одной из групп, производит изменение группы в базе данных и обновление списка групп, если такая группа уже существует в базе данных, то появится сообщение «Такая группа уже существует». При нажатии на кнопку «Удалить», в списке групп под одной из групп, произойдет удаление группы из базы данных и обновление списка групп. При нажатии на кнопку «Выход» производится выход со страницы.

Переходя на страницу «Добавить предмет», администратор может увидеть поле для записи «Наименование предмета», кнопку «Добавить» и лист, где выведен список предметов из базы данных и кнопки «Изменить» и «Удалить». При заполнении поля «Наименование предмета», нажатие на кнопку «Добавить» производит добавление предмета в базу данных и обновление списка предметов, если такой предмет уже существует в базе данных, то появится сообщение «Такой предмет уже существует». При заполнении поля «Наименование предмета», нажатие на кнопку «Изменить», в списке предметов под одним из предметов, производит изменение предмета в базе данных и обновление списка предметов, если такой предмет уже существует в базе данных, то появится сообщение «Такой предмет уже существует». При нажатии на кнопку «Удалить», в списке предметов под одним из предметов, произойдет удаление предмета из базы данных и обновление списка предмета. При нажатии на кнопку «Выход» производится выход со страницы.

Переходя на страницу «Добавить преподавателя», администратор может увидеть поля для записи «Фамилия», «Имя», «Отчество» и «Номер телефона», кнопку «Добавить» и лист, где выведен список преподавателей из базы данных и кнопки «Изменить» и «Удалить». При заполнении полей «Фамилия», «Имя», «Отчество» и «Номер телефона», нажатие на кнопку «Добавить» производит добавление преподавателя в базу данных и обновление списка преподавателей, если такой преподаватель уже существует в базе данных, то появится сообщение «Такой преподаватель уже существует». При заполнении полей «Фамилия», «Имя», «Отчество» и «Номер телефона», нажатие на кнопку «Изменить», в списке преподавателей под одним из преподавателей, производит изменение преподавателя в базе данных и обновление списка преподавателей, если такой преподавателей уже существует в базе данных, то появится сообщение «Такой преподаватель уже существует». При нажатии на кнопку «Удалить», в списке преподавателей под одним из преподавателей, произойдет удаление преподавателя из базы данных и обновление списка преподавателей. При нажатии на кнопку «Выход» производится выход со страницы.

При нажатии на кнопку «Создать Excel файл» производится автоматическое формирование графика учебного процесса, вывод групп из базы данных и предметов для соответствующей группы.

На панели находится кнопка «Выйти», нажав на неё администратор сможет выйти из системы, откроется страница авторизации.

### Требование к дизайну

К внешнему виду приложения был предъявлен определенный ряд требований, предоставленных заказчиком:

* + Минималистичный плоский дизайн;
  + Использование неярких цветов: черный, белый, синий, светло-голубой;
  + Внешний вид приложения API должен состоять из адреса вкладки, кода запроса, отображения необходимого метода, отображение схемы.
  + Внешний вид приложения для расчета стоимости ликвидации «Мобильный офис»: синий фон в стилистике компании, белые кладки, черный текст.
  + Вкладки должны быть выполнены в едином стиле;
* Все визуальные компоненты должны быть выровнены, доступны, иметь соизмеримый масштаб и не оставлять много свободного пространства. Должны отсутствовать орфографические и грамматические ошибки.
* Программный комплекс должно быть выполнено в следующих цветах и с изображениями:
* #ffffff;
* фоновая картинка;
* иконка;
* логотип на странице авторизации.

Более подробно с внешним видом программы можно ознакомиться в ПРИЛОЖЕНИЕ А. ЭСКИЗНЫЙ ПРОЕКТ.

### Требования к работе пользователя

Во время изучения предметной области и обсуждения необходимых требования к программе, заказчик выделил роли: «Администратор» и «Пользователь», имеющих набор следующих функций:

Администратор:

* Авторизация;
* Внесение данных необходимых для групп;
* Внесения данных необходимых для предметов;
* Внесение данных необходимых для преподавателей;
* Просмотр данных о группах;
* Просмотр данных о предметах;
* Просмотр данных о преподавателях;
* Формирование графика учебного процесса.

Пользователь:

* Авторизация;
* Просмотр готового документа Excel файла.

### Входные и выходные данные

В Таблице 4 представлены входные данные программы.

Таблица - Входные данные

| Имя | Ограничение | Точность представления | Формат | Форма вывода |
| --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 |
| Окно «Авторизации» | | | | |
| Логин | [а-я, А-Я, 0-9, +, =, -, \_, (, ), \*, !, ?, @]; {0, 50} | Нет | string | Текстовое поле |
| Пароль | [а-я, А-Я, 0-9, +, =, -, \_, (, ), \*, !, ?, @]; {0, 50} | Нет | string | Текстовое поле |
| Окно «Добавить группу» | | | | |
| Наименование группы | [а-я, А-Я, 0-9, -, ;, /]; {0, 30} | Нет | string | Текстовое поле |
| Окно «Добавить предмет» | | | | |
| Наименование предмета | [а-я, А-Я, -]; {0, 30} | Нет | string | Текстовое поле |
| Окно «Добавить преподавателя» | | | | |
| Фамилия | [а-я, А-Я, 0-9, -]; {0, 30} | Нет | string | Текстовое поле |
| Имя | [а-я, А-Я, 0-9, -]; {0, 30} | Нет | string | Текстовое поле |
| Отчество | [а-я, А-Я, 0-9, -]; {0, 30} | Нет | string | Текстовое поле |
| Номер телефона | [0-9, (, ), -]; {0, 30} | Нет | string | Текстовое поле |

В Таблице 5 представлены выходные данные программного комплекса.

Таблица - Выходные данные

| Имя | Ограничение | Точность представления | Формат | Форма вывода |
| --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 |
| Окно «Авторизации» | | | | |
| Вход в систему | Нет | Нет | Нет | Нет |
| Сообщение об ошибке | [a-z, A-Z, А-Я, а-я, [0 –9]; {1, 100} | Нет | string | Текстовое поле |
| Форма добавления групп | | | | |
| Список групп | [a-z, A-Z, А-Я, а-я, [0 –9]; {1, 100} | Нет | string | Поле вывода |
| Сообщение об ошибке | [a-z, A-Z, А-Я, а-я, [0 –9]; {1, 100} | Нет | string | Текстовое поле |
| Форма добавления предметов | | | | |
| Список предметов | [a-z, A-Z, А-Я, а-я, [0 –9]; {1, 100} | Нет | string | Поле вывода |
| Сообщение об ошибке | [a-z, A-Z, А-Я, а-я, [0 –9]; {1, 100} | Нет | string | Текстовое поле |
| Форма добавление преподавателей | | | | |
| Список преподавателей | [a-z, A-Z, А-Я, а-я, [0 –9]; {1, 100} | Нет | string | Поле вывода |
| Сообщение об ошибке | [a-z, A-Z, А-Я, а-я, [0 –9]; {1, 100} | Нет | string | Текстовое поле |
| Главная страница | | | | |
| Расписание | [a-z, A-Z, А-Я, а-я, [0 –9]; {1, 100} | Нет | .xlsx | Документ вывода |

### Метод

Для проектирования и разработки реляционной базы данных, которая будет хранить данные для графика учебного процесса используется технология создания миграций, которая включают в себя определённую модель для создания объектов в базе. Данная технология позволяет создавать, отслеживать и управлять версиями базы данных при её проектировании.

