МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждения образования «БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ

ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет информационных технологий

Кафедра программной инженерии

Специальность 1-40 01 01 Программное обеспечение информационных технологий

Направление специальности 1-40 01 01 10 Программное обеспечение информационных технологий (программирование интернет приложений)

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

**КУРСОВОГО ПРОЕКТА**

по дисциплине «Объектно-ориентированные технологии программирования и стандарты проектирования»

Тема Программное средство «Виртуальный деканат»

Исполнитель

студент (ка) 2 курса группы 5 Никоненко Андрей Романович

(Ф.И.О.)

Руководитель работы ассистент Панченко О.Л.

(учен. степень, звание, должность, подпись, Ф.И.О.)

Курсовой проект защищен с оценкой

Председатель Панченко О.Л.

(подпись)

Минск 2024

**Содержание**

[ВВЕДЕНИЕ 4](#_Toc166836107)

[1 Постановка задачи и анализ аналогов и прототипов 5](#_Toc166836108)

[1.1 Анализ аналогов и прототипов 5](#_Toc166836109)

[1.2 Постановка задачи 9](#_Toc166836110)

[1.3 Вывод по разделу 10](#_Toc166836111)

[2 Анализ требований к программному средству и разработка функциональных требований 11](#_Toc166836112)

[2.1 Описание средств разработки 11](#_Toc166836113)

[2.1.1 Microsoft Visual Studio 2022 11](#_Toc166836114)

[2.1.2 Платформа .NET Framework 8.0 11](#_Toc166836115)

[2.1.3 Расширяемый язык разметки XAML 11](#_Toc166836116)

[2.1.4 Технология Entity Framework 12](#_Toc166836117)

[2.1.5 Технология Fluent API 12](#_Toc166836118)

[2.1.6 Технология WPF 12](#_Toc166836119)

[2.1.7 SQLite 12](#_Toc166836120)

[2.2 Спецификация функциональных требований к программному средству 12](#_Toc166836121)

[2.3 Спецификация функциональных требований 13](#_Toc166836122)

[3 Проектирование программного средства 14](#_Toc166836123)

[3.1 Общая структура 14](#_Toc166836124)

[3.2 Проектирование схемы базы данных 14](#_Toc166836125)

[3.3 Взаимоотношения между классами 15](#_Toc166836126)

[3.4 Проектирование архитектуры проекта 15](#_Toc166836127)

[3.5 Проектирование последовательностей проекта 15](#_Toc166836128)

[4 Реализация программного средства 16](#_Toc166836129)

[4.1 Основные классы программного средства 16](#_Toc166836130)

[4.2 Описание классов и методов программного средства 16](#_Toc166836131)

[4.2.1 Выполнение входа в аккаунт 16](#_Toc166836132)

[4.2.2 Взаимодействие со списком студентов 16](#_Toc166836133)

[4.2.3 Взаимодействие с расписанием занятий 16](#_Toc166836134)

[4.2.4 Взаимодействие с профилем студента 17](#_Toc166836135)

[4.2.5 Взаимодействие с лентой новостей 17](#_Toc166836136)

[4.2.6 Взаимодействие с аттестацией студентов 17](#_Toc166836137)

[5 Тестирование, проверка работоспособности и анализ полученных результатов 18](#_Toc166836138)

[5.1 Тестирование авторизации 18](#_Toc166836139)

[5.2 Тестирование создания профиля студента 19](#_Toc166836140)

[5.3 Тестирование создания аудитории 21](#_Toc166836141)

[5.4 Тестирование создания преподавателя 22](#_Toc166836142)

[5.5 Тестирование добавления предмета в расписание 23](#_Toc166836143)

[5.6 Тестирование изменения информации об аттестации студента 24](#_Toc166836144)

[5.7 Тестирование создания заявки на получение справки 25](#_Toc166836145)

[6 Руководство по установке и использованию 26](#_Toc166836146)

[Заключение 30](#_Toc166836147)

[Список литературы 31](#_Toc166836148)

[ПРИЛОЖЕНИЕ А 32](#_Toc166836149)

[ПРИЛОЖЕНИЕ Б 33](#_Toc166836150)

[ПРИЛОЖЕНИЕ В 34](#_Toc166836151)

[ПРИЛОЖЕНИЕ Г 35](#_Toc166836152)

ВВЕДЕНИЕ

В наше время информационные технологии несут в себе ключ к улучшению и автоматизации множества процессов, в том числе управления образовательными учреждениями. Проект «Виртуальный деканат» на платформе Windows Presentation Foundation (WPF) создан для того, чтобы упростить и ускорить работу деканата, предоставляя удобный интерфейс для студентов и преподавателей, а также обеспечивая эффективное взаимодействие между различными подразделениями университета.

Целью данного курсового проекта является разработка программного решения, которое позволит централизованно управлять учебным процессом, включая расписание занятий, учет аттестации студентов. Использование WPF обусловлено его гибкостью и мощными возможностями по созданию настраиваемых пользовательских интерфейсов, что является важным аспектом для создания интуитивно понятного и функционального приложения.

В рамках проекта будут рассмотрены следующие задачи:

* аналитический обзор литературы;
* анализ требований к программному средству и разработка функциональных требований;
* проектирование программного средства;
* реализация программного средства;
* тестирование, проверка работоспособности и анализ результатов;
* руководство по установке и использованию.

Проект «Виртуальный деканат» нацелен на создание эффективной и безопасной системы, которая будет способствовать повышению качества образовательного процесса и удовлетворению потребностей всех заинтересованных сторон.

1 Постановка задачи и анализ аналогов и прототипов

1.1 Анализ аналогов и прототипов

При разработке программного средства “Виртуальный деканат” важно рассмотреть существующие аналоги и альтернативные решения, которые также предназначены для решения подобных задач. Это поможет выявить преимущества и недостатки конкурирующих продуктов и определить, каким образом наше программное средство может выделиться на фоне аналогов.

Аналог 1: Электронный деканат ФГБОУ ВО СибГУФК [1]

На главной странице пользователя встречает приятная страница с информацией для кого и для чего предназначен данный сайт. Страница представлена на рисунке 1.1.

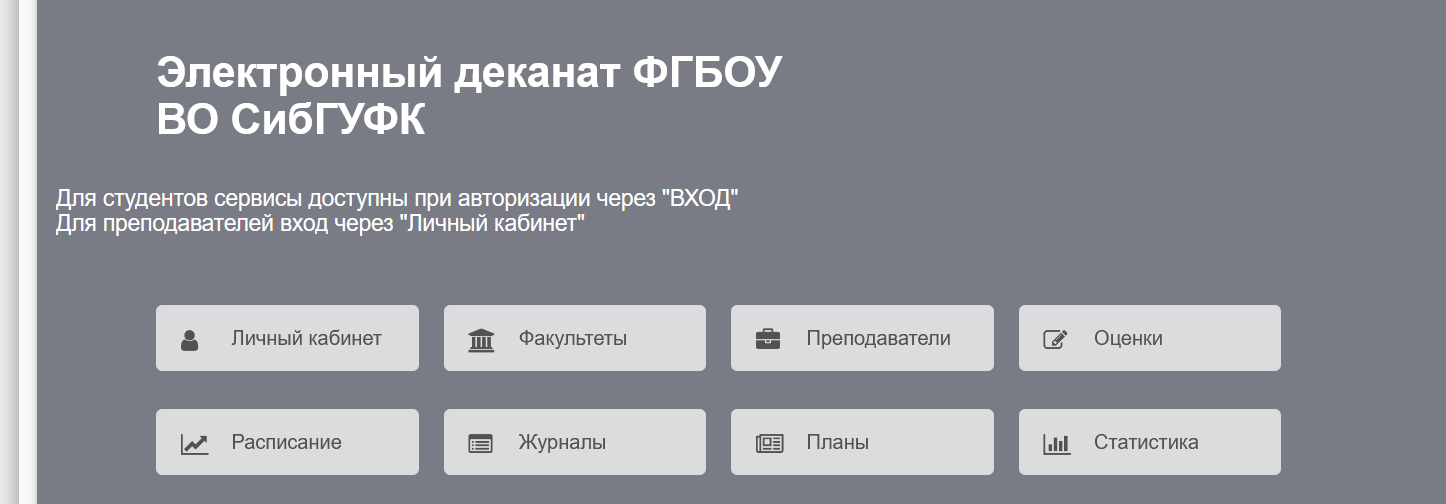


Рисунок 1.1 – Главная страница электронного деканата ФГБОУ ВО СибГУФК

Расписание занятий представлено приятным интерфейсом, с простым поиском и фильтрами. Страница представлена на рисунке 1.2

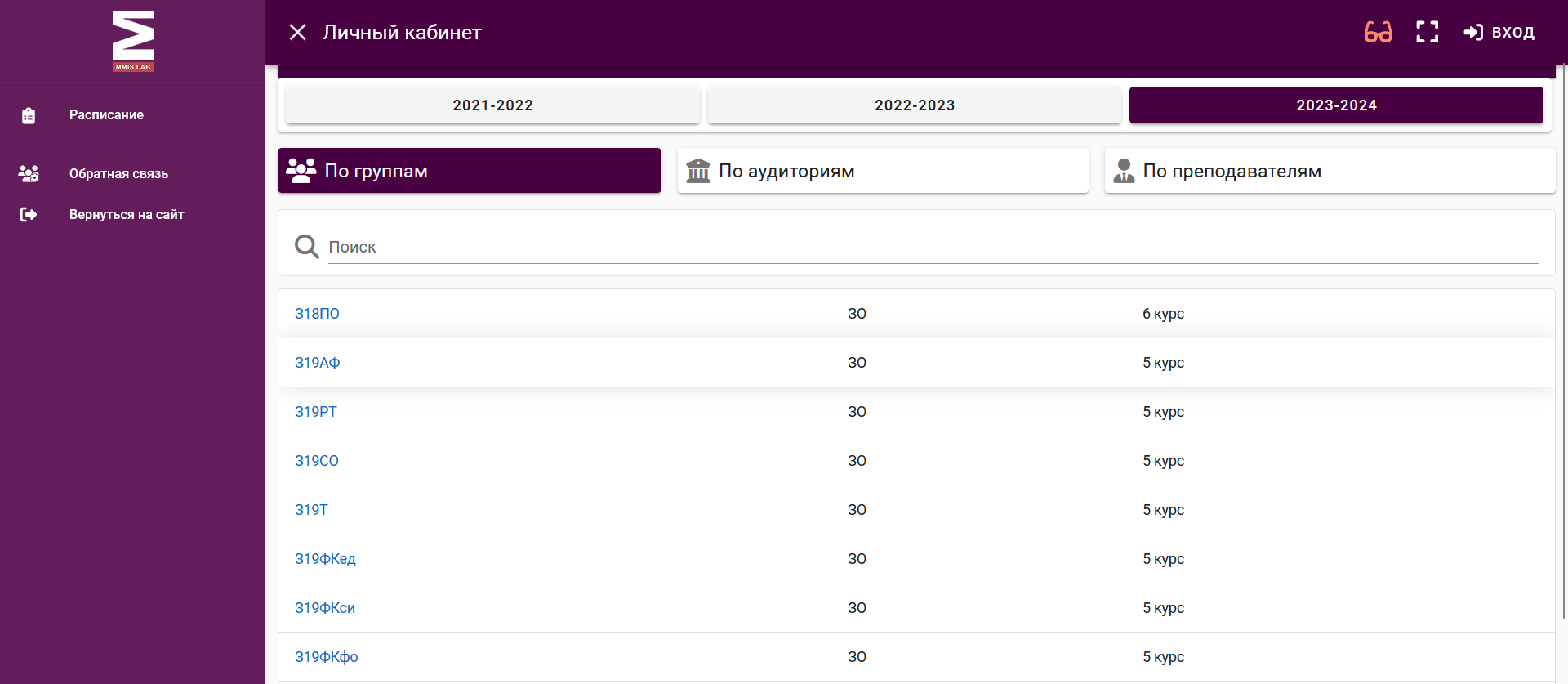


Рисунок 1.2 – Страница с расписанием

При выборе группы отображается недельное расписание занятий с названием дисциплины, преподавателем и номером аудитории. На странице также возможно выбрать номер недели, текущую неделю, год, семестр. Имеется возможность распечатать расписание или сохранить его на устройство. Функционал представлен на рисунке 1.3.

