МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПЕУБЛИКИ БЕЛАРУСЬ

Учреждение образования   
«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ

Факультет Информационных технологий

Кафедра Программной инженерии

Специальность 1-40 05 01 Информационные системы и технологии

Направление специальности 1-40 01 02 03 Информационные системы

и технологии (издательско-полиграфический комплекс)

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

**КУРСОВОГО ПРОЕКТА:**

по дисциплине «Объектно-ориентированное программирование»

Тема «Электронный журнал для ВУЗа»

Исполнитель

Студент 2 курса группы 2 Полынская Юлия Викторовна

(Ф.И.О.)

Руководитель работы ассистент Чевжик Е.А. (учен. степень, звание, должность, подпись, Ф.И.О.)

Курсовой проект защищен с оценкой

Председатель Пацей Н.В.

(подпись)

Минск 2019

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПЕУБЛИКИ БЕЛАРУСЬ

Учреждение образования   
«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет информационных технологий  
Кафедра информационных систем и технологий

Утверждаю

Заведующий кафедрой ПИ

\_\_\_\_\_\_\_\_\_\_\_\_ Н.В Пацей

подпись инициалы и фамилия

“\_\_\_”\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2019г.

**ЗАДАНИЕ**

**к курсовому проектированию**

**по дисциплине** "Объектно-ориентированное программирование"

|  |  |
| --- | --- |
| Специальность: 1-40 01 05 03 Информационные системы и технологии  Студент: Полынская Ю.В. | Группа: 2 |
| **Тема:** Электронный журнал для ВУЗа | |

**1. Срок сдачи студентом законченной работы**: "май 2019 г."

**2. Исходные данные к проекту:**

**2.1**. Функционально ПС должно:

* Управлять базой данных: сохранять введенные данные, извлекать информацию об учебном процессе;
* Сохранять, изменять и удалять данные;
* Формировать списки студентов, изучающих дисциплину;
* Вести учет посещения занятий;
* Вести учет cдачи работ;
* Подсчитывать количество пропущенных часов;
* Формировать расписание по учебным предметам.

**2.2.** Программное средство должно быть выполнено с использованием ООП, взаимодействовать с базой данных, реализовано под ОС Windows. Отображение, бизнес-логика и хранилище данных должны быть максимально независимы друг от друга для возможности расширения. Управление программой должно быть интуитивно понятным и удобным. Использовать архитектурные шаблоны и шаблоны проектирования.

**3. Примерное содержание расчетно-пояснительной записки**

(перечень вопросов, подлежащих разработке)

* Введение;
* Обзор литературы и аналогов (алгоритмы решения, обзор литературы);
* Постановка задачи;
* Проектирование программного средства;
* Руководство пользователя;
* Тестирование программного средства;
* Заключение;
* Список используемых источников;
* Приложения.

**4. Форма представления выполненной курсовой работы:**

* Теоретическая часть курсового проекта должны быть представлены в формате MS Word. Оформление записки должно быть согласно правилам.
* Необходимые схемы, диаграммы и рисунки допускается делать в MS Office Visio, VS или других редакторах.
* Листинги программы представляются частично в приложении.
* К записке необходимо приложить CD (DVD).

**Календарный план**

|  |  |  |  |
| --- | --- | --- | --- |
| №  п/п | Наименование этапов курсового проекта | Срок выполнения этапов проекта | Примечание |
| 1 | Введение | 21.02.2019 |  |
| 2 | Аналитический обзор литературы по теме проекта. Изучение требований, определение вариантов использования | 7.03.2019 |  |
| 3 | Анализ и проектирование архитектуры приложения (построение диаграмм, проектирование бизнес-слоя, представления и данных) | 21.03.2019 |  |
| 4 | Проектирование структуры базы данных. Разработка дизайна пользовательского интерфейса | 04.04.2019 |  |
| 5 | Кодирование программного средства | 18.04.2019 |  |
| 6 | Тестирование и отладка программного средства | 02.05.2019 |  |
| 7 | Оформление пояснительной записки | 07.05.2019 |  |
| 8 | Сдача проекта | 20.05.2019 |  |

**5. Дата выдачи задания** 21.02.2019

Руководитель\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ *Чевжик Е.А.*

(подпись)

Задание принял к исполнению \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ *Полынская Ю.В.*

(дата и подпись студента)

Оглавление

[**ВВЕДЕНИЕ** 5](#_Toc10031021)

[**1.Анализ прототипов, литературных источников и формирование требований к проектируемому программному средству** 7](#_Toc10031022)

[**2.Анализ требований к программному средству и разработка функциональных требований** 13](#_Toc10031023)

[**3.Разработка архитектуры проекта** 14](#_Toc10031024)

[**3.1 Обобщенная структура** 14](#_Toc10031025)

[**3.2 Диаграммы UML, взаимосвязь всех компонентов** 16](#_Toc10031026)

[**4.Разработка функциональной модели и модели данных ПС** 20](#_Toc10031027)

[**4.1. Модель базы данных** 20](#_Toc10031028)

[**4.2. Выполняемые функции** 24](#_Toc10031029)

[**5.Тестирование, проверка работоспособности и анализ полученных результатов** 25](#_Toc10031030)

[**5.1 Тестирование** 25](#_Toc10031031)

[**6.Руководство по использованию программного средства** 27](#_Toc10031032)

[**ЗАКЛЮЧЕНИЕ** 31](#_Toc10031033)

[**СПИСОК ЛИТЕРАТУРЫ** 32](#_Toc10031034)

## **ВВЕДЕНИЕ**

Век высоких технологий диктует нам все больше новых тенденций и максимально оптимизирует процессы, которые раннее занимали приличную часть времени. Общество все больше использует технологии в повседневной жизни и вряд ли сможет отказаться от них. Нас же интересует сфера образования.

Важным направлением в развитии образования является разработка и внедрение единой системы электронного документооборота. Сервисы «Электронный журнал/Электронный дневник» позволяют организовать индивидуальный учет результатов освоения обучающимися образовательных программ в электронном виде, а также хранение в архивах данных об этих результатах. Такие журналы позволяют в кратчайшие сроки делиться изменениями и обновлениями данных, что способствует повышению продуктивности.

Программный продукт «Электронный журнал для ВУЗа» должен удовлетворять этим требованиям.

Уже многие образовательные учреждения перешли на электронные системы хранения информации. Как пример можно привести такие университеты как: БГУ, ГрГУ, ГГМУ и другие.

Так же во многих статьях, посвященных внедрению информационных технологий в учебный процесс, говориться о том, что «…Для студента основным инструментом формирования его информационной культуры, получения им необходимых общекультурных и профессиональных компетенций стали информационные, облачные и интернет-технологии. Для преподавателя, практически решающего задачу объективной оценки учебных достижений студента, важными элементами современного учебного процесса стали методы компьютерного тестирования, балльно-рейтинговая система оценивания и система зачетных единиц. Эффективное использование этих систем возможно с помощью нового инструмента – электронного журнала.»

Актуальность разработки и применения в вузе электронного журнала определяется следующими факторами:

* Внедряя балльно-рейтинговые системы, многие вузы требуют от преподавателей использовать журналы успеваемости (электронные журналы) как инструмент оценки знаний, навыков и умений студентов;
* Требования к уровню подготовки студентов и процедуры их оценки должны быть открытыми и понятными для студентов, их родителей и будущих работодателей;
* Переход от разовых экзаменов в конце учебного курса к оценке учебных достижений студента на всех промежуточных этапах обучения по курсу существенно повысит объективность оценки этих достижений и сделает практически невозможным какой-либо обман.

В современных системах электронных журналов обычно реализованы технологии личных кабинетов с различным набором функций, зависящих от прав доступа и той роли, которую выполняет пользователь (преподаватель, студент, администратор), применяются технологии социальных сетей, передачи файлов и сообщений.

Цели проекта:

* Формирование навыков проектирования баз данных;
* Формирование навыков проектирования и разработки пользовательского интерфейса;
* Развитие навыков тестирования продукта;

Задача проекта:

Разработать архитектуру приложения, создать интерфейс, взаимодействие с которым будет понятно пользователю любого возраста. Построить базу данный и интерфейс основываясь на обзоре аналогов и выполнить тестирование готового продукта.