В процессе разработки программы используются следующие методы программирования:

* Объектно-ориентированное программирование - основанная на представлении программы в виде совокупности взаимодействующих объектов, каждый из которых является экземпляром определённого класса, а классы образуют иерархию наследования (Рисунок 6);

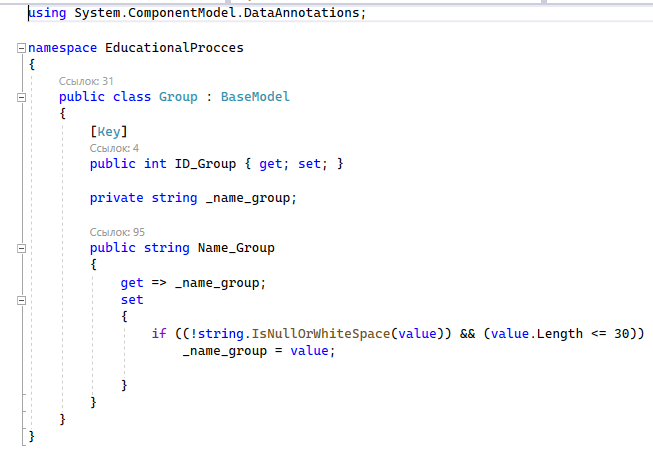


Рисунок - Объектно-ориентированное программирование

* Императивное программирование — это парадигма программирования (стиль написания исходного кода компьютерной программы), для которой характерно следующее: в исходном коде программы записываются инструкции (команды); инструкции должны выполняться последовательно; при выполнении инструкции данные, полученные при выполнении предыдущих инструкций, могут читаться из памяти; данные, полученные при выполнении инструкции, могут записываться в память (Рисунок 7);



Рисунок - Императивное программирование

* Модульное программирование – разделение программы на независимые модули. Главное особенностью этого метода является разработка программного продукта в виде независимых блоков, которые имеют имя - модули. Использование модульного программирования позволяет упростить тестирование программы и обнаружение ошибок (Рисунок 8);

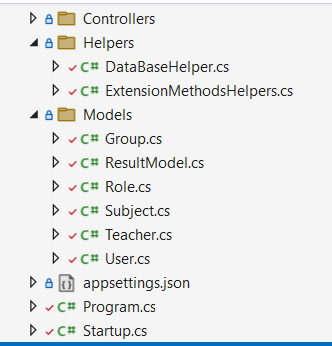


Рисунок - Модульное программирование

* API - использование программного интерфейса приложения, представляющий собой набор методов, классов и функций, обеспечивающих взаимодействие внутри программы (Рисунок 9).

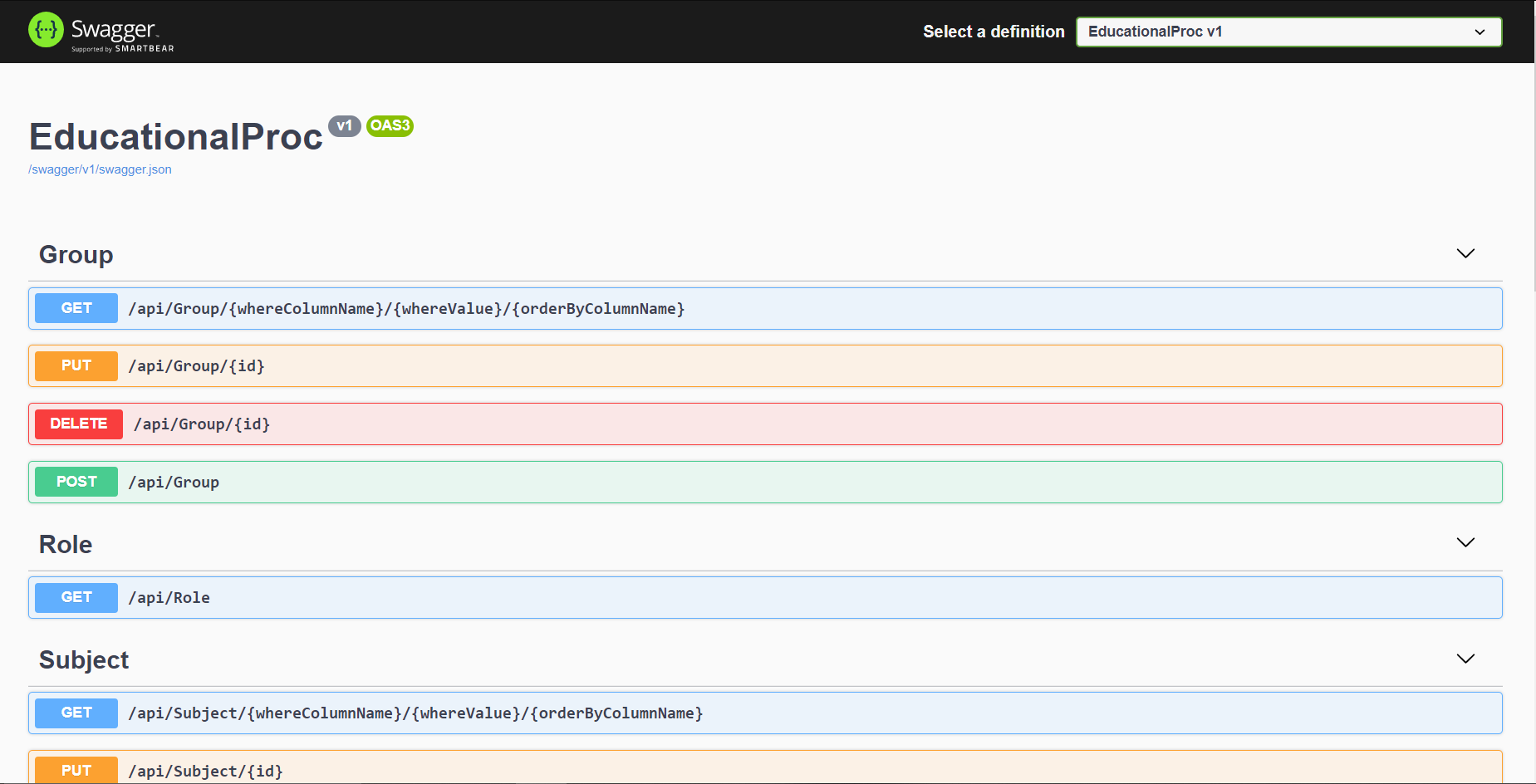


Рисунок - API

### Тесты

Тестирование систем проводилось в несколько этапов. На Рисунке 10 показана схема тестирования системы.



Рисунок - Схема тестирования

Результаты тестовых испытаний продемонстрированы в ПРИЛОЖЕНИЕ Б. СЦЕНАРИЙ ТЕСТОВЫХ ИСПЫТАНИЙ.

### Контроль целостности данных

Контроль целостности данных предоставляет возможность обнаружения аномалий в программном коде, из-за которых могут возникать ошибки критического характера или незначительные ошибки.

В Таблице 6 представлены различные ситуации и аномалии, которые могут возникнуть в процессе эксплуатации программы учебного процесса «EducationalProcces».

Таблица - Контроль целостности данных

| № | Ситуация | Аномалия | Реакция программа | Примечание |
| --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 |
| 1 | Введение неверного логина и пароля | Данные, которые ввел пользователь нет в базе данных | Программа при помощи механизма валидации и строки подключения ищет необходимую информацию в базе данных. Если данные будут не найдет, то появится сообщение «Неверный логин или пароль», в случае если неправильно введен только пароль появится сообщение «Неверный пароль» | Такой механизм проверки будет происходить даже если пользователь заполнит только одно поле. |
| 2 | Устройство, с помощью которого открывается учебный процесс, не подключено к сети Интернет | Так ка устройство не подключено к сети Интернет и не может связаться с сторонними ресурса при помощи API (Application programming interface) и получить данные для обработки и вывода их на странице происходит внутренняя ошибка | Специальный механизм  определяет подключено  ли устройство к сети  Интернет. Если  устройство не может  получить доступ к  стороннему ресурсу, то специальный контроллер переправляет пользователя на специальный представление (вид), на котором будет отображено специальное сообщение для пользователя | Данный механизм  перенаправления  пользователя на  специальное  преставления (вид),  работает для каждого  стороннего ресурса,  который  взаимодействует с  учебным процессом  при помощи API  (Application  programming  interface) |
| 3 | Созданием новой группы | Пользователь заполняет не все поля | Добавление данных в базу данных не происходит из-за не введённых данных, появляется сообщение об ошибке |  |
| 4 | Созданием нового задания | Пользователь заполняет не все поля | Добавление данных в базу данных не происходит из-за не введённых данных, появляется сообщение об ошибке |  |
| 5 | Созданием нового пользователя | Пользователь заполняет не все поля | Добавление данных в базу данных не происходит из-за не введённых данных, появляется сообщение об ошибке |  |

## Проектирование

### Схема баз данных

На Рисунке 11 представлена логическая схема данных приложения. На схеме приведены следующие таблицы баз данных:

