Частное учреждение образования

«Колледж бизнеса и права»

РАЗРАБОТКА ЭЛЕКТРОННОГО ОРГАНАЙЗЕРА ДЛЯ ПРЕДСЕДАТЕЛЯ ЦИКЛОВОЙ КОМИССИИ ЧУО "КОЛЛЕДЖ БИЗНЕСА И ПРАВА"

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовому проекту по учебному предмету

«Конструирование программ и языки программирования»

КП Т.292005.401

Руководитель проекта ( Е.Н.Коропа )

Обучающаяся ( В.В.Внук )

2025

**Этикетка**

для курсовых проектов

**Курсовой проект**

Тема Разработка электронного органайзера для председателя цикловой комиссии ЧУО "Колледж бизнеса и права"

КП Т.292005.401

Разработан Внук В.В.

Утвержден Коропа Е.Н.

Руководитель: Коропа Е.Н.

Технические средства: ПК с ЦПУ частотой от 2 ГГц и ОП от 4 ГБ, монитор, клавиатура, мышь

Программные средства: ОС Microsoft Windows 11 Pro, Microsoft Visual Studio,

.NET 9.0, .NET Framework 4.8

**Состав документа:**

Пояснительная записка – Внук ПЗ.docx

Графическая часть –

Папка с проектом – Курсовая1.0

Установочный пакет программного средства – setup.exe

Сведения о защите информации: логин отсутствует, пароль 13022007

СОДЕРЖАНИЕ

*Изм.*

*Изм.*

*Изм.*

*Изм.*

*Изм.*

*Изм.*

*Изм.*

*Изм.*

*Лист*

*Лист*

*Лист*

*Лист*

*Лист*

*Лист*

*Лист*

*Лист*

*№ докум.*

*№ докум.*

*№ докум.*

*№ докум.*

*№ докум.*

*№ докум.*

*№ докум.*

*№ докум.*

*Подпись*

*Подпись*

*Подпись*

*Подпись*

*Подпись*

*Подпись*

*Подпись*

*Подпись*

*Дата*

*Дата*

*Дата*

*Дата*

*Дата*

*Дата*

*Дата*

*Дата*

*Лист*

*Лист*

*Лист*

*Лист*

*Лист*

*Лист*

*Лист*

*Лист*

*3*

*3*

*3*

*3*

*2*

*3*

*3*

*3*

КП Т.292005.401 ПЗ

*КП Т.716007.401 ПЗ*

*КП Т.716009.401 ПЗ*

*КП Т.716007.401 ПЗ*

*ОП Т.893023*

*КП Т.716007.401 ПЗ*

*КП Т.716009.401 ПЗ*

*КП Т.716007.401 ПЗ*

*Разраб.*

*Разраб.*

*Разраб.*

*Разраб.*

*Разраб.*

*Разраб.*

*Разраб.*

*Разраб.*

*Внук В.В.*

*Провер.*

*Провер.*

*Провер.*

*Провер.*

*Провер.*

*Провер.*

*Провер.*

*Провер.*

*Коропа Е.Н.*

*Т. контр.*

*Т. контр.*

*Т. контр.*

*Т. контр.*

*Т. контр.*

*Т. контр.*

*Т. контр.*

*Т. контр.*

*Н. контр.*

*Н. контр.*

*Н. контр.*

*Н. контр.*

*Н. контр.*

*Н. контр.*

*Н. контр.*

*Н. контр.*

*Утверд.*

*Утверд.*

*Утверд.*

*Утверд.*

*Утверд.*

*Утверд.*

*Утверд.*

*Утверд.*

*Коропа Е.Н.*

РАЗРАБОТКА ЭЛЕКТРОННОГО ОРГАНАЙЗЕРА ДЛЯ ПРЕДСЕДАТЕЛЯ ЦИКЛОВОЙ КОМИССИИ ЧУО "КОЛЛЕДЖ БИЗНЕСА И ПРАВА"

*Лит.*

*У*

*Лит.*

*У*

*Лит.*

*У*

*Лит.*

*У*

*Лит.*

*У*

*Лит.*

*У*

*Лит.*

*У*

*Лит.*

*У*

*Листов*

*Листов*

*Листов*

*Листов*

*Листов*

*Листов*

*Листов*

*Листов*

*45*

*55*

*50*

*50*

*55*

*50*

*КБП*

*КБП*

*КБП*

*КБП*

*КБП*

*КБП*

*КБП*

*КБП*

[Введение 4](#_Toc199161117)

[1 Описание задачи 6](#_Toc199161118)

[1.1 Анализ предметной области 6](#_Toc199161119)

[1.2 Постановка задачи 6](#_Toc199161120)

[2 Проектирование системы 8](#_Toc199161121)

[2.1 Требования к приложению 8](#_Toc199161122)

[2.2 Проектирование задачи 8](#_Toc199161123)

[Основные сущности системы: 8](#_Toc199161124)

[2.3 Концептуальный прототип 10](#_Toc199161125)

[3 Описание реализации программного средства 14](#_Toc199161126)

[3.1 Инструменты разработки и применяемые технологии 14](#_Toc199161127)

[3.2 Порядок авторизации пользователей 15](#_Toc199161128)

[3.3 Организация данных 15](#_Toc199161129)

[3.4 Функции: логическая и физическая организация 17](#_Toc199161130)

[3.5 Входные и выходные данные 21](#_Toc199161131)

[3.6 Функциональное тестирование 21](#_Toc199161132)

[3.7 Описание справочной системы 24](#_Toc199161133)

[4 Применение 27](#_Toc199161134)

[4.1 Назначение программного средства 27](#_Toc199161135)

[4.2 Условия применения 27](#_Toc199161136)

[Заключение 29](#_Toc199161137)

[Список использованных источников 31](#_Toc199161138)

[ПРИЛОЖЕНИЕ А 32](#_Toc199161139)

[ПРИЛОЖЕНИЕ Б 46](#_Toc199161140)

# Введение

В современных образовательных учреждениях одной из ключевых задач становится эффективное управление данными о преподавателях, их дисциплинах, планах работ и выдаваемых заданиях. Решение этой задачи способствует прозрачности процессов, оптимизации планирования и формированию необходимой документации. Основной целью разработки является создание системы, автоматизирующей ведение базы данных преподавателей, дисциплин, планов работ и заданий, а также обеспечивающей безопасность данных и удобство взаимодействия с системой.

Актуальность проекта определяется необходимостью внедрения современных цифровых решений в образовательную среду. Автоматизация учета и управления информацией позволяет снизить временные затраты, минимизировать вероятность ошибок при работе с данными и повысить удобство эксплуатации программного средства. Таким образом, внедрение данной системы имеет значимое практическое значение и способствует совершенствованию учебной и административной деятельности.

В рамках выполнения курсового проекта будет подготовлена пояснительная записка, включающая четыре раздела.

В первом разделе «Описание задачи» рассматривается предметная область и формулируется основная задача, решаемая в ходе разработки. Особое внимание уделяется анализу существующих аналогов, а также определяются требования к безопасности и периодичности использования системы.

Во втором разделе «Проектирование системы» формулируются требования к функционалу приложения, моделируется архитектура программного средства, описывается структура базы данных, а также представляется концептуальный прототип пользовательского интерфейса с учетом организации меню, кнопочных форм и справочной системы.

Третий раздел «Описание реализации программного средства» посвящён выбору инструментов и технологий разработки, описанию структуры данных, реализации ключевых функций - таких как поиск, добавление, редактирование и удаление записей, выдача и управление заданиями, автоматизация создания шаблонов протоколов. В разделе также рассматриваются вопросы тестирования системы и подготовки справочной документации.

Четвертый раздел «Применение» содержит информацию о назначении системы и условиях её использования, демонстрируя сферы применения и влияние на организационные процессы образовательного учреждения.

В заключении проводится анализ разработанного программного средства, сопоставление поставленных задач с достигнутыми результатами, а также рассматриваются возможные несоответствия и направления дальнейшего развития системы.

Приложение А содержит текст программы.

Приложение Б содержит изображения с результатами работы программы.

В графической части представлены диаграммы вариантов использования, деятельности и классов.

# Описание задачи

## Анализ предметной области

В рамках данной системы исследуется организация и автоматизация работы председателя цикловой комиссии.

Органайзер знаний — это компактный документ или система, которая содержит сводку ключевых знаний и информации по учебным темам, предназначенный для систематизации и упрощения доступа к важным данным. В контексте преподавателя органайзер помогает структурировать учебный материал, планы и другую методическую информацию, облегчая подготовку и проведение занятий. Он не заменяет учебный курс, а служит вспомогательным инструментом для эффективного управления учебным процессом.

Председатель цикловой комиссии — это опытный педагог, назначаемый директором учебного заведения, который осуществляет планирование, организацию и руководство работой цикловой комиссии. Его основные обязанности включают:

* анализ проблем и результатов методической работы закрепленных преподавателей;
* анализ формы и содержания посещенных уроков и других видов деятельности;
* прогнозирование последствий запланированных мероприятий;
* разработка предложений по организации и содержанию аттестации педагогов;
* составление отчетов о профессиональном самообразовании преподавателей и их участии в курсах повышения квалификации.

Планы — это документы, отражающие структуру и содержание учебного процесса, включая учебные планы и планы работ преподавателей. Учебные планы разрабатываются по специальностям и включают государственный компонент и компонент учреждения образования, отражая профилизацию и особенности подготовки специалистов. Они могут быть примерными, учебными планами учреждений или экспериментальными.

Протоколы — официальные документы, фиксирующие результаты заседаний и решений цикловой комиссии. В курсовом проекте предусмотрена автоматизация создания шаблонов протоколов в формате Word для упрощения ведения документации.

Учебный предмет — это систематизированная совокупность знаний, умений и навыков, отобранных из определенной отрасли науки, техники или искусства для изучения в учебном заведении. Он может быть общеобразовательным или специальным, определяющим профиль подготовки специалиста. Учебный предмет является основной структурной единицей учебно-воспитательного процесса и реализует содержание обучения через учебные планы и программы.

В рамках учебного предмета преподаватель должен:

* разрабатывать и использовать учебную программу, которая определяет цели, содержание и методы обучения по предмету;
* составлять КТП (календарно-тематическое планирование), где расписаны темы занятий, сроки и формы контроля;
* подбирать и использовать соответствующую литературу и учебные материалы, обеспечивающие качественное освоение учебного предмета.

Электронный органайзер для председателя цикловой комиссии должен обеспечивать:

* безопасный вход в систему с паролями;
* ведение базы данных преподавателей, их дисциплин и планов работы с возможностью поиска, добавления, редактирования и удаления данных;
* управление заданиями преподавателям с установлением сроков сдачи;
* редактирование планов работ;
* автоматизированное создание протоколов в формате Word;
* удобный и лаконичный интерфейс с меню, панелями инструментов и справочной системой.

Это позволит председателю ЦК эффективно организовывать методическую работу, контролировать учебный процесс и документацию, что соответствует его функциональным обязанностям и требованиям учебного заведения.