# **1.Анализ прототипов, литературных источников и формирование требований к проектируемому программному средству**

**1.1. Журнал преподавателя**

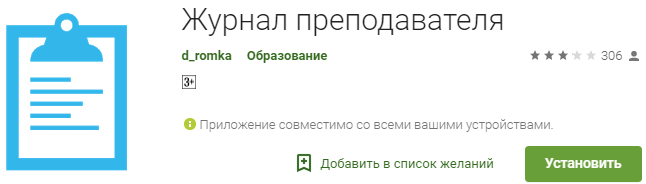


Рис.1.1.1 – Иконка аналога приложения

Программа предназначена для преподавателей университетов. Предоставляет возможность вести журналы посещаемости и успеваемости учащихся.

Дополнительные возможности:

* настраиваемая сводная ведомость;
* студенты могут быть зачислены в несколько групп/подгрупп одновременно;
* настройка графического представления журнала (таблицы);
* импорт записей о студентах/группах из файлов;
* сохранение/восстановление базы данных (SD-card).
* графа "Среднее значение";
* синхронизация с календарем (событие календаря должно содержать название группы, а также аббревиатуры дисциплины и вида занятия).

Интерфейс:

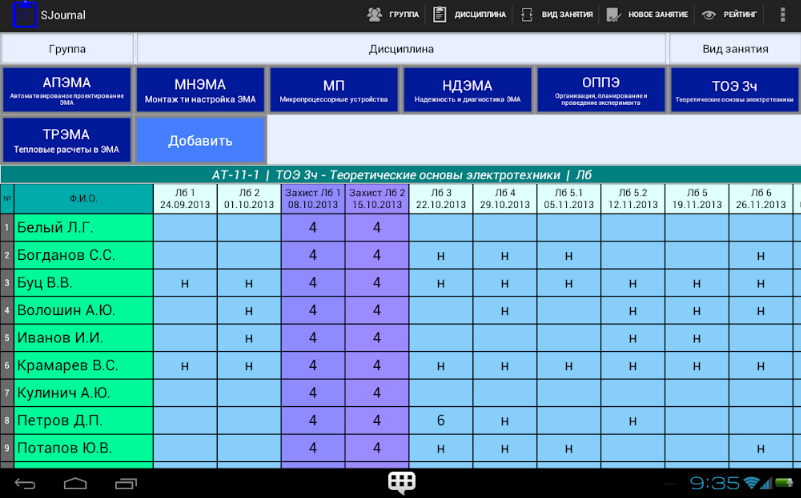


Рис.1.1.2 – Страница со списком группы и выводом дисциплин

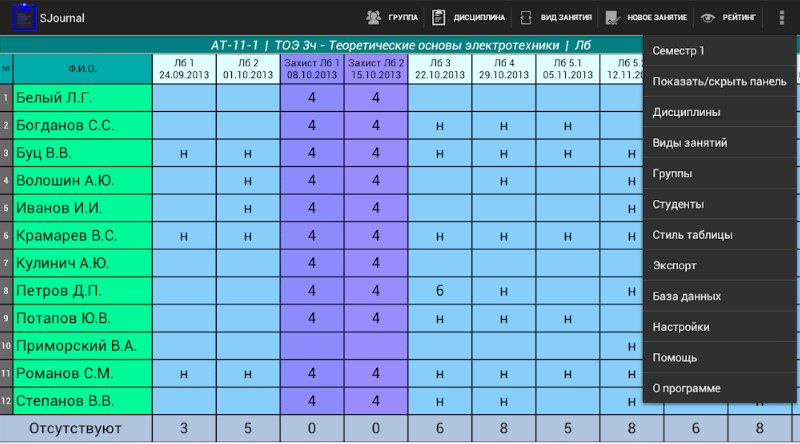


Рис.1.1.3 – Страница со списком группы и выводом возможных действий пользователя

Вывод: Визуально приложение не привлекает. По функционалу тоже есть вопросы. Учитывая отзывы пользователей, могу сказать, что приложение не примитивное и не совсем адаптированное для комфортной работы.

**1.2. iTeacherBook-журнал преподавателя**

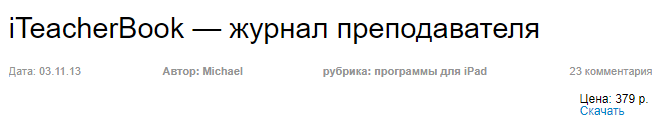
 

Рис.1.4 – Иконка аналога приложения

Красивое и простое приложение поможет сохранить расписание и задания на неделю в удобной форме.

Особенности приложения:

* простое и удобное расписание;
* множество настроек, включая дни недели и количество занятий;
* возможность сохранять задания и заметки;
* создание заданий и прописывание их атрибутов;
* выставление баллов за задания;
* отслеживайте процент посещаемости;
* вы будите видеть свое расписание на текущую неделю, причем можно сделать так, чтобы в нем отображались события из системного календаря;
* можно задавать праздники и правила переноса занятий;
* в режиме текущего дня приложение автоматически показывает идущее сейчас занятие.

Интерфейс:

Осталось создать расписание, то есть для каждого курса указать время и место проведения занятий. Это очень похоже на обычную работу с календарными событиями. Занятия могут быть разных типов, периодичность настраивается без проблем (раз в неделю, раз в две недели — как хотите), допускаются некоторые оформительские элементы. В настройках приложения можно указать продолжительность занятия, а заданный для семестра временной интервал укажет, сколько именно повторений будет создано.



Рис.1.2.1 – Процесс создания расписания

Здесь можно отметить посещаемость. Предусмотрены статусы «присутствовал» и «отсутствовал», а также два расширенных: «опоздал» и «отсутствовал по уважительной причине». Возможно, некоторым преподавателям не хватит статуса «присутствовал, был у доски»

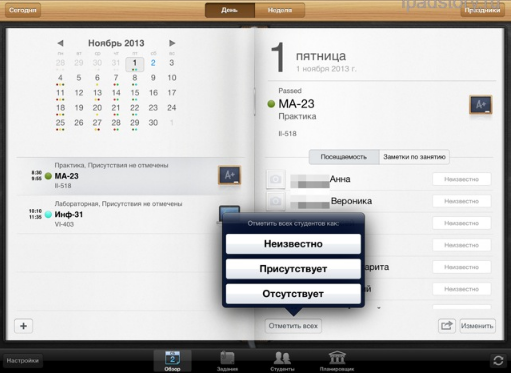


Рис.1.2.2 – Отметка посещаемости

Ну и самая главная задача преподавательского журнала — учёт студенческих успехов. Это делается посредством заданий, под каковое понятие подпадает всё, что подлежит оцениванию. Контрольные работы, домашние задания, индивидуальные задания, лабораторные работы, коллоквиумы, зачёты, экзамены…

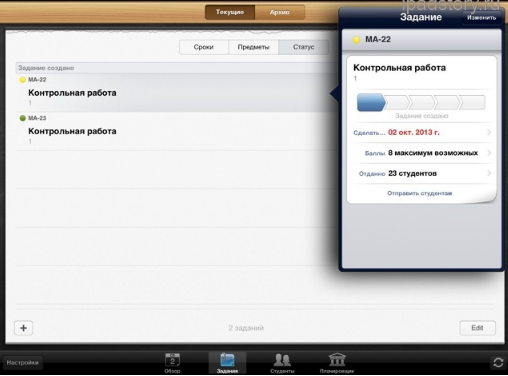


Рис.1.2.3 – Создание заданий

Вывод: хорошое приложение для контроля посещаемости и успеваемости. Комфортное и примитивное в использовании.

**1.3. Magellan – система управления учебным процессом**



Рис.1.8 – Иконка аналога приложения

* понятный и удобный пользовательский интерфейс;
* наличие встроенного инструмента по созданию и редактированию печатных форм;
* возможность формирования всех журналов регистрации справок, документов, приказов, гибкая система документооборота;
* **вы сможете гибко настраивать права доступа** для преподавателей, методистов и администраторов;
* **с помощью большого количества встроенных отчётов вы легко получите картину по обучающимся;**
* отличная работоспособность с огромным массивом данных.