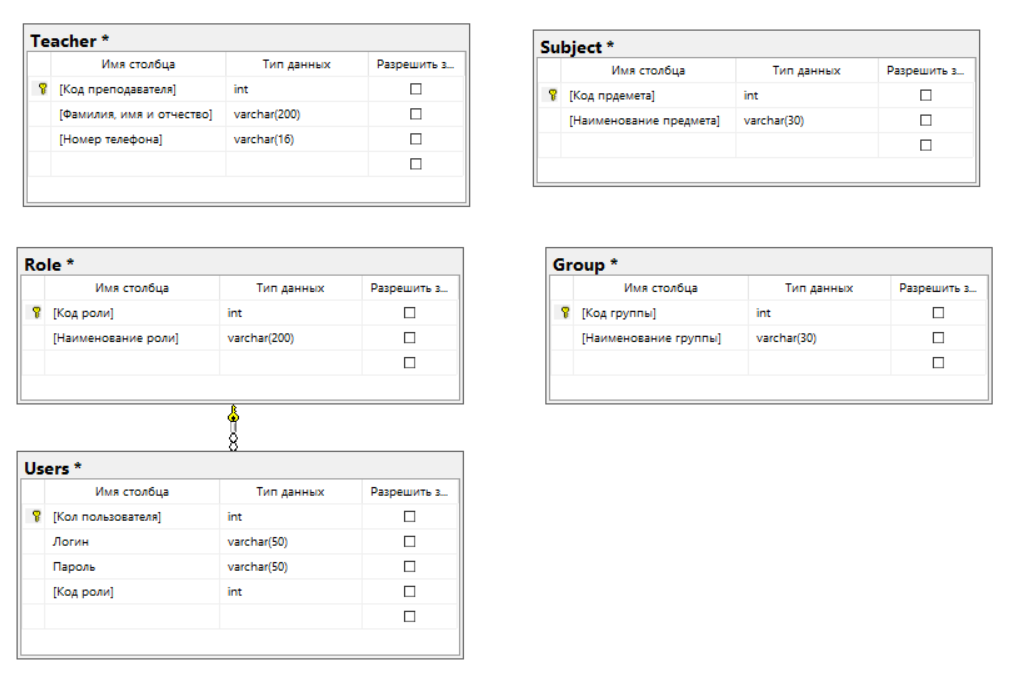


Рисунок - Логическая схема данных

На Рисунке 12 представлена схема базы данных, с которой взаимодействует график учебного процесса «EducationalProcces».



Рисунок – Физическая схема данных

База данных состоит из 5 сущностей (таблиц), которые имеют определённый набор атрибутов (полей) для записи информации. Каждая таблицы имеет своё уникальное поле идентификатор, по которым формируются связи между таблицами. Помимо сущностей в данной базе данных содержится представление (логическая таблица), которая служит для формирования и вывода информации пользователю в понятном виде.

В Таблице 7 представлен словарь данных для разработанной базы данных, с которой взаимодействует программа.

Таблица - Словарь данных

| Ключ | Поле | Тип данных  Размер | Пустое значение | Описание |
| --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 |
| Таблица Group | | | | |
| PK | ID\_Group | int | Нет | Код записи новой группы |
|  | Name\_Group | varchar (30) | Нет | Наименование новой группы |
| Таблица Role | | | | |
| PK | ID\_Role | int | Нет | Код записи новой роли |
|  | Name\_Role | varchar (200) | Нет | Наименование новой роли |
| Таблица Subject | | | | |
| PK | ID\_Subject | int | Нет | Код записи нового занятия |
|  | Name\_Subject | varchar (30) | Нет | Наименование нового занятия |
| Таблица Teacher | | | | |
| PK | ID\_Teacher | int | Нет | Код записи нового преподавателя |
|  | FIO | varchar (200) | Нет | Имя, Фамилии, Отчества нового преподавателя |
|  | Phone | varchar (16) | Нет | Номер телефона новой преподавателя |
| Таблица Users | | | | |
| PK | ID\_User | int | Нет | Код записи нового пользователя |
|  | Login | varchar (50) | Нет | Логин нового пользователя |
|  | Password | varchar (50) | Нет | Пароль нового пользователя |
| FK | Role\_ID | int | Нет | Код записи роли |

### 

### Диаграмма классов

На Рисунке 13 приставлена диаграмма классов приложения «EducationalProcces», на ней графически изображены классы, которые стоят из полей, методов и свойств.

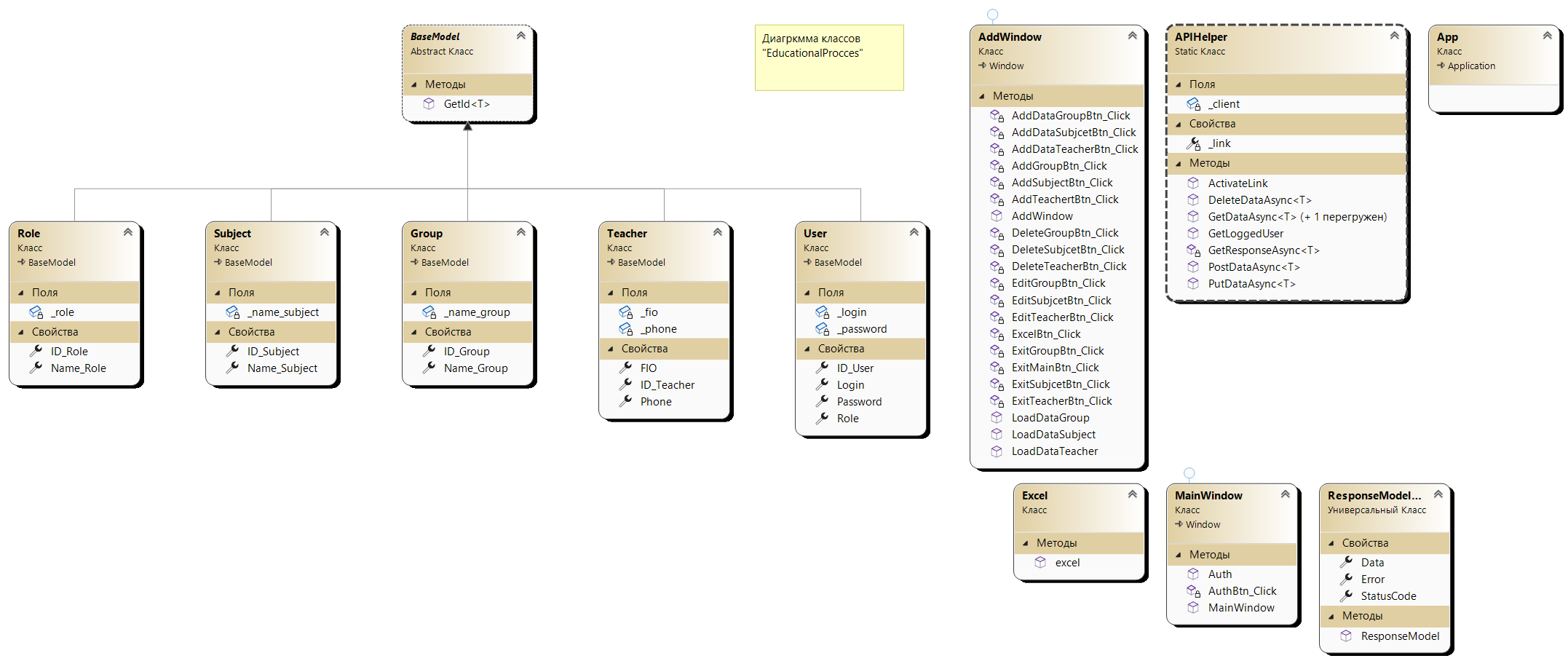


Рисунок - Диаграмма классов

В Таблице 8 представлено описание диаграммы классов проекта.