## Постановка задачи

На основе анализа процессов управления данными преподавателей, дисциплин, планов работ и заданий выделены следующие функции, подлежащие автоматизации в системе:

* авторизация и безопасность данных. Реализация механизма входа в систему по заранее выданным паролям. Обеспечение защиты данных преподавателей и планов работ;
* управление базой данных. Ведение базы данных преподавателей, дисциплин и планов работ (добавление, редактирование, удаление записей). Реализация поиска по ключевым параметрам: преподаватель, дисциплина, срок выполнения задания;
* управление заданиями. Выдача заданий преподавателям с установлением сроков сдачи. Корректировка заданий (изменение описания, сроков, статуса выполнения);
* работа с планами работ. Создание, редактирование и структурирование планов работ. Автоматизация формирования шаблонов протоколов в формате Word;
* интерфейс и справочная система. Организация лаконичного интерфейса с использованием меню, кнопочных форм и панелей инструментов. Интеграция встроенной справочной системы для упрощения взаимодействия с приложением.

Существуют решения для управления задачами (например, «Trello», «Asana», «Google Class») и базами данных («Microsoft Access», «MS SQL Server», «Airtable»), однако они не соответствуют специфическим требованиям образовательных учреждений по следующим причинам:

* отсутствие специализированного функционала для управления планами работ и дисциплинами;
* невозможность автоматической генерации протоколов в формате Word.

Система ориентирована на администраторов учебных учреждений для управления данными и преподавателей для взаимодействия с заданиями и планами работ.

Система будет иметь интуитивно понятный интерфейс и встроенную справочную систему для упрощения работы с программным средством.

# Проектирование системы

## Требования к приложению

В разрабатываемой системе требуется реализовать шифрование паролей с помощью хеширования. Необходимо реализовать механизм авторизации с использованием заранее выданных паролей. Интерфейс должен быть лаконичным, без лишнего визуального шума. Все ключевые функции должны быть доступны за минимальное количество действий. Размеры текстовых полей должны быть ограничены для предотвращения ошибок или перегрузки системы.

Элементы управления должны быть расположены логически, с разделением по функциональным блокам. Кнопки и текстовые поля должны быть достаточно крупными для удобного взаимодействия. Необходимо использовать шрифт Arial размером 16 пикселей. Цветовая палитра должна включать в себя преимущественно нейтральные тона (белый, темно-серый).

Для использования приложения требуется операционная система Windows 11, объёмом памяти минимум 2 ГБ. Необходима установленная платформа .NET Framework версии 4.8 или выше.

(Какие требования предъявляются электронным органайзерам, изначально должен быть список с предметами и часами)

## Проектирование задачи

Для отражения предметной области была спроектирована ER-диаграмма, которая описывает ключевые сущности и их взаимосвязи.

## Основные сущности системы:

* **преподаватель;**
* **дисциплина;**
* **план работ;**
* **задание.**

Сущность «Teachers» имеет атрибуты «Name».

Сущность «Disciplines» имеет атрибуты «Name».

Сущность «Work plans» имеет атрибуты «Date».

Сущность «Tasks» имеет атрибуты «Name», «Description», «Issuance Date» «Due Date» и «Status».

Диаграмма сущность-связь представлена на рисунке 2.1.

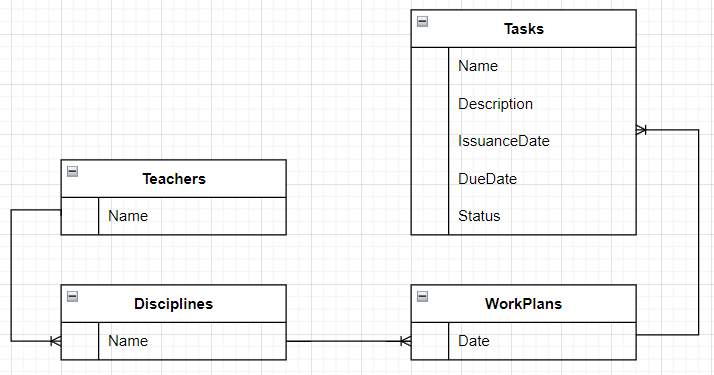


Рисунок 2.1 – Диаграмма сущность-связь

Разрабатываемая система будет предоставлять пользователю следующие возможности:

* авторизоваться в системе с использованием заранее выданного пароля;
* просматривать, добавлять, редактировать и удалять записи о преподавателях;
* просматривать, добавлять, редактировать и удалять сведения о дисциплинах;
* осуществлять поиск по базе данных преподавателей;
* просматривать, добавлять, редактировать и удалять планы работ;
* выдавать задания преподавателям с указанием сроков сдачи;
* редактировать и удалять ранее выданные задания;
* автоматически формировать шаблоны протоколов и других документов в формате Word;
* экспортировать сформированные документы в Word для дальнейшей работы;
* использовать удобный интерфейс с меню, кнопочными формами и панелями инструментов для навигации по системе;
* просматривать справочную информацию по работе с программным обеспечением.

Подробное описание данных функций представлено на диаграмме вариантов использования, деятельности и классов представленных в графической части на листе 1-3.

## Концептуальный прототип

Система меню должна содержать пункты «Преподаватели», «Задания», «Документы», «Протоколы» и «Справка». Основной функцией меню является быстрый доступ к ключевым компонентам системы. Меню должно быть реализовано в виде горизонтальной панели, при нажатии на которую система переключает страницы.

Контекстное меню должно реализовано на страницах и предоставлять быстрый доступ к функциям добавления, удаления и редактирования.

Диалоговое окно должно открываться при добавлении преподавателя для запроса на ввод имени. Также диалоговое окно должно выводиться при добавлении нового предмета для указания названия предмета и преподавателя.

При создании должны быть использованы следующие элементы управления:

* кнопки, для взаимодействия с основными функциями (например для добавления предмета, редактирования заданий);
* текстовые поля, для ввода данных;
* списки выбора, для фильтрации;
* таблицы, для просмотра выданных заданий и работы с базой данных;
* элемент управления для выбора даты, для выбора даты сдачи заданий.

Прототипы страниц представлены на рисунках 2.2 – 2.8.