Интерфейс:

**Электронный журнал успеваемости** позволит вам хранить и редактировать данные по текущей успеваемости и посещаемости обучающихся, формировать на их основе фактически выполненную аудиторную нагрузку преподавателей и рейтинги обучающихся, выявлять хронических прогульщиков.

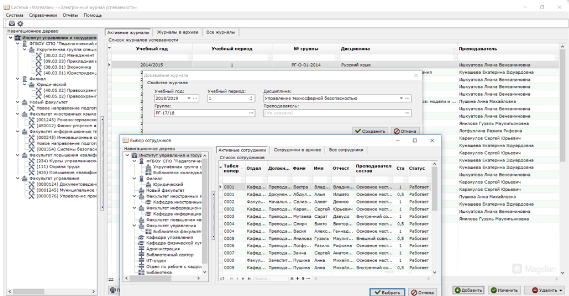


Рис.1.3.1 – Создание электронного журнала – закрепление преподавателя за журналом

**С помощью модуля вы сможете создавать и выбирать типы занятий (лекции, практические, лабораторные, зачеты, экзамены).**

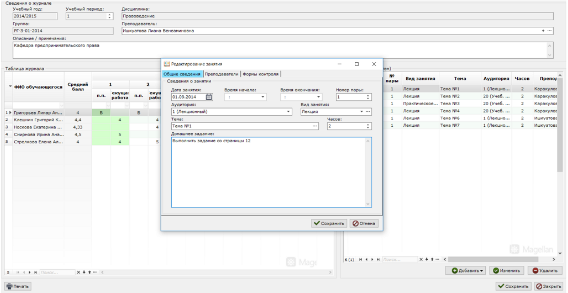


Рис.1.3.2 – Окно редактирования занятий в электронном журнале

Кроме того, у вас появятся [личный кабинет обучающегося](https://magellanius.ru/studentboard/), в котором обучающийся может посмотреть свои оценки, и [личный кабинет сотрудника](http://magellanius.ru/teacherboard/), где преподаватель может редактировать доступные ему журналы успеваемости, создавая новые занятия, указывая успеваемость и посещаемость.

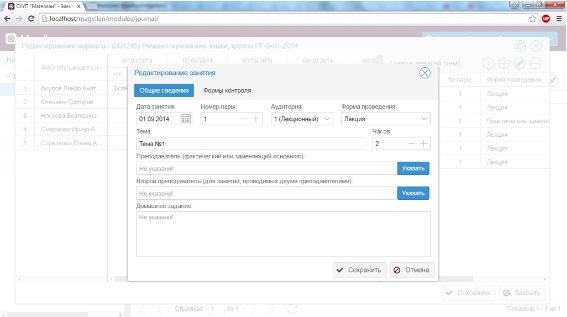


Рис.1.3.3 – Окно редактирования занятий в веб-версии модуля

Вывод: Приложение простое в использование. Функционал минимальный, но оптимальный для студентов и преподавателей. Ничего лишнего.

Данные приложения являются аналогами того, что мы будем создавать. Но просмотрев функционал и отзывы пользователей можно отметить для себя плюсы и минусы таких приложений. Можно взять на заметку не перегруженный, аккуратный и понятный интерфейс, адаптивный функционал и идеи по оптимизации данных.

Требования к электронному журналу:

* Возможность ведения учета успеваемости каждого студента вуза;
* Разграничение прав доступа к базе данных для различных пользователей;
* Возможность одновременного использования базы данных несколькими пользователями как в режиме просмотра, так и в режиме редактирования данных;
* Возможность анализа успеваемости студента, группы;
* Интерфейс приложения должен обеспечивать простой и понятный способ заполнения электронного журнала.

# **2.Анализ требований к программному средству и разработка функциональных требований**

Разрабатываемое приложение должно содержать форму авторизации. Следует разработать два уровня доступа: уровень доступа для преподавателей и уровень доступа для студентов. Преподавателю дополнительно предоставляется возможность добавления студентов, дисциплин, оценок, пропусков и изменения расписания.

Форма реализации продукта – десктоп приложение.

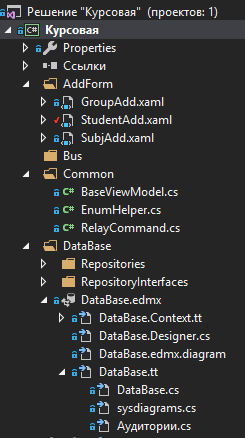
Десктоп приложение должно быть реализовано с использованием языка С#. Технология для реализации – WPF. База данных приложения должно взаимодействовать с Microsoft SQL server.

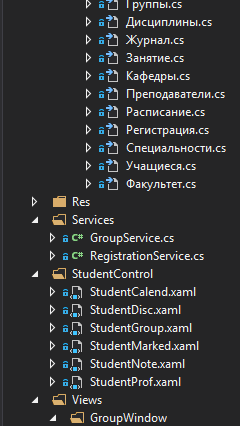
В приложении требуется организовать систему для взаимодействия преподавателя и студентов, с возможностью контроля своей успеваемости. Необходимо построить удобную систему, содержащую личные данные преподавателя/студента, а так же информацию о посещении дисциплин и успеваемости по дисциплинам, так же возможность вносить изменения в расписание (переносы, отмены занятий).  
 Сделать возможность редактирования личных данных.  
 При нажатии на знак меню (в верхнем правом углу) должна открываться панель с расшифровкой обозначений меню.

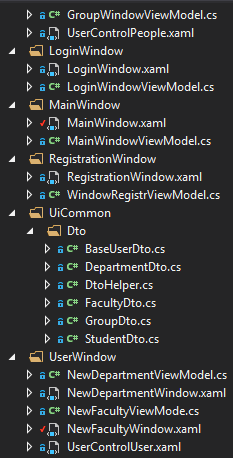
# **3.Разработка архитектуры проекта**

## **3.1 Обобщенная структура**

Решение проекта представлено одним проектом Курсовая, имеющим структуру, представленную на рисунке 3.1.1:







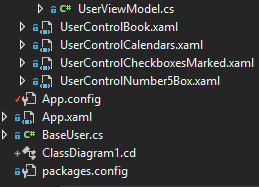


Рисунок 3.1.1 – Структура проекта

Таблица 3.1

|  |  |
| --- | --- |
| Properties | Свойства проекта, содержит информацию о сборке, используемых ресурсах и настройках |
| Ссылки | Перечень сборок, используемых в проекте |
| Properties | Папка со свойствами проекта |
| App.config | Файл с параметрами проекта |
| App.xaml | Класс Application |
| GroopAdd.xaml | Окно добавления группы |
| StudentAdd.xaml | Окно добавления студента |
| SubjAdd.xaml | Окно добавления дисциплины |
| Папка Database | Все классы таблиц, репозитории и их интерфейсы |
| BaseViewModel.cs | Базовый класс для всех viewmodel |
| EnumHelper.cs | Класс перечислений |
| RelayCommand.cs | Класс для обработки событий или команд |
| Services | Папка с сервисами |
| Res | Папка с ресурсами |
| StudentControl | Папка с контролами для взаимодействия студента с приложением и преподавателем |
| GroupWindowViewModel.cs | Класс viewmodel для контрола заполнения групп |
| UserControlPeople.xaml | Контрол, содержащий информацию о группах с кнопками для изменения данных |
| LoginWindow.xaml | Окно входа/регистрации |
| LoginWindowViewModel | Класс viewmodel для авторизации |
| RegistrationWindow.xaml | Окно регистрации |
| WindowRegistrViewModel.cs | Класс viewmodel для регистрации |
| MainWindow.xaml | Начальная страница приложения |
| MainWindowViewModel.cs | Класс viewmodel для начальной страницы |
| Папка UiCommon | Класс который содержит методы конвертирования объектов базы данных в объекты представления |
| UserControlUser.xaml | Страница профиля |
| UserViewModel.cs | Класс viewmodel для страницы профиля |
| NewFacultyWindow.xaml | Окно добавления факультета |
| NewFacultyViewMode.cs | Класс viewmodel для окна добавления факультета |
| NewDepartmentWindow.xaml | Окно добавления специальности |
| NewDepartmentViewModel.cs | Класс viewmodel для окна добавления специальности |
| UserControlBook.xaml | Страница добавления дисциплин |
| UserControlCalendars.xaml | Страница добавления расписания |
| UserControlCheckboxesMarked.xaml | Страница пропусков |
| UserControlNumber5Box.xaml | Страница оценок |
| BaseUser.cs | Класс для добавления пользователей |
| ClassDiagram1.cd | Класс диаграмм |

## **3.2 Диаграммы UML, взаимосвязь всех компонентов**

Диаграмма UML – это графическое представление выбора элементов, изображаемое чаще всего в виде графа с вершинами (сущностями) и рёбрами (отношениями). Диаграммы рисуют для визуализации. Основная цель диаграмм – визуализация разрабатываемой системы с разных точек зрения.