Таблица - Описание диаграммы классов

| №п/п | Наименование | Описание |
| --- | --- | --- |
| 1 | 2 | 3 |
| BaseModel | | |
| Методы | | |
| 1 | GetId<T> | Метод, в которой передаются значения |
| Role | | |
| Поля | | |
| 1 | \_role | Наименование роли |
| Subject | | |
| Поля | | |
| 1 | \_name\_subject | Наименование предмета |
| Group | | |
| Поля | | |
| 1 | \_name\_group | Наименование группы |
| Teacher | | |
| Поля | | |
| 1 | \_fio | Фамилия, имя и отчество преподавателя |
| 2 | \_phone | Номер телефона преподавателя |
| User | | |
| Поля | | |
| 1 | \_login | Логин (никнейм) пользователя |
| 2 | \_password | Пароль пользователя |
| AddWindow | | |
| Методы | | |
| 1 | AddDataGroupBtn\_Click | Добавление новой группы в БД |
| 2 | AddDataSubjectBtn\_Click | Добавление нового предмета в БД |
| 3 | AddDataTeacherBtn\_Click | Добавление нового преподавателя в БД |
| 4 | AddGroupBtn\_Click | Открытие панели DataGroup |
| 5 | AddSubjectBtn\_Click | Открытие панели DataSubject |
| 6 | AddTeacherBtn\_Click | Открытие панели DataTeacher |
| 7 | AddWindow | Обращение к методу класса APIHelper |
| 8 | DeleteGroupBtn\_Click | Удаление группы из БД |
| 9 | DeleteSubjectBtn\_Click | Удаление предмета из БД |
| 10 | DeleteTeacherBtn\_Click | Удаление преподавателя из БД |
| 11 | EditGroupBtn\_Click | Редактирование группы в БД |
| 12 | EditSubjectBtn\_Click | Редактирование предмета в БД |
| 13 | EditTeacherBtn\_Click | Редактирование преподавателя в БД |
| 14 | ExcelBtn\_Click | Обращение к методу из класса Excel |
| 15 | ExitGroupBtn\_Click | Закрытие из панели DataGroup |
| 16 | ExitMainBtn\_Click | Закрытие из панели Main |
| 17 | ExitSubjectBtn\_Click | Закрытие из панели DataSubject |
| 18 | ExitTeacherBtn\_Click | Закрытие из панели DataTeacher |
| 19 | LoadDataGroup | Вывод данных о группах из базы данных в DataGroup через API |
| 20 | LoadDataSubject | Вывод данных о предметах из базы данных в DataSubject через API |
| 21 | LoadDataTeacher | Вывод данных о преподавателях из базы данных в DataTeacher через API |
| APIHelper | | |
| Поля | | |
| 1 | \_client | Новый HttpClient |
| Методы | | |
| 1 | ActivateLink | Подключение к json файлу |
| 2 | DeleteDataAsync<T> | Работа с данными при удалении по ID |
| 3 | GetDataAsync<T> | Работа с данными для получения их из базы данных |
| 4 | GetLoggedUser | Работа с данными для валидации |
| 5 | GetResponseAsync<T> | Работа с данными |
| 6 | PostDataAsync<T> | Работа с данными при добавлении новых данных |
| 7 | PutDataAsync<T> | Работа с данными при изменении по ID |
| Excel | | |
| Методы | | |
| 1 | excel | Создание Excel файла с графиком учебного процесса |
| MainWindow | | |
| Методы | | |
| 1 | Auth | Валидация (проверка) вводимых данных |
| 2 | AuthBtn\_Click | Обращение к методу Auth |
| 3 | MainWindow | Обращение к методу класса APIHelper |
| ReaponseModel<T> | | |
| Методы | | |
| 1 | ResponseModel | Проверка данных, получаемых с API |

### Функциональная схема задачи

На Рисунке 14 представлена функциональная схема для пользователя или программы, на которой графически изображены возможности пользователя в рамках программного комплекса графика учебного процесса «EducationalProcces».

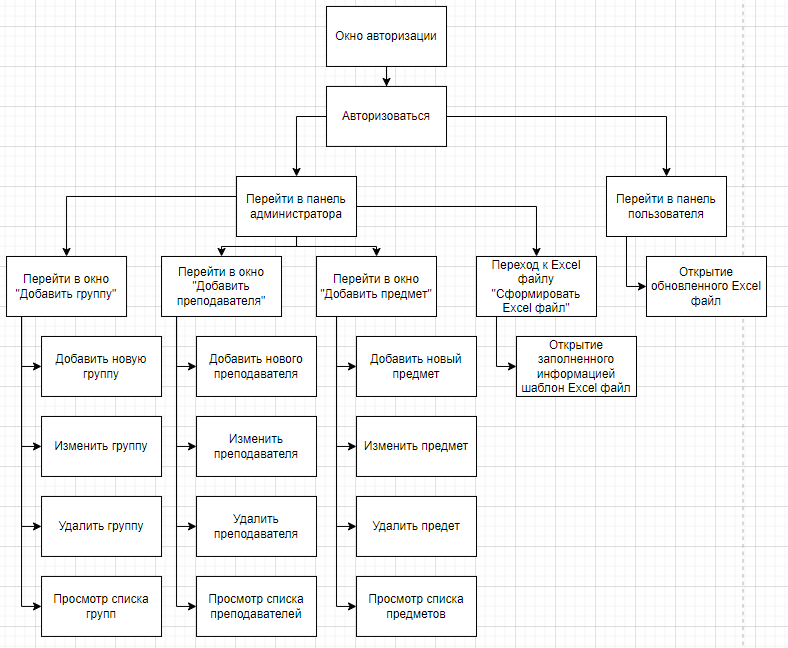


Рисунок - Функциональная схема

Предоставляется авторизация пользователям.

После авторизации как администратор появляется возможность перехода на окна: «Добавить группу», «Добавить предмет» и «Добавить преподавателя», так же при нажатии на кнопку «Сформировать Excel файл» производится автоматическое создание Excel файла с графиком учебного процесса.

В окне «Добавить группу» присутствует просмотр списка групп, добавление/изменение/удаление групп (поля: наименование группы).

В окне «Добавить предмет» присутствует просмотр списка предметов, добавление/изменение/удаление предметов (поля: наименование предмета).

В окне «Добавить преподавателя» присутствует просмотр списка преподавателей, добавление/изменение/удаление преподавателей (поля: фамилия, имя, отчество и номер телефона преподавателя).

После авторизации как пользователь производится переход в Excel файл с графиком учебного процесса.

Пояснение к функциональной схеме приложения представлены в Таблице 9.

Таблица - Описание функциональной схемы

| № | Наименование | Описание |
| --- | --- | --- |
| 1 | 2 | 3 |
| 1 | Авторизоваться | Предназначена для того, чтобы авторизоваться в системе |
| 2 | Перейти в панель пользователя | Предназначена для того, чтобы перейти в окно «Панель пользователя» |
| 3 | Перейти в панель администратора | Предназначена для того, чтобы перейти в окно «Панель администратора» |
| 4 | Перейти в окно «Добавить группу» | Предназначена для того, чтобы перейти администратору в окно «Добавить группу» |
| 5 | Добавить новую группу | Предназначена для того, чтобы администратора добавить новую группу |
| 6 | Изменить группу | Предназначена для того, чтобы администратора изменить группу |
| 7 | Удалить группу | Предназначена для того, чтобы администратора удалить группу |
| 8 | Просмотр списка групп | Предназначена для того, чтобы администратора просмотреть список групп |
| 9 | Перейти в окно «Добавить преподавателя» | Предназначена для того, чтобы перейти администратору в окно «Добавить преподавателя» |
| 10 | Добавить нового преподавателя | Предназначена для того, чтобы администратора добавить нового преподавателя |
| 11 | Изменить преподавателя | Предназначена для того, чтобы администратора изменить преподавателя |
| 12 | Удалить преподавателя | Предназначена для того, чтобы администратора удалить преподавателя |
| 13 | Просмотр списка преподавателей | Предназначена для того, чтобы администратора просмотреть список преподавателей |
| 14 | Перейти в окно «Добавить предмет» | Предназначена для того, чтобы перейти администратору в окно «Добавить предмет» |
| 15 | Добавить новый предмет | Предназначена для того, чтобы администратору добавить новый предмет |
| 16 | Изменить предмет | Предназначена для того, чтобы администратору изменить предмет |
| 17 | Удалить предмет | Предназначена для того, чтобы администратору удалить предмет |
| 18 | Просмотр списка предметов | Предназначена для того, чтобы администратора просмотреть администратору список предметов |
| 19 | Переход к Excel файлу «Сформировать Excel файл» | Предназначена для того, чтобы администратора перейти в Excel файл «Сформировать Excel файл» |
| 20 | Открытие заполненного информацией шаблон Excel файл | Предназначена для того, чтобы администратор открыл автоматически заполненный информацией шаблон Excel файла |
| 21 | Открытие обновленного Excel файл | Предназначена для того, чтобы пользователь открыл обновленный Excel файла |

### Структурная схема программы

На Рисунке 15 представлен общий план структурной схемы учебного процесса «EducationalProcces», в которой графически показано взаимодействие всех программных модулей и их взаимодействия друг с другом.