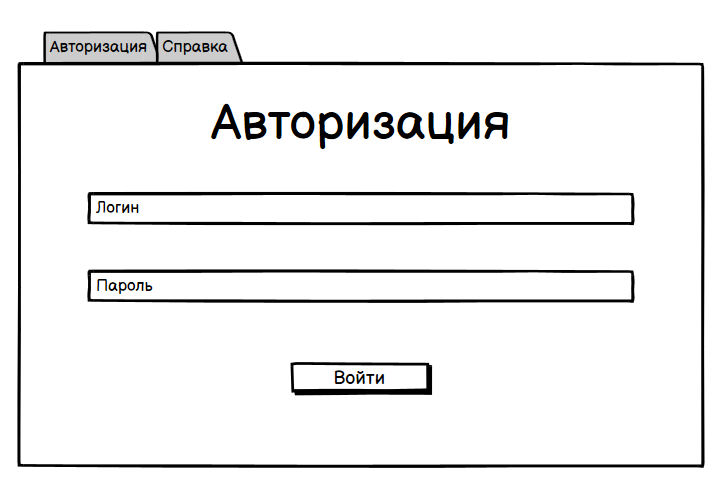


Рисунок 2.2 – Прототип страницы авторизации

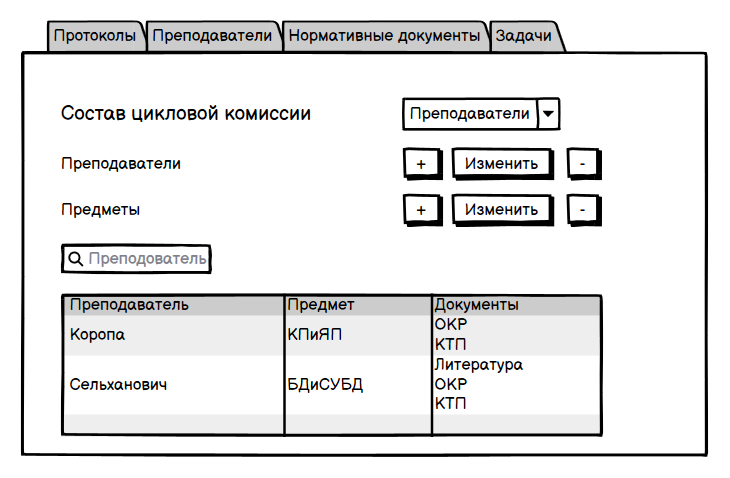


Рисунок 2.3 – Прототип страницы преподавателей

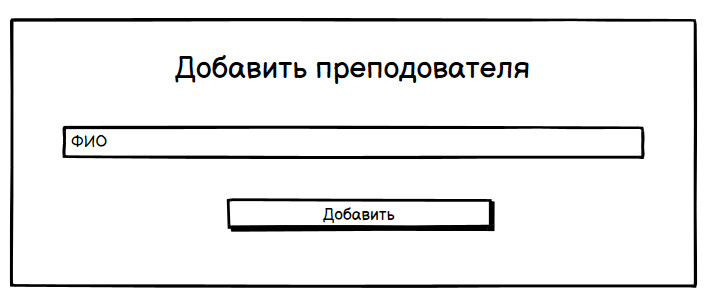


Рисунок 2.4 – Прототип страницы добавления преподавателя

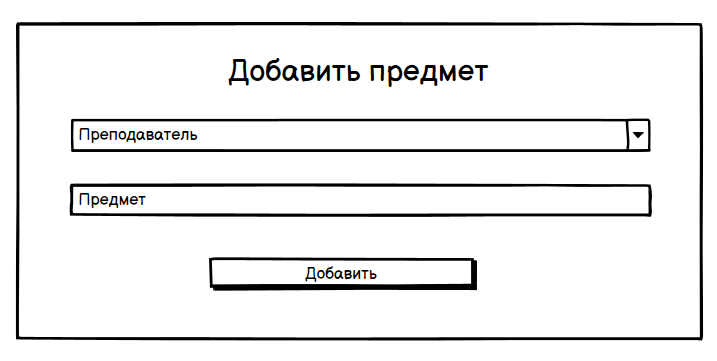


Рисунок 2.5 – Прототип страницы добавления предмета

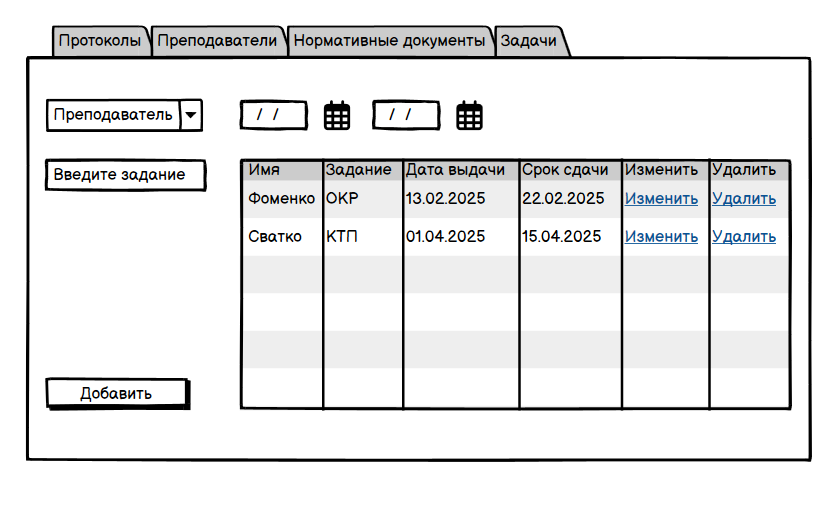


Рисунок 2.6 – Прототип страницы выдачи заданий преподавателям

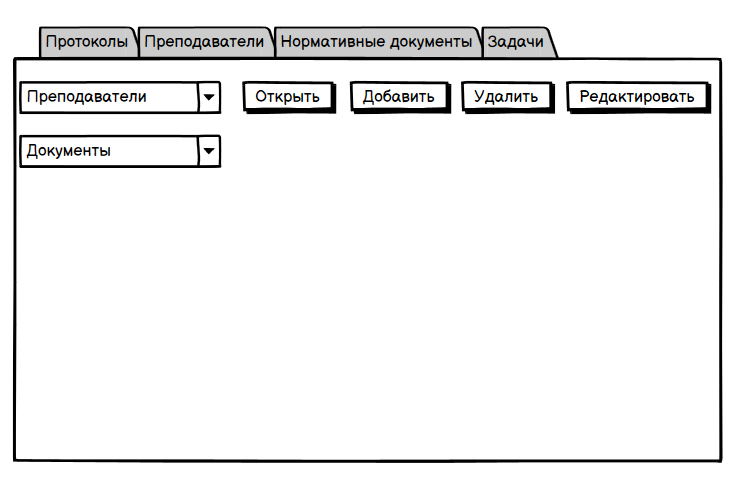


Рисунок 2.7 – Прототип страницы редактирования планов работ

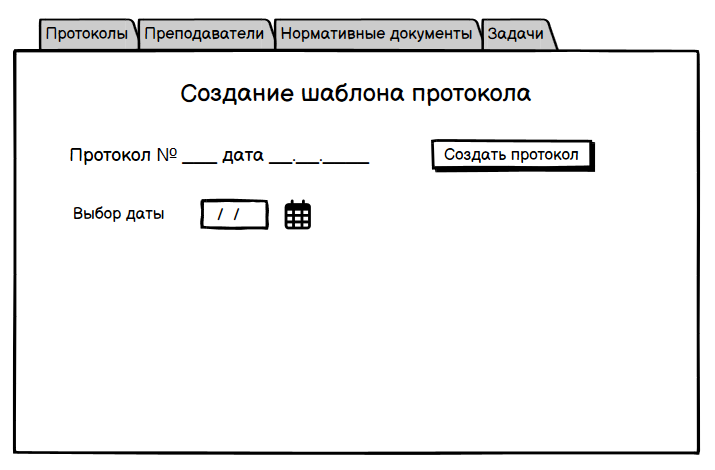


Рисунок 2.8 – Прототип страницы создания шаблона протоколов

# Описание реализации программного средства

## Инструменты разработки и применяемые технологии

При разработке программного обеспечения, электронного органайзера для председателя цикловой комиссии были выбраны следующие технологии:

* среда разработки – Microsoft Visual Studio 2022;
* язык программирования – C#;
* фреймворк – WPF (Windows Presentation Foundation);
* система управления базами данных – Microsoft SQL Server 2022.

Microsoft Visual Studio поддерживает разработку программного обеспечения для различных платформ, включая десктопные приложения, веб-приложения, мобильные решения и облачные сервисы. Среда обеспечивает удобные механизмы работы с кодом, автоматизированную проверку, интеллектуальные подсказки и возможности рефакторинга. [8]

Выбор C# обусловлен его мощными возможностями для создания десктопных приложений, богатой экосистемой библиотек и удобством работы с данными. [5]

WPF используется для разработки современного и функционального графического интерфейса, позволяя создать удобное взаимодействие пользователя с системой, реализовать гибкость дизайна и поддержку сложных визуальных элементов. [9]

Microsoft SQL Server обеспечивает надежное хранение данных, высокую скорость обработки запросов и возможность структурированного анализа информации, что критически важно для работы с данными о преподавательской нагрузке. [6]

C# и WPF активно применяются в разработке корпоративных приложений, таких как финансовые системы, CRM и ERP-системы.

Microsoft SQL Server используется в банковской сфере, медицинских учреждениях и предприятиях для ведения учета данных, управления статистикой и формирования аналитических отчетов.

В данном проекте C# является основным языком для логики обработки данных, анализа информации и взаимодействия с базой данных. WPF применяется для отображения расписания, формирования отчетов и удобного интерфейса взаимодействия преподавателя с программой. Microsoft SQL Server служит хранилищем данных о преподавателях, предметах, заданиях и планах работ.

Выбор данных технологий полностью оправдан задачами проекта: требуется интерактивный пользовательский интерфейс, надежная база данных и эффективные механизмы учета и отчетности. Использование C#, WPF и Microsoft SQL Server позволяет достичь высокой производительности, удобства работы с данными и расширяемости функционала приложения.

## Порядок авторизации пользователей

В данном программном продукте предусмотрена **роль председателя цикловой комиссии**. Он имеет доступ к следующим функциям:

* загрузка и изменение планов работ;
* управление базой данный (добавление, удаление и редактирование информации об преподавателях, предметах и заданиях);
* автоматическая генерация протоколов с номером и датой создания.

Процесс авторизации обеспечивает безопасный доступ к функционалу приложения, позволяя единственному пользователю войти в систему с помощью заранее выданного пароля, хранящегося внутри программы. Это гарантирует, что только авторизованный пользователь может получить доступ и управлять данными приложения, исключая возможность несанкционированного доступа или изменений. Аутентификация пользователя происходит путём проверки введённого пароля с сохранённым в программе значением. Для безопасности пароль хранится в хешированном виде.

## Организация данных

В ходе разработки программного средства для хранения данных было принято решение использовать реляционную модель, в которой каждой сущности, рассмотренной на ERD диаграмме, соответствует своя таблица. Она позволяет эффективно управлять информацией электронного органайзера председателя цикловой комиссии.

В основе системы три таблицы:

* таблица «Teachers» – служит для хранения данных о преподавателях. Детальное описание этой таблицы представлено в таблице 3.1;
* таблица «Disciplines» – служит для хранения данных о предметах, которые ведёт преподаватель. Детальное описание этой таблицы представлено в таблице 3.2;

таблица «Tasks» – служит для хранения данных о выданных заданиях преподавателям, дате выдачи и сроке выполнения. Детальное описание этой таблицы представлено в таблице 3.3.

Таблица 3.1 – Описание таблицы Teachers

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название поля | Тип данных | Длина поля | Разрешение значения NULL | Описание поля |
| IDTeacher | int | MAX | Нет | Уникальный идентификатор преподавателя |
| Name | nchar | 30 | Нет | ФИО преподавателя |

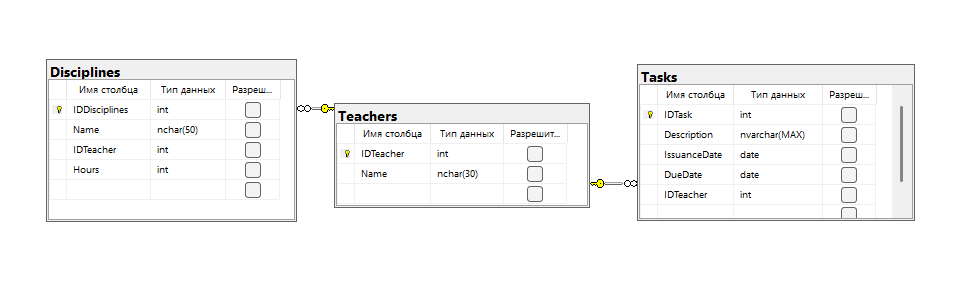
Таблица 3.2 – Описание таблицы Disciplines

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название поля | Тип данных | Длина поля | Разрешение значения NULL | Описание поля |
| IDDisciplines | int | MAX | Нет | Уникальный идентификатор предмета |
| Name | nchar | 50 | Нет | Название предмета |
| Hours | int | MAX | Нет | Количество часов у предмета |
| IDTeacher | int | MAX | Нет | Уникальный идентификатор преподавателя, который ведёт данный предмет |

Таблица 3.3 – Описание таблицы Tasks

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название поля | Тип данных | Длина поля | Разрешение значения NULL | Описание поля |
| IDTask | int | MAX | Нет | Уникальный идентификатор задания |
| Description | nvarchar | MAX | Нет | Описание задания для преподавателя |
| IssuenceDate | date | MAX | Нет | Дата выдачи задания |
| DueDate | date | MAX | Нет | Дата сдачи задания |
| IDTeacher | int | MAX | Нет | Уникальный идентификатор преподавателя, который ведёт данный предмет |

Схема базы данных представлена на рисунке 3.1.

Рисунок 3.1 – Схема базы данных

## Функции: логическая и физическая организация

В ходе разработки электронного органайзера для председателя цикловой комиссии было разработано множество функций, из которых основными являются функции, представленные ниже.

Функция автоматической генерации шаблона протокола в формате Word. Код функции представлен ниже.

public void ReplacePlaceholders(string templatePath, string outputPath, string number, string date)

{

var wordApp = new Word.Application(); // Создаем новый экземпляр приложения Word;

var document = wordApp.Documents.Open(templatePath); // Открываем документ-шаблон по указанному пути;

FindAndReplace(wordApp, "{{номер}}", number); // Заменяем плейсхолдер "{{номер}}" на значение number;

FindAndReplace(wordApp, "{{дата}}", date); // Заменяем плейсхолдер "{{дата}}" на значение date;

document.SaveAs2(outputPath); // Сохраняем изменённый документ по указанному пути;

document.Close(); // Закрываем документ;

wordApp.Quit(); // Закрываем приложение Word;

var process = new Process(); // Создаем новый процесс для открытия файла;

process.StartInfo = new ProcessStartInfo

{

FileName = outputPath, // Указываем путь к файлу для открытия;

UseShellExecute = true // Используем оболочку Windows для запуска файла;

};

process.Start(); // Запускаем процесс (открываем документ);

}

private void FindAndReplace(Word.Application wordApp, string findText, string replaceText)

{

wordApp.Selection.Find.ClearFormatting(); // Очищаем форматирование поиска;

wordApp.Selection.Find.Text = findText; // Устанавливаем текст для поиска;

wordApp.Selection.Find.Replacement.ClearFormatting(); // Очищаем форматирование замены;

wordApp.Selection.Find.Replacement.Text = replaceText; // Устанавливаем текст для замены;

object replaceAll = Word.WdReplace.wdReplaceAll; // Задаем замену всех вхождений;

wordApp.Selection.Find.Execute(

Replace: ref replaceAll, // Выполняем замену всех вхождений;

Forward: true, // Ищем в направлении вперед;

Wrap: Word.WdFindWrap.wdFindContinue // Продолжаем поиск с начала документа после достижения конца;

);

}

private void AddProtocolBtn\_Click(object sender, RoutedEventArgs e)