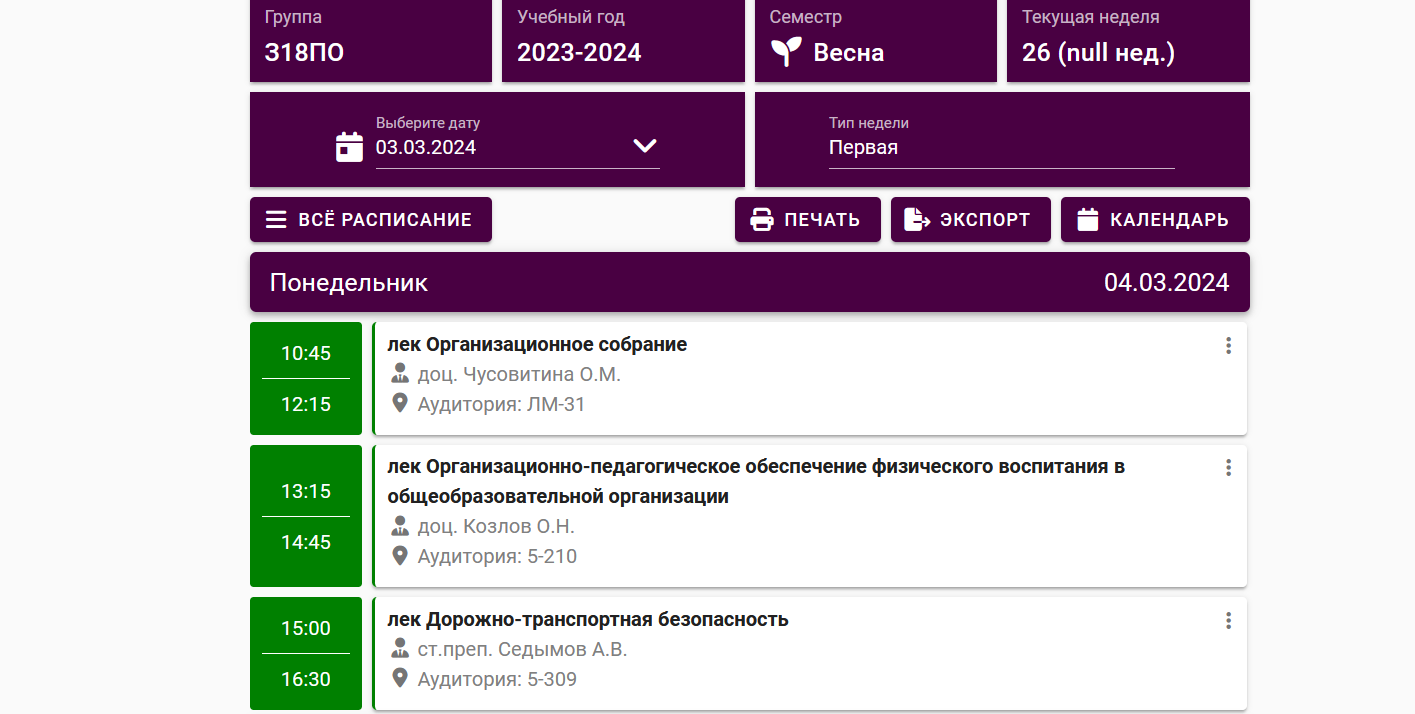


Рисунок 1.3 – Страница с расписанием

Аналог 2: Веб-сайт БГУИР [2]

Веб-сайт выполнен в строгих оттенках, главная страница не перегружена информацией, навигационное меню не включает в себя слишком большое количество ссылок, все уточняющие ссылки являются элементами выпадающего списка, появляющегося при наведении на один из блоков меню. На странице присутствует поиск, что упрощает и ускоряет работу с сайтом. Рисунок 1.4.

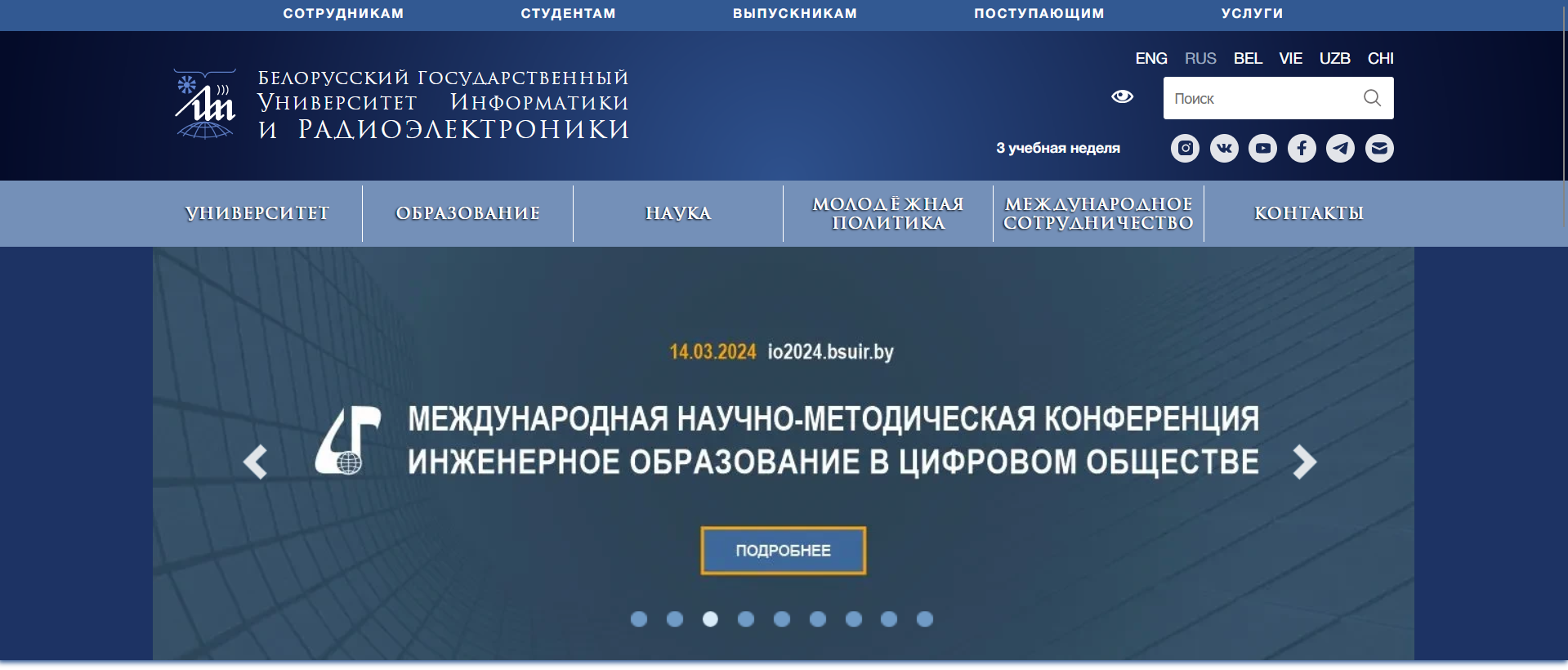


Рисунок 1.4 – Главная страница веб-сайта БГУИР

Вкладка с расписанием ничего не показывает, если не выбрать преподавателя или группу, что может показаться непонятным новым пользователям. На рисунке 1.5 представлен скриншот.

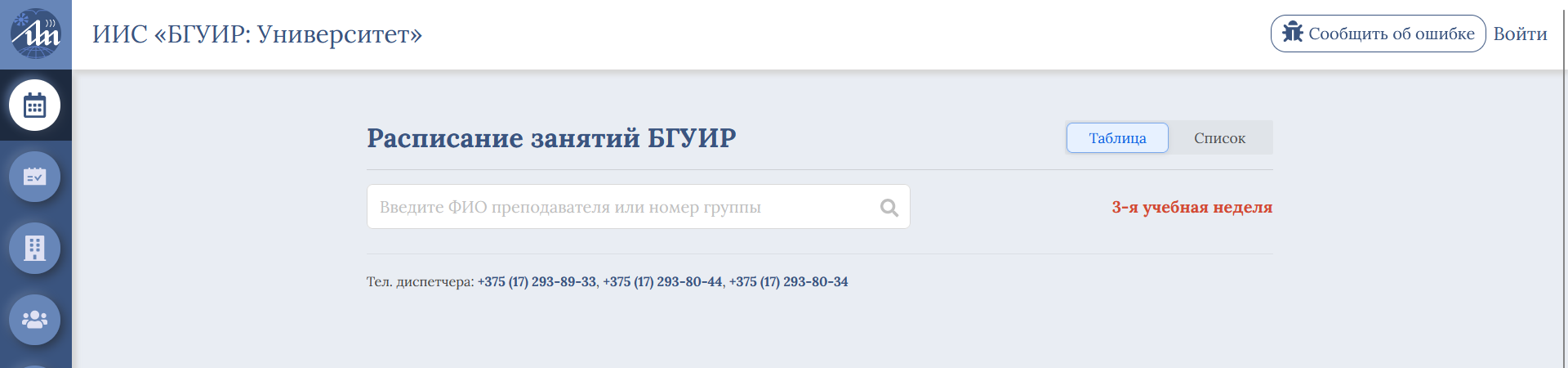


Рисунок 1.5 – Вкладка с расписанием занятий

Однако, выбрав преподавателя, пользователь видит краткую информацию о нем, расписание его занятий, может выставить необходимые параметры поиска. Скриншот представлен на рисунке 1.6

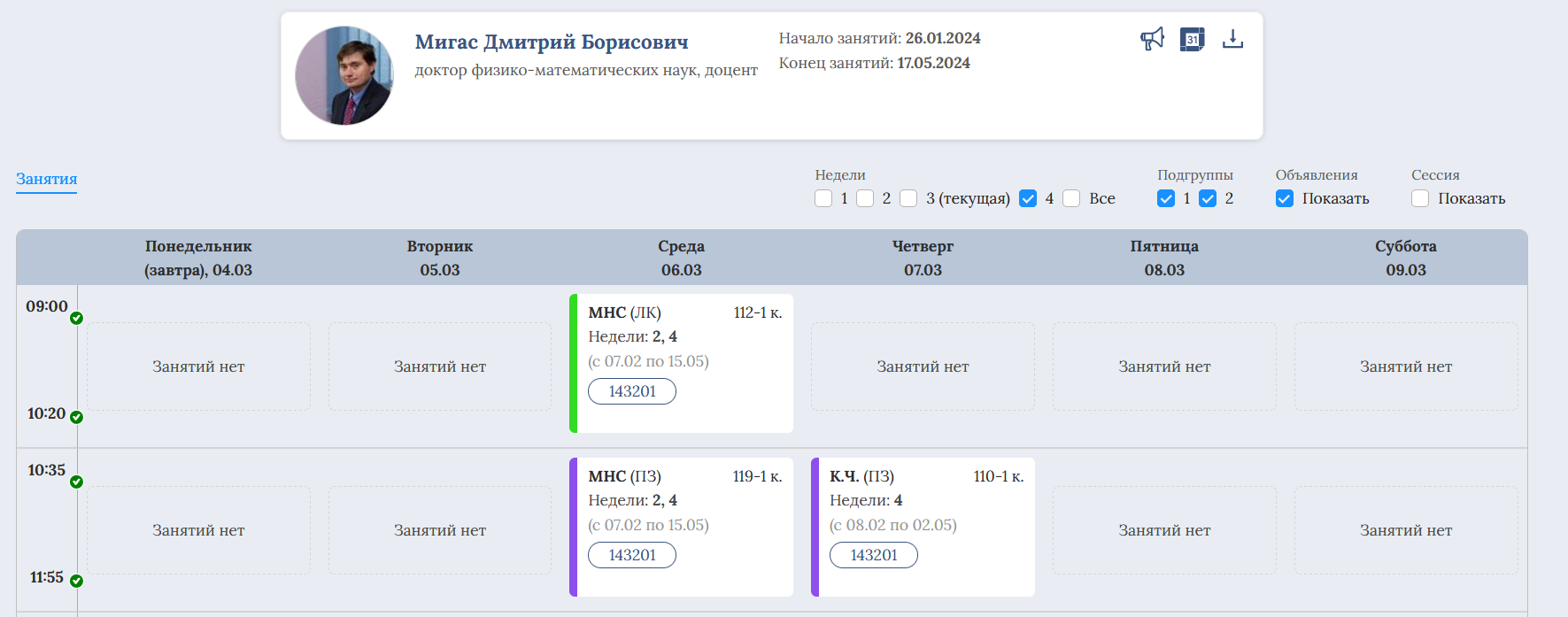


Рисунок 1.6 – Вкладка с расписанием занятий и выбранным преподавателем

Также присутствует возможность поиска студентов из списка по фамилии с использованием фильтров.