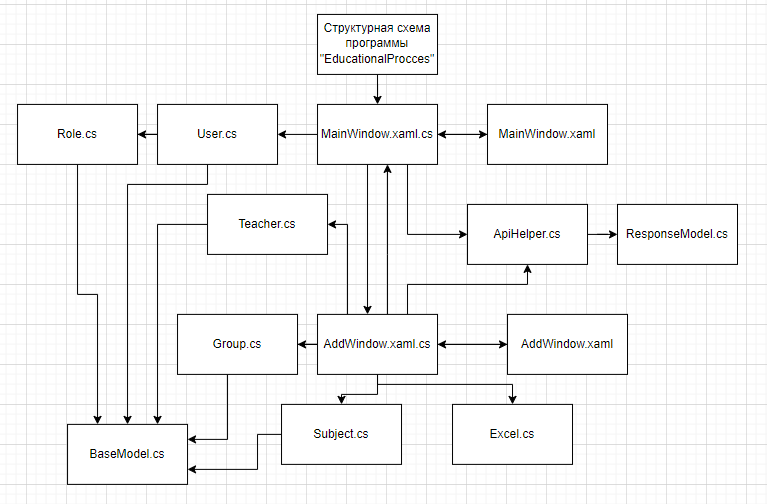


Рисунок - Структурная схема учебного процесса «EducationalProcces»

Пояснение к структурной схеме приложения представлены в Таблице 10.

Таблица - Описание структурной схемы программы

| № | Название модуля | Описание модуля |
| --- | --- | --- |
| 1 | 2 | 5 |
| 1 | MainWindow.xaml.cs | Логика главного окна |
| 2 | MainWindow.xaml | Разметка окна авторизации |
| 2 | ApiHelper.cs | Помощник для работы с API, содержит строку подключения к API, методы работы с данными |
| 3 | ResponseModel.cs | Класс проверки данных, получаемых с API |
| 4 | User.cs | Модуль пользователя для API |
| 5 | Role.cs | Модуль роли для API |
| 6 | AddWindow.xaml.cs | Логика главного окна |
| 7 | AddWindow.xaml | Разметка главного окна, где располагаются вкладки: Добавить группу, Добавить предмет, Добавить преподавателя |
| 7 | Teacher.cs | Модуль преподавателя для API |
| 8 | Group.cs | Модуль групп для API |
| 9 | Subject.cs | Модуль предметов для API |
| 10 | BaseModel.cs | Общая модель, в которую передаются значения |
| 11 | Excel.cs | Модуль для создания Excel документа |

### Схема архитектуры программы

Приложение работает в архитектуре, представленной на Рисунке 16. Для работы с данными необходимо подключение к API.

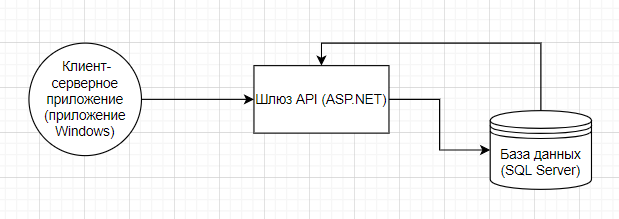


Рисунок - Схема архитектуры приложения

Приложение имеет связь с шлюзом API, представляющее собой приложение для администрирования базой данных SQL Server.

### Схема пользовательского интерфейса

На Рисунке 17 представлена схема интерфейса приложения. Схема интерфейса отображает все возможные переходы между окнами в программе, которые доступны администратору. На схеме приведены следующие окна:

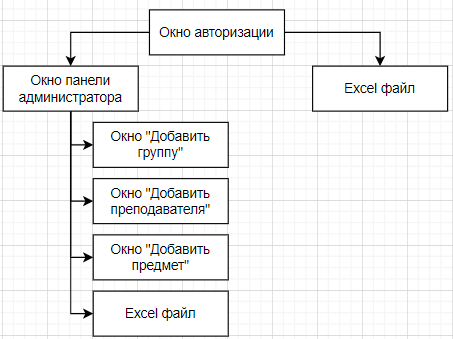


Рисунок - Схема интерфейса

В схеме интерфейса показано, что после открытия страницы авторизация в зависимости от роли пользователь может переходить по страницам, используя панель управления. Подробное описание схемы интерфейса представлено в Таблице 11.

Таблица - Описание схемы интерфейса

| № | Название функции | Описание |
| --- | --- | --- |
| 1 | 2 | 3 |
| 1 | Окно «Добавить группу» | Предназначена для добавления, изменения и удаления данных о группах в базе данных |
| 2 | Окно «Добавить преподавателя» | Предназначена для добавления, изменения и удаления данных о преподавателях в базе данных |
| 3 | Окно «Добавить предмет» | Предназначена для добавления, изменения и удаления данных о предметах в базе данных |
| 4 | Excel файл | Автоматическое создание графика учебного процесса |
| 5 | Excel файл | Просмотр обновления данных графика учебного процесса |

## Результаты работы программы

В результате выполнения поставленной задачи, было разработано требуемое программное решение, которое является создание учебного процесса для написания (формирования) и редактирования графика и отображение данных в нем. Подробное описание работы программы приведено в ПРИЛОЖЕНИЕ В. РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ. На Рисунке 18 Представлено окно «Добавить группу».

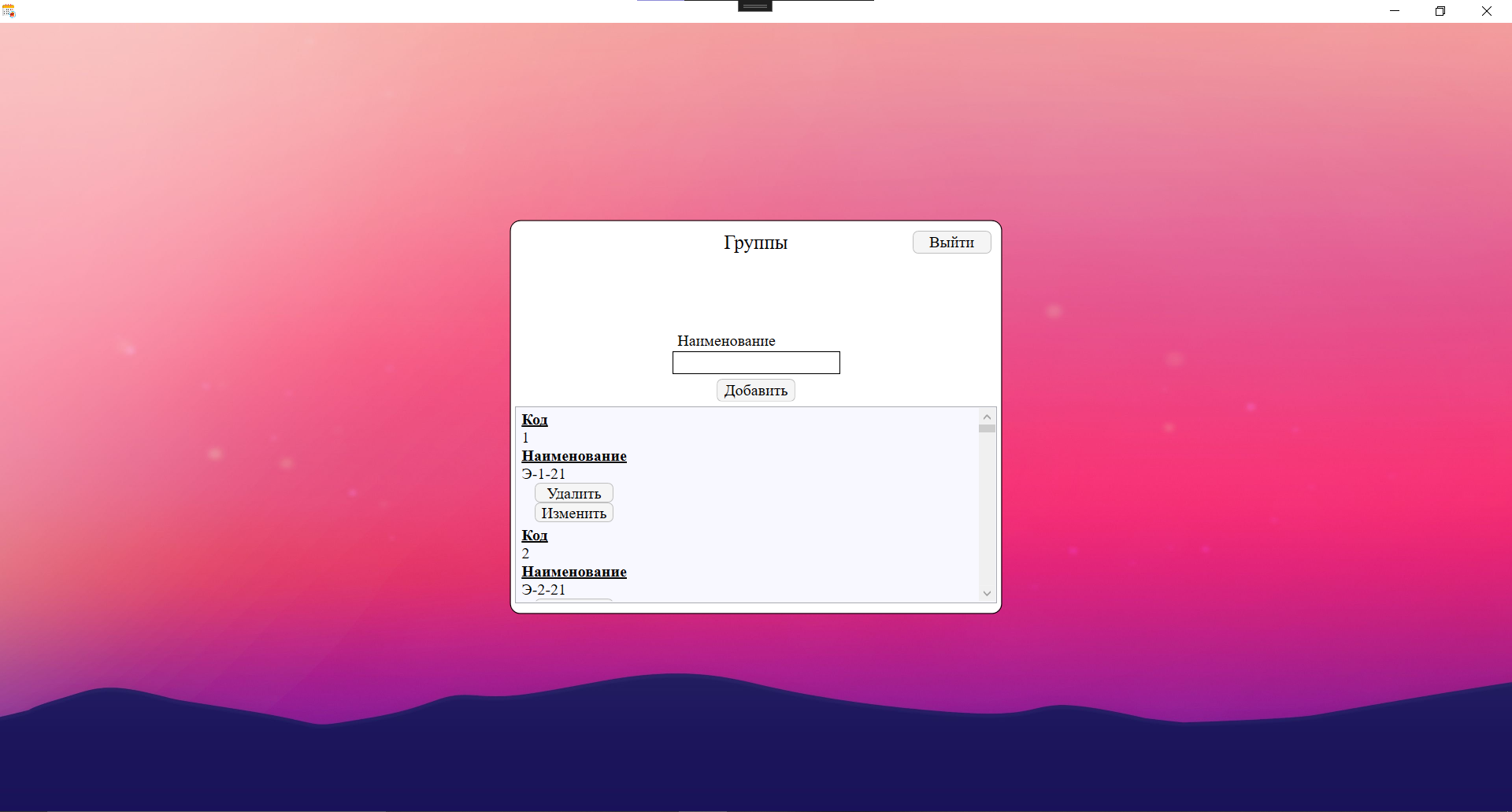


Рисунок - Окно «Добавить группу»

На Рисунке 19 представлено окно «Добавить предмет», где производится добавление предмета и повторное добавление такого же предмета. Появилось сообщение об ошибке «Такой предмет уже существует».