На диаграмме 3.2.1 представлены принципы работы приложения с точки зрения студента и преподавателя соответственно:

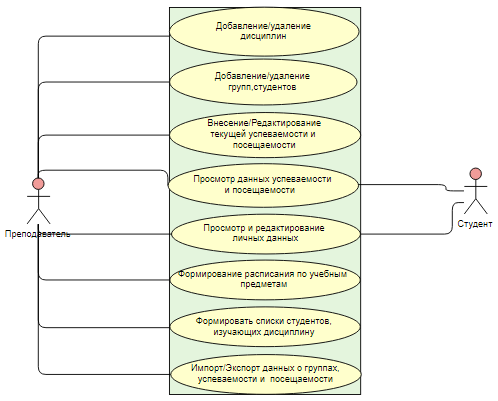
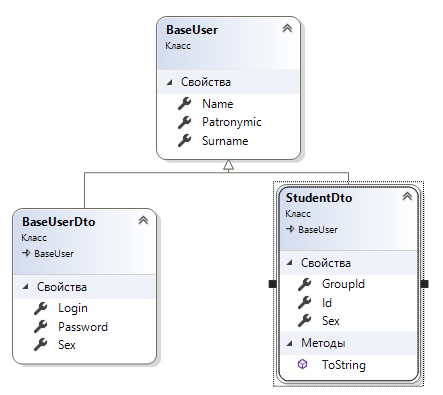
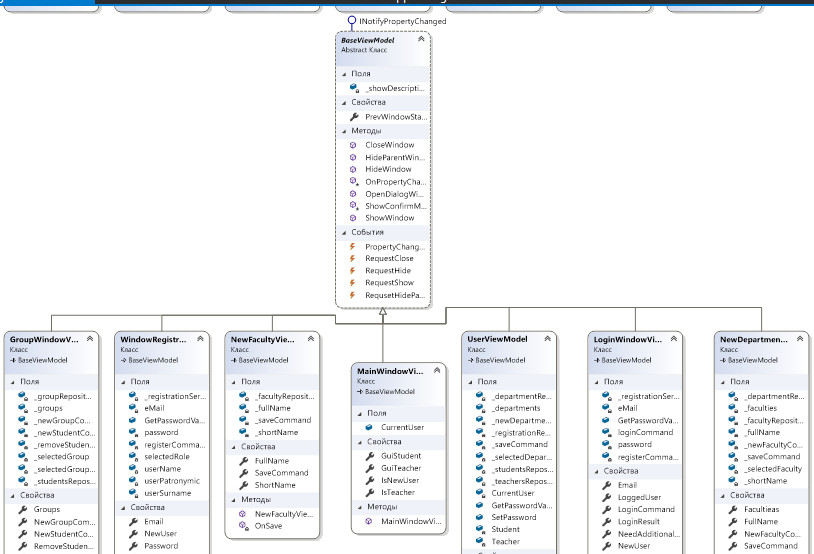
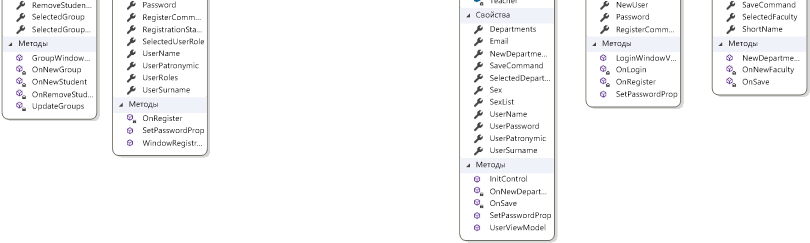


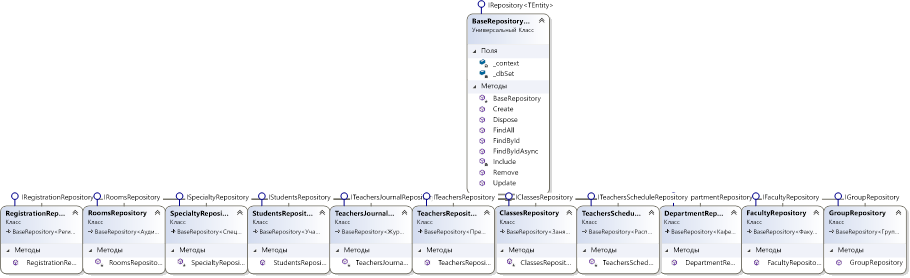
Рисунок 3.2.1 – Принцип работы студента и преподавателя

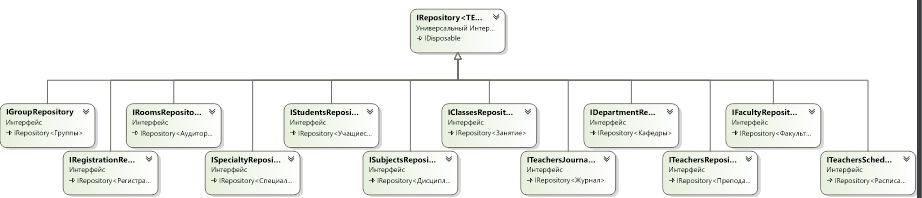
На рисунке 3.2.2 представлена диаграмма классов:











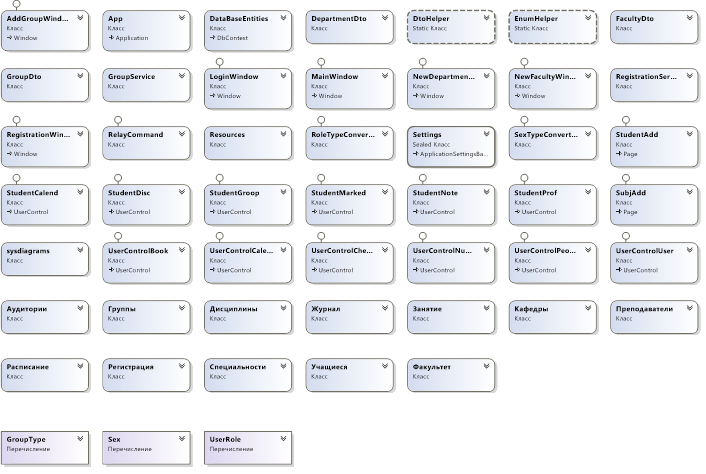


Рисунок 3.2.2 – диаграмма классов

# **4.Разработка функциональной модели и модели данных ПС**

## **4.1. Модель базы данных**

Для создания базы данных использовалась система управления реляционными базами данных Microsoft SQL Server 2017.