{

string Unumber = ProtocolNum.Text; // Получаем номер протокола из текстового поля;

string Udate = ProtocolDate.Text; // Получаем дату протокола из текстового поля;

string outputDir = string.Format(@"C:\Papka\Протокол\_№{0}\_{1}.docx", Unumber, Udate); // Формируем путь для сохранения нового документа;

var replacer = new WordReplacer(); // Создаем экземпляр класса для замены плейсхолдеров;

replacer.ReplacePlaceholders(

templatePath: @"C:\Users\LocalUSer\Desktop\Подопытный.docx", // Путь к шаблону документа;

outputPath: outputDir, // Путь для сохранения нового документа;

number: Unumber, // Значение для замены плейсхолдера {{номер}};

date: Udate // Значение для замены плейсхолдера {{дата}};

);

}

Функция для редактирования планов работ. Код функции представлен ниже.

foreach (var d in teacher) // Проходим по каждому преподавателю из коллекции teacher;

{

string path = Path.Combine(AppDomain.CurrentDomain.BaseDirectory, "PlansWork"); // Формируем базовый путь к папке "PlansWork" в директории приложения;

path += $"\\{d.TrimEnd()}"; // Добавляем к пути имя преподавателя с удалением пробелов справа;

if (!Directory.Exists(path)) // Проверяем, существует ли такая папка;

{

Directory.CreateDirectory(path); // Если папка не существует, создаем её;

}

}

List<string> MyStringList = new List<string> { "ИТК", "ОКР", "Литература", "Учебная программа" }; // Создаем список с названиями документов;

Documents.ItemsSource = MyStringList; // Привязываем список к элементу управления Documents для отображения;

foreach (var d in teacher) // Для каждого преподавателя из коллекции teacher;

{

foreach(var t in MyStringList) // Для каждого типа документа из списка MyStringList;

{

string path = Path.Combine(AppDomain.CurrentDomain.BaseDirectory, "PlansWork"); // Формируем базовый путь к папке "PlansWork";

path += $"\\{d.TrimEnd()}\\{t.TrimEnd()}"; // Добавляем к пути имя преподавателя и тип документа, удаляя пробелы справа;

if (!Directory.Exists(path)) // Проверяем, существует ли такая папка;

{

Directory.CreateDirectory(path); // Если папка не существует, создаем её;

}

}

}

private void Open\_Click(object sender, RoutedEventArgs e) // Обработчик события нажатия кнопки "Открыть";

{

string path = Path.Combine(AppDomain.CurrentDomain.BaseDirectory, "PlansWork"); // Формируем базовый путь к папке "PlansWork";

if ((TeacherDoc.SelectedItem != null && Documents.SelectedItem != null) || (TeacherDoc.SelectedItem != null && Documents.SelectedItem == null)) // Проверяем, выбран ли преподаватель и, возможно, документ;

{

if (TeacherDoc.SelectedItem != null) // Если выбран преподаватель;

{

path += "\\" + TeacherDoc.SelectedItem.ToString().TrimEnd(); // Добавляем имя преподавателя к пути;

}

if (Documents.SelectedItem != null) // Если выбран документ;

{

path += "\\" + Documents.SelectedItem.ToString(); // Добавляем название документа к пути;

}

Process.Start(path); // Открываем папку или файл по сформированному пути;

}

else

{

System.Windows.MessageBox.Show("Выберите преподавателя."); // Если преподаватель не выбран, выводим сообщение об ошибке;

}

}

private void Delete\_Click(object sender, RoutedEventArgs e) // Обработчик события нажатия кнопки "Удалить";

{

if (FileBox.SelectedItem != null) // Проверяем, выбран ли файл для удаления;

{

string path = Path.Combine(AppDomain.CurrentDomain.BaseDirectory, "PlansWork"); // Формируем базовый путь к папке "PlansWork";

path += "\\" + TeacherDoc.SelectedItem.ToString().TrimEnd() + "\\" + Documents.SelectedItem.ToString() + "\\" + FileBox.SelectedItem.ToString(); // Формируем полный путь к выбранному файлу;

if (System.Windows.MessageBox.Show("Вы точно хотите удалить данный файл?", "Вопрос", MessageBoxButton.YesNo, MessageBoxImage.Question) == MessageBoxResult.Yes) // Запрашиваем подтверждение удаления у пользователя;

{

File.Delete(path); // Удаляем файл;

System.Windows.MessageBox.Show("Файл был успешно удалён."); // Сообщаем об успешном удалении;

}

path = Path.Combine(AppDomain.CurrentDomain.BaseDirectory, "PlansWork"); // Формируем путь к папке с документами преподавателя;

path += "\\" + TeacherDoc.SelectedItem.ToString().TrimEnd() + "\\" + Documents.SelectedItem.ToString(); // Добавляем имя преподавателя и тип документа;

var files = Directory.GetFiles(path); // Получаем список файлов в указанной папке;

FileBox.ItemsSource = files.Select(f => new FileInfo(f).Name).ToList(); // Обновляем список файлов в интерфейсе;

}

}

private void DocChange\_Change(object sender, SelectionChangedEventArgs e) // Обработчик изменения выбора в списках преподавателей или документов;

{

if (TeacherDoc.SelectedItem != null && Documents.SelectedItem != null) // Проверяем, что выбраны и преподаватель, и документ;

{

string path = Path.Combine(AppDomain.CurrentDomain.BaseDirectory, "PlansWork"); // Формируем базовый путь к папке "PlansWork";

path += "\\" + TeacherDoc.SelectedItem.ToString().TrimEnd() + "\\" + Documents.SelectedItem.ToString(); // Добавляем имя преподавателя и тип документа;

var files = Directory.GetFiles(path); // Получаем список файлов в указанной папке;

FileBox.ItemsSource = files.Select(f => new FileInfo(f).Name).ToList(); // Обновляем список файлов в интерфейсе;

}

}

Функция фильтрации преподавателей по фамилии и очищение таблицы с фильтрами. Код функции представлен ниже.

private void ApplyFilterBtn\_Click(object sender, RoutedEventArgs e)

{

if(FilterComboBox.SelectedItem != null) // Проверяем, выбран ли элемент в фильтре;

{

var item = FilterComboBox.SelectedItem.ToString(); // Получаем выбранное имя преподавателя;

// Находим ID преподавателя по имени из базы данных;

var teachId = KBPClassBetaEntities1.GetContext().Teachers.FirstOrDefault(t => t.Name == item).IDTeacher;

var list = new List<SubjectToTeacher>();

// Получаем все дисциплины, связанные с выбранным преподавателем, и формируем список для отображения;

foreach(var i in KBPClassBetaEntities1.GetContext().Disciplines.Where(d => d.IDTeacher == teachId).ToList())

{

list.Add(new SubjectToTeacher(i));

}

TeacherGrid.ItemsSource = list; // Обновляем источник данных таблицы с отфильтрованными данными;

}

}

private void ClearFiltersBtn\_Click(object sender, RoutedEventArgs e)

{

var list = new List<SubjectToTeacher>(); // Загружаем все дисциплины без фильтрации и формируем список для отображения;

foreach (var i in KBPClassBetaEntities1.GetContext().Disciplines.ToList())

{

list.Add(new SubjectToTeacher(i));

}

TeacherGrid.ItemsSource = list; // Обновляем источник данных таблицы, показывая все данные;}

Все остальные функции представлены в приложении А.

## Входные и выходные данные

В качестве входных данных приложение принимает введённые вручную номер и дату протокола заседания цикловой комиссии.

В качестве выходных данных выступает отчёт, который имеет следующую структуру:

* заголовок, в котором указано наименование учреждения;
* название документа, номер и дата заседания;
* состав цикловой комиссии;
* повестка дня;
* основные разделы протокола.

Пример такого отчёта представлен в приложении Б на рисунке Б.1.

## Функциональное тестирование

Функциональное тестирование играет ключевую роль в обеспечении качества программного обеспечения, проверяя его соответствие заявленным требованиям. Оно проводится для выявления ошибок в работе системы, тестирования ее бизнес-логики, оценки удобства взаимодействия с пользователем и гарантии стабильности перед внедрением. В ходе функционального тестирования проверяются основные функции приложения, его корректность при обработке данных, взаимодействие модулей и реакция на некорректные входные данные.

Для проверки корректности работы приложения были разработаны тест-кейсы, которые представлены в таблице 3.4.

Таблица 3.4 – Тест-кейсы

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Номер | Описание | Шаги | Ожидаемый результат | Фактический результат |
| Т1 | Проверка добавления преподавателей | 1. Открыть приложение; 2. Ввести пароль «13022007»; 3. Нажать на кнопку «Войти»; 4. Нажать на кнопку «Преподаватели»; 5. Нажать на кнопку «Добавить»; 6. Ввести фамилию и инициалы преподавателя в поле для ввода; 7. Нажать на кнопку «Добавить»; | Приложение должно добавить преподавателя и отобразить его в таблице | Соответствует ожидаемому, представлен в приложении В на рисунке В.1 |

Продолжение таблицы 3.4

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Номер | Описание | Шаги | Ожидаемый результат | Фактический результат |
| Т2 | Проверка фильтрации преподавателей | 1. Открыть приложение; 2. Ввести пароль «13022007»; 3. Нажать на кнопку «Войти»; 4. Выбрать необходимое ФИО в поле выбора; 5. Нажать на кнопку «Применить». | Приложение должно отобразить отфильтрованную таблицу преподавателей с их предметами | Соответствует ожидаемому, представлен в приложении В на рисунке В.2 |
| Т3 | Проверка фильтрации без выбора преподавателя | 1. Открыть приложение; 2. Ввести пароль «13022007»; 3. Нажать на кнопку «Войти»; 4. Нажать на кнопку «Применить фильтр». | Приложение должно оповестить об обязательном выборе преподавателя для фильтрации | Соответствует ожидаемому, представлен в приложении В на рисунке В.3 |
| Т4 | Проверка выдачи заданий преподавателям | 1. Открыть приложение; 2. Ввести пароль «13022007»; 3. Нажать на кнопку «Войти»; 4. Перейти на страницу «Задания»; 5. Выбрать ФИО в поле выбора; 6. Ввести дату выдачи в поле даты; 7. Ввести срок сдачи во второе поле даты; 8. Ввести текст задания в поле ввода; 9. Нажать на кнопку «Добавить». | Приложение должно отобразить созданное задание в таблице | Соответствует ожидаемому, представлен в приложении В на рисунке В.4 |
| Т5 | Проверка выдачи заданий преподавателям без ввода задания | 1. Открыть приложение; 2. Ввести пароль «13022007»; 3. Нажать на кнопку «Войти»; 4. Перейти на страницу «Задания»; 5. Выбрать ФИО в поле выбора; 6. Ввести дату выдачи в поле даты; | Приложение должно отобразить сообщение о требовании ввести задание | Соответствует ожидаемому, представлен в приложении В на рисунке В.5 |
| Продолжение таблицы 3.4 | | | | |
| Номер | Описание | Шаги | Ожидаемый результат | Фактический результат |
|  |  | 1. Ввести срок сдачи во второе поле даты; 2. Нажать на кнопку «Добавить». |  |  |
| Т6 | Проверка выдачи заданий преподавателям без ввода срока сдачи | 1. Открыть приложение; 2. Ввести пароль «13022007»; 3. Нажать на кнопку «Войти»; 4. Перейти на страницу «Задания»; 5. Ввести текст задания в поле ввода; 6. Нажать на кнопку «Добавить». | Приложение должно потребовать выбрать дату выдачи и срок сдачи | Соответствует ожидаемому, представлен в приложении В на рисунке В.6 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Т7 | Проверка работы открытия папки документов преподавателя | 1. Открыть приложение; 2. Ввести пароль «13022007»; 3. Нажать на кнопку «Войти»; 4. Перейти на страницу «Документы»; 5. В поле выбора нажать «Сельханович М.А.»; 6. В поле выбора нажать «ОКР»; 7. Нажать на кнопку «Открыть». | Приложение должно открыть на компьютере соответствующую папку с документами | Соответствует ожидаемому, представлен в приложении В на рисунке В.7 |
| Т8 | Проверка работы удаления документов из папки преподавателя | 1. Открыть приложение; 2. Ввести пароль «13022007»; 3. Нажать на кнопку «Войти»; 4. Перейти на страницу «Документы»; 5. В поле выбора нажать «Сельханович М.А.»; 6. В поле выбора нажать «ОКР»; | Приложение должно загрузить в таблицу результаты запроса | Соответствует ожидаемому, представлен в приложении В на рисунке В.8 |