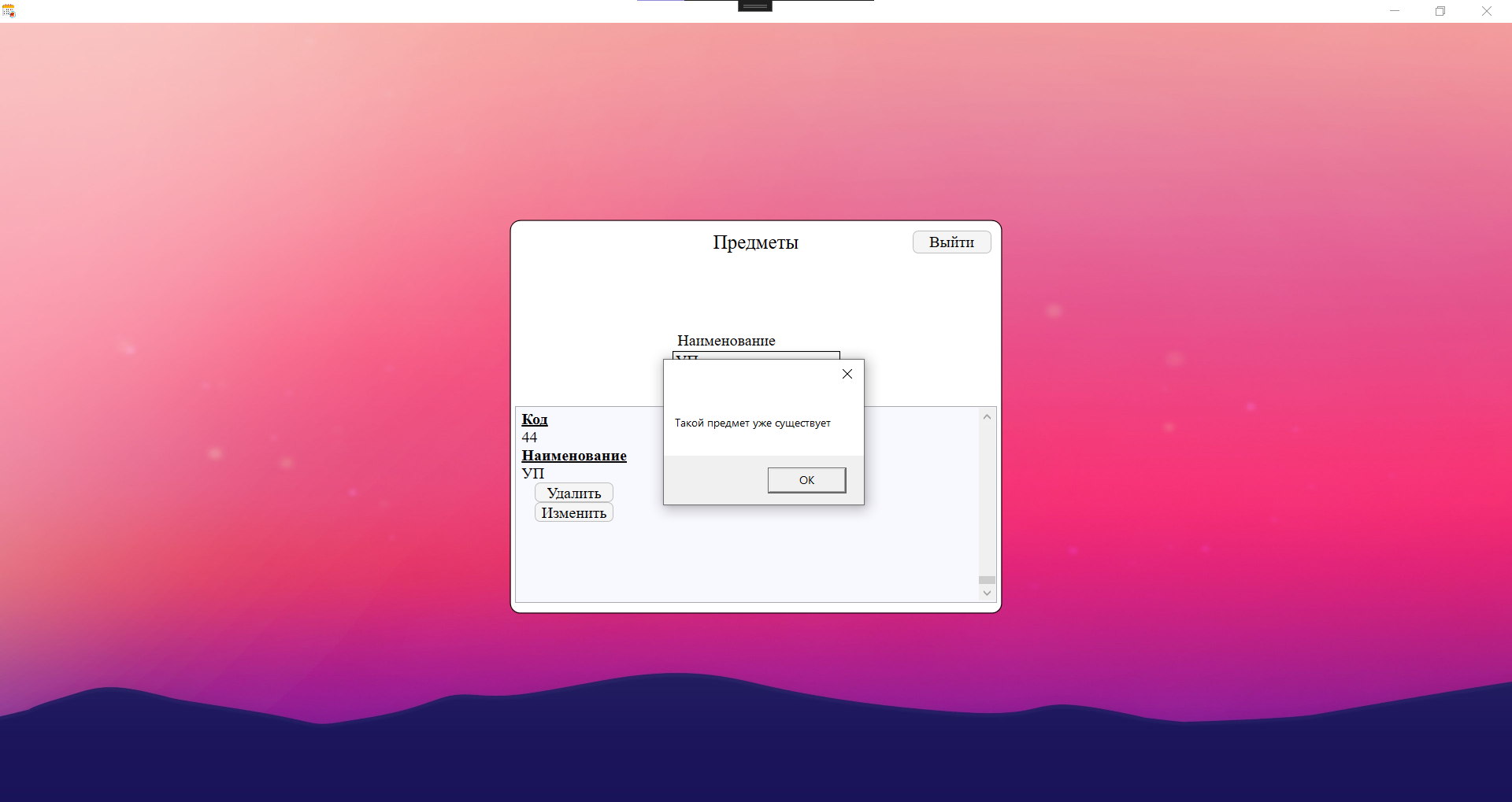


Рисунок - Окно «Добавить предмет», функция добавления предмета

Программа состоит из основной части, а именно из реляционной базы данных, которые размещаются на сервере.

Рассмотреть состав и структуру базы данных можно в пункте 2.3.1. Схема базы данных ([ссылка](#_Логическая_схема)), а также можете ознакомится с исходным скриптом базы данных, который описан в Приложении Г. Скрипт базы данных. Узнать о взаимодействии пользователя с графиком учебного процесса «EducationalProcces» можно из Приложения В. Руководство пользователя.

# ТЕХНОЛОГИЧЕСКАЯ ЧАСТЬ

## Инструментальные средства разработки

Для разработки графика учебного процесса использовалась среда разработки Microsoft Visual Studio 2022, которая позволяет писать код на языке C# с системой WPF для построения клиентских приложений Windows, а также удобно осуществлять отладку.

Swagger - язык описания интерфейсов для описания RESTful API, выраженных с помощью JSON.

Для работы с документацией использовался текстовый процессор, предназначенный для создания, просмотра и редактирования текстовых документов, с локальным применением простейших форм таблично-матричных алгоритмов. Выпускается корпорацией Microsoft в составе пакета Microsoft Office. Первая версия была написана Ричардом Броди для IBM PC, использующих DOS, в 1983 году. Текущей версией является Microsoft Office Word 2019 для Windows и macOS.

Для создания схем использовался векторный графический редактор Diagrams, редактор диаграмм и блок-схем для различных операционных систем.

В качестве операционной системы использовалась операционная система для персональных компьютеров и рабочих станций, разработанная корпорацией Microsoft в рамках семейства Windows NT. После Windows 8.1 система получила номер 10, минуя 9.

Для работы с базами данных использовалась SQL Server Management Studio, которая предоставляет средства для настройки, наблюдения и администрирования экземпляров SQL Server и баз данных. С помощью SSMS можно развертывать, отслеживать и обновлять компоненты уровня данных, используемые вашими приложениями, а также создавать запросы и скрипты.

В качестве сервера базы данных использовался SQL Server 2017. SQL Server - это система управления реляционными базами данных, или RDBMS, разработанная и продаваемая Microsoft. Подобно другому программному обеспечению СУБД, SQL Server построен на основе SQL, стандартного языка программирования для взаимодействия с реляционными базами данных.

## Отладка программы

Отладка учебного процесса «EducationalProcces» производилось при помощи встроенных инструментов Microsoft Visual Studio. Также для отладки серверной часть был применено специальное расширение, а именно Xdebug, которое позволило отслеживать протекающие процессы в ходе работы программы, что позволило уменьшить количество критических ошибок и нарушений в целостности данных.

При незнании способа или решения исправления ошибки, которая возникла при разработке учебного процесса, использовались дополнительная информация, взятая из специализированной литературы, профильных сайтов и официальных мануалах. С открытыми источниками и литературой можно ознакомится в разделе Список используемых материалов.

Пример самых распространённых ошибок продемонстрирован на Рисунке 9.

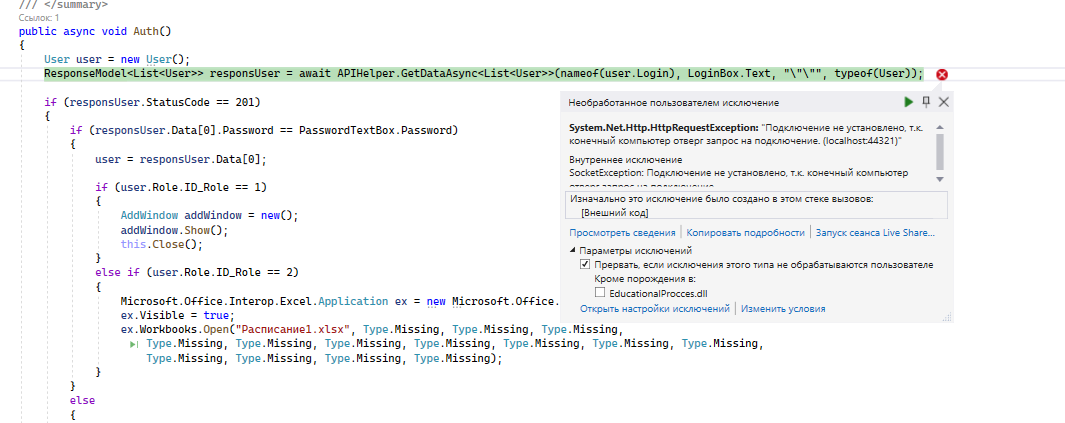


Рисунок - Ошибка в подключении к API

## Защитное программирование

Особое защитное программирование не применялось при разработке учебного процесса «EducationalProcces». На определённом этапе проектирования программы были заданы основные свойства для каждого поля, с которыми взаимодействует пользователь.