База данных состоит из 12 таблиц, перечисленных ниже (рисунок 4.1.1):

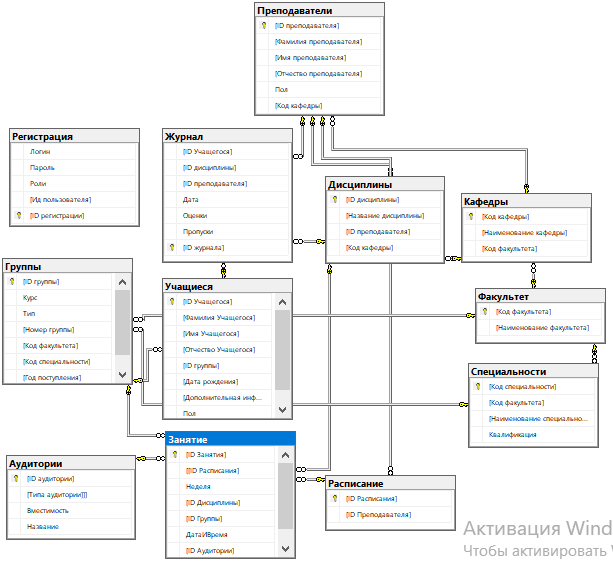


Рисунок 4.1.1 – Взаимосвязь таблиц данных

Диаграмма на рисунке 4.1.1 иллюстрирует структуру таблицы Аудитории, которая содержит информацию о кабинетах и их вместимости:

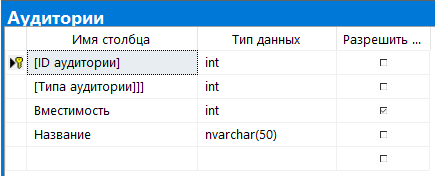


Рисунок 4.1.2 – Структура таблицы Аудитории

Для всех полей, кроме Вместимость установлено ограничение not null, т.к. при регистрации эти поля должны быть заполнены обязательно; для остальных же – default.

Для хранения данных о студентах определенного курса и группы была создана таблица Группы, представленная на рисунке 4.1.3:

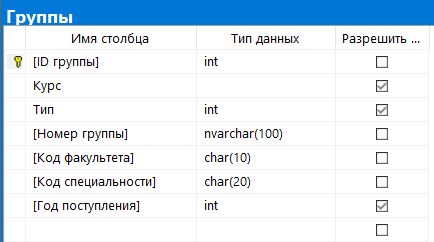


Рисунок 4.1.3 – Структура таблицы Группы

Для хранения данных о преподавателях и о дисциплинах, которые те ведут, была создана таблица Дисциплины, представленная на рисунке 4.1.4:

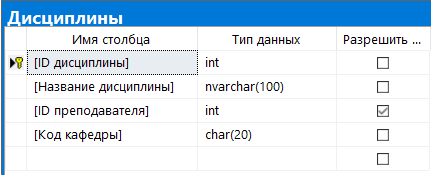


Рисунок 4.1.4 – Структура таблицы Дисциплины

Для хранения данных о студентах, их посещении и успеваемости была создана таблица Журнал, представленная на рисунке 4.1.5:

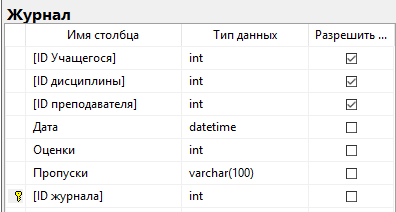


Рисунок 4.1.5 – Структура таблицы Журнал

Для хранения данных о расписании определенного преподавателя была создана таблица Занятие, представленная на рисунке 4.1.6:

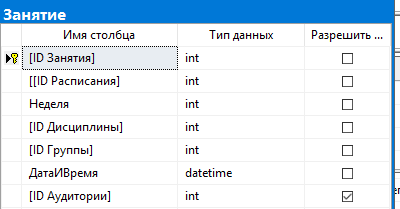


Рисунок 4.1.6 – Структура таблицы Занятие

Для хранения данных о кафедре была создана таблица Кафедры, представленная на рисунке 4.1.7:

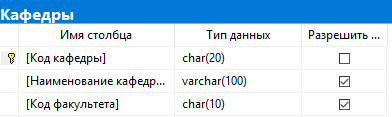


Рисунок 4.1.7 – Структура таблицы Кафедры

Для хранения данных о преподавателях и их предметной области была создана таблица Группы, представленная на рисунке 4.1.8:

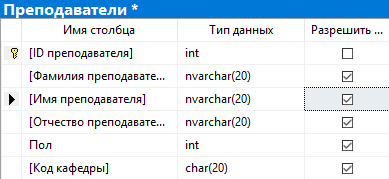


Рисунок 4.1.8 – Структура таблицы Преподаватели

Для хранения данных о занятиях определенного преподавателя была создана таблица Расписание, представленная на рисунке 4.1.9:

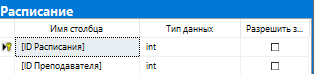


Рисунок 4.1.9 – Структура таблицы Расписание

Для хранения данных при регистрации была создана таблица Регистрация, представленная на рисунке 4.1.10:

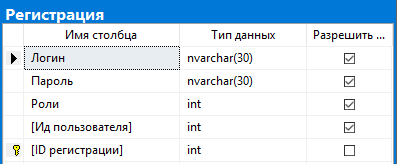


Рисунок 4.1.10 – Структура таблицы Регистрация

Для хранения данных о факультете была создана таблица Факультет, представленная на рисунке 4.1.11:

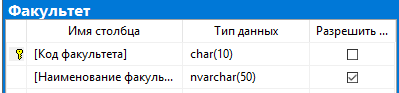


Рисунок 4.1.11 – Структура таблицы Факультет

Для хранения данных о студентах была создана таблица Учащиеся, представленная на рисунке 4.1.12:

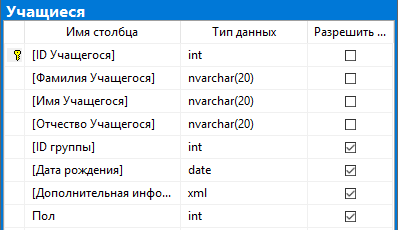


Рисунок 4.1.12 – Структура таблицы Учащиеся

Для хранения данных о специальностях факультета была создана таблица Специальности, представленная на рисунке 4.1.13:

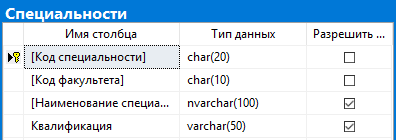


Рисунок 4.1.13 – Структура таблицы Специальности

## **4.2. Выполняемые функции**

Данное приложение предоставляет минимальные требования к выполнению функций приложения и выполняет следующие функции:

* регистрация с последующей авторизацией пользователя;
* возможность изменения личных данных пользователя;
* добавление и удаление студентов в группах;
* импорт записей о студентах/группах из файлов;
* формировать расписание по учебным предметам;
* сохранять, изменять и удалять данные;
* формировать списки студентов, изучающих дисциплину;
* вести учет посещения занятий;
* вести учет cдачи работ;
* подсчитывать количество пропущенных часов.

# **5.Тестирование, проверка работоспособности и анализ полученных результатов**

## **5.1 Тестирование**

Для оценки работоспособности разработанного приложения были проведены тесты использование памяти и использование ЦП.

Тестирование использование памяти проводится с целью обнаружить проблемы, как утечка памяти.

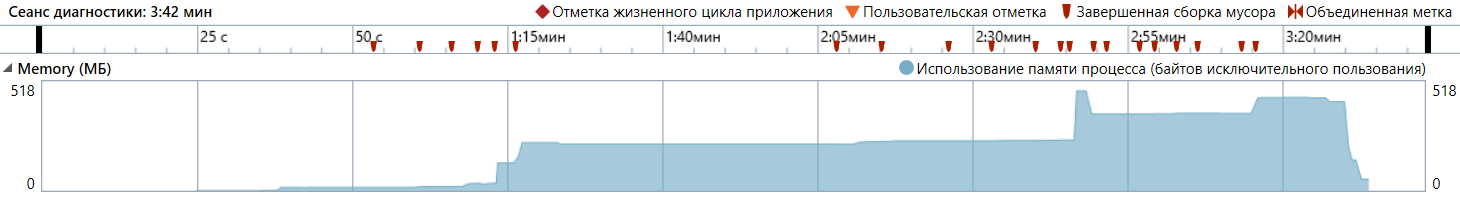
****

Рисунок 5.1.1 – Тестирование памяти

Как показано на Рисунке 5.1.1 тестирование приложения на проблему утечку памяти пройдено успешно. Приложения в процессе своей работы использует 150-190 мб памяти.

Тестирование использование ЦП проводится с целью нахождения степени загруженности центрального процессора при работе программы.