Аналог 3: Веб-сайт БГТУ [3]

Веб-сайт выполнен в строгих тёмно-синих цветах, что представляет его со стороны государственного учреждения. Главная страница встречает пользователя большой картинкой и удобным навигационным меню. Скриншот главной страницы сайта представлен на рисунке 1.7.

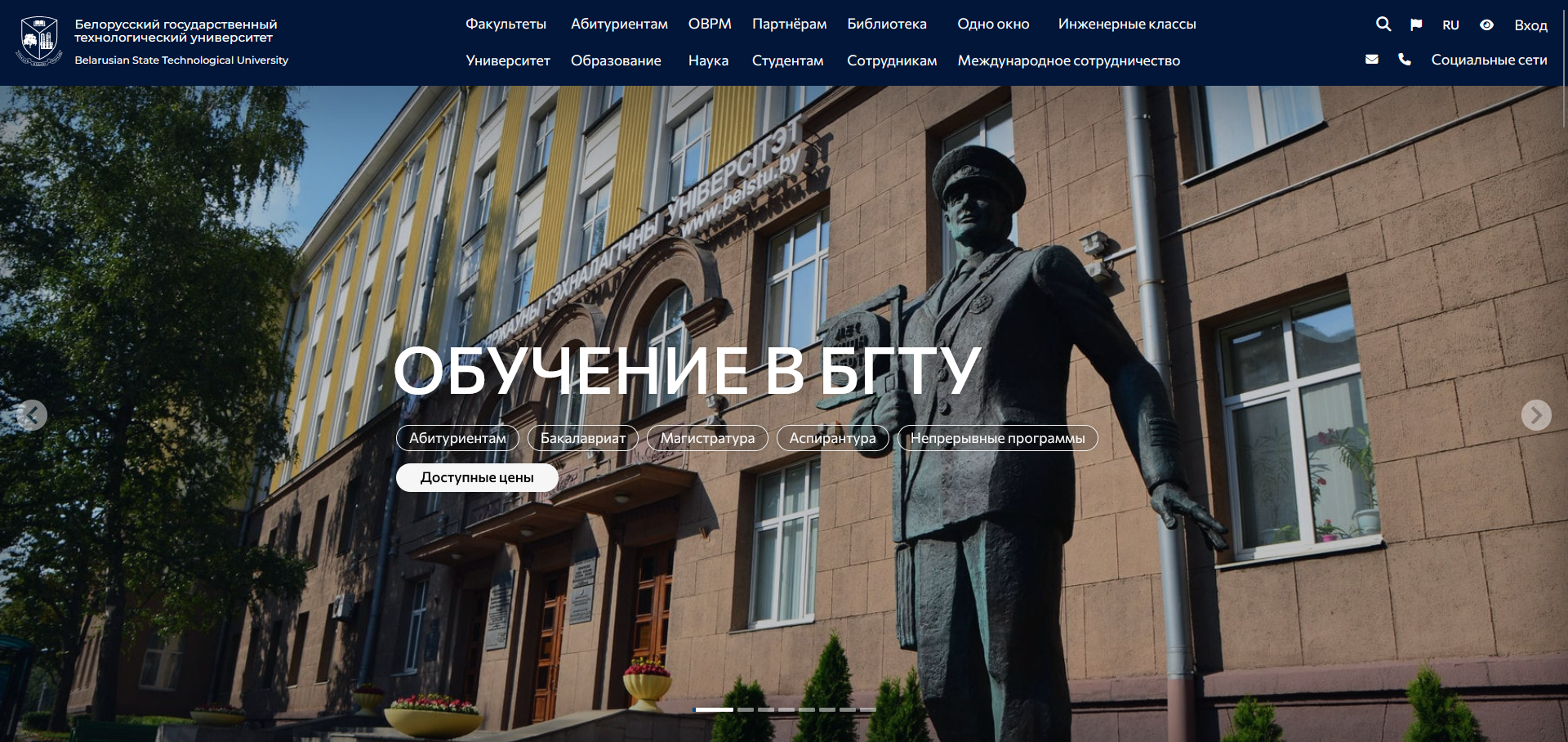


Рисунок 1.7 – Главная страница веб-сайта БГТУ

Веб-сайт предоставляет возможность просмотра расписания студентов различных факультетов, специальностей, курсов и групп. На страницах можно найти репозиторий публикаций, научный журнал, информацию о проведении конференций, защите диссертаций и уникальном оборудовании. Можно просмотреть информацию о стипендиях, практике, студенческой жизни, студгородке, выдающихся студентах.

Целевая аудитория веб-сайта БГТУ не только студенты, но и преподаватели, абитуриенты, магистранты, бакалавриаты, аспиранты.

Аналог 4: Веб-сайт ГРГУ [4]

Данный веб-сайт включает в себя простой доступ к расписанию занятий с выбором необходимого факультета, курса, группы. Скриншот представлен на рисунке 1.8.

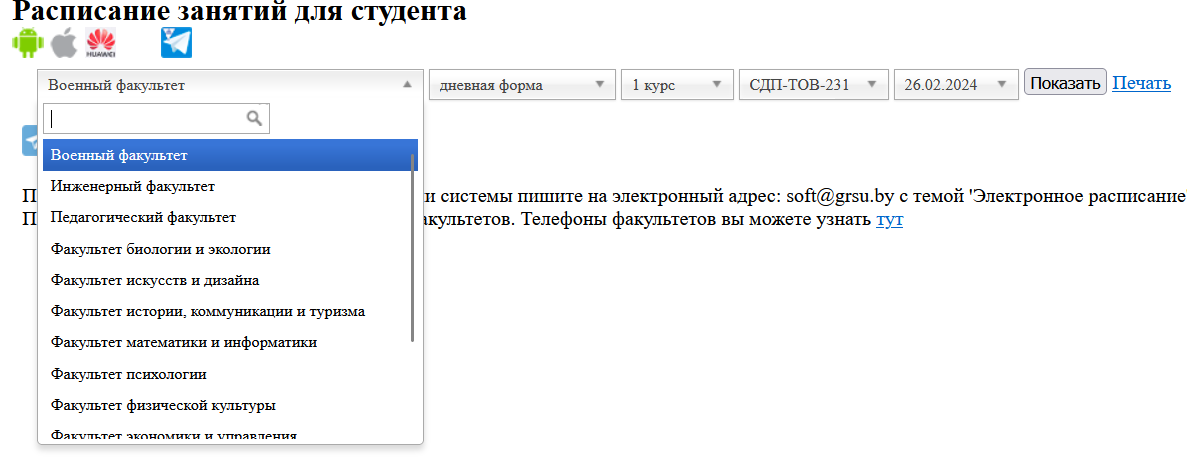


Рисунок 1.8 – Расписание занятий для студента

Также имеется возможность просмотра расписания занятий со стороны преподавателя, выбрав необходимую фамилию и дату, что значительно упрощает доступ преподавателям. На рисунке 1.9 представлен скриншот расписания для преподавателя

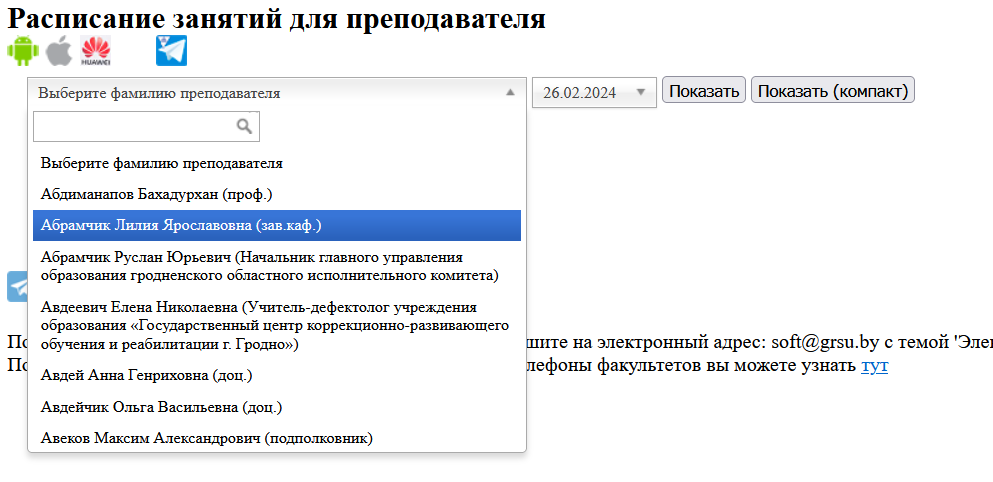


Рисунок 1.9 – Расписание занятий для преподавателя

Навигационное меню сайта включает в себя слишком много пунктов, что было бы корректнее распределить на выпадающие списки. Навигационное меню представлено на рисунке 1.10.

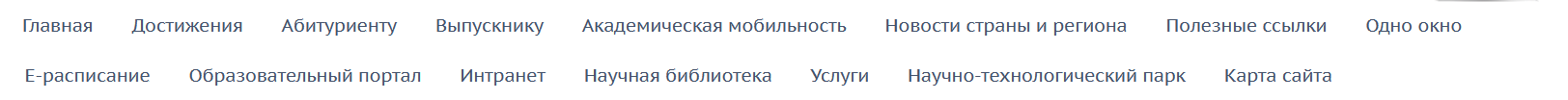


Рисунок 1.10 – Навигационное меню

Аналог 5: Деканат

Необходимо рассмотреть не только интернет-ресурсы, но и реальный деканат факультета. Преимуществом данного выбора является возможность на месте уточнить интересующие вопросы, которые могут быть не рассмотрены в условиях интернет-ресурсов.

Среди недостатков данной альтернативы можно выделить количество затраченного времени и наличие очереди на больших перерывах.

1.2 Постановка задачи

На основе анализа аналогичных решений и с учётом темы проекта был составлен список функций, которые должна выполнять разрабатываемая программа:

Функции деканата:

* + создание, изменение расписания занятий студентов;
  + добавление новостей;
  + добавление и редактирование списка студентов, сортировка, поиск в списке;
  + определение студента-старосты;
  + закрытие запроса на справку об обучении от студента.

Функции студента:

* + просмотр добавленных новостей;
  + просмотр расписания занятий;
  + отправление запроса на справку об обучении.