На основе выявленных свойств была создана система валидации (проверки) каждого поля, в которые пользователь вводи логин (никнейм) и пароль, который не должен отображаться в интерфейсе системы. Работа данного механизма продемонстрирована на Рисунке 20.

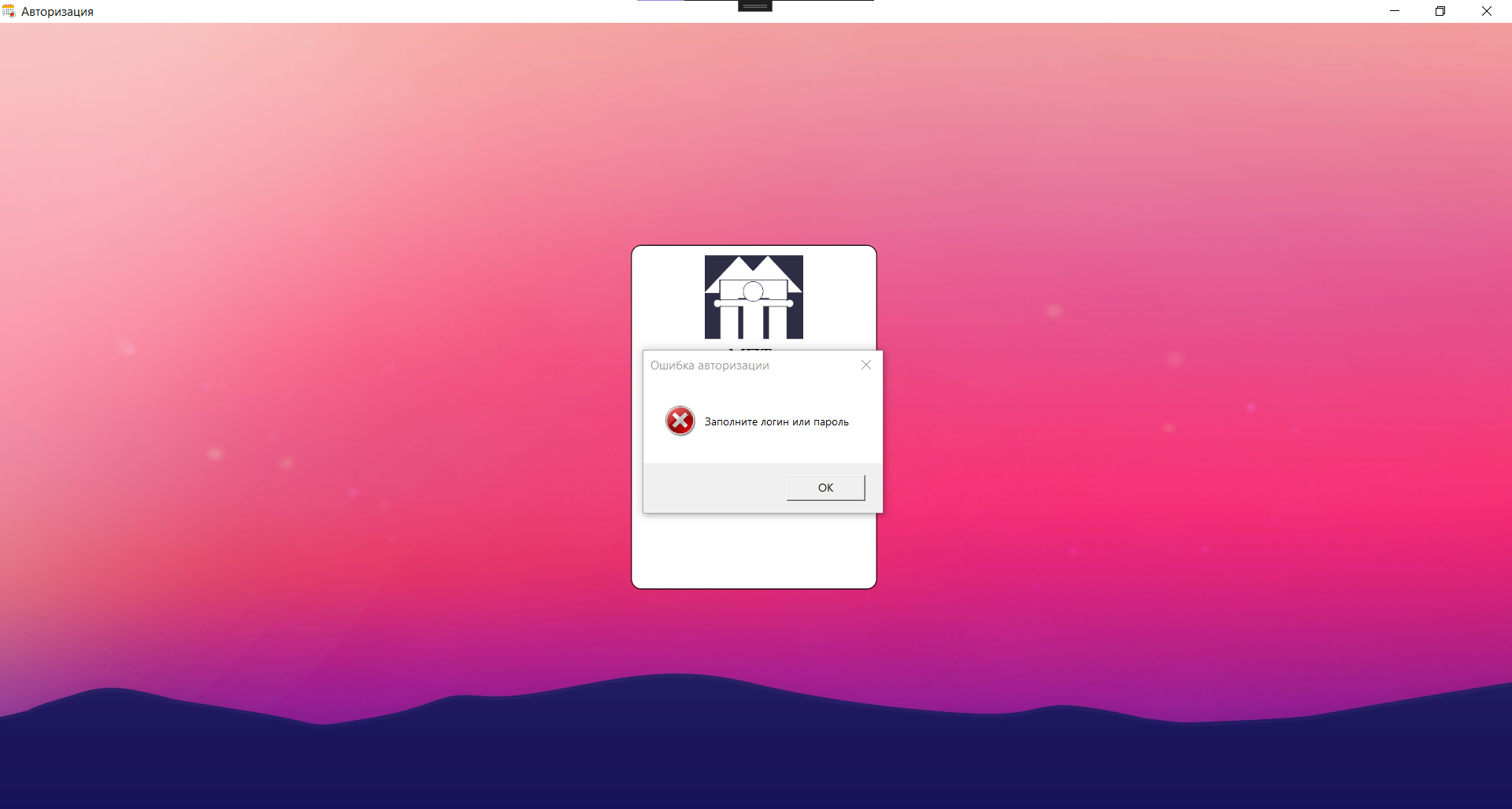


Рисунок - Проверка валидации данных

Для реализации полной защиты система должна удовлетворять такие требования, как масштабируемость, минимальная вероятность сбоев, наличие средств защиты данных от потери и не санкционируемого доступа.

## Характеристики программы

### Список подключенных библиотек

В процессе разработки были использованы следующие библиотеки:

* Microsoft.EntityFrameworkCore;
* Microsoft.EntityFrameworkCore.SqlServer;
* Microsoft.EntityFrameworkCore.Tools;
* Microsoft.Extensions.Configuration.Json;
* Microsoft.SqlServer.SqlManagementObjects.

### Список модулей

В процессе разработки были созданы программные модули, указанные в Таблице 11.

Таблица - Описание модулей

| Наименование | Назначение | Размер (в КБ) | Количество строк |
| --- | --- | --- | --- |
| 1 | 2 | 3 | 4 |
| EducationalProcApi | | | |
| Program.cs | Основной класс проекта | 1 | 24 |
| Startup.cs | Класс настроек проекта | 2 | 60 |
| GroupController.cs | Контроллер группы | 7 | 142 |
| TacherController.cs | Контроллер преподавателя | 7 | 144 |
| SubjectController.cs | Контроллер предмета | 7 | 142 |
| RoleController.cs | Контроллер роли | 1 | 33 |
| UserController.cs | Контроллер пользователя | 9 | 198 |
| DataBaseHelper.cs | Класс для подключения к базе данных | 2 | 40 |
| ExtensionMethodsHelper.cs | Файл методов расширения | 4 | 82 |
| Group.cs | Модель группы | 1 | 23 |
| Subject.cs | Модель предмета | 1 | 23 |
| Teacher.cs | Модель преподавателя | 1 | 34 |
| ResultModel.cs | Модель результата | 1 | 7 |
| Role.cs | Модель роли | 1 | 23 |
| User.cs | Модель пользователя | 1 | 40 |
| EducationalProcces | | | |
| ApiHelper.cs | Класс работы с API | 6 | 116 |
| Group.cs | Модель филиала | 1 | 23 |
| Subject.cs | Модель кабинета | 1 | 23 |
| Teacher.cs | Модель оборудования | 1 | 33 |
| Role.cs | Модель роли | 1 | 23 |
| User.cs | Модель пользователя | 1 | 40 |
| BaseModel.cs | Класс базовой модели | 1 | 7 |
| ResponseModel.cs | Модель ответа от API | 1 | 16 |
| MainWindows.cs | Окно «Авторизации» | 4 | 84 |
| MainWindows.xaml | Разметка для окна «Авторизация» | 4 | 90 |
| AddWindow.cs | Окно «Главная страница» | 21 | 458 |
| AddWindow.xaml | Разметка для окна «Главная страница» | 20 | 392 |
| BtnStyle.xaml | Разметка для стиля кнопок | 2 | 19 |
| Excel.cs | Класс создания Excel файла | 375 | 4766 |

# ЗАКЛЮЧЕНИЕ

Во время выполнения выпускной квалификационной работы была изучена актуальная проблема в области автоматизации процесса создания графика учебного процесса, добавления, изменения и удаления данных для него. Основным аспектом в рассматриваемой области, является организация всего процесса и предоставления удобной кроссплатформенной платформы для его реализации.

В ходе решения поставленной задачи был разработан учебный процесс «EducationalProcces». Данное решения состоит из двух частей, а именно реляционная база данных, которая имеет определенный набор сущностей и которые взаимодействуют между друг с другом посредством специальных связей. Данная база данных была реализована для хранения информации о группах, о преподавателях и о занятиях (которые хранятся на сервере), авторах и редакторах.

Второй частью рассматриваемого программного решения считается разработанный программы, которая напрямую взаимодействует с базой данных и с пользователями. Данные ресурс предназначен для отображения информации в понятной человеку форме, а также предоставляет функциональный набор возможностей.

При выполнении выпускной квалификационной работы были закреплены и улучшены знания в области написания программных решений на .NetCore с использованием языка C#.