Продолжение таблицы 3.4

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Номер | Описание | Шаги | Ожидаемый результат | Фактический результат |
|  |  | 1. В поле выбора «Доступные документы» выбрать «ОКР №1»; 2. Нажать на кнопку «Удалить». |  |  |
| Т9 | Проверка создания протокола | 1. Открыть приложение; 2. Ввести пароль «13022007»; 3. Нажать на кнопку «Войти»; 4. Перейти на страницу «Протоколы»; 5. В поле ввода указать номер; 6. В поле дата указать дату заседания; 7. Нажать на кнопку «Создать протокол». | Приложение должно открыть шаблон протокола в Word | Соответствует ожидаемому, представлен в приложении В на рисунке В.9 |
| Т10 | Проверка создания протокола без указания даты | 1. Открыть приложение; 2. Ввести пароль «13022007»; 3. Нажать на кнопку «Войти»; 4. Перейти на страницу «Протоколы»; 5. В поле ввода указать номер; 6. Нажать на кнопку «Создать протокол». | Приложение должно открыть шаблон протокола Word c указанием текущей даты | Соответствует ожидаемому, представлен в приложении В на рисунке В.10 |

## Описание справочной системы

Справочная система для электронного органайзера председателя цикловой комиссии была разработана с помощью Dr.Explain – программного обеспечения, предназначенного для создания документации, справочных систем и руководств. Оно позволяет разработчикам и техническим специалистам автоматически генерировать справку для программ, веб-сайтов и мобильных приложений.

Справочная система содержит следующие разделы:

* раздел «Окно аутентификации»;
* раздел «Главное окно».

Раздел «Окно аутентификации» содержит один подраздел – подраздел «Страница авторизации» – содержит подробное описание процесса авторизации.

Раздел «Главное окно» содержит четыре подраздела:

* подраздел «Преподаватели» – содержит подробное описание функций, которые доступны на странице «Преподаватели»;
* подраздел «Задания» – содержит подробное описание функций, которые доступны на странице «Задания»;
* подраздел «Документы» – содержит подробное описание функций, которые доступны на странице «Документы»;
* подраздел «Протоколы» – содержит подробное описание функций, которые доступны на странице «Протоколы».

Открыть справочную систему в приложении можно на любой странице, нажав на кнопку «Справка», которая расположена в правом верхнем углу приложения.

Содержание справочной системы представлено на рисунках 3.2 – 3.3.



Рисунок 3.2 – Начальная страница справочной системы

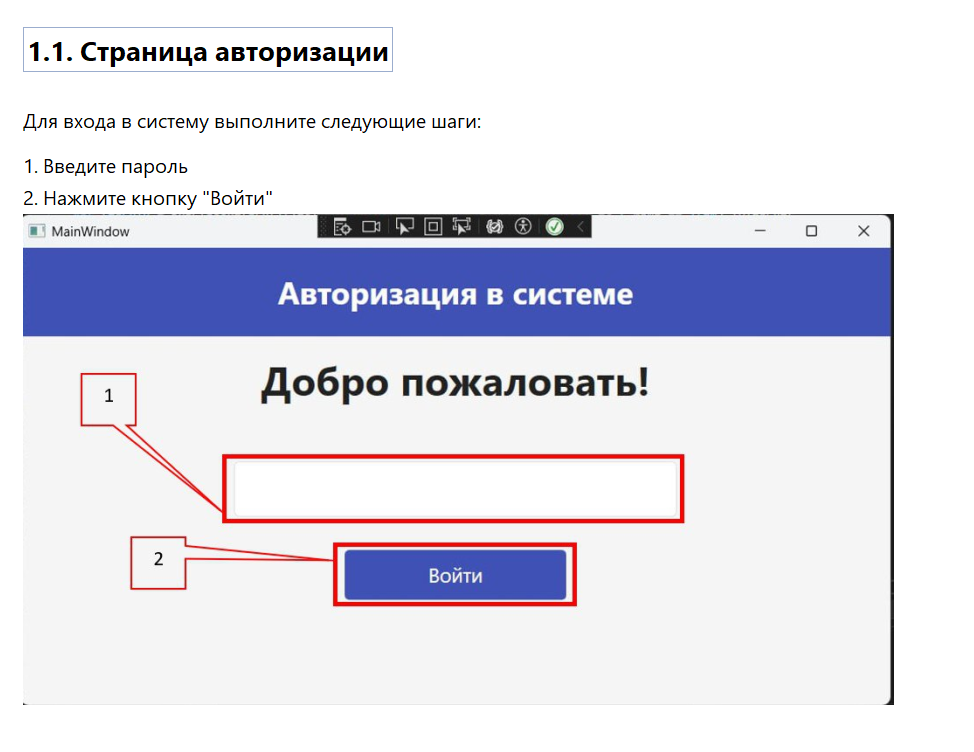


Рисунок 3.3 – Содержание справочной системы

# Применение

## Назначение программного средства

Разработанное программное средство (ПС) получило название «KBPOrganizer». Оно предназначено для упрощения и автоматизации управленческих задач председателя цикловой комиссии ЧУО «Колледж бизнеса и права», обеспечивая эффективное ведение базы данных преподавателей, дисциплин и планов работ, а также удобное управление заданиями и документацией.

Основные функции ПС:

функция входа в систему по заранее выданным паролям с обеспечением безопасности данных;

функция ведения базы данных преподавателей, их дисциплин и планов работ с возможностью поиска, добавления, редактирования и удаления записей;

функция выдачи заданий преподавателям с установлением сроков сдачи, включая возможности добавления, редактирования и удаления заданий;

функция редактирования планов работ (добавление, удаление, изменение записей) с удобным доступом к информации;

функция автоматического создания шаблонов протоколов в формате Word для упрощения документооборота;

функция организации лаконичного и интуитивно понятного интерфейса с меню, кнопочными формами и панелями инструментов;

функция справочной системы для помощи пользователям в работе с приложением.

Разработанная система ориентирована на потребности председателя цикловой комиссии ЧУО «Колледж бизнеса и права» и обеспечивает надежность, безопасность и удобство эксплуатации в рамках учебного процесса данного учреждения.

## Условия применения

Для корректной работы программного средства на компьютере пользователя должно быть установлено следующее ПО:

* операционная система «Windows 7» и выше;
* программа «Microsoft Word» 2016 года выпуска и позже;
* система управления базами данных «Microsoft SQL Server» 2022 года;
* среда выполнения .NET Framework версии 4.7.2 и выше;
* средство установки MSI Installer.

Чтобы установить программу необходимо запустить установочный файл «KBPOrganizer.msi». После его запуска откроется мастер установки и далее необходимо следовать его инструкциям.

При первом запуске программа автоматически развернёт базу данных на компьютере пользователя и после этого можно будет полноценно пользоваться установленным приложением.

# Заключение

Разработка электронного органайзера для председателя цикловой комиссии ЧУО «Колледж бизнеса и права» была направлена на автоматизацию и упрощение управления учебным процессом, а также на повышение эффективности работы комиссии. В ходе реализации проекта была создана система, обеспечивающая безопасный вход по заранее выданным паролям, что гарантирует защиту данных и разграничение доступа пользователей.

Важной частью функционала стало ведение базы данных преподавателей, их дисциплин и планов работ с возможностью удобного поиска, добавления, редактирования и удаления записей. Реализована возможность выдачи заданий преподавателям с установлением сроков сдачи и соответствующими функциями управления этими заданиями. Для удобства работы с планами работ создан интуитивно понятный интерфейс, позволяющий редактировать, добавлять и удалять записи, обеспечивая быстрый доступ к необходимой информации.

Особое внимание уделено автоматизации создания протоколов в формате Word на основе шаблонов, что значительно упрощает процесс документооборота и снижает вероятность ошибок при оформлении документации. Интерфейс программного средства выполнен в лаконичном стиле с использованием меню, кнопочных форм и панелей инструментов, что повышает удобство и скорость работы пользователя. Дополнительно реализована справочная система, помогающая быстро освоить функционал приложения.

Разработанное программное средство полностью соответствует поставленным требованиям, обеспечивая надежность, безопасность и удобство эксплуатации. Все заявленные функции успешно реализованы, что позволяет существенно сократить время на организационные задачи и повысить качество управления учебным процессом в колледже.

Преимущества данного программного средства:

* обеспечение безопасности данных посредством авторизации по заранее выданным паролям;
* комплексное ведение базы данных преподавателей, дисциплин и планов работ с широкими возможностями управления;
* автоматизация выдачи и контроля выполнения заданий преподавателями;
* удобный и интуитивно понятный интерфейс с лаконичным дизайном;
* автоматическое формирование протоколов в формате Word для упрощения документации;
* наличие встроенной справочной системы для поддержки пользователей.

Недостатки данного программного средства:

* Ограниченная область применения — система разработана специально для нужд ЧУО «Колледж бизнеса и права»;
* Для работы с протоколами требуется наличие установленного Microsoft Word.

Электронный органайзер значительно повышает эффективность работы председателя цикловой комиссии, снижает нагрузку на администрацию и способствует более качественному ведению учебной документации.

# Список использованных источников

1. Багласова, Т.Г. Методические указания по оформлению курсовых проектов, дипломных проектов и отчетов для учащихся специальности 2-40 01 01 «Программное обеспечение информационных технологий» // Т.Г. Багласова, К.О. Якимович и др. – Минск : КБП, 2022. – 41 с.
2. Общие требования к тестовым документам : ГОСТ 2.105-95. – Введ. 01.01.1996. – Минск : Межгос. совет по стандартизации, метрологии и сертификации, 1995. – 84 с.
3. Объектно-ориентированный анализ и проектирование с примерами приложений / Г. Буч [и др.]. – 3-е изд. – М. : ООО «И.Д. Вильямс» , 2008 . – 720 с.
4. Программа и методика испытаний. Требования к содержанию, оформлению и контролю качества : ГОСТ 19.301-2000. – Введ. 01.09.2001. – Минск : Межгос. совет по стандартизации, метрологии и сертификации, 2000. – 14 с.
5. Руководство по C#. [Электронный ресурс]. – Режим доступа: https://metanit.com/sharp/. – Дата доступа: 12.04.2025.
6. Руководство по SQL Server. [Электронный ресурс]. – Режим доступа: https://metanit.com/sql/sqlserver/. – Дата доступа: 10.05.2025.
7. Руководство по SQL. [Электронный ресурс]. – Режим доступа: https://metanit.com/sql/. – Дата доступа: 01.05.2025.
8. Руководство по Visual Studio. [Электронный ресурс]. – Режим доступа: <https://visualstudio.microsoft.com/ru/>. – Дата доступа: 01.04.2025.
9. Руководство по WPF. [Электронный ресурс]. – Режим доступа: https://metanit.com/sharp/wpf/. – Дата доступа: 20.04.2025.
10. Текст программы. Требования к содержанию, оформлению и контролю качества : ГОСТ 19.401-2000. – Введ. 01.09.2001. – Минск : Межгос. совет по стандартизации, метрологии и сертификации, 2000. – 16 с.

# ПРИЛОЖЕНИЕ А

**(обязательное)**

**Текст программных модулей**

//Код файла MainWindow.cs

public partial class MainWindow : Window

{

public MainWindow()

{

InitializeComponent();

}

private void PasswordBox\_Click(object sender, MouseButtonEventArgs e)

{

PasswordBox.Password = "";

}

private void EnterBtn\_Click(object sender, RoutedEventArgs e)

{

if(PasswordBox.Password =="13022007")

{

new ProgramMain().Show();

this.Close();

}

else

{

MessageBox.Show("Неверный пароль");

}

}

}

//Код файла ProgramMain.cs

public partial class ProgramMain : System.Windows.Window

{

class WordReplacer

{

public void ReplacePlaceholders(string templatePath, string outputPath, string number, string date)

{

var wordApp = new Word.Application();

var document = wordApp.Documents.Open(templatePath);

FindAndReplace(wordApp, "{{номер}}", number);

FindAndReplace(wordApp, "{{дата}}", date);

document.SaveAs2(outputPath);

document.Close();

wordApp.Quit();

var process = new Process();

process.StartInfo = new ProcessStartInfo

{

FileName = outputPath,

UseShellExecute = true

};

process.Start();

}

private void FindAndReplace(Word.Application wordApp, string findText, string replaceText)

{

wordApp.Selection.Find.ClearFormatting();

wordApp.Selection.Find.Text = findText;

wordApp.Selection.Find.Replacement.ClearFormatting();

wordApp.Selection.Find.Replacement.Text = replaceText;

object replaceAll = Word.WdReplace.wdReplaceAll;

wordApp.Selection.Find.Execute(

Replace: ref replaceAll,

Forward: true,

Wrap: Word.WdFindWrap.wdFindContinue

);

}

}

public ProgramMain()

{

InitializeComponent();

List<String> teacher = new List<String>();

teacher = KBPClassBetaEntities1.GetContext().Teachers.Select(n => n.Name).ToList();

taskForTeacher = new ObservableCollection<TaskForTeacher>();

var taskLoad = KBPClassBetaEntities1.GetContext().Tasks.ToList();

foreach (var d in taskLoad)

{

taskForTeacher.Add(new TaskForTeacher(d));

}

TaskGrid.ItemsSource = taskForTeacher;

TeacherComboBox.ItemsSource = teacher;

TeacherDoc.ItemsSource = teacher;

FilterComboBox.ItemsSource = teacher;

foreach (var d in teacher)

{

string path = Path.Combine(AppDomain.CurrentDomain.BaseDirectory, "PlansWork");

path += $"\\{d.TrimEnd()}";

if (!Directory.Exists(path))

{

Directory.CreateDirectory(path);

}

}

List<string> MyStringList = new List<string> { "ИТК", "ОКР", "Литература","Учебная программа" };

Documents.ItemsSource = MyStringList;

foreach (var d in teacher)

{

foreach(var t in MyStringList)

{

string path = Path.Combine(AppDomain.CurrentDomain.BaseDirectory, "PlansWork");

path += $"\\{d.TrimEnd()}\\" + $"{t.TrimEnd()}";

if (!Directory.Exists(path))

{

Directory.CreateDirectory(path);

}

}

}

disciplines = new ObservableCollection<SubjectToTeacher>();

var disciplinesDistinct = KBPClassBetaEntities1.GetContext().Disciplines.ToList();

foreach (var d in disciplinesDistinct)

{

disciplines.Add(new SubjectToTeacher(d));

}

TeacherGrid.ItemsSource = disciplines;

}

static ObservableCollection<SubjectToTeacher> disciplines;

static ObservableCollection<TaskForTeacher> taskForTeacher;

private void TeacherPageBtn\_Click(object sender, RoutedEventArgs e)

{

TeacherPage teacherPage = new TeacherPage();

teacherPage.Show();

this.Close();

}

private void SubjectsBtn\_Click(object sender, RoutedEventArgs e)

{

SubjectPage subjectsPage = new SubjectPage();

subjectsPage.Show();

this.Close();

}

private void AddTaskBtn\_Click(object sender, RoutedEventArgs e)

{

string teacher = TeacherComboBox.SelectedItem.ToString();

DateTime issuanseD = IssuanceDate.SelectedDate.Value;

DateTime dueD = DueDate.SelectedDate.Value;

Tasks tasks = new Tasks();

tasks.Description = DescriptionTextBox.Text;

tasks.IssuanceDate= issuanseD;

tasks.DueDate= dueD;

tasks.IDTeacher = KBPClassBetaEntities1.GetContext().Teachers.FirstOrDefault(t => t.Name == teacher).IDTeacher;

KBPClassBetaEntities1.GetContext().Tasks.Add(tasks);

try

{

KBPClassBetaEntities1.GetContext().SaveChanges();

}

catch (DbEntityValidationException ex)

{

foreach (var error in ex.EntityValidationErrors)

{

foreach (var validationError in error.ValidationErrors)

{

System.Windows.MessageBox.Show($"{validationError.PropertyName}\n{validationError.ErrorMessage}");

}

}

throw;

}

taskForTeacher = new ObservableCollection<TaskForTeacher>();

var TaskAdd = KBPClassBetaEntities1.GetContext().Tasks.ToList();

foreach (var d in TaskAdd)

{

taskForTeacher.Add(new TaskForTeacher(d));

}

TaskGrid.ItemsSource = taskForTeacher;

}

private void ChangeBtn\_Click(object sender, RoutedEventArgs e)

{

var buttom = sender as Button;

var curTask = buttom?.DataContext as TaskForTeacher;

EditTaskPage addedit = new EditTaskPage(curTask);

if (addedit.ShowDialog() == true)

{

taskForTeacher = new ObservableCollection<TaskForTeacher>();

var taskReload = KBPClassBetaEntities1.GetContext().Tasks.ToList();

foreach (var d in taskReload)

{

taskForTeacher.Add(new TaskForTeacher(d));

}

TaskGrid.ItemsSource = taskForTeacher;

}

}

private void DeleteBtn\_Click(object sender, RoutedEventArgs e)

{

if (System.Windows.MessageBox.Show($"Вы точно хотите удалить данный предмет?", "Внимание", MessageBoxButton.YesNo, MessageBoxImage.Question) == MessageBoxResult.Yes)

{

try

{

var but = sender as Button;

var tsk = but?.DataContext as TaskForTeacher;

var r = KBPClassBetaEntities1.GetContext().Tasks.FirstOrDefault(c => c.IDTask == tsk.id);

KBPClassBetaEntities1.GetContext().Tasks.Remove(r);

KBPClassBetaEntities1.GetContext().SaveChanges();

taskForTeacher = new ObservableCollection<TaskForTeacher>();

var taski = KBPClassBetaEntities1.GetContext().Tasks.ToList();

foreach (var t in taski)

{

taskForTeacher.Add(new TaskForTeacher(t));

}

TaskGrid.ItemsSource = taskForTeacher;

}

catch (System.Exception ex)

{

System.Windows.MessageBox.Show(ex.Message.ToString());

}

}

}

private void AddProtocolBtn\_Click(object sender, RoutedEventArgs e)

{

string Unumber = ProtocolNum.Text;

string Udate = ProtocolDate.Text;

string outputDir = string.Format(@"C:\Papka\Протокол\_№{0}\_{1}.docx", Unumber, Udate);

var replacer = new WordReplacer();

replacer.ReplacePlaceholders(

templatePath: @"C:\Users\LocalUSer\Desktop\Подопытный.docx",

outputPath: outputDir,

number: Unumber,

date: Udate

);

}

private void ViewTeachersBtn\_Click(object sender, RoutedEventArgs e)

{

string filePath = "C:\\Users\\LocalUSer\\Desktop\\Курсовая зачем мне вообще это надо\\Состав ЦК.docx";

try

{

Process.Start(new ProcessStartInfo()

{

FileName = filePath,

UseShellExecute = true

});

}

catch (Exception ex)

{

System.Windows.MessageBox.Show("Не удалось открыть файл: " + ex.Message);

}

}

private void Open\_Click(object sender, RoutedEventArgs e)

{

string path = Path.Combine(AppDomain.CurrentDomain.BaseDirectory, "PlansWork");

if((TeacherDoc.SelectedItem != null && Documents.SelectedItem != null) || (TeacherDoc.SelectedItem != null && Documents.SelectedItem == null))

{

if(TeacherDoc.SelectedItem != null)

{

path += "\\" + TeacherDoc.SelectedItem.ToString().TrimEnd();

}

if (Documents.SelectedItem != null)

{

path += "\\" + Documents.SelectedItem.ToString();

}

Process.Start(path);

}

else

{

System.Windows.MessageBox.Show("Выберите преподавателя.");

}

}

private void Delete\_Click(object sender, RoutedEventArgs e)

{

if(FileBox.SelectedItem != null)

{

string path = Path.Combine(AppDomain.CurrentDomain.BaseDirectory, "PlansWork");

path += "\\" + TeacherDoc.SelectedItem.ToString().TrimEnd() + "\\" + Documents.SelectedItem.ToString() + "\\" + FileBox.SelectedItem.ToString();

if (System.Windows.MessageBox.Show("Вы точно хотите удалить данный файл?", "Вопрос", MessageBoxButton.YesNo, MessageBoxImage.Question) == MessageBoxResult.Yes)

{

File.Delete(path);

System.Windows.MessageBox.Show("Файл был успешно удаён.");

}

path = Path.Combine(AppDomain.CurrentDomain.BaseDirectory, "PlansWork");

path += "\\" + TeacherDoc.SelectedItem.ToString().TrimEnd() + "\\" + Documents.SelectedItem.ToString();

var files = Directory.GetFiles(path);

FileBox.ItemsSource = files.Select(f => new FileInfo(f).Name).ToList();

}

}

private void DocChange\_Change(object sender, SelectionChangedEventArgs e)

{

if (TeacherDoc.SelectedItem != null && Documents.SelectedItem != null)

{

string path = Path.Combine(AppDomain.CurrentDomain.BaseDirectory, "PlansWork");

path += "\\" + TeacherDoc.SelectedItem.ToString().TrimEnd() + "\\" + Documents.SelectedItem.ToString();

var files = Directory.GetFiles(path);

FileBox.ItemsSource = files.Select(f => new FileInfo(f).Name).ToList();

}

}

private void ApplyFilterBtn\_Click(object sender, RoutedEventArgs e)

{

if(FilterComboBox.SelectedItem != null)

{

var item = FilterComboBox.SelectedItem.ToString();

var teachId = KBPClassBetaEntities1.GetContext().Teachers.FirstOrDefault(t => t.Name == item).IDTeacher;

var list = new List<SubjectToTeacher>();

foreach(var i in KBPClassBetaEntities1.GetContext().Disciplines.Where(d => d.IDTeacher == teachId).ToList())

{

list.Add(new SubjectToTeacher(i));

}

TeacherGrid.ItemsSource = list;

}

}

private void ClearFiltersBtn\_Click(object sender, RoutedEventArgs e)

{

var list = new List<SubjectToTeacher>();

foreach (var i in KBPClassBetaEntities1.GetContext().Disciplines.ToList())

{

list.Add(new SubjectToTeacher(i));

}

TeacherGrid.ItemsSource = list;

}

}

//Код файла AddSubjectPage.cs

public partial class AddSubjectPage : Window

{

public AddSubjectPage()

{

InitializeComponent();

List<String> teacher = new List<String>();

teacher = KBPClassBetaEntities1.GetContext().Teachers.Select(n => n.Name).ToList();

Teacher.ItemsSource = teacher;

}

private void Button\_Click(object sender, RoutedEventArgs e)

{

Disciplines subject = new Disciplines();

if(Subject.Text == "")

{

MessageBox.Show("Введите название предмета");

}

else if(int.TryParse(Subject.Text, out int p))

{

MessageBox.Show("Поле не должно содержать цифры");

}

else if (!int.TryParse(Hour.Text, out int t))

{

MessageBox.Show("Введите корректное кол-во часов");

}

else if(Teacher.SelectedItem == null)

{

MessageBox.Show("Заполните поле учителя");

}

else

{

string teach = Teacher.SelectedItem.ToString();

string subjName = Subject.Text;

int countHour = Convert.ToInt32(Hour.Text);

Disciplines discipline = new Disciplines();

discipline.Name = subjName;

discipline.IDTeacher = KBPClassBetaEntities1.GetContext().Teachers.FirstOrDefault(m => m.Name == teach).IDTeacher;

discipline.Hours = countHour;

KBPClassBetaEntities1.GetContext().Disciplines.Add(discipline);

KBPClassBetaEntities1.GetContext().SaveChanges();

this.DialogResult = true;

this.Close();

}

}

}

//Код файла AddTeacherPage.cs

public partial class AddTeacherPage : Window

{

public AddTeacherPage()

{

InitializeComponent();

}

private void Button\_Click(object sender, RoutedEventArgs e)

{

Teachers teacher = new Teachers();

bool f = false;

var teachers = KBPClassBetaEntities1.GetContext().Teachers.ToList();

foreach (var t in teachers)

{

if (t.Name.Trim().CompareTo(TeacherName.Text.TrimEnd()) == 0)

{

f = true;

}

}

if (TeacherName.Text == "")

{

MessageBox.Show("Введите имя преподавателя");

}

else if (int.TryParse(TeacherName.Text, out int p))

{

MessageBox.Show("Поле не должно содержать цифры");

}

else if (f)

{

MessageBox.Show($"Преподаватель {TeacherName.Text} уже добавлен");

}

else

{

teacher.Name = TeacherName.Text.TrimEnd();

KBPClassBetaEntities1.GetContext().Teachers.Add(teacher);

KBPClassBetaEntities1.GetContext().SaveChanges();

this.DialogResult = true;

this.Close();

}

}

}

//Код файла EditSubjectPage.cs

public partial class EditSubjectPage : Window

{

public EditSubjectPage()

{

InitializeComponent();

}

SubjectToTeacher d;

public EditSubjectPage(SubjectToTeacher subject)

{

InitializeComponent();

d = subject;

var tech = KBPClassBetaEntities1.GetContext().Teachers.Select(cc => cc.Name).ToList();

Teacher.ItemsSource = tech;

Subject.Text=subject.Name;

Hours.Text=subject.Hour.ToString();

foreach (var c in Teacher.ItemsSource)

{

if (c.ToString() == subject.idTeacher)

{

Teacher.SelectedItem = c;

break;

}

}

}

private void Button\_Click(object sender, RoutedEventArgs e)

{

string teacherName = Teacher.Text;

string subjc = Subject.Text;

int hours = Convert.ToInt32(Hours.Text);

var disp = KBPClassBetaEntities1.GetContext().Disciplines.FirstOrDefault(c => c.IDDisciplines == d.id);

disp.IDTeacher = KBPClassBetaEntities1.GetContext().Teachers.FirstOrDefault(c => c.Name == teacherName).IDTeacher;

disp.Name = subjc;

disp.Hours = hours;

KBPClassBetaEntities1.GetContext().SaveChanges();

this.DialogResult = true;

this.Close();

}

}

//Код файла EditTaskPage.cs

public partial class EditTaskPage : Window

{

public EditTaskPage()

{

InitializeComponent();

}

TaskForTeacher tft;

public EditTaskPage(TaskForTeacher tasking)

{

InitializeComponent();

tft = tasking;

var tech = KBPClassBetaEntities1.GetContext().Teachers.Select(cc => cc.Name).ToList();

TeacherComboBox.ItemsSource = tech;

IssuanceDate.Text = tasking.issuence;

DueDate.Text = tasking.due;

DescriptionTextBox.Text = tasking.descript;

foreach (var c in TeacherComboBox.ItemsSource)

{

if (c.ToString() == tasking.nameTeacher)

{

TeacherComboBox.SelectedItem = c;

break;

}

}

}

private void AddTaskBtn\_Click(object sender, RoutedEventArgs e)

{

string teacherName = TeacherComboBox.Text;

DateTime issuenseD = IssuanceDate.SelectedDate.Value;

DateTime dueD = DueDate.SelectedDate.Value;

string ttask = DescriptionTextBox.Text;

var tasks = KBPClassBetaEntities1.GetContext().Tasks.FirstOrDefault(c => c.IDTask == tft.id);

tasks.IDTeacher = KBPClassBetaEntities1.GetContext().Teachers.FirstOrDefault(c => c.Name == teacherName).IDTeacher;

tasks.IssuanceDate = issuenseD;

tasks.DueDate = dueD;

tasks.Description = ttask;

KBPClassBetaEntities1.GetContext().SaveChanges();

this.DialogResult = true;

this.Close();

}

}

//Код файла EditTeacherPage.cs

public partial class EditTeacherPage : Window

{

public EditTeacherPage()

{

InitializeComponent();

}

private Teachers teach;

public EditTeacherPage(Teachers teacher)

{

InitializeComponent();

this.teach = teacher;

TeacherName.Text = teacher.Name;

}

private void Button\_Click(object sender, RoutedEventArgs e)

{

string teachName = TeacherName.Text;

Teachers teacher = new Teachers();

teach.Name = teachName;

KBPClassBetaEntities1.GetContext().SaveChanges();

this.DialogResult = true;

this.Close();

}

}

//Код файла SubjectPage.cs

public partial class SubjectPage : Window

{

public SubjectPage()

{

InitializeComponent();

disciplines = new ObservableCollection<SubjectToTeacher>();

var disciplinesDistinct = KBPClassBetaEntities1.GetContext().Disciplines.ToList();

foreach (var d in disciplinesDistinct)

{

disciplines.Add(new SubjectToTeacher(d));

}

SubjectGrid.ItemsSource = disciplines;

}

static ObservableCollection<SubjectToTeacher> disciplines;

private void ExitBtn\_Click(object sender, RoutedEventArgs e)

{

ProgramMain programMain = new ProgramMain();

programMain.Show();

this.Close();

}

private void AddBtn\_Click(object sender, RoutedEventArgs e)

{

AddSubjectPage addSubjectPage = new AddSubjectPage();

if (addSubjectPage.ShowDialog() == true)

{

disciplines = new ObservableCollection<SubjectToTeacher>();

var disciplinesDistinct = KBPClassBetaEntities1.GetContext().Disciplines.ToList();

foreach (var d in disciplinesDistinct)

{

disciplines.Add(new SubjectToTeacher(d));

}

SubjectGrid.ItemsSource = disciplines;

}

}

private void ChangeBtn\_Click(object sender, RoutedEventArgs e)

{

var buttom = sender as Button;

var curSubject = buttom?.DataContext as SubjectToTeacher;

EditSubjectPage addedit = new EditSubjectPage(curSubject);

if (addedit.ShowDialog() == true)

{

disciplines = new ObservableCollection<SubjectToTeacher>();

var disciplinesReload = KBPClassBetaEntities1.GetContext().Disciplines.ToList();

foreach (var d in disciplinesReload)

{

disciplines.Add(new SubjectToTeacher(d));

}

SubjectGrid.ItemsSource = disciplines;

}

}

private void DeleteBtn\_Click(object sender, RoutedEventArgs e)

{

if (MessageBox.Show($"Вы точно хотите удалить данный предмет?", "Внимание", MessageBoxButton.YesNo, MessageBoxImage.Question) == MessageBoxResult.Yes)

{

try

{

var but = sender as Button;

var subj = but?.DataContext as SubjectToTeacher;

var r = KBPClassBetaEntities1.GetContext().Disciplines.FirstOrDefault(c => c.IDDisciplines == subj.id);

KBPClassBetaEntities1.GetContext().Disciplines.Remove(r);

KBPClassBetaEntities1.GetContext().SaveChanges();

MessageBox.Show("Данные удалены");

disciplines = new ObservableCollection<SubjectToTeacher>();

var disciplinesDistinct = KBPClassBetaEntities1.GetContext().Disciplines.ToList();

foreach (var d in disciplinesDistinct)

{

disciplines.Add(new SubjectToTeacher(d));

}

SubjectGrid.ItemsSource = disciplines;

}

catch (System.Exception ex)

{

MessageBox.Show(ex.Message.ToString());

}

}

}

}

//Код файла TeacherPage.cs

public partial class TeacherPage : Window

{

public TeacherPage()

{

InitializeComponent();

TeacherGrid.ItemsSource = KBPClassBetaEntities1.GetContext().Teachers.ToList();

}

private void ExitBtn\_Click(object sender, RoutedEventArgs e)

{

ProgramMain programMain = new ProgramMain();

programMain.Show();

this.Close();

}

private void AddBtn\_Click(object sender, RoutedEventArgs e)

{

AddTeacherPage addTeacherPage = new AddTeacherPage();

if (addTeacherPage.ShowDialog() == true)

{

TeacherGrid.ItemsSource = KBPClassBetaEntities1.GetContext().Teachers.ToList();

}

}

private void ChangeBtn\_Click(object sender, RoutedEventArgs e)

{

var buttom = sender as System.Windows.Controls.Button;

var curTeacher = buttom?.DataContext as Teachers;

EditTeacherPage addedit = new EditTeacherPage(curTeacher);

if (addedit.ShowDialog() == true)

{

TeacherGrid.ItemsSource = KBPClassBetaEntities1.GetContext().Teachers.ToList();

}

}

private void DeleteBtn\_Click(object sender, RoutedEventArgs e)

{

var teacherForRemoving = TeacherGrid.SelectedItems.Cast<Teachers>().ToList();

if (MessageBox.Show($"Вы точно хотите удалить следующие {teacherForRemoving.Count} элементов?", "Внимание",

MessageBoxButton.YesNo, MessageBoxImage.Question) == MessageBoxResult.Yes)

{

try

{

KBPClassBetaEntities1.GetContext().Teachers.RemoveRange(teacherForRemoving);

KBPClassBetaEntities1.GetContext().SaveChanges();

MessageBox.Show("Данные удалены");

TeacherGrid.ItemsSource = KBPClassBetaEntities1.GetContext().Teachers.ToList();

}

catch (System.Exception ex)

{

MessageBox.Show(ex.Message.ToString());

}

}

}

}

# ПРИЛОЖЕНИЕ Б

**(справочное)**

**Формы входных и выходных документов**

КП Т.292005.401 ГЧ

Диаграмма вариантов использования

КБП

У

Т. Контр.