Функции студента-старосты:

* + заполнение электронного журнала;
  + функции студента.

1.3 Вывод по разделу

После оценки всех веб-сайтов можно заметить, что многие из них имеют схожую функциональность, но аналоги проекта практически отсутствуют, что определенно является плюсом. Проанализировав аналоги веб-сайтов с тематикой виртуального деканата были выделены основные задачи проекта: разработка приложения, которое предоставляет возможность студентам быстро узнать расписание занятий, просмотреть новости факультета, оставить заявку на справку об обучении по месту требования и всё это не выходя из дома, а для деканата – возможность следить за образовательным процессом, назначать старост в группах, менять расписание занятий, обрабатывать заявки на справки добавлять и удалять студентов в группы, создавать новые группы и т.п. Интерфейс должен быть удобным и понятным.

2 Анализ требований к программному средству и разработка функциональных требований

2.1 Описание средств разработки

При разработке приложения были использованы:

* интегрированная среда разработки Microsoft Visual Studio 2022;
* платформа .NET Framework 8.0;
* расширяемый язык разметки XAML;
* технология Entity Framework;
* технология Fluent API;
* технология WPF;
* SQLite;
* язык программирования С#.

2.1.1 Microsoft Visual Studio 2022

Microsoft Visual Studio 2022 представляет собой комплексный инструмент для создания, тестирования и развертывания кода. Эта среда поддерживает разработку множества типов приложений, включая как консольные, так и настольные, с применением технологий WinForms и WPF.

2.1.2 Платформа .NET Framework 8.0

.NET Framework 8.0 — это современная версия платформы от Microsoft, выпущенная для разработки и запуска разнообразных приложений, включая веб-сервисы и настольные программы. Эта платформа предоставляет унифицированную объектно-ориентированную программную среду, которая позволяет безопасно выполнять код, разработанный на множестве поддерживаемых языков программирования. В её основе лежит улучшенная версия Common Language Runtime, обеспечивающая повышенную производительность и оптимизацию для современных вычислительных систем [6].

2.1.3 Расширяемый язык разметки XAML

XAML (eXtensible Application Markup Language) — это декларативный язык разметки, основанный на XML, разработанный Microsoft для программирования приложений. Он позволяет разработчикам описывать пользовательский интерфейс в декларативной форме, отделяя логику приложения от его визуального представления. XAML широко используется в .NET Framework, в частности в Windows Presentation Foundation (WPF), Xamarin.Forms и Windows Workflow Foundation (WWF).

XAML значительно упрощает процесс разработки интерфейса, позволяя разработчикам использовать различные элементы управления, такие как кнопки, текстовые поля и слайдеры, а также поддерживает сложные макеты и стилизацию, что делает его мощным инструментом для создания богатых и интерактивных пользовательских интерфейсов [8].

2.1.4 Технология Entity Framework

Entity Framework (EF) - это современный объектно-реляционный маппер (ORM), который позволяет создавать чистый, переносимый и высокоуровневый слой доступа к данным с использованием .NET (C#) для различных баз данных, включая SQL Database (на локальных серверах и в Azure), SQLite, MySQL, PostgreSQL и Azure Cosmos DB. Он поддерживает запросы LINQ, отслеживание изменений, обновления и миграции схем [9].

2.1.5 Технология Fluent API

При настройке отображения модели базы данных в Entity Framework Core могут использоваться два подхода: аннотации и Fluent API. В данном проекте для реализации поставленных технических задач был выбран последний подход. Технология Entity Framework Core также способна самостоятельно сопоставить классы моделей с таблицами базы данных при использования согласовании об именовании, однако Fluent API является набором методов, предоставляющим их более тонкую настройку императивным методом [9].

2.1.6 Технология WPF

Для предоставления пользовательского интерфейса и разграничения дизайна и бизнес-логики используется технология Microsoft WPF – аналог WinForms, система для построения клиентских приложений Windows с возможностями взаимодействия с пользователем и графическая подсистема в составе .NET Framework, использующая язык разметки XAML[8].

2.1.7 SQLite

SQLite - это библиотека на языке программирования C, которая реализует компактный, быстрый, самодостаточный, высоконадежный и полнофункциональный SQL-движок базы данных. SQLite является самым используемым движком базы данных в мире и встроен во все мобильные телефоны, большинство компьютеров и множество других приложений, которыми люди пользуются каждый день[10].

2.2 Спецификация функциональных требований к программному средству

Для студента:

* + просмотр добавленных новостей;
  + просмотр расписания занятий;
  + отправление запроса на справку об обучении;
  + просмотр информации об аттестации;
  + просмотр личного профиля;
  + авторизация.

Для студента-старосты:

* + заполнение электронного журнала;
  + функции студента.

Для деканата:

* + создание, изменение расписания занятий студентов;
  + добавление новостей;
  + добавление и редактирование списка студентов, сортировка, поиск в списке;
  + определение студента-старосты;
  + закрытие запроса на справку об обучении от студента;
  + авторизация;
  + применение фильтров к списку студентов.

2.3 Спецификация функциональных требований

Для функциональности ПС необходимо создание базы данных для хранения информации приложения. Подробно база данных описано в следующем разделе.

В программном средстве необходимо реализовать авторизацию студентов для доступа ко всем его возможностям. Для авторизации входными параметрами являются логин и пароль, которые хранятся в базе данных. Для создания аккаунтов студентов необходимо, чтобы пользователь деканата создал профиль студента, указав его имя, фамилию, отчество, логин и пароль.

Студент может просматривать расписание занятий, аттестацию своей группы, новости, добавленные деканатом, создавать заявку на получение справки об обучении по месту требования и просматривать статус их обработки.

Студент-староста обладает всеми возможностями студента, а также имеет возможность выставления оценок и пропусков студентам своей группы в аттестации.

Деканат может создать профили студентов, просматривать список студентов, информацию о студенте, редактировать информация о студенте, создать группу, расписание занятий, удалить предмет из расписания, добавить новый предмет в расписание, выставить оценки и пропуски студента любой группы в аттестации.

3 Проектирование программного средства

3.1 Общая структура

Программное средство «Виртуальный деканат» имеет структуру, представленную на рисунке 3.1.

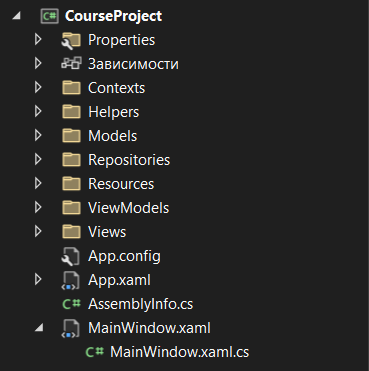


Рисунок 3.1 – Структура проекта

Папка «Contexts» содержит в себе контекст базы данных.

Папка «Helpers» содержит в себе классы-помощники, такие как конвертеры, определение класса Command, перечисления.

Папка «Models» содержит модели для реализации паттерна MVVM.

Папка «Repositories» содержит репозитории для реализации паттерна UnitOfWork.

Папка «Resources» содержит в себе ресурсы для работы с XAML, например: изображения, стили.

Папка «ViewModels» содержит в себе классы, организующие связь между моделями и представлениями в паттерне MVVM.

Папка «Views» содержит в себе представления, которые отображаются в приложении.

3.2 Проектирование схемы базы данных

Для хранения и управления данными приложения была разработана схема базы данных. Данная схема определяет структуру данных, основные таблицы и связи между ними. Логическая схема базы данных представлена в приложении А.

Таблица «Users» хранит информацию о зарегистрированных пользователях.

Таблица «Students» хранит информацию о студентах.

Таблица «Requests» хранит информацию о запросах студента.

Таблица «Auditoriums» хранит информацию об аудиториях.

Таблица Subjects хранит информацию о предметах, добавленных в расписание.

Таблица «Timetables» хранит расписания, созданные для определенных групп.

Таблица «Groups» хранит информацию о группах определенного курса.

Таблица «Specialities» хранит информацию о специальностях.

Таблица «Teachers» хранит информацию о преподавателях определенного курса.

Таблица «News» хранит информацию о добавленных новостях.

Таблица «Progresses» хранит информацию о аттестации студентов.

Таблица «Courses» хранит информацию о курсах определенной специальности.

3.3 Взаимоотношения между классами

Диаграмма классов служит для визуализации архитектуры приложения, отображая классы и их взаимосвязи. Каждый класс в этой структуре обладает определенной функциональностью и минимально взаимодействует с остальными. Схематическое представление этой структуры представлено в приложении Б. Переформулировано, это описание подчеркивает самостоятельность классов и их специализированные роли в программе.

3.4 Проектирование архитектуры проекта

Диаграмма использования — это инструмент, который демонстрирует различные функции программной системы и доступность этих функций для разных категорий пользователей. Она включает в себя элементы, называемые актёрами и вариантами использования. Актёры представляют группы пользователей или другие системы, которые взаимодействуют с разрабатываемой системой. Варианты использования описывают конкретные функции, доступные актёрам, и последовательность действий, которые система выполняет в ответ на запросы актёров. Эта диаграмма помогает понять, какие возможности системы доступны для каждой группы пользователей и как они могут их использовать. Схематическое изображение диаграммы использования представлено в приложении В.

3.5 Проектирование последовательностей проекта

Диаграмма последовательностей UML демонстрирует взаимодействие компонентов системы в рамках определённого сценария. Она визуализирует последовательность обмена сообщениями между объектами для достижения функциональной цели. Временная ось представлена вертикальными пунктирными линиями, называемыми линиями жизни, которые указывают на продолжительность существования объекта в системе. Объекты отображаются прямоугольниками, а их взаимодействие – через стрелки, соединяющие линии жизни, что отражает поток сообщений. Эта диаграмма представлена в приложении Г.

4 Реализация программного средства

4.1 Основные классы программного средства

Для выполнения технической задачи программного средства «Виртуальный деканат» необходимо реализовать следующие функции:

* выполнение входа в аккаунт;
* взаимодействие со списком студентов;
* взаимодействие с расписанием занятий;
* взаимодействие с профилем студента;
* взаимодействие с лентой новостей;
* взаимодействие с аттестацией студентов.

Далее подробно рассмотрены каждые из необходимых функций.

4.2 Описание классов и методов программного средства

4.2.1 Выполнение входа в аккаунт

Функционал для возможности входа в аккаунт, а также проверки корректности ввода данных, реализован в классе AuthorizationVM.

Основной командой класса является команда checkAuthorization, которая вызывается нажатием на кнопку «Войти» в окне авторизации. Данная команда вызывает методы checkLogin() и checkPassword(), которые проверяют корректность ввода данных и сравнивают значения с базой данных. Если данные совпали с данными из базы данных, то вышеупомянутая команда переводит пользователя на основное окно, закрывая при этом окно авторизации.

4.2.2 Взаимодействие со списком студентов

Функционал для взаимодействия пользователя со списком студентов реализован в классе StudentsVM.

Данный класс содержит в себе команды для фильтрации списка студентов по выбранным специальности, курсу, группе, для поиска студентов по фамилии, для добавления и удаления студентов в список группы.

Функционал взаимодействия со списком студентов доступен только пользователю с ролью dean.

4.2.3 Взаимодействие с расписанием занятий

Функционал для взаимодействия пользователя с расписанием занятий представлен в следующих классах: TimetableVM, AddingSubjectVM, CreateAuditVM и CreateTeacherVM.

Класс TimetableVM содержит в себе команды для загрузки расписания определенной группы, а также команды для заполнения и изменения расписания занятий, которые доступны только пользователям с ролью dean.

Класс AddingSubjectVM содержит в себе свойства и команды, которые необходимы для создания нового предмета в расписании занятий, данный класс доступен только пользователю с ролью dean.

Классы CreateAuditVM и CreateTeacherVM содержат в себе свойства и команды для создания новых аудитории и преподавателя, классы доступны только пользователю с ролью dean.