После оценки итогов и полученного результата работы программного решения, был сделан вывод о целесообразности использования такого подхода для автоматизации процесса создания графика учебного плана. Данное приложение позволить сформировать единую систему, в которой взаимодействую пользователи, возможностями и методами их решения.

# СПИСОК ИСПОЛЬЗОВАННЫХ МАТЕРИАЛОВ

1. ГОСТ 19404- 79 ЕСПД. Пояснительная записка. ПЕРЕИЗДАНИЕ Января 2010 г. (Дата обращения: 25.05.2022)

1. МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ ДЛЯ ВЫПОЛНЕНИЯ ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЫ по профессиональному модулю ПМ.03. «Участие в интеграции программных модулей». (Дата обращения: 25.05.2022)
2. ГОСТ 7.80-2000 СИБИД. Библиографическая запись. Заголовок. Общие требования и правила составления. (Дата обращения: 25.05.2022)
3. ГОСТ Р 7.0.5-2008 БИБЛИОГРАФИЧЕСКАЯ ССЫЛКА. Общие требования и правила составления. (Дата обращения: 25.05.2022)
4. ГОСТ 19.101-77 ЕСПД. Виды программ и программных документов. (Дата обращения: 01.05.2022)
5. ГОСТ 19.103-77 ЕСПД. Обозначение программ и программных документов. (Дата обращения: 25.05.2022)
6. ГОСТ 19.105-78 ЕСПД. Общие требования к программным документам. (Дата обращения: 25.05.2022)
7. C# для профессионалов. Тонкости программирования, Автор: Джон Скит, Год издания: 2014, Количество страниц 608. (Дата обращения: 25.04.2022)
8. Программист-прагматик. Путь от подмастерья к мастеру, Авторы: Эндрю Хант, Дэвид Томас, Год издания: 2000, Количество страниц 289. (Дата обращения: 25.04.2022)
9. ​Идеальный программист. Как стать профессионалом разработки ПО, Автор: Роберт Мартин, Год издания: 2012, Количество страниц 240. (Дата обращения: 26.04.2022)
10. C# Smorgasbord, Авторы: Filip Ekberg, Год издания: 2012, Количество страниц 278. (Дата обращения: 27.04.2022)
11. Паттерны проектирования, Авторы: Фримен Э., Робсон Э., Сьерра К., Бейтс Б., Год издания: 2018, Количество страниц 656. (Дата обращения: 28.04.2022)
12. Путь программиста, Авторы: Сонмез Джон, Год издания: 2018, Количество страниц 448. (Дата обращения: 28.04.2022)
13. Искусство автономного тестирования с примерами на C#, Авторы: Ошероув Р., Год издания: 2014, Количество страниц 360. (Дата обращения: 28.04.2022)
14. C# 7 и .NET Core. Кросс-платформенная разработка для профессионалов, Авторы: Прайс Марк, Год издания 2018, Количество страниц 640. (Дата обращения: 30.04.2022)
15. Язык программирования C# 7 и платформы .NET и .NET Core, Авторы: Джепикс Филипп и Троелсен Эндрю, Год издания 2018, Количество страниц 1328. (Дата обращения: 30.04.2022)
16. Официальный сайт Московского приборостроительного техникума. URL: https://mpt.ru/studentu/raspisanie-zanyatiy/ (Дата обращения: 25.04.2022).
17. Официальный сайт РЭУ им. Г.В. Плеханова. Учебные планы. URL: https://www.rea.ru/ru/org/colleges/instrcol/Pages/%D0%9E %D0%B1%D1%80%D0%B0%D0%B7%D0%BE%D0%B2%D0%B0%D0% BD%D0%B8%D0%B5.aspx#PPSSZ (Дата обращения: 27.04.2022).
18. Что такое автоматизация. Википедия. URL: https://ru.wikipedia.org/wiki/%D0%90%D0%B2%D1%82%D0%BE%D0%B C%D0%B0%D1%82%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D1 %8F (Дата обращения: 30.05.2022).
19. Автоматизация документооборота. URL: https://www.cleverence.ru/articles/bukhgalteriya/avtomatizatsiyadokumentooborota-na-predpriyatii-avtomatizirovannye-sistemy upravleniyadokumentami-/ (Дата обращения: 28.05.2022).
20. Единая система программной документации. Пояснительная записка. URL: https://docs.cntd.ru/document/1200007671 (Дата обращения: 27.05.2022).
21. Сравнение MySQL vs MariaDB. URL: https://losst.ru/sravnenie-mysql-vs-mariadb. (Дата обращения: 08.05.2022).
22. График учебного процесса. URL: http://www.пгатк.рф/students/dnevnoe-otdelenie/the-schedule-of-the-educational-process/ (Дата обращения: 09.05.2022).
23. Create Excel file in WPF. URL: https://www.youtube.com/watch?v=XfSI4kSXhdk (Дата обращения: 09.05.2022).
24. ПРОГРАММНАЯ РАБОТА С ТАБЛИЦАМИ EXCEL С ПОМОЩЬЮ БИБЛИОТЕКИ MICROSOFT.OFFICE.INTEROP.EXCEL. URL: https://nationalteam.worldskills.ru/skills/programmnaya-rabota-s-tablitsami-excel-s-pomoshchyu-biblioteki-microsoft-office-interop-excel/ (Дата обращения: 10.05.2022).
25. Работа с Excel с помощью C# (Microsoft.Office.Interop.Excel). URL: https://razilov-code.ru/2017/12/13/microsoft-office-interop-excel/ (Дата обращения: 10.05.2022).
26. Практическое руководство. Добавление диаграмм классов в проекты. URL: https://docs.microsoft.com/ru-ru/visualstudio/ide/class-designer/how-to-add-class-diagrams-to-projects?view=vs-2019 (Дата обращения: 30.05.2022).
27. Диаграммы классов в Visual Studio. URL: https://professorweb.ru/my/programs/visual-studio/level3/3\_6.php (Дата обращения: 30.05.2022).
28. Работа с excel на примерах C# с помощью Microsoft.Office.Interop.Excel. URL: https://www.nookery.ru/c-work-c-excel/ (Дата обращения: 30.05.2022).
29. Web API. URL: https://metanit.com/sharp/mvc/12.1.php (Дата обращения: 15.05.2022).
30. Учебник. Создание веб-API с помощью ASP.NET Core. URL: https://docs.microsoft.com/ru-ru/aspnet/core/tutorials/first-web-api?view=aspnetcore-5.0&tabs=visual-studio (Дата обращения: 15.05.2022).
31. СОЗДАНИЕ КАРКАСА ПРИЛОЖЕНИЯ. СОЗДАНИЕ И ИСПОЛЬЗОВАНИЕ СТИЛЕЙ. URL: https://nationalteam.worldskills.ru/skills/sozdanie-karkasa-prilozheniya-sozdanie-i-ispolzovanie-stiley/ (Дата обращения: 25.04.2022).
32. ПРОГРАММНЫЕ РЕШЕНИЯ ДЛЯ БИЗНЕСА. URL: https://nationalteam.worldskills.ru/skills/programmnye-resheniya-dlya-biznesa/ (Дата обращения: 25.04.2022).
33. СОЗДАНИЕ БАЗЫ ДАННЫХ. URL: https://nationalteam.worldskills.ru/skills/sozdanie-bazy-dannykh/ (Дата обращения: 27.04.2022).
34. РАЗРАБОТКА API. URL: https://nationalteam.worldskills.ru/skills/razrabotka-api/ (Дата обращения: 25.04.2022).
35. ПРОЕКТИРОВАНИЕ ER-ДИАГРАММЫ. URL: https://nationalteam.worldskills.ru/skills/proektirovanie-er-diagrammy/ (Дата обращения: 17.05.2022).
36. РАБОТА С СИСТЕМОЙ КОНТРОЛЯ ВЕРСИЙ. URL: https://nationalteam.worldskills.ru/skills/rabota-s-sistemoy-kontrolya-versiy/ (Дата обращения: 22.04.2022).
37. МОДУЛЬНОЕ ТЕСТИРОВАНИЕ (UNIT-TESTS). URL: https://nationalteam.worldskills.ru/skills/modulnoe-testirovanie-unit-tests/ (Дата обращения: 27.05.2022).
38. ПРОЕКТИРОВАНИЕ ДИАГРАММЫ КЛАССОВ UML (CLASS DIAGRAM). URL: https://nationalteam.worldskills.ru/skills/proektirovanie-diagrammy-klassov-uml-class-diagram/ (дата обращения: 30.05.2022).