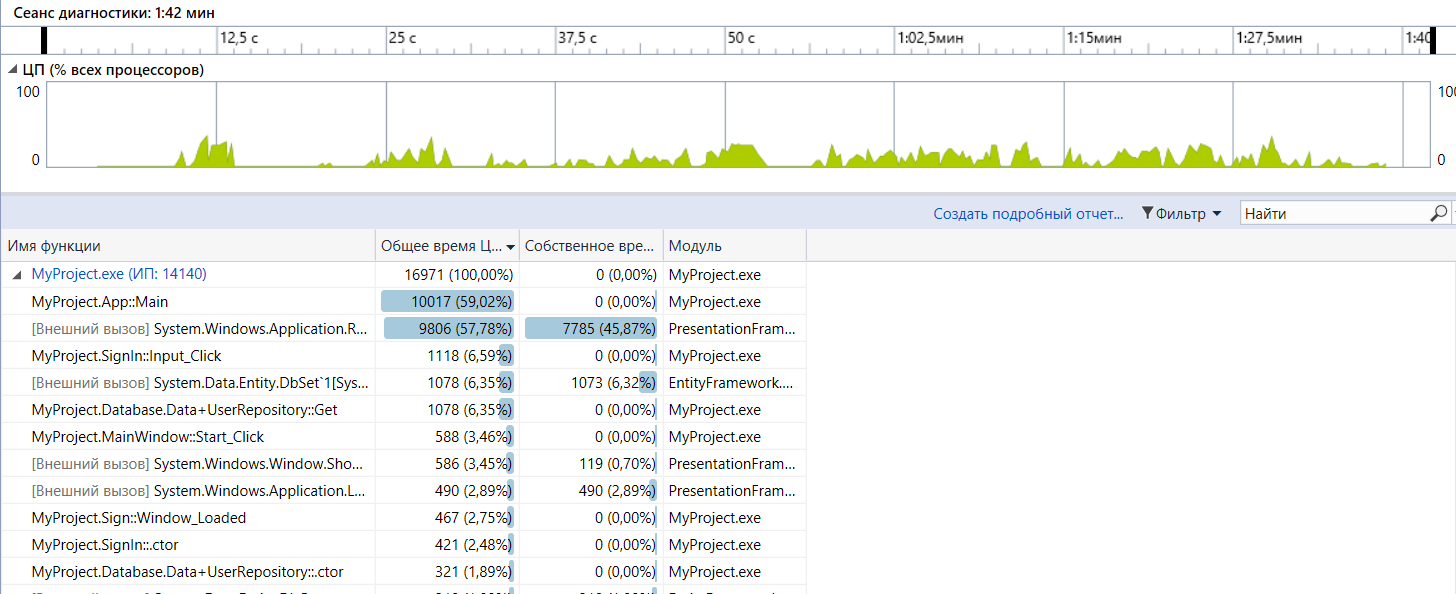


Рисунок 5.1.2 – Тестирование ЦП

Как показано на Рисунке 5.1.2 тестирование приложения на загруженность процессора основная нагрузка на приложение (59%) при запуске приложения. Стоит заметить, что нагрузка на процессор не постоянная, а в среднем 0.5-0.9 секунды при том, что все страницы генерируются из базы данных данный результат можно считать хорошим.

На рисунке 5.1.3 показана временная шкала выполнения процессов в приложении

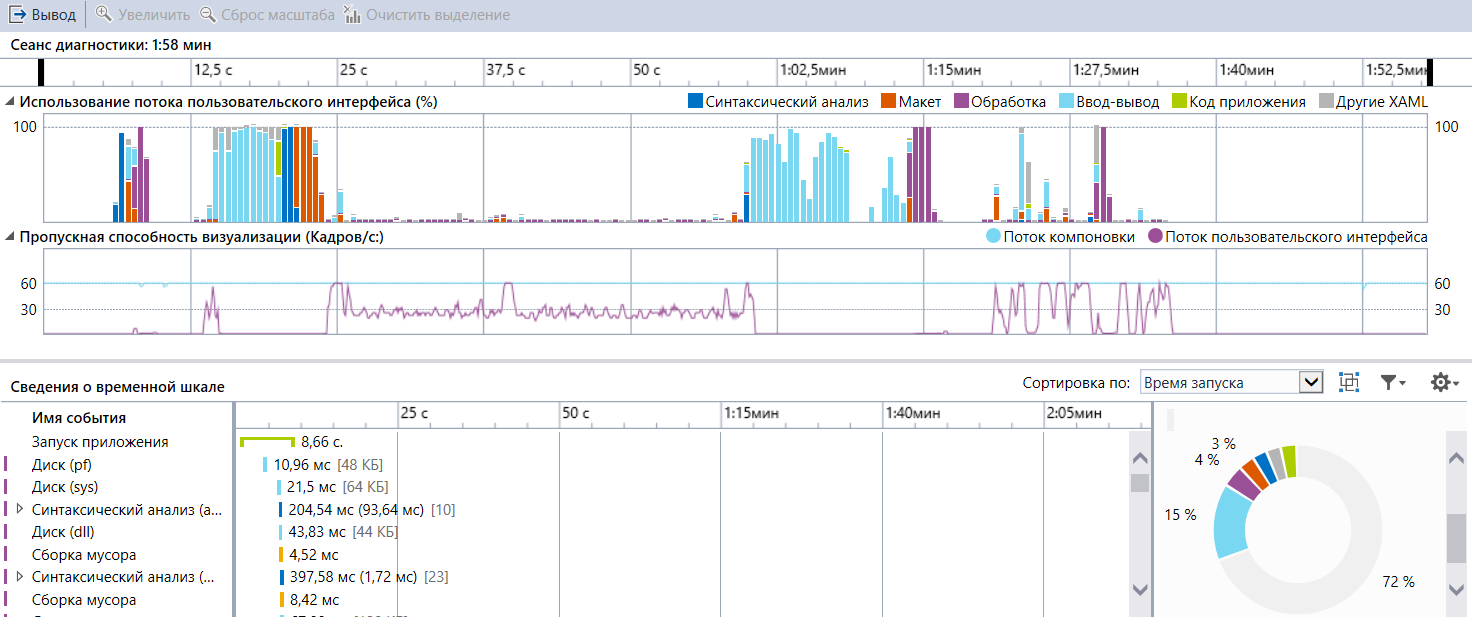


Рисунок 5.1.3 – Временная шкала

По результату тестирование и получение временной шкалы программного средства можно сделать вывод что из во время работы нашего приложения 72% времени свободно от процессов, что говорит о хорошей оптимизации приложения.

Протестировав приложение можно сделать вывод, что программное средство «Электронный журнал для ВУЗа» является нетребовательным к системным характеристикам приложением и может запуститься практически на любом устройстве с оперативной памятью не ниже 512 мб.

# **6.Руководство по использованию программного средства**

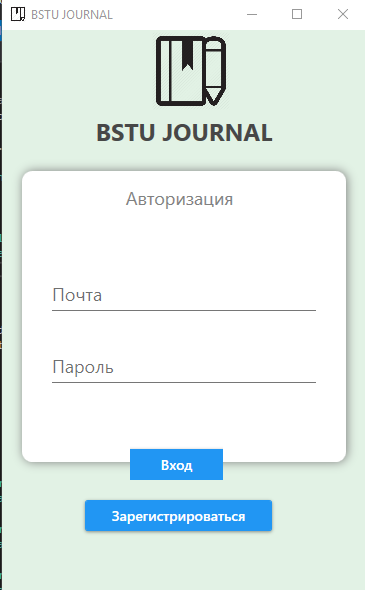
При запуске приложения открывается окно входа/регистрации (рис.6.1 и 6.2):  
 