4.2.4 Взаимодействие с профилем студента

Взаимодействие с профилем студента представлено в классах ProfileVM и CreateStudentVM.

Класс ProfileVM содержит в себе свойства для отображения информации о студенте пользователю с ролью student.

Класс CreateStudentVM содержит в себе свойства для отображения информации о студенте пользователю с ролью dean, а также команды для изменения и сохранения информации. Установленные значения проверяются с помощью сеттера свойств, хранящих соответствующие значения.

4.2.5 Взаимодействие с лентой новостей

Функционал для взаимодействия пользователя с лентой новостей представлен

в классах NewsVM и CreateNewsVM.

Класс NewsVM содержит в себе свойства для отображения информации о новостях, а также изменяемые свойства, которые доступны только пользователю с ролью dean.

Класс CreateNewsVM содержит в себе команды и свойства для добавления новой новости, которые доступны только пользователю с ролью dean.

4.2.6 Взаимодействие с аттестацией студентов

Взаимодействие с аттестацией студентов представлено классом PerfomanceVM.

Класс PerfomanceVM содержит в себе свойства и команды для загрузки аттестации определенной группы и студента, а также редактирования оценок и количества пропусков, доступных только пользователям с ролью dean или студенту-старосте.

4.3 Вывод по разделу

В данной главе были рассмотрены основные классы для взаимодействия пользователя с различными функциями приложения. Функционал приложения основан на MVVM паттерне. Суть паттерна в том, что имеется класс-модель, которая предоставляет данные для взаимодействия, данные представляются на странице с помощью views, а для корректного представления информации осуществляется обработка с помощью viewmodels.

5 Тестирование, проверка работоспособности и анализ полученных результатов

5.1 Тестирование авторизации

При авторизации возможны ситуации, когда пользователь вводит данные, которых не существует в базе данных, например: несуществующий логин или неверный пароль. Такие ситуации обрабатываются программным средством посредством отображения текста в форме авторизации с указанием ошибки. Пример обработки исключительных ситуаций представлен на рисунках 5.1-5.2.

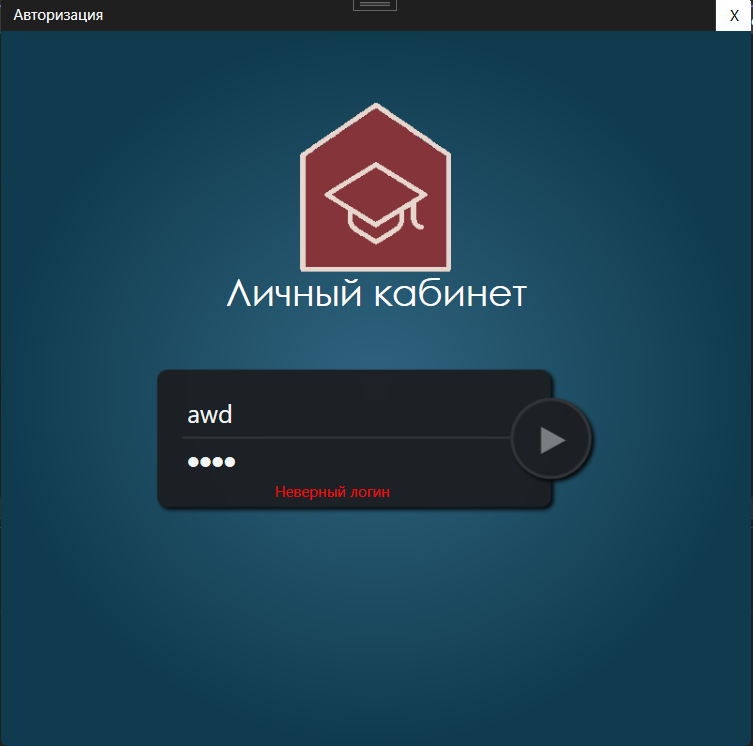


Рисунок 5.1 – Обработка несуществующего логина

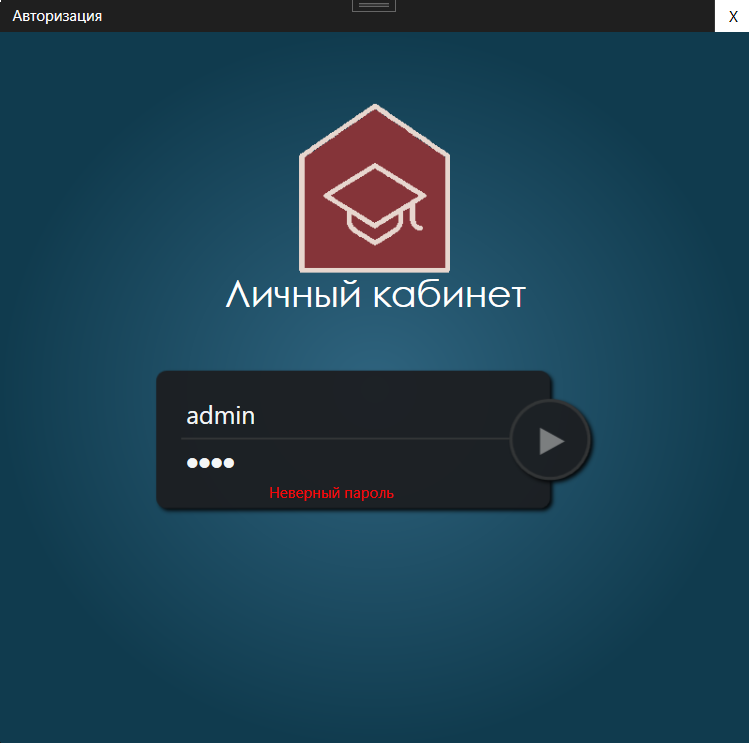


Рисунок 5.2 – Обработка неверного пароля

5.2 Тестирование создания профиля студента

При создании профиля студента могут возникнуть ошибки, связанные с форматом ввода данных и установки значений логина и пароля. Все поля при создании профиля должны быть заполнены, логин должен состоять из символов английского алфавита любого регистра и цифр, длиной от 5 символов, а пароль должен состоять из минимум 8-ми символов и содержать в себе хотя бы по одному символу английского алфавита обоих регистров и цифру. В случае невыполнения условий пользователь будет уведомлён об ошибке посредством всплывающего окна с текстом ошибки. Пример обработки ошибок, связанных с форматом данных, представлен на рисунках 5.3-5.7.

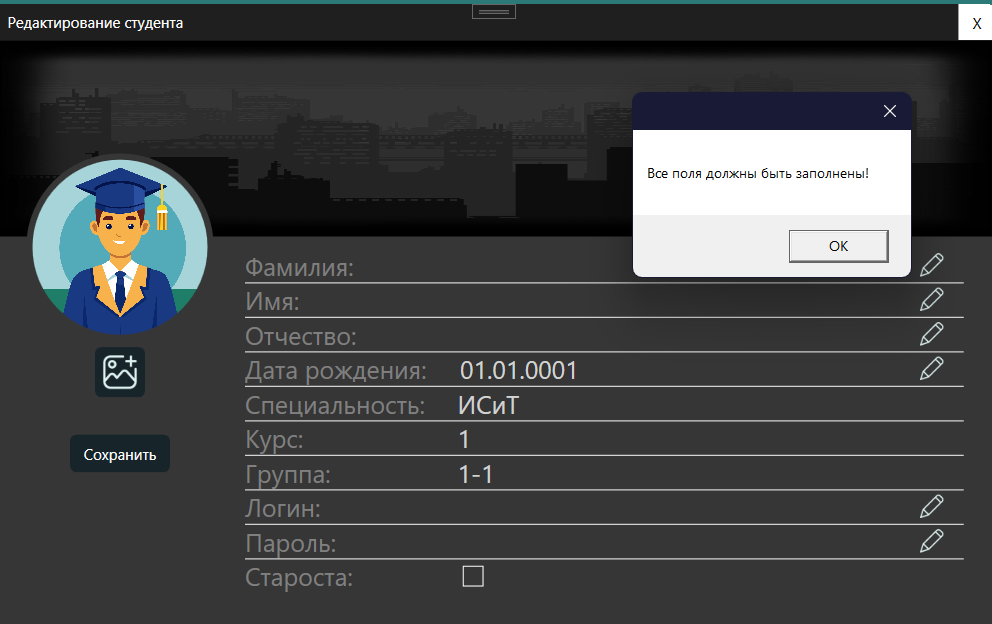


Рисунок 5.3 – Обрабокта создания студента с незаполненными полями

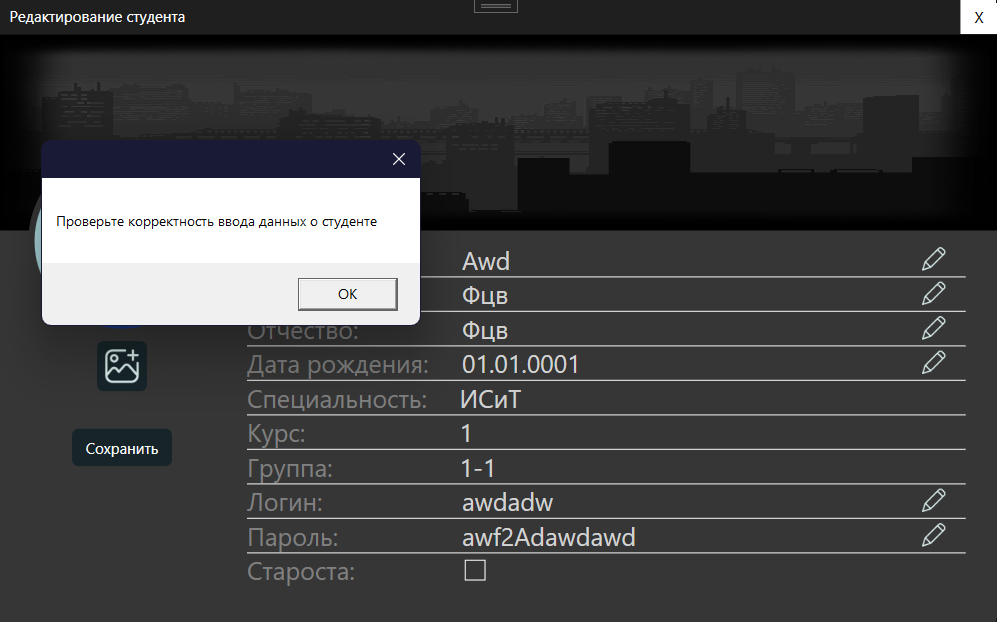


Рисунок 5.4 – Обработка создания студента с некорректным значением

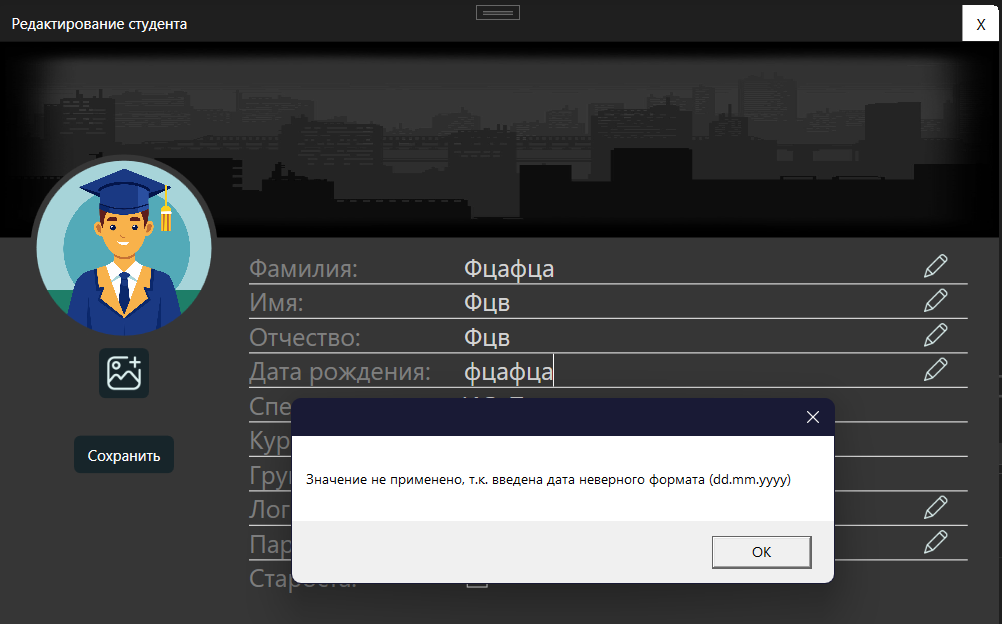


Рисунок 5.5 – Обработка создания студента с некорректным заполнением поля «Дата рождения»

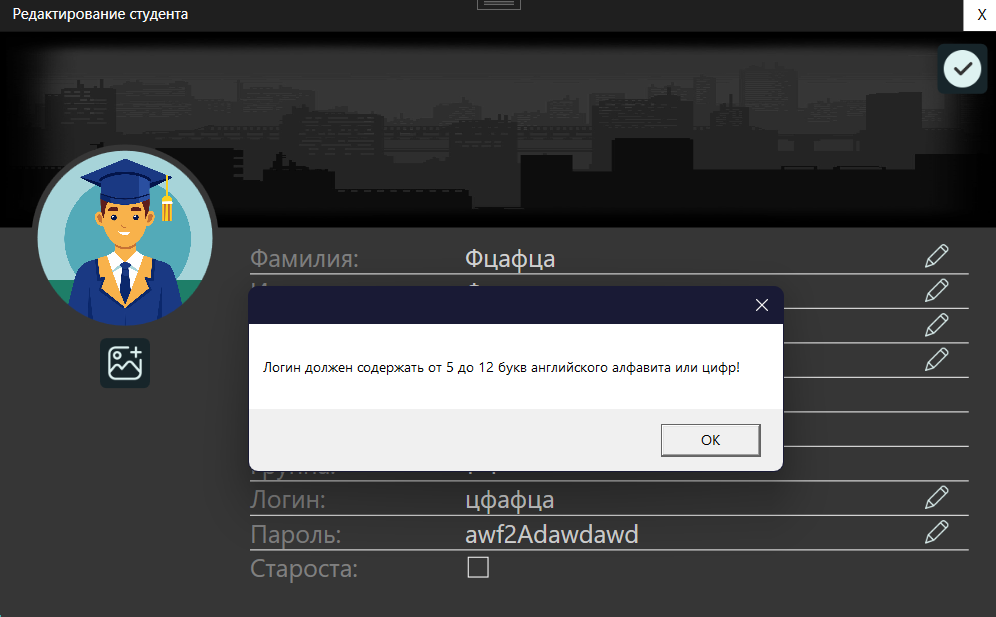


Рисунок 5.6 – Обработка создания студента с некорректным заполнением поля «Логин»

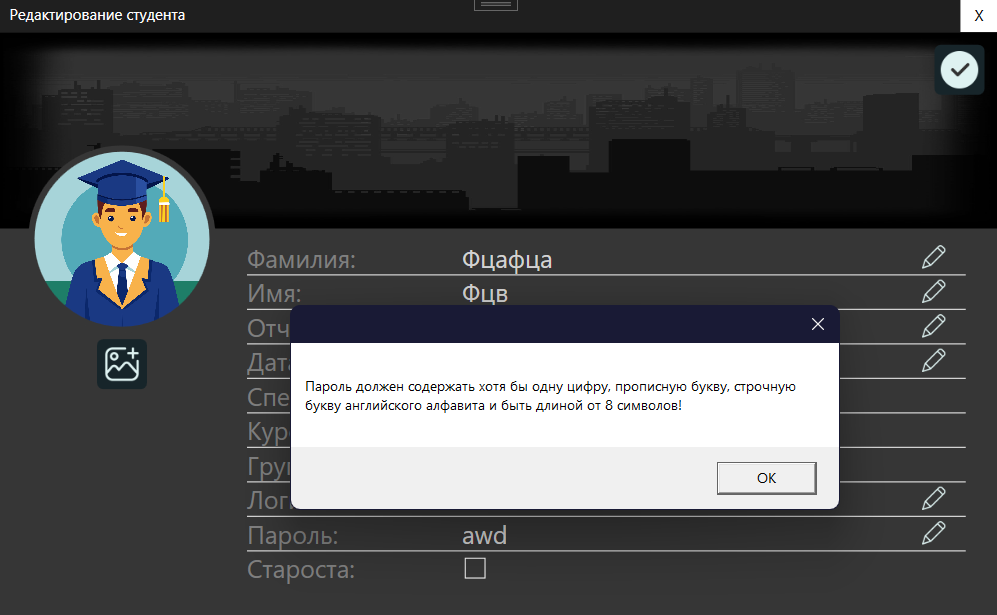


Рисунок 5.7 – Обработка создания студента с некорректным заполнением поля «Пароль»

5.3 Тестирование создания аудитории

При создании новой аудитории могут возникнуть ошибки, связанные с форматом ввода номера аудитории. Номер аудитории и тип аудитории должны быть указаны, формат ввода номера: 000-0. При несоблюдении условий пользователь будет уведомлён об ошибке посредством появления всплывающего окна с текстом ошибки. Примеры обработки ошибок, связанных с добавлением аудитории представлены на рисунках 5.8-5.10.

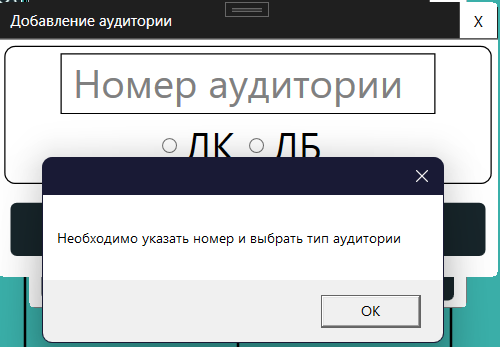


Рисунок 5.8 – Обработка создания аудитории без ввода значений

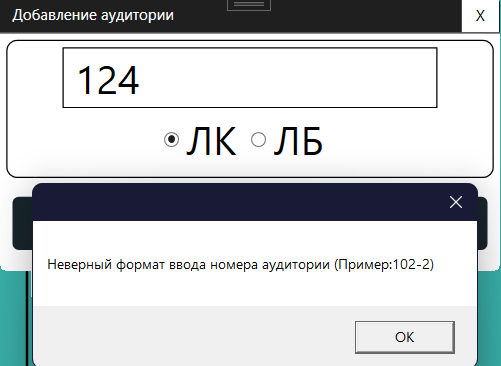


Рисунок 5.9 – Обработка создания аудитории с вводом некорректного значения в поле «Номер аудитории»

5.4 Тестирование создания преподавателя

При создании преподавателя пользователь может столкнуться с ошибкой, связанной с форматом ввода данных о преподавателе. Все поля формы должны быть заполнены и состоять из символов русского алфавита, если условия не соблюдены, то пользователь будет уведомлён об ошибке, связанной с форматом ввода данных. Обработка исключений создания преподавателя представлена на рисунках 5.10-5.11.

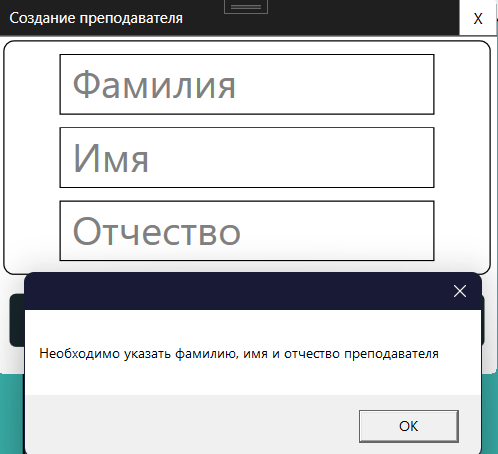


Рисунок 5.10 – Обработка создания преподавателя без заполнения полей

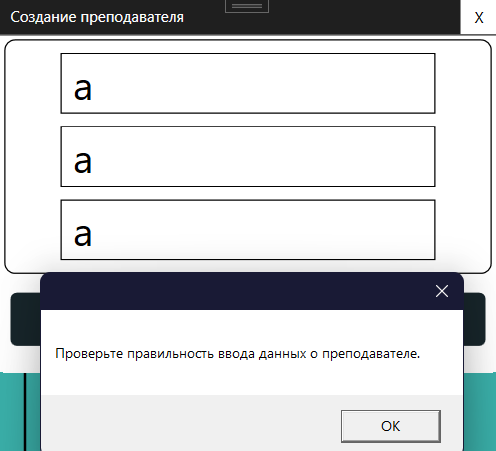


Рисунок 5.11 – Обработка создания преподавателя с некорректным заполнением полей формы

5.5 Тестирование добавления предмета в расписание

При добавлении предмета в расписание пользователь должен выбрать не занятое в расписании время начала занятия, учитывать, что название предмета должно включать только символы русского алфавита, и должны быть заполнены все поля формы. Обработка исключений представлена на рисунках 5.12-5.14.

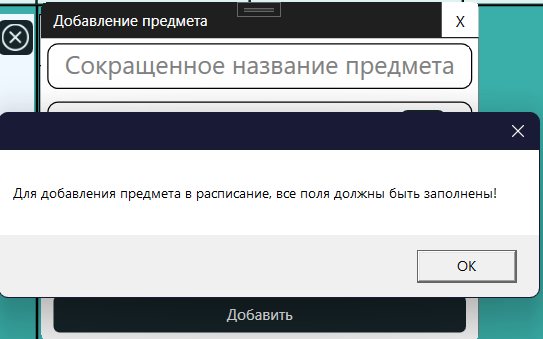


Рисунок 5.12 – Обработка исключения, связанного с не заполнением полей формы



Рисунок 5.13 – Обработка исключения, связанного с выбором занятого времени

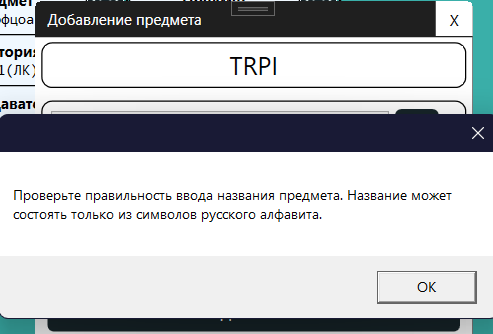


Рисунок 5.14 – Обработка исключения, связанного с заполнением названия предмета некорректным значением

5.6 Тестирование изменения информации об аттестации студента

При выставлении оценок и пропусков необходимо учитывать, что оценка студента – число, которое не может быть больше 10 и меньше 0, при вводе некорректного значения пользователь будет уведомлён об ошибке посредством появления всплывающего окна с указанием текста ошибки. При вводе в поля значений, не являющихся цифрами, значения вводиться не будут. Обработка ошибки связанной с выставлением оценки выше 10 представлена на рисунке 5.15.

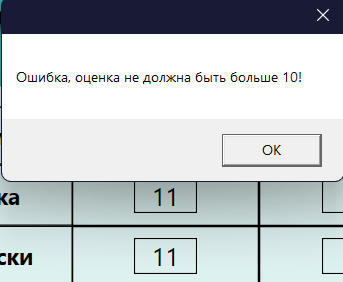
14

Рисунок 5.15 – Обработка ошибки, связанной с вводом оценки значение которой больше 10

5.7 Тестирование создания заявки на получение справки

При создании заявки на получение справки пользователь должен учитывать те заявки, которые у него уже имеются. В случае создания заявки с тем же местом требования, пользователь будет уведомлён об ошибке посредством появления всплывающего окна с текстом ошибки. Обработка исключения представлена на рисунке 5.16.

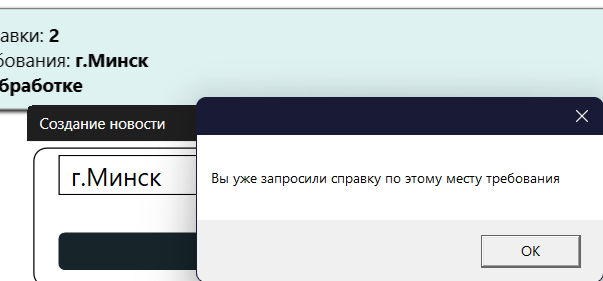


Рисунок 5.16 – Обработка исключения, связанного с созданием заявки по тому же месту требования

6 Руководство по установке и использованию

6.1 Руководство по установке и использованию

При запуске инсталлятора с названием AppInstaller, на устройство пользователя будет установлено программное средство виртуальный деканат.

При запуске программного средства пользователя встретит окно авторизации, в котором пользователю необходимо ввести логин и пароль. Окно авторизации представлено на рисунке 6.1.

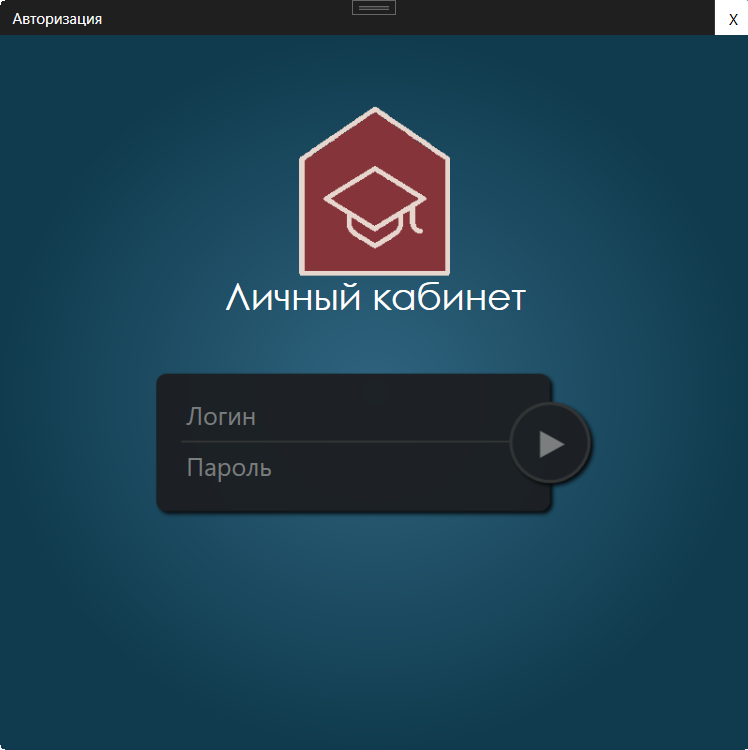


Рисунок 6.1 – Окно авторизации

После успешной авторизации пользователь будет перенаправлен на страницу с лентой новостей и навигационным меню, с помощью которого пользователю доступен переход по страницам. Страница с лентой новостей представлена на рисунке 6.2.

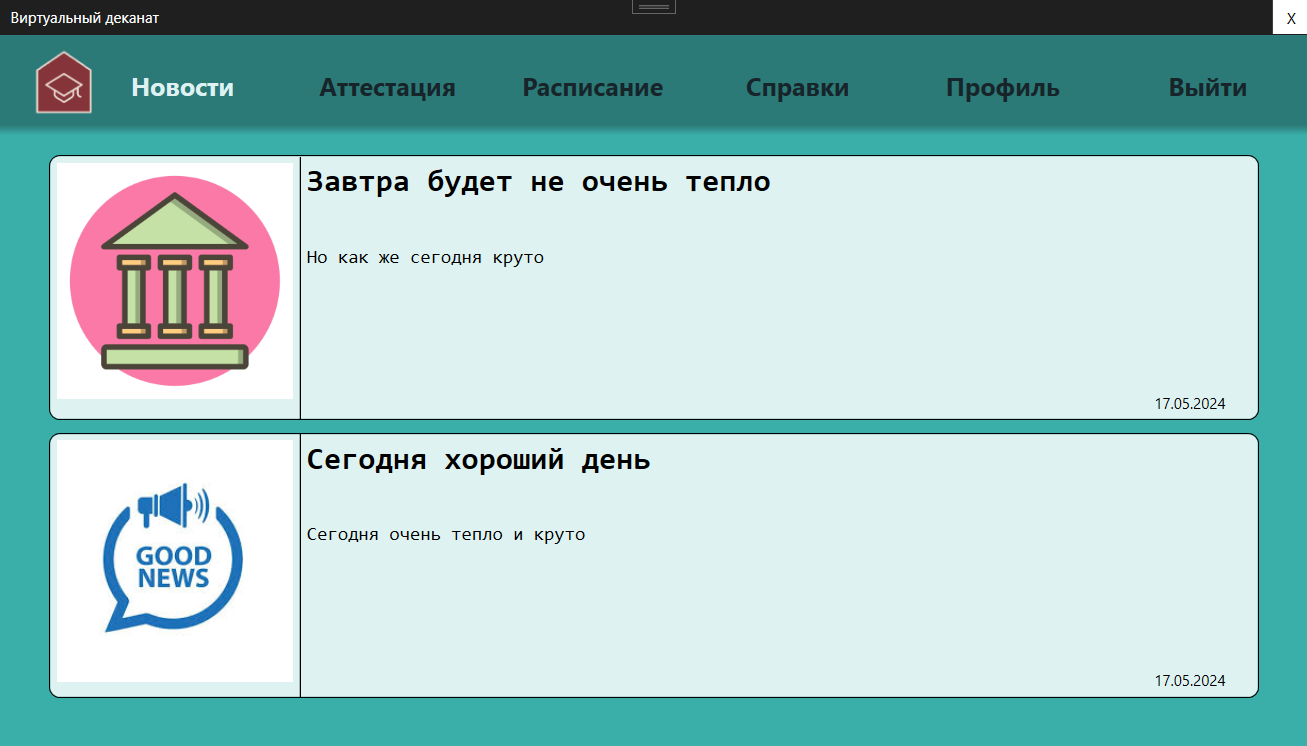


Рисунок 6.2 – Окно «Новости»

При выборе страницы «Аттестация» в навигационном меню пользователь будет перенаправлен на страницу с аттестацией студентов, если пользователь является студентом, на странице будет отображена только его группа, в противном случае, на странице будет возможность выбора специальности, курса и группы. Страница «Аттестация» представлена на рисунках 6.3.



Рисунок 6.3 – Окно «Аттестации»

При выборе страницы «Расписание» пользователь будет перенаправлен на страницу с расписанием студентов, для отображения расписания пользователю необходимо выбрать специальность, курс и группу. Страница с расписанием занятий представлена на рисунке 6.4.



Рисунок 6.4 – Окно «Расписание»

При выборе страницы «Справки» пользователь будет перенаправлен на страницу со своими справками. Страница со справками представлена на рисунке 6.5.

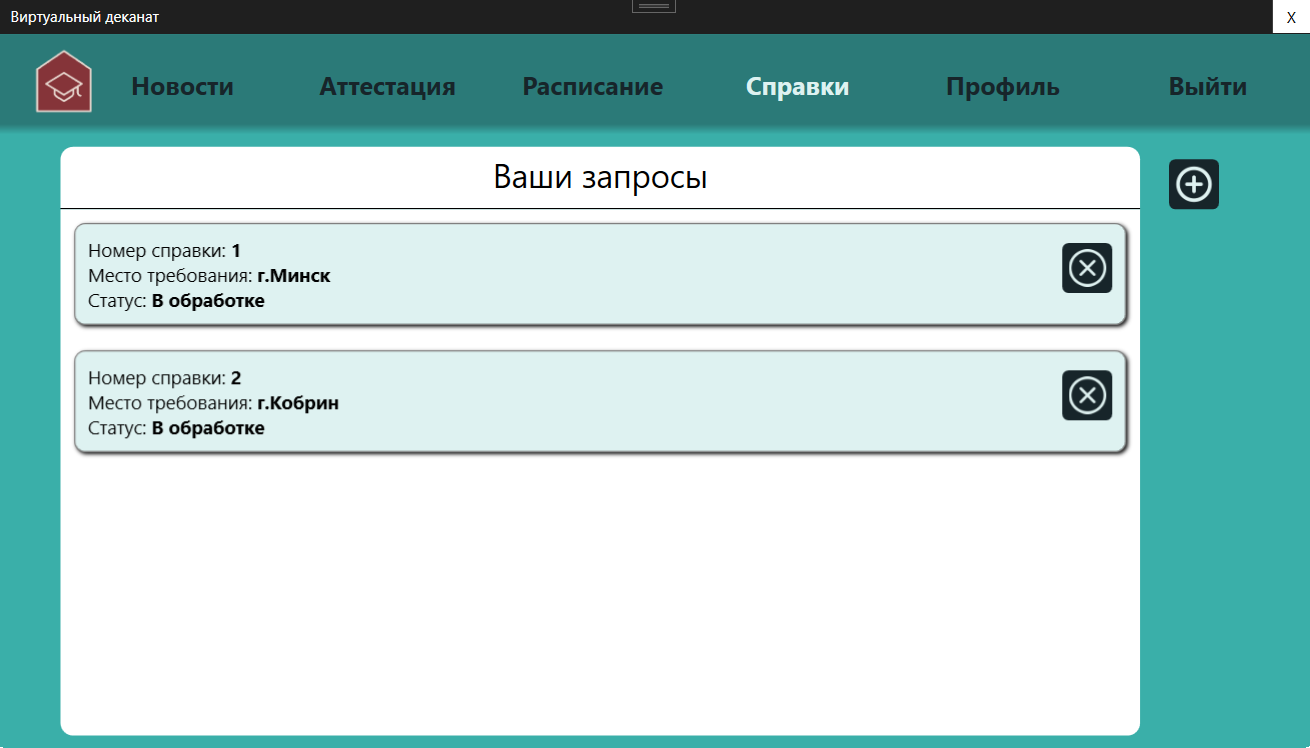


Рисунок 6.5 – Окно «Справки»

При выборе страницы «Профиль» пользователь будет перенаправлен на страницу со своим профилем. Страница с профилем представлена на рисунке 6.6.

