Частное учреждение образования

«Колледж бизнеса и права»

РАЗРАБОТКА СИСТЕМЫ ФОРМИРОВАНИЯ ОТЧЕТОВ ПО ОТРАБОТАННЫМ ЧАСАМ ПРЕПОДАВАТЕЛЕЙ ЧУО «КОЛЛЕДЖ БИЗНЕСА И ПРАВА»

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовому проекту по учебному предмету

«Конструирование программ и языки программирования»

КП Т.292001.401

Руководитель проекта ( Е.Н.Коропа )

Обучающийся ( Д.Р.Буев )

2025

**Этикетка**

для курсовых проектов

**Курсовой проект**

Тема Разработка системы формирования отчётов по отработанным часам преподавателей ЧУО «Колледж бизнеса и права»

КП Т.292001.401

Разработан Буев Д.Р.

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 – Буев ПЗ.docx

Графическая часть –

Папка с проектом – Kursach2.0

Установочный пакет программного средства – setup.exe

Сведения о защите информации: логин отсутствует, пароль отсутствует

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.292001.401 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.*

*Буев Д.Р.*

*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 СИСТЕМЫ ФОРМИРОВАНИЕ ОТЧЕТОВ ПО ОТРАБОТАННЫМ ЧАСАМ ПРЕПОДАВАТЕЛЕЙ 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*

*15*

*55*

*50*

*50*

*55*

*50*

*КБП*

*КБП*

*КБП*

*КБП*

*КБП*

*КБП*

*КБП*

*КБП*

[Введение 4](#_Toc198279003)

[1 Описание задачи 5](#_Toc198279004)

[1.1 Анализ предметной области 5](#_Toc198279005)

[1.2 Постановка задачи 6](#_Toc198279006)

[2 Проектирование системы 7](#_Toc198279007)

[2.1 Требования к приложению 7](#_Toc198279008)

[2.2 Проектирование задачи 7](#_Toc198279009)

[2.3 Концептуальный прототип 8](#_Toc198279010)

[3 Описание реализации программного средства 13](#_Toc198279011)

[3.1 Инструменты разработки и применяемые технологии 13](#_Toc198279012)

[3.2 Порядок авторизации пользователей 13](#_Toc198279013)

[3.3 Организация данных 14](#_Toc198279014)

[3.4 Функции: логическая и физическая организация 16](#_Toc198279015)

[3.5 Входные и выходные данные 19](#_Toc198279016)

[3.6 Функциональное тестирование 20](#_Toc198279017)

[3.7 Описание справочной системы 21](#_Toc198279018)

[4 Применение 22](#_Toc198279019)

[4.1 Назначение программного средства 22](#_Toc198279020)

[4.2 Условия применения 22](#_Toc198279021)

[Заключение 23](#_Toc198279022)

[Список использованных источников 24](#_Toc198279023)

[ПРИЛОЖЕНИЕ А 25](#_Toc198279024)

[ПРИЛОЖЕНИЕ Б 26](#_Toc198279025)

# Введение

В процессе деятельности образовательных учреждений важной задачей является учет и анализ отработанных часов преподавателей. Эффективное решение этой задачи способствует прозрачности учета, оптимизации планирования и формирования отчетной документации. Основной целью разработки является создание системы формирования отчетов, которая автоматизирует процесс учета и упрощает составление отчетов об отработанных часах.

Актуальность проекта обусловлена необходимостью перехода на современные цифровые технологии в образовательной среде. Автоматизация процессов учета позволяет сократить временные затраты, минимизировать ошибки при вводе данных и обеспечить удобство в эксплуатации системы. Таким образом, данная система имеет значимое практическое применение и является важным этапом совершенствования учебной и административной деятельности.

При разработке курсового проекта будет составлена пояснительная записка, содержащая четыре раздела.

В первом разделе «Описание задачи» рассматривается предметная область и формулируется основная задача, которую необходимо решить в рамках разработки. Особое внимание уделяется анализу текущих аналогов, а также периодичности использования системы.

Во втором разделе «Проектирование системы» приводятся требования к функциональности приложения и осуществляется моделирование архитектуры программного средства. Также описывается структура базы данных и представляется концептуальный прототип пользовательского интерфейса.

Третий раздел «Описание реализации программного средства» посвящен выбору инструментов разработки и технологий, описанию структуры данных и реализации ключевых функций. Включены аспекты логической и физической организации данных, тестирование системы, а также описание справочной документации.

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

Заключение представляет собой анализ созданного программного средства, сопоставление поставленной задачи с достигнутым результатом и обзор возможных несоответствий, если таковые имеются.

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 исследуются процессы учёта рабочих часов преподавателей, включающие:

* регистрация преподавателей для авторизованного доступа к системе;
* получение информации (импорт) о расписании занятий преподавателей с официального сайта учреждения;
* учет фактического времени работы, а также возможность редактирования данных для исправления ошибок. Преподаватель либо загружает расписание с сайта колледжа, либо вводит его вручную. В результате данные о проведённых занятиях поступают в систему и сохраняются в базе данных, создавая основу для дальнейшего учёта. Каждое занятие фиксируется в базе данных, включая дату, время, дисциплину и количество часов. Это позволяет системе точно отслеживать преподавательскую нагрузку и формировать отчетность без необходимости ручного подсчета. Когда преподаватель переходит на страницу составления отчёта и нажимает кнопку «Составить отчёт», приложение автоматически анализирует сохранённые данные, подсчитывает количество отработанных часов по каждому предмету за определённый период и формирует отчет. После обработки данных приложение генерирует документ в формате Word, в котором структурировано представлена информация о преподавательской нагрузке.
* составление отчетов о рабочих часах, распределенных по месяцам и предметам. Эти процессы являются ключевыми для административной работы образовательного учреждения и должны быть упрощены с помощью автоматизированной системы.