Рисунок 6.1 – окно входа Рисунок 6.2 – окно регистрации

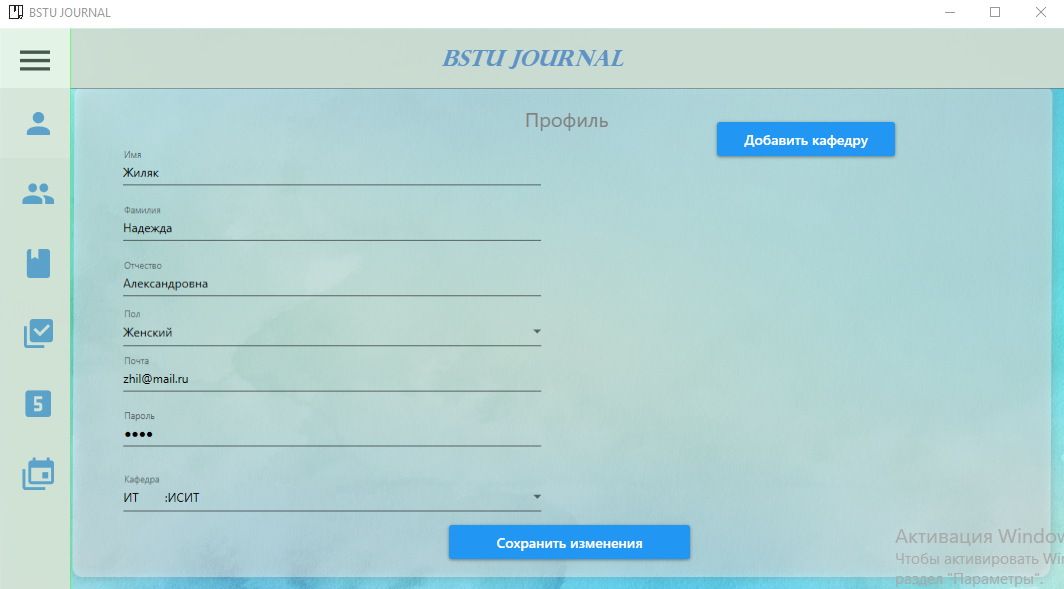
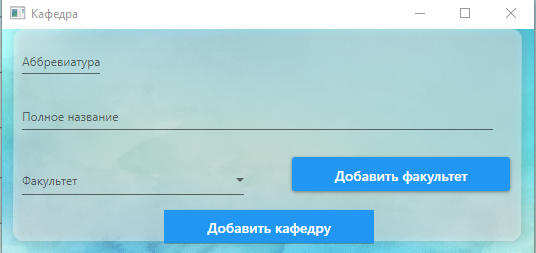
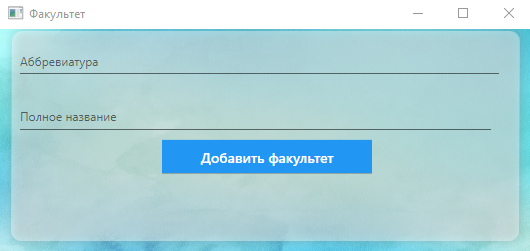
После входа/регистрации откроется окно персональной страницы (рис. 6.3):  


Рисунок 6.2 – профиль преподавателя

Форму можно считать универсальной для университета, так как кафедра и факультет добавляется вручную, далее загружается в бд, а после выбирается в выпадающем меню (рис. 6.3 и 6.4):

   
Рисунок 6.3 – окно добавления кафедры Рисунок 6.4 – окно добавления факультета

Следующий раздел – Группы. В данном разделе мы можем создать группы и добавить/удалить студентов из данной таблицы (рис.6.5, 6.6, 6.7):

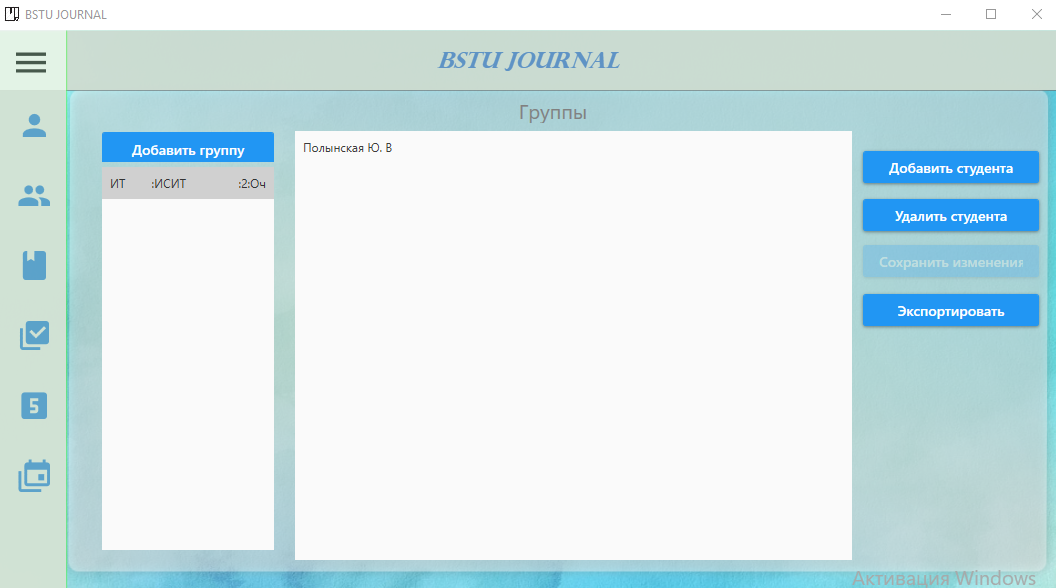
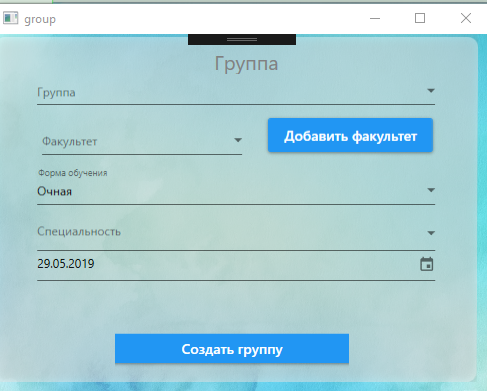
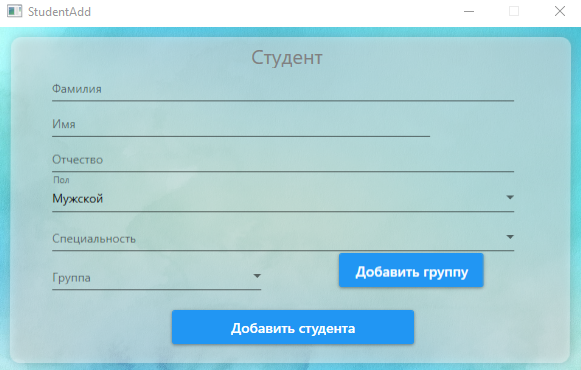


Рисунок 6.5 – Списки группы

  Рисунок 6.6 – окно добавления группы Рисунок 6.7 – окно добавления студентов

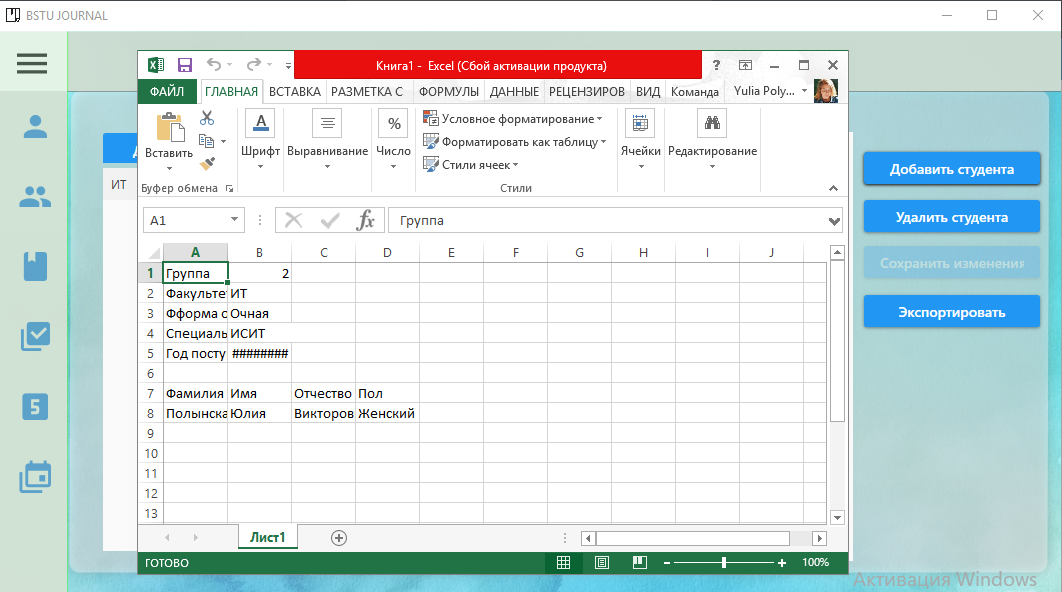
Так же для удобства сделан экспорт введенных данных, то есть заполненных раннее форм (рис. 6.6 и 6.7), которые выводятся в Excel таблицы (рис. 6.8):