Разраб. Разраб.

Внук В.В.

Провер. Провер.

Коропа Е.Н.

Реценз.

Утверд.

Коропа Е.Н.

Н. Контр.

*РАЗРАБОТКА ЭЛЕКТРОННОГО ОРГАНАЙЗЕРА ДЛЯ ПРЕДСЕДАТЕЛЯ ЦИКЛОВОЙ КОМИССИИ ЧУО "КОЛЛЕДЖ БИЗНЕСА И ПРАВА"*

№ докум.

Подпись

Дата

Изм.

Лист

Масса

Лит.

Масштаб

Лист 1

Листов 3

Инв.№подл.

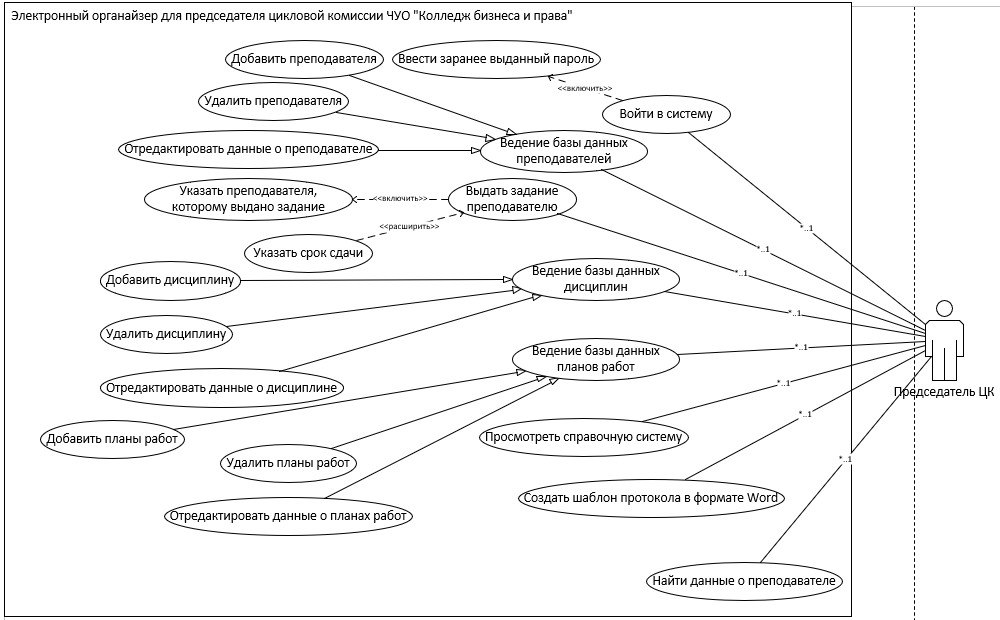
Подп. и дата

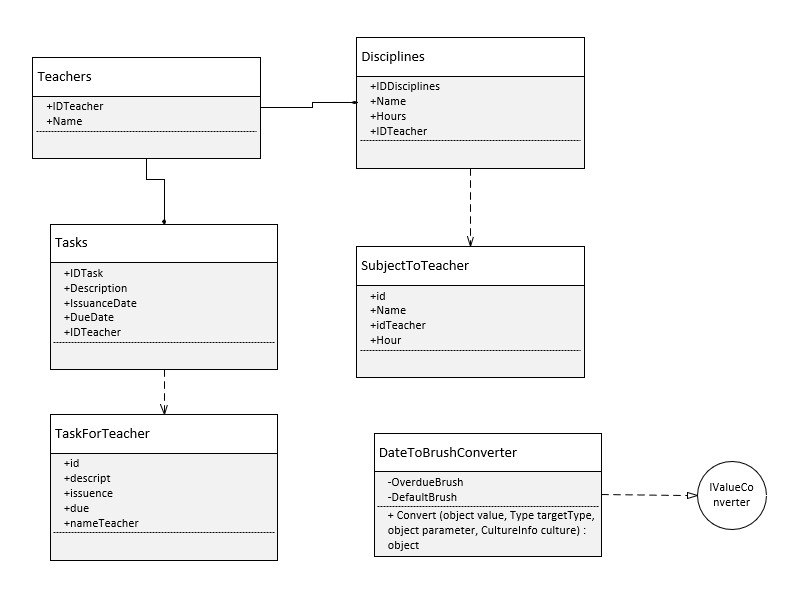
Взам.инв.№

Инв.№дубл.

Подп. и дата

КП Т.292005.401 ГЧ





КП Т.292005.401 ГЧ

Диаграмма классов

КБП

У

Т. Контр.

Разраб. Разраб.

Внук В.В.

Провер. Провер.

Коропа Е.Н.

Реценз.

Утверд.

Коропа Е.Н.

Н. Контр.

*РАЗРАБОТКА ЭЛЕКТРОННОГО ОРГАНАЙЗЕРА ДЛЯ ПРЕДСЕДАТЕЛЯ ЦИКЛОВОЙ КОМИССИИ ЧУО "КОЛЛЕДЖ БИЗНЕСА И ПРАВА"*

№ докум.

Подпись

Дата

Изм.

Лист

Масса

Лит.

Масштаб

Лист 2

Листов 3

Инв.№подл.

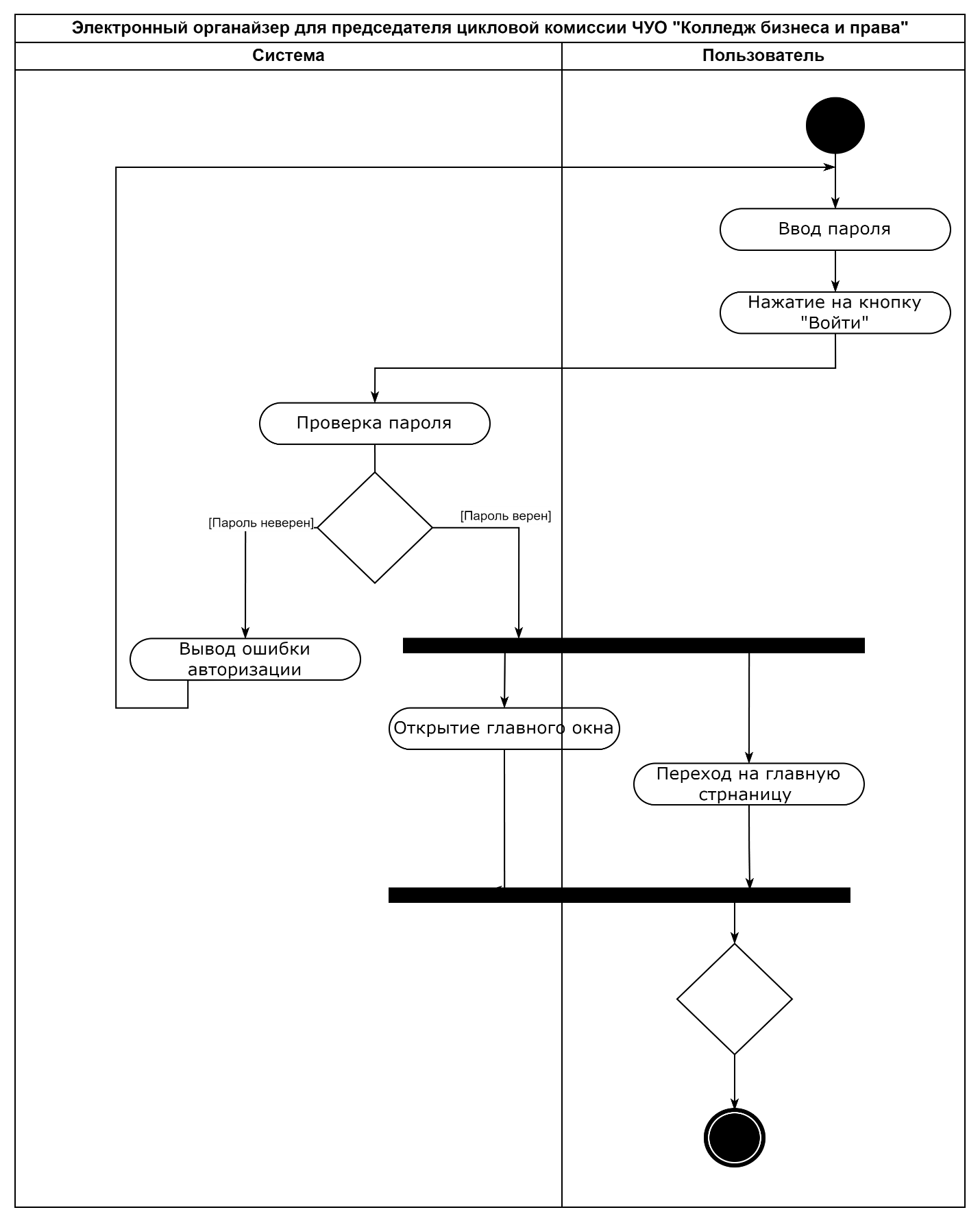
Подп. и дата

Взам.инв.№

Инв.№дубл.

Подп. и дата

КП Т.292005.401 ГЧ



КП Т.292005.401 ГЧ

Диаграмма деятельности

КБП

У

Т. Контр.

Разраб. Разраб.

Внук В.В.

Провер. Провер.

Коропа Е.Н.

Реценз.

Утверд.

Коропа Е.Н.

Н. Контр.

*РАЗРАБОТКА ЭЛЕКТРОННОГО ОРГАНАЙЗЕРА ДЛЯ ПРЕДСЕДАТЕЛЯ ЦИКЛОВОЙ КОМИССИИ ЧУО "КОЛЛЕДЖ БИЗНЕСА И ПРАВА"*

№ докум.

Подпись

Дата

Изм.

Лист

Масса

Лит.

Масштаб

Лист 3

Листов 3

Инв.№подл.

Подп. и дата

Взам.инв.№

Инв.№дубл.

Подп. и дата

КП Т.292005.401 ГЧ