Промежуточные данные формируются путём автоматического извлечения расписания с сайта колледжа и сохранением его в базе данных для последующей обработки.

Итоговые данные формируются путём подсчёта отработанных часов за каждую дату по каждому предмету, подсчёта общего количества отработанных часов по каждому предмету, подсчёта оставшихся часов каждого предмета и экспорт отчёта в формат Word.

Данное программное средство позволит минимизировать ошибки, увеличить скорость обработки и обеспечит удобный интерфейс для работы с большими объёмами данных.

Целевой аудиторией данной системы являются преподаватели ЧУО «Колледж бизнеса и права».

## Постановка задачи

На основе анализа бизнес-процессов и задач, связанных с учетом и анализом рабочего времени преподавателей, выделены следующие функции, которые подлежат автоматизации в разрабатываемой системе:

* авторизация и регистрация пользователей. Реализация надежного механизма для регистрации новых преподавателей с сохранением конфиденциальности их данных. Авторизация преподавателей с проверкой введенных учетных данных. Возможность изменения пароля для повышения безопасности учетной записи;
* извлечение и управление расписанием. Автоматический парсинг расписания преподавателей с официального сайта колледжа. Сохранение расписания в базе данных для анализа и последующего использования. Интерфейс для просмотра и редактирования расписания с удобной навигацией;
* учет и корректировка отработанных часов. Ввод данных о часах, отработанных преподавателями. Возможность добавления, удаления и изменения введенной информации для обеспечения точности данных. Поддержка фильтрации данных по предмету, преподавателю и дате;
* формирование отчетов. Генерация отчетов об отработанных часах по каждому преподавателю за заданный период. Экспорт отчетов в удобные форматы. Расчет итоговых данных, таких как количество часов, отработанных по предметам;
* интерфейс и справочная система. Лаконичный и удобный интерфейс для взаимодействия с системой, включающий меню, кнопочные формы и панели инструментов. Создание справочной системы для объяснения функционала программы и упрощения работы с системой.

На рынке существуют различные аналоги, обеспечивающие учет рабочего времени преподавателей, например Excel, Google Sheets или CRM-системы Bitrix24 и Asana, но ни один из перечисленных аналогов не соответствует требованиям ЧУО «Колледж бизнеса и права».

Предполагаемая система будет интегрирована с сайтом колледжа для автоматического извлечения расписания, чего не предполагают вышеуказанные системы. Система будет поддерживать подсчёт отработанных часов и экспорт их в Word. Также в системе будет присутствовать функционал для редактирования и фильтрации данных. 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 уникальных учетом данных. 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. Необходим доступ к интернету, а также установленная платформа .NET Framework версии 4.8 или выше.

Для установки программы будет создано установочное средство.

## Проектирование задачи

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

Основные сущности системы:

* преподаватели;
* предметы;
* проведённые часы.

Сущность «Преподаватель» имеет атрибуты «Логин», «Зашифрованный пароль» и «Ссылка» для извлечения расписания с сайта колледжа.

Сущность «Предмет» имеет атрибуты «Название», «Имя преподавателя» и «Общее количество часов».

Сущность «Проведённый час» имеет атрибуты «Название предмета», «Дата» и «Имя преподавателя».

Связи между преподавателем и проведёнными часами, а также между предметом и проведёнными часами позволяет преподавателю проводить по несколько предметов в один день.

Диаграмма сущность-связь представлена на рисунке 2.1.

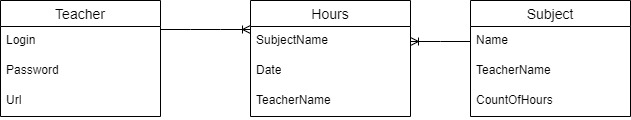


Рисунок 2.1 – Диаграмма сущность-связь

Разрабатываемая система будет представлять возможность пользователю выполнять следующие действия:

* авторизоваться;
* зарегистрироваться;
* автоматически извлечь расписание с сайта колледжа;
* сохранить извлечённую информацию в файл;
* сохранить извлечённую информацию в базу данных;
* просмотреть расписание (посредством нажатия на кнопки < и >);
* изменить пароль;
* изменить ссылку для парсинга расписания;
* сформировать месячный отчёт по отработанным часам, включая подсчёт отработанных часов за каждую дату;
* экспорт отчёта в Word;
* добавить информацию об отработанных часах;
* удалить информацию об отработанных часах;
* изменить информацию об отработанных часах;
* отфильтровать данные об отработанных часах по предмету, дате и группе;
* просмотреть справку.

Подробное описание данных функций представлено на диаграмме вариантов использования, представленной в графической части на листе 1. Остальные диаграммы сюда же

## Концептуальный прототип

Система меню должна содержать пункты «Расписание», «Часы», «Отчёт», «Профиль» 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 учебного часа для запроса на ввод названия предмета, группы, даты и количества часов. Также диалоговое окно должно выводиться при добавлении нового предмета для указания общего количества часов.

При создании должны быть использованы следующие элементы управления:

* кнопки, для взаимодействия с основными функциями (например парнсинг данных сайта, переход по недельному расписанию);
* текстовые поля, для ввода данных;
* списки выбора, для фильтрации;
* таблицы, для просмотра расписания и работы с базой данных;
* элемент управления для выбора даты, для выбора даты проведения предметов.

Прототипы страниц представлены на рисунках 2.2 – 2.8.

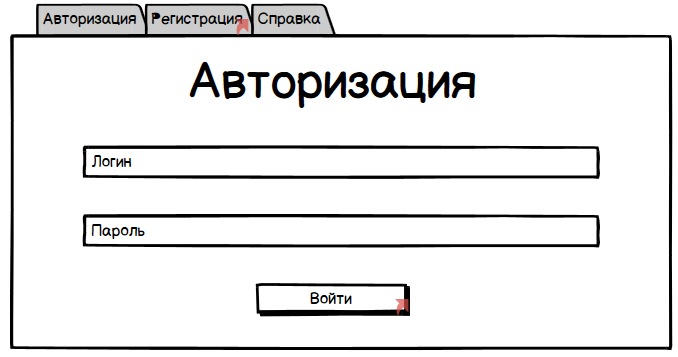


Рисунок 2.2 – Прототип страницы авторизации

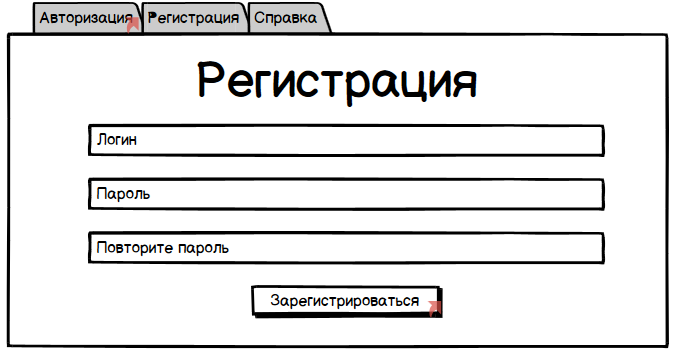


Рисунок 2.3 – Прототип страницы регистрации



Рисунок 2.4 – Прототип страницы расписания

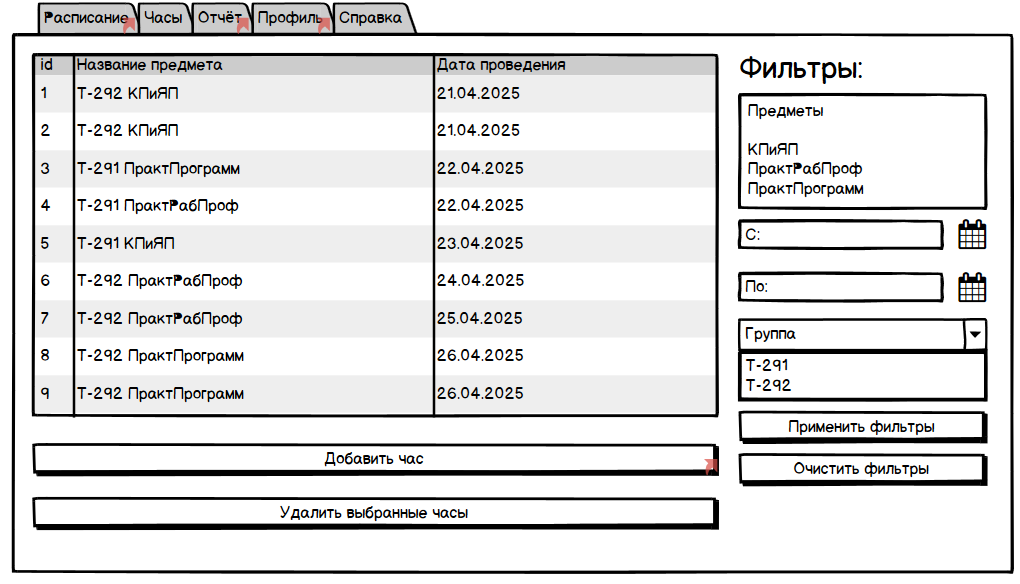


Рисунок 2.5 – Прототип страницы часов

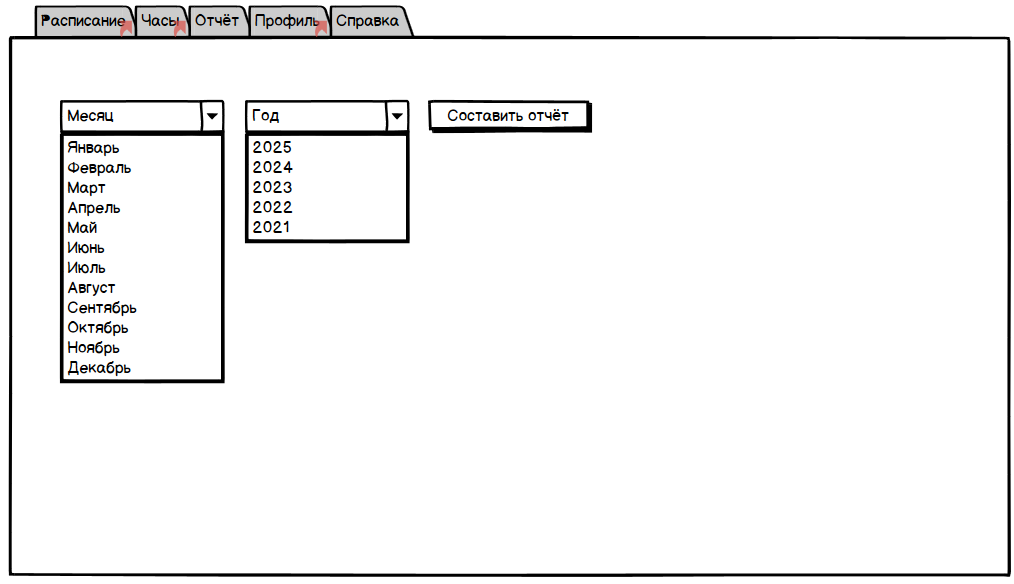


Рисунок 2.6 – Прототип страницы отчёта

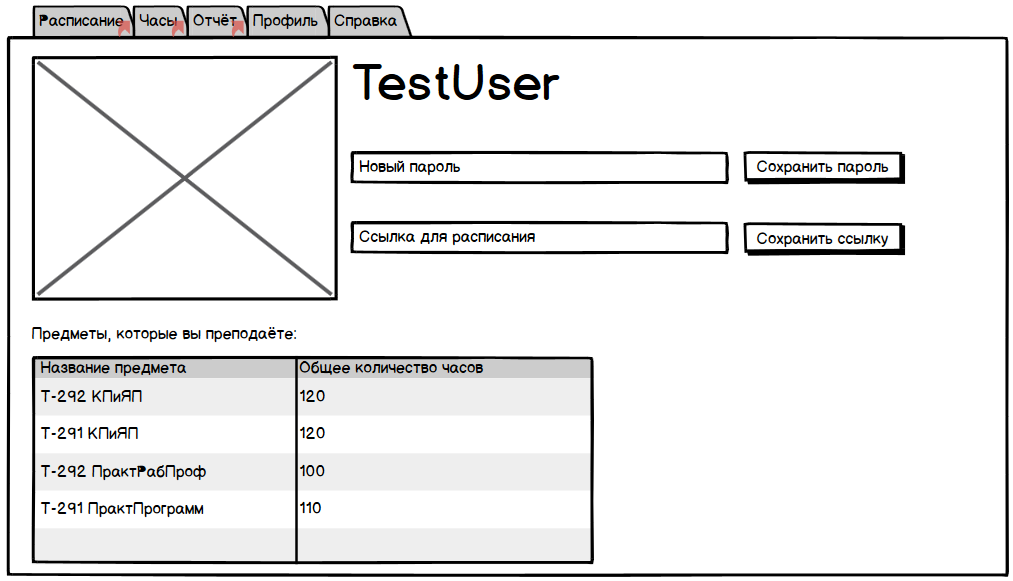


Рисунок 2.7 – Прототип страницы профиля

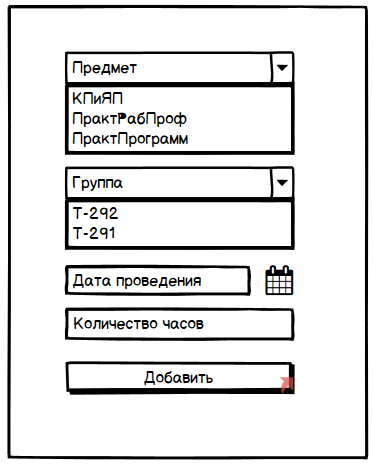


Рисунок 2.8 – Прототип страницы добавления часов

# Описание реализации программного средства

## Инструменты разработки и применяемые технологии

При разработке программного обеспечения для учета отработанных часов преподавателей были выбраны следующие технологии:

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

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.

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

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

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.

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

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. SQL Server служит хранилищем данных о занятиях, преподавательской нагрузке и результатах обработки информации.

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

## Порядок авторизации пользователей

В данном программном продукте предусмотрена **роль преподавателя**. Он имеет доступ к следующим функциям:

* загрузка недельного расписания с сайта колледжа;
* управление базой данный (добавление, удаление и редактирование информации об отработанных часах);
* генерация месячного отчёта по отработанным часам;
* редактирование своего профиля (изменение пароля и ссылки для загрузки расписания с сайта колледжа).

Процесс аутентификации и авторизации обеспечивает безопасный доступ к функционалу приложения, гарантируя, что каждый преподаватель получает индивидуальные права доступа и может управлять только своими данными, исключая возможность несанкционированного редактирования информации другими преподавателями.

## 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 о преподавателях и их отработанных часах.

В основе системы пять таблиц:

* таблица «Teachers» – служит для хранения данных о преподавателях. Детальное описание этой таблицы представлено в таблице 3.1;
* таблица «Subjects» – служит для хранения данных о предметах, которые ведёт преподаватель. Детальное описание этой таблицы представлено в таблице 3.2;
* таблица «AllHours» – служит для хранения общего количества часов по каждому предмету, который ведёт преподаватель. Детальное описание этой таблицы представлено в таблице 3.3;
* таблица «Hours» – служит для хранения информации о проведённых часах. Детальное описание этой таблицы представлено в таблице 3.4;
* таблица «Dates» – служит для хранения дат, по которым проходили предметы преподавателя. Детальное описание этой таблицы представлено в таблице 3.5.

Таблица 3.1 – Детальное описание таблицы Teachers

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название поля | Тип данных | Длина поля | Разрешение значения NULL | Описание поля |
| id | int | MAX | Нет | Уникальный идентификатор преподавателя |
| login | varchar | 50 | Нет | Логин преподавателя |
| password | varchar | 50 | Нет | Зашифрованный пароль преподавателя |
| url | varchar | MAX | Да | Ссылка для парсинга раписания преподавателя |

Таблица 3.2 – Детальное описание таблицы Subjects

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название поля | Тип данных | Длина поля | Разрешение значения NULL | Описание поля |
| id | int | MAX | Нет | Уникальный идентификатор предмета |
| name | varchar | 50 | Нет | Название предмета |
| teacherId | int | MAX | Нет | Уникальный идентификатор преподавателя, который ведёт данный предмет |

Таблица 3.3 – Детальное описание таблицы AllHours

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название поля | Тип данных | Длина поля | Разрешение значения NULL | Описание поля |
| id | int | MAX | Нет | Уникальный идентификатор предмета |
| subjectName | varchar | 50 | Нет | Название предмета |
| teacherId | int | MAX | Нет | Уникальный идентификатор преподавателя, который ведёт данный предмет |
| countOfHours | int | MAX | Нет | Общее количество часов предмета |

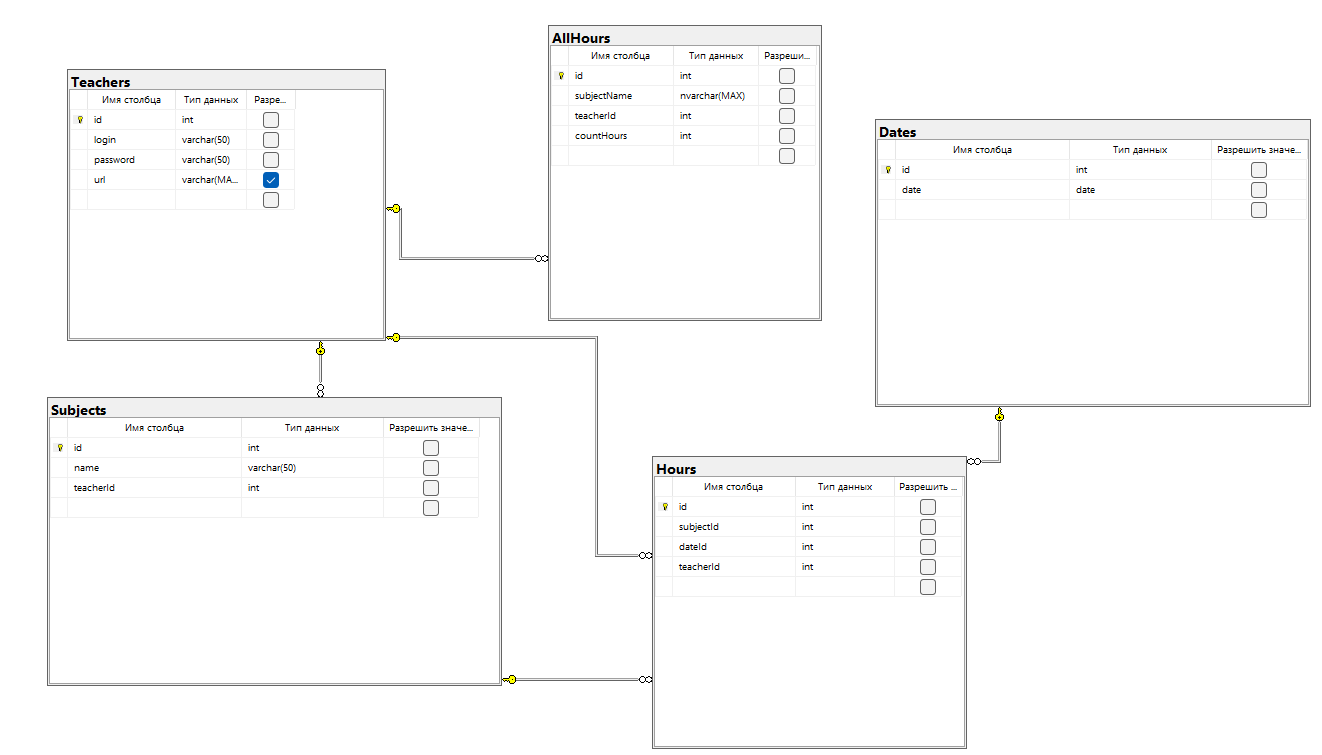
Таблица 3.4 – Детальное описание таблицы Hours

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название поля | Тип данных | Длина поля | Разрешение значения NULL | Описание поля |
| id | int | MAX | Нет | Уникальный идентификатор отработанного часа |
| subjectId | int | MAX | Нет | Уникальный идентификатор предмета |
| dateId | int | MAX | Нет | Уникальный идентификатор даты |
| teacherId | int | MAX | Нет | Уникальный идентификатор преподавателя, который ведёт предмет |

Таблица 3.5 – Детальное описание таблицы Dates

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название поля | Тип данных | Длина поля | Разрешение значения NULL | Описание поля |
| id | int | MAX | Нет | Уникальный идентификатор даты |
| date | date | MAX | Нет | Дата проведения предмета |

Схема базы данных представлена на рисунке 3.1.

Рисунок 3.1 – Схема базы данных

## Функции: логическая и физическая организация

В ходе разработки системы для формирования отчётов по отработанным часам преподавателя было разработано множество функций, из которых основными являются функции, представленные ниже.

Функция хеширования паролей, которая вызывается в окнах регистрации (авторизации) и предназначена для шифрования паролей. Код функции ниже:

public static ulong HashFunc(string str)

{

int key = 11; // Ключ для степени (можно изменить для получения других значений хеша)

int count = 9; // Определяет размер хеш-таблицы (10^count)

ulong size = (ulong)Math.Pow(10, count); // Вычисляем размер хеш-таблицы

ulong hash\_code, t\_hash = 0; // Переменные для хранения промежуточного и конечного хеша

for (int i = 0; i < str.Length; i++) // Проход по каждому символу строки

{

t\_hash += (ulong)Math.Pow(key, i) \* (ulong)str[i]; // Добавляем значение символа, умноженное на key^i

t\_hash %= size; // Обрезаем значение, чтобы избежать переполнения

}

hash\_code = t\_hash % size; // Получаем окончательный хеш, ограниченный размером таблицы

return hash\_code; // Возвращаем вычисленный хеш-код

}

Функция загрузки расписания с сайта колледжа, которая служит для парсинга расписания преподавателя и вызывается при нажатии на кнопку «Загрузить расписание». Код функции представлен ниже:

public async static Task<List<string>> GetDayOfWeek(string url, string subjectsTag, string groupsTag)

{

// Создаём конфигурацию для парсинга, используя встроенный загрузчик страниц

var config = Configuration.Default.WithDefaultLoader();

// Создаём новый контекст для обработки веб-страницы

var context = BrowsingContext.New(config);

// Асинхронно загружаем HTML-документ по указанному URL

var doc = await context.OpenAsync(url);

// Создаём список для хранения извлечённых данных (предмет + группа)

var day = new List<string>();

// Получаем список элементов, соответствующих тэгу subjectsTag (например, названия предметов)

var links1 = doc.QuerySelectorAll(subjectsTag);

// Получаем список элементов, соответствующих тэгу groupsTag (например, номера групп)

var links2 = doc.QuerySelectorAll(groupsTag);

// Проходим по извлечённым элементам и объединяем информацию о группе и предмете

for (int j = 0; j < links1.Length; j++)

{

day.Add($"{links2[j].TextContent} {links1[j].TextContent}"); // Формируем строку "Группа Предмет"

}

// Возвращаем список полученных данных

return day;

}

Функция составления отчётов по отработанным часам, которая вызывается при нажатии на кнопку «Составить отчёт» и служит для формирования отчёта по отработанным часам преподавателя за указанный им период.

public void GenerateReport(int teacherId, List<List<string>> resultReport, string month, string year)

{

// Получаем путь к каталогу проекта и создаём папку "Отчёты"

string projectDir = AppDomain.CurrentDomain.BaseDirectory;

string dir = System.IO.Path.Combine(projectDir, "Отчёты");

// Настраиваем диалог для сохранения файла

var save = new SaveFileDialog

{

Title = "Сохранить файл как",

Filter = "Документы Word (\*.docx)|\*.docx",

FileName = $"Отчёт.docx",

InitialDirectory = dir

};

string file = $"Отчёт.docx";

// Если пользователь подтвердил сохранение

if (save.ShowDialog() == true)

{

file = save.FileName;

// Перебираем список отчётов и вычисляем количество вычтенных часов

for (int i = 0; i < resultReport.Count; i++)

{

int sum = 0;

for (int j = 1; j < resultReport[i].Count; j++)

{

string[] temp = resultReport[i][j].Split('-');

if (int.TryParse(temp[1].Trim(), out int num))

{

sum += num;

}

}

// Получаем информацию о преподавателе по имени предмета

string subName = resultReport[i][0];

var sbj = AllHoursRepository.GetTeacherHourByName(teacherId, subName);

// Добавляем итоговые данные в отчёт

resultReport[i].Add($"Всего часов: {sbj.countHours}");

resultReport[i].Add($"Вычтено: {sum}");

resultReport[i].Add($"Осталось: {sbj.countHours - sum}");

// Обновляем данные в репозитории

AllHoursRepository.SubtractHours(sbj, sum);

}

// Определяем максимальное количество строк в таблице

int[] arr = new int[resultReport.Count];

for (int i = 0; i < resultReport.Count; i++)

{

arr[i] = resultReport[i].Count;

}

int max = arr.Max();

// Создаём объект Word

Microsoft.Office.Interop.Word.Application word = new Microsoft.Office.Interop.Word.Application();

Microsoft.Office.Interop.Word.Document doc = word.Documents.Add();

// Настраиваем параметры страницы (A3, альбомный режим)

doc.PageSetup.PaperSize = WdPaperSize.wdPaperA3;

doc.PageSetup.Orientation = WdOrientation.wdOrientLandscape;

// Добавляем заголовок отчёта

Microsoft.Office.Interop.Word.Paragraph paragraph = doc.Content.Paragraphs.Add();

paragraph.Range.Text = $"Отчёт по отработанным часам за {month} {year} года.";

paragraph.Range.Font.Size = 16;

paragraph.Range.Font.Bold = 1;

paragraph.Alignment = WdParagraphAlignment.wdAlignParagraphCenter;

paragraph.Range.InsertParagraphAfter();

// Создаём таблицу для отчёта

Microsoft.Office.Interop.Word.Range tableRange = doc.Content.Paragraphs.Add().Range;

Microsoft.Office.Interop.Word.Table table = doc.Tables.Add(tableRange, max, resultReport.Count);

// Настраиваем границы таблицы

table.Borders.Enable = 1;

table.Borders.OutsideLineStyle = WdLineStyle.wdLineStyleSingle;

table.Borders.InsideLineStyle = WdLineStyle.wdLineStyleSingle;

table.AutoFitBehavior(WdAutoFitBehavior.wdAutoFitContent);

table.Rows.Alignment = WdRowAlignment.wdAlignRowCenter;

// Заполняем таблицу данными из `resultReport`

for (int col = 0; col < resultReport.Count; col++)

{

for (int row = 0; row < resultReport[col].Count; row++)

{

table.Cell(row + 1, col + 1).Range.Text = resultReport[col][row];

table.Cell(row + 1, col + 1).Range.Font.Bold = 0;

}

}

// Добавляем дату создания отчёта

paragraph.Range.Text = $"Дата создания: {DateTime.Now.ToShortDateString()}";

paragraph.Range.Font.Size = 16;

paragraph.Range.Font.Bold = 1;

paragraph.Alignment = WdParagraphAlignment.wdAlignParagraphLeft;

paragraph.Range.InsertParagraphAfter();

// Сохраняем и закрываем документ

doc.SaveAs2(file);

doc.Close();

word.Quit();

}

}

Все остальные функции представлены в приложении А.

## Входные и выходные данные

В качестве входных данных приложение принимает введённое вручную или загруженное расписание занятий, а также период, в который эти занятия были проведены.

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

* заголовок, в котором указан месяц и год, за который был составлен данный отчёт;
* таблица с данными об отработанных часах;
* дата создания отчёта.

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

## Функциональное тестирование

Для проверки корректности работы приложения были разработаны тест-кейсы, которые представлены в таблице 3.6.

Таблица 3.6 – Тест-кейсы

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Описание | Шаги | Ожидаемый результат | Фактический результат |
| Т1 | Проверка парсинга расписания по корректной ссылке | 1. Открыть приложение; 2. Ввести логин «TestUser»; 3. Ввести пароль «UserTest230225»; 4. Нажать на кнопку «Войти»; 5. Зайти на сайт колледжа: 6. Открыть страницу с расписанием преподавателя: 7. Скопировать ссылку данной страницы; 8. Закрыть сайт; 9. Вернуться в приложение; 10. Перейти на страницу «Профиль»; 11. Установить скопированную ссылку в поле «Ссылка для парсинга расписания»; 12. Нажать на кнопку «Сохранить»; 13. Перейти на страницу «Расписание»; 14. Нажать на кнопку «Загрузить расписание». | Приложение должно загрузить недельное расписание и отобразить его в таблице | Представлен в приложении Б на рисунке Б.2 |

Продолжение таблицы 3.6

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Описание | Шаги | Ожидаемый результат | Фактический результат |
| Т2 | Проверка парсинга расписания по не корректной ссылке | 1. Открыть приложение; 2. Ввести логин «TestUser»; 3. Ввести пароль «UserTest230225»; 4. Нажать на кнопку «Войти»; 5. Перейти на страницу «Профиль»; 6. Установить ссылку «https://www.google.by/?hl=ru» в поле «Ссылка для парсинга расписания»; 7. Нажать на кнопку «Сохранить»; 8. Перейти на страницу «Расписание»;   Нажать на кнопку «Загрузить расписание». | Приложение должно оповестить об ошибке загрузки расписания | Представлен в приложении Б на рисунке Б.3 |
| Т3 | Проверка парсинга расписания с исправным интернет-соединением | 1. Подключиться к сети интернет; 2. Открыть приложение; 3. Ввести логин «TestUser»; 4. Ввести пароль «UserTest230225»; 5. Нажать на кнопку «Войти»; 6. Зайти на сайт колледжа: 7. Открыть страницу с расписанием преподавателя: 8. Скопировать ссылку данной страницы; 9. Закрыть сайт; 10. Вернуться в приложение; 11. Перейти на страницу «Профиль»; 12. Установить скопированную ссылку в поле «Ссылка для парсинга расписания»; 13. Нажать на кнопку «Сохранить»; 14. Перейти на страницу «Расписание»; 15. Нажать на кнопку «Загрузить расписание». | Приложение должно загрузить недельное расписание и отобразить его в таблице | Представлен в приложении Б на рисунке Б.4 |

Продолжение таблицы 3.6

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Описание | Шаги | Ожидаемый результат | Фактический результат |
| Т4 | Проверка парсинга расписания с неисправным интернет-соединением | 1. Отключиться от сети интернет; 2. Открыть приложение; 3. Ввести логин «TestUser»; 4. Ввести пароль «UserTest230225»; 5. Нажать на кнопку «Войти»; 6. Перейти на страницу «Профиль»; 7. Установить ссылку «https://www.google.by/?hl=ru» в поле «Ссылка для парсинга расписания»; 8. Нажать на кнопку «Сохранить»; 9. Перейти на страницу «Расписание»; 10. Нажать на кнопку «Загрузить расписание». | Приложение должно оповестить об ошибке загрузки расписания | Представлен в приложении Б на рисунке Б.5 |
| Т5 | Проверка составления отчёта с выбранными месяцем и годом | 1. Открыть приложение; 2. Ввести логин «TestUser»; 3. Ввести пароль «UserTest230225»; 4. Нажать на кнопку «Войти»; 5. Перейти на страницу «Отчёт»; 6. Выбрать месяц «Апрель» в поле «Месяц»; 7. Выбрать год «2025» в поле «Год»; 8. Нажать на кнопку «Составить отчёт»; 9. Указать путь для сохранения отчёта; 10. Нажать на кнопку «Сохранить». | Приложение должно оповестить об успешном составлении и сохранении отчёта | Представлен в приложении Б на рисунке Б.6 |
| Т6 | Проверка составления отчёта с не выбранными месяцем и годом | 1. Открыть приложение; 2. Ввести логин «TestUser»; 3. Ввести пароль «UserTest230225»; 4. Нажать на кнопку «Войти»; 5. Перейти на страницу «Отчёт»; 6. Нажать на кнопку «Составить отчёт». | Приложение должно потребовать выбрать месяц и год | Представлен в приложении Б на рисунке Б.7 |

Продолжение таблицы 3.6.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Описание | Шаги | Ожидаемый результат | Фактический результат |
| Т7 | Проверка работы фильтров с не корректными параметрами | 1. Открыть приложение; 2. Ввести логин «TestUser»; 3. Ввести пароль «UserTest230225»; 4. Нажать на кнопку «Войти»; 5. Перейти на страницу «Часы»; 6. Указать в поле «С:» дату «16.05.2025»; 7. Указать в поле «До:» дату «14.05.2025»; 8. Нажать на кнопку «Применить фильтры». | Приложение должно оповестить об ошибке | Представлен в приложении Б на рисунке Б.8 |
| Т8 | Проверка работы фильтров с корректными параметрами | 1. Открыть приложение; 2. Ввести логин «TestUser»; 3. Ввести пароль «UserTest230225»; 4. Нажать на кнопку «Войти»; 5. Перейти на страницу «Часы»; 6. Указать в поле «С:» дату «01.04.2025»; 7. Указать в поле «До:» дату «29.04.2025»; 8. Нажать на кнопку «Применить фильтры». | Приложение должно загрузить в таблицу результаты запроса | Представлен в приложении Б на рисунке Б.9 |
| Т9 | Проверка добавления предмета с корректными данными | 1. Открыть приложение; 2. Ввести логин «TestUser»; 3. Ввести пароль «UserTest230225»; 4. Нажать на кнопку «Войти»; 5. Перейти на страницу «Часы»; 6. Нажать на кнопку «Добавить час»; 7. Ввести в поле «Предмет:» значение «КПиЯП»; 8. Ввести в поле «Группа:» значение «Т-292»; 9. Выбрать дату «14.05.2025»; 10. В поле «Количество часов» указать значение «2»; 11. Нажать на кнопку «Добавить». | Приложение должно оповестить об успешном добавлении и обновить таблицу | Представлен в приложении Б на рисунке Б.10 |

Продолжение таблицы 3.6

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Т10 | Проверка добавления предмета с не корректными данными | 1. Открыть приложение; 2. Ввести логин «TestUser»; 3. Ввести пароль «UserTest230225»; 4. Нажать на кнопку «Войти»; 5. Перейти на страницу «Часы»; 6. Нажать на кнопку «Добавить час»; 7. Ничего не указывать в поле «Предмет:»; 8. Ввести в поле «Группа:» значение «Т-292»; 9. Не выбирать дату проведения предмета; 10. В поле «Количество часов» указать значение «2»; 11. Нажать на кнопку «Добавить». | Приложение должно оповестить об ошибке с указанием всех некорректно заполненных полей. | Представлен в приложении Б на рисунке Б.11 |

## Описание справочной системы

# Применение

## Назначение программного средства

## Условия применения

# Заключение

# Список использованных источников

# ПРИЛОЖЕНИЕ А

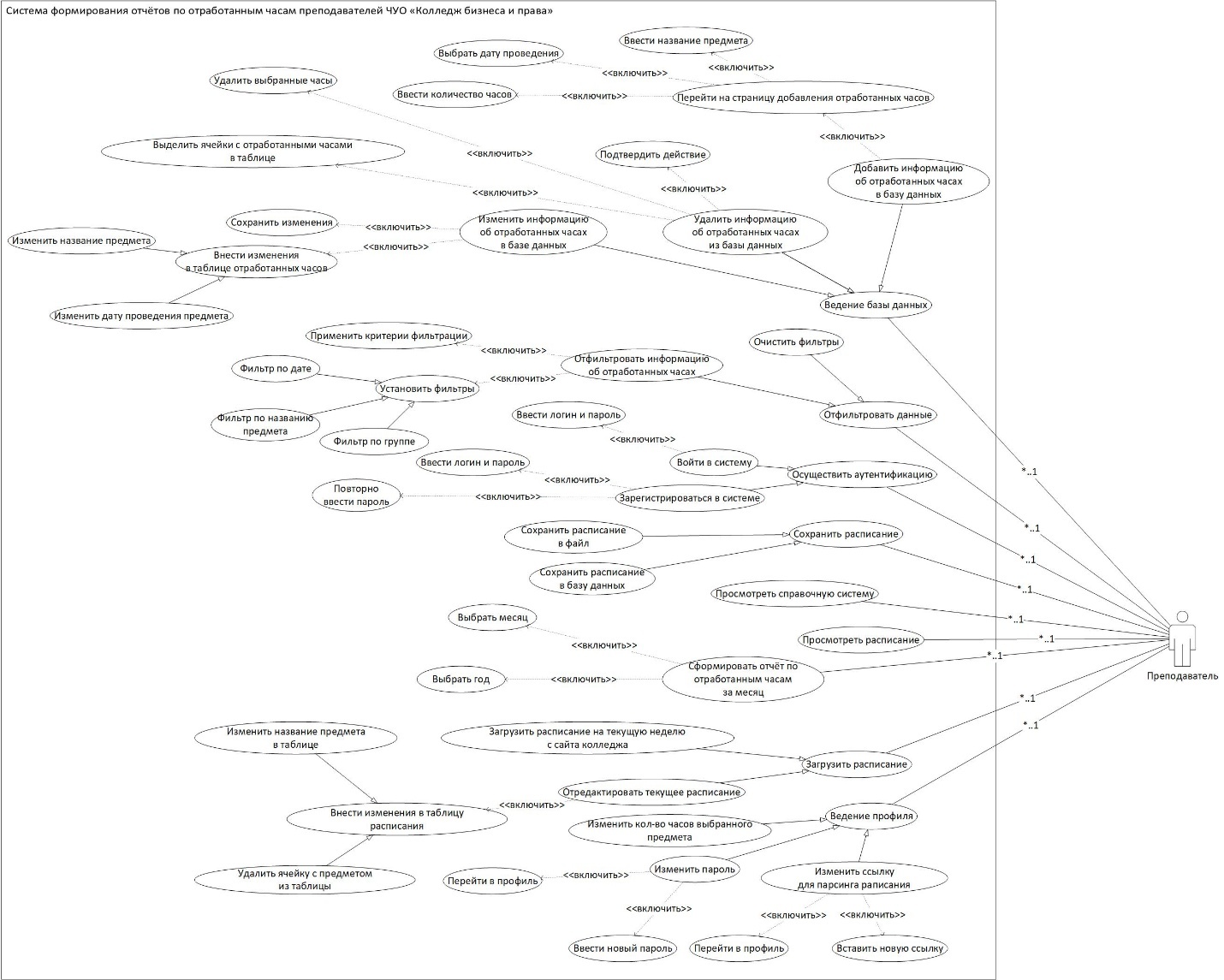
**(обязательное)**

**Текст программных модулей**

# ПРИЛОЖЕНИЕ Б

**(справочное)**

**Результаты работы приложения**



КП Т.292001.401 ГЧ

КБП

У

Т. Контр.

Разраб. Разраб.

Буев Д.Р.

Провер. Провер.

Коропа Е.Н.

Реценз.

Утверд.

Н. Контр.

*РАЗРАБОТКА СИСТЕМЫ ФОРМИРОВАНИЕ ОТЧЕТОВ ПО ОТРАБОТАННЫМ ЧАСАМ ПРЕПОДАВАТЕЛЕЙ 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.292001.401 ГЧ

*Диаграмма вариантов использования*

*Диаграмма вариантов использования*

Диаграмма вариантов использования

КП Т.292001.401 ГЧ

КБП

У

Т. Контр.

Разраб. Разраб.

Буев Д.Р.

Провер. Провер.

Коропа Е.Н.

Реценз.

Утверд.

Н. Контр.

*РАЗРАБОТКА СИСТЕМЫ ФОРМИРОВАНИЕ ОТЧЕТОВ ПО ОТРАБОТАННЫМ ЧАСАМ ПРЕПОДАВАТЕЛЕЙ ЧУО «КОЛЛЕДЖ БИЗНЕСА И ПРАВА»*

№ докум.

Подпись

Дата

Изм.

Лист

Масса

Лит.

Масштаб

Лист 2

Листов 3

Инв.№подл.