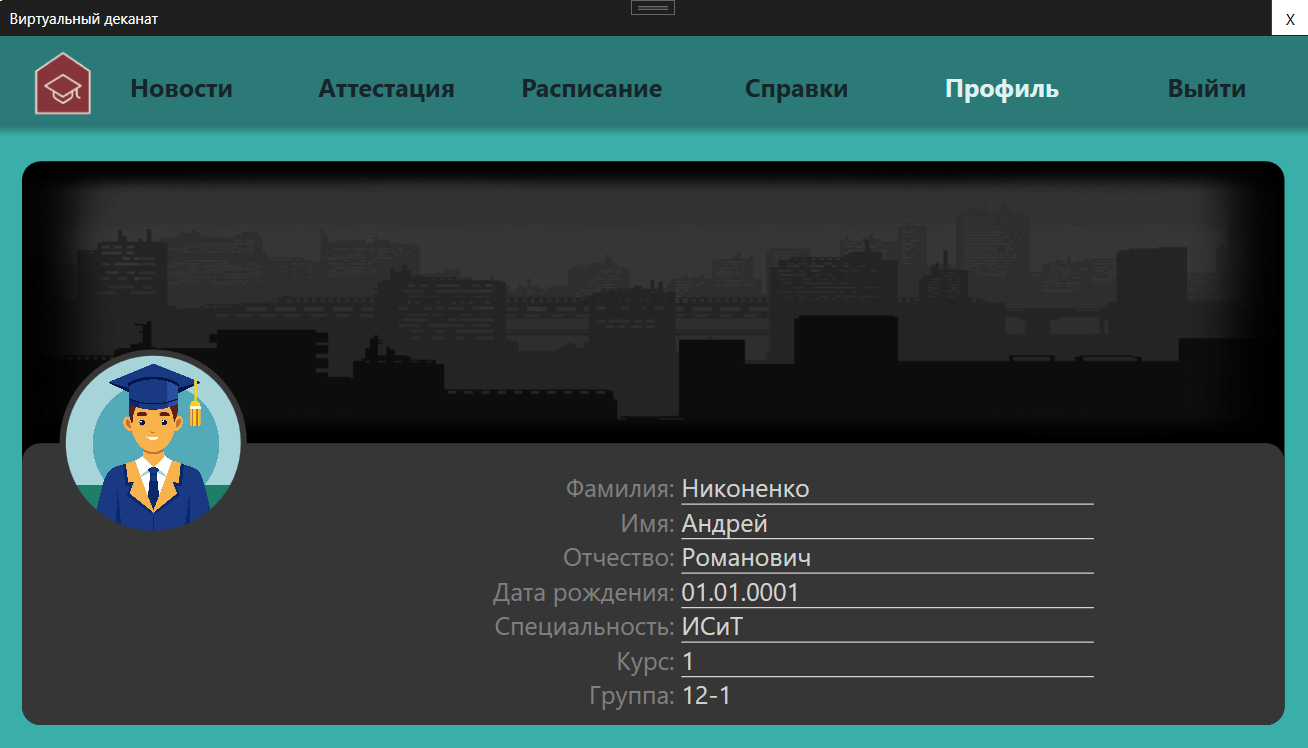


Рисунок 6.6 – Окно «Профиль»

6.2 Вывод по разделу

В данном разделе было разобрано руководство по установке приложения, а также основной функционал, доступный пользователю.

С помощью данного функционала пользователь получает возможность ускоренного доступа к информации, которая ему необходима. А для пользователя с ролью dean предоставляется возможность просмотра информации об студентах, обучающихся в университете.

Заключение

В ходе выполнения курсового проекта было разработано программное средство «Виртуальный деканат» на языке программирования C# с использованием технологий Entity Framework Core, WPF.

При разработке программного средства были выполнены все пункты из указанного списка предполагаемого основного функционала приложения, а именно:

* создана база данных;
* реализована возможность войти в существующую учетную запись;
* реализована возможность авторизированным пользователям просмотра информации о созданных заявках на получение справки об обучении, создание заявки;
* реализована возможность просмотра личной страницы пользователя с возможность её редактирования со стороны деканата;
* реализована возможность просмотра расписания и изменения расписания со стороны деканата;
* реализована возможность просмотра аттестации и изменения информации об аттестации со стороны деканата и студента-старосты.

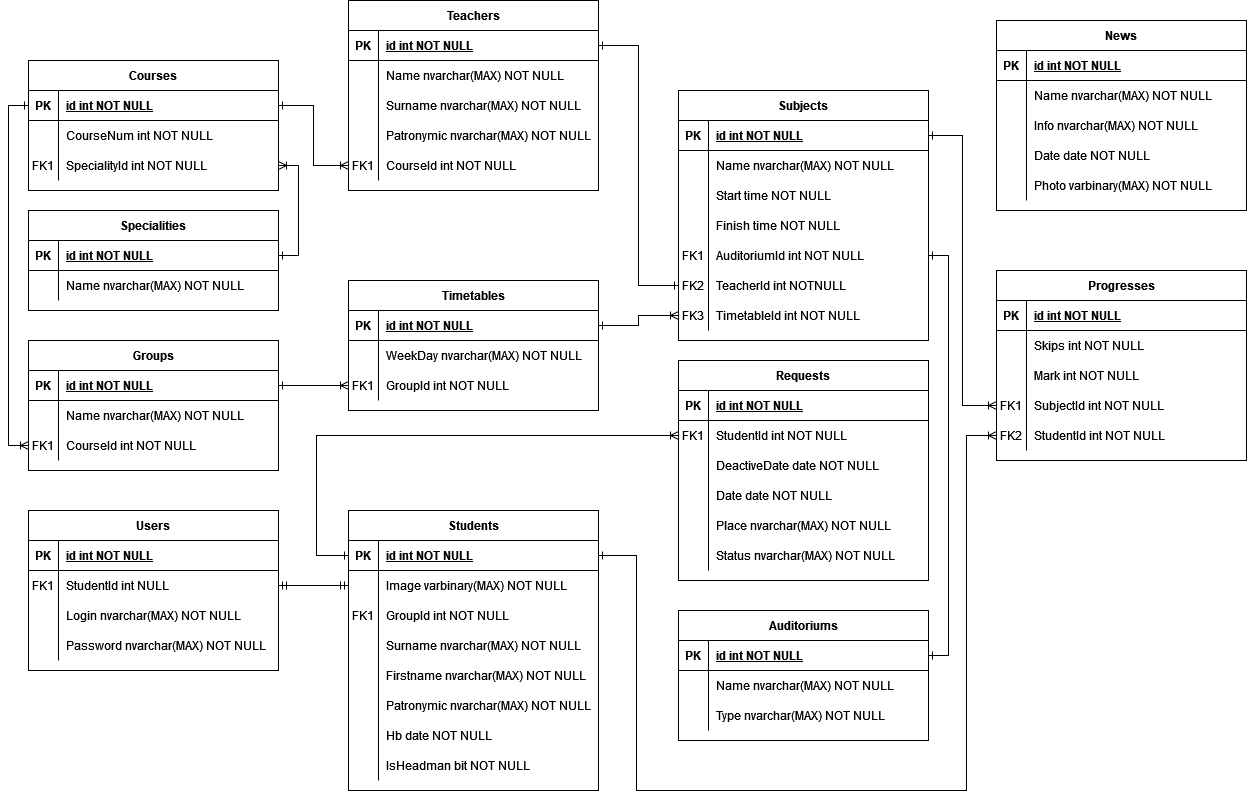
Тестирование и результаты: Проведенное тестирование подтвердило надежность и корректность работы программы, что доказывает ее соответствие поставленным требованиям и готовность к внедрению в образовательный процесс.

Разработанный “Виртуальный деканат” является значительным шагом в цифровизации учебного заведения и создает основу для дальнейшего развития и оптимизации учебного процесса.

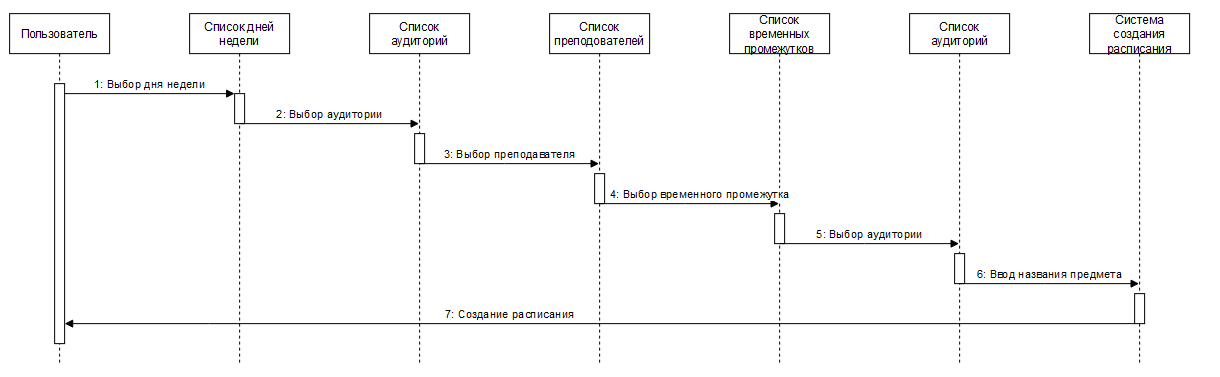
Список литературы

1. <https://sibsport.ru/>
2. https://www.bsuir.by
3. <https://belstu.by>
4. https://www.grsu.by/e-raspisanie
5. Microsoft Visual Studio [Электронный ресурс] – https://ru.wikipedia.org/wiki/Microsoft\_Visual\_Studio – Дата доступа 23.02.2022
6. Полное руководство по языку программирования C# 7.0 и платформе .NET 4.7. Режим доступа: https://metanit.com/sharp/tutorial/ – Дата доступа: 14.03.2022
7. Пацей, Н.В. Курс лекций по языку программирования C# / Н. В. Пацей. – Минск: БГТУ, 2018. – 175 с.
8. Руководство по WPF // [Электронный ресурс]. – Режим доступа: https://metanit.com/sharp/wpf/ – Дата доступа: 28.03.2022
9. Работа с Entity Framework Core [Электронный ресурс] – https://professorweb.ru/my/entity-framework/6/level1/ – Дата доступа 03.05.2022
10. Блинова, Е.А. Курс лекций по Базам данным / Е.А. Блинова. – Минск: БГТУ, 2019. – 175 с.

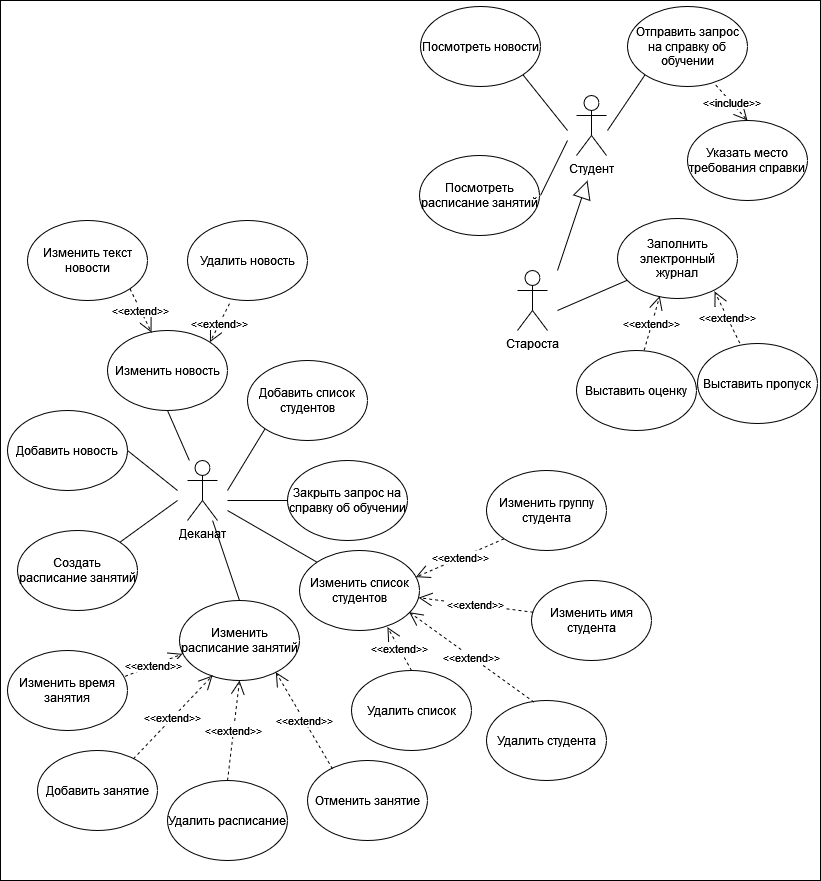
ПРИЛОЖЕНИЕ А



ПРИЛОЖЕНИЕ Б



ПРИЛОЖЕНИЕ В



ПРИЛОЖЕНИЕ Г