Рисунок 6.8 – Экспорт в Excel

Раздел Дисциплины позволяет преподавателю создать нужные Дисциплины, которые далее тот сможет использовать при создании расписания, а также просмотривать то,что вел (рис. 6.9):

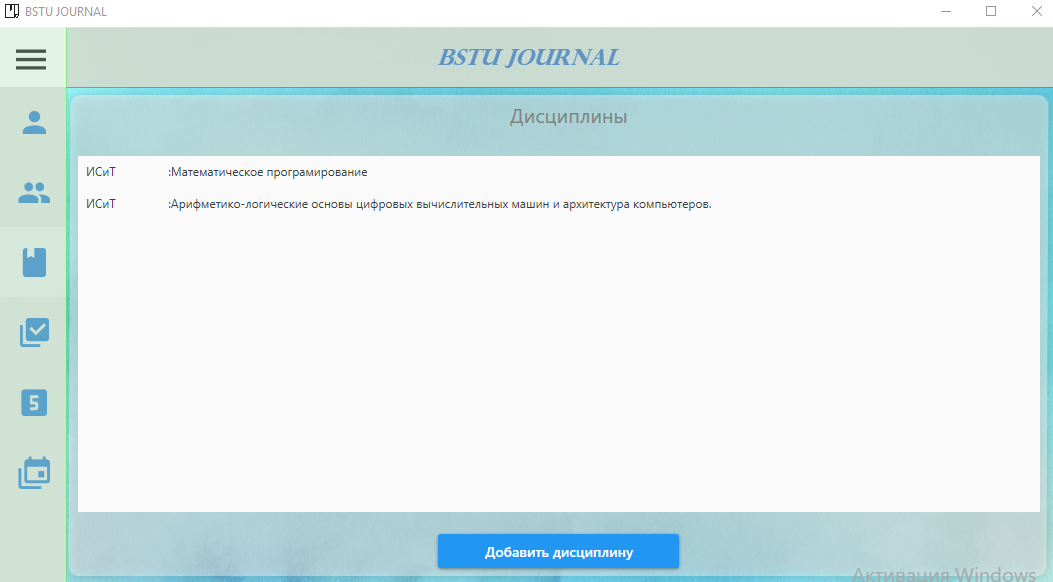
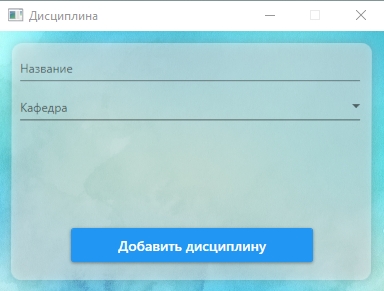


Рисунок 6.9 – Дисциплины



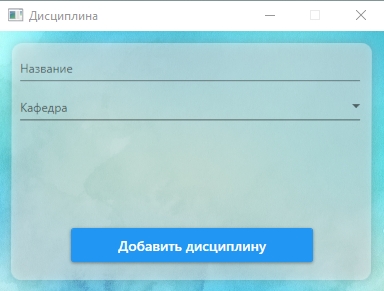


Рисунок 6.10 – Добавление дисциплин

Раздел Успеваемость нужен для того, чтобы преподаватель мог отслеживать посещаемость и успеваемость студентов:

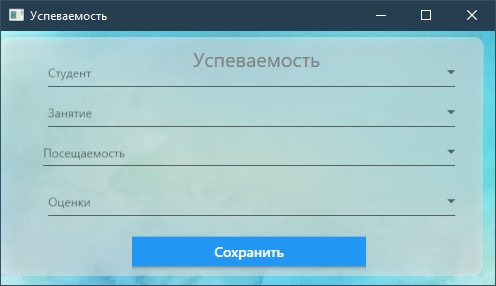


Рисунок 6.11 – Успеваемость и посещаемость

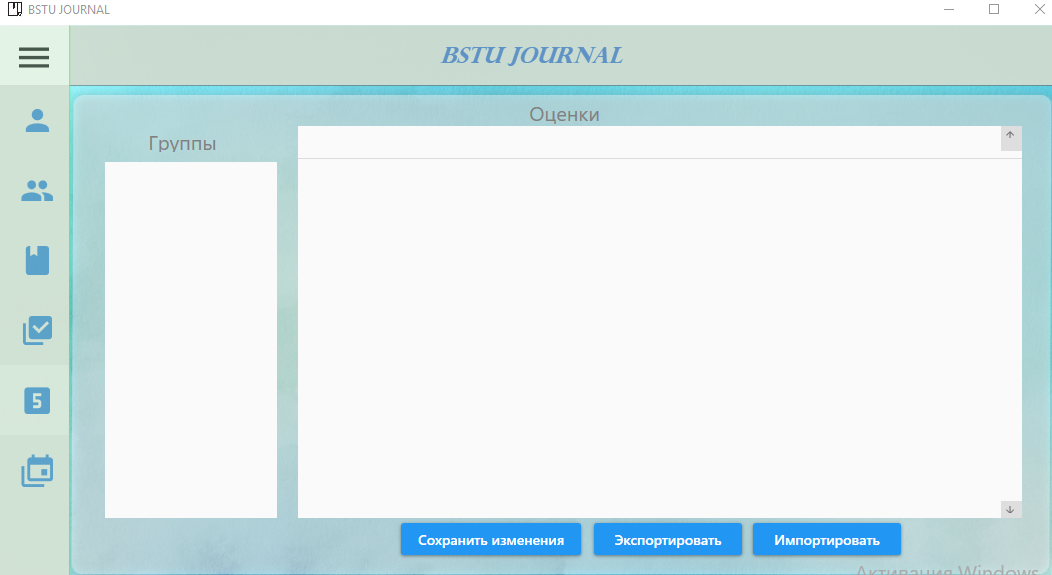


Рисунок 6.12 – Добавление оценки и пропуска

Расписание создается преподавателем для личного контроля, отметки возможных переносов или замен:

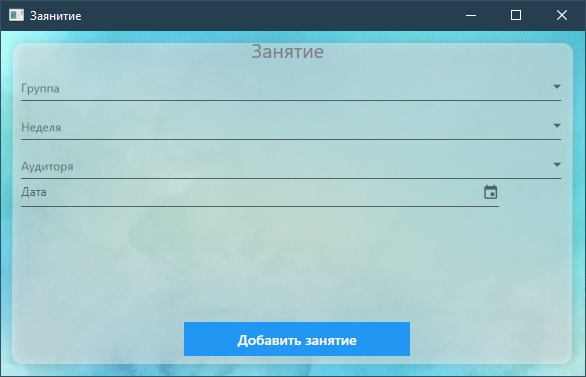


Рисунок 6.13 – Создание расписания

  
Рисунок 6.14 – Расписание

# **ЗАКЛЮЧЕНИЕ**

В ходе выполнения работы были достигнуты следующие результаты:

* изучена работа с технологией wpf, практиковалось написание кода на языке c#;
* исследованы обучающие приложения-аналоги;
* сформулирована модель данных и приведено её графическое представление;
* разработано руководство по использованию приложения;
* подготовлена база данных с использованием mssql-сервера.

# **СПИСОК ЛИТЕРАТУРЫ**

1. Язык C# и .NET Framework [Электронный ресурс] – http://professorweb.ru/my/csharp/charp\_theory/level1/infonet.php.
2. Руководство по ADO.NET и работе с базами данных [Электронный ресурс] – https://metanit.com/sharp/adonet/. Дата доступа: 10.05.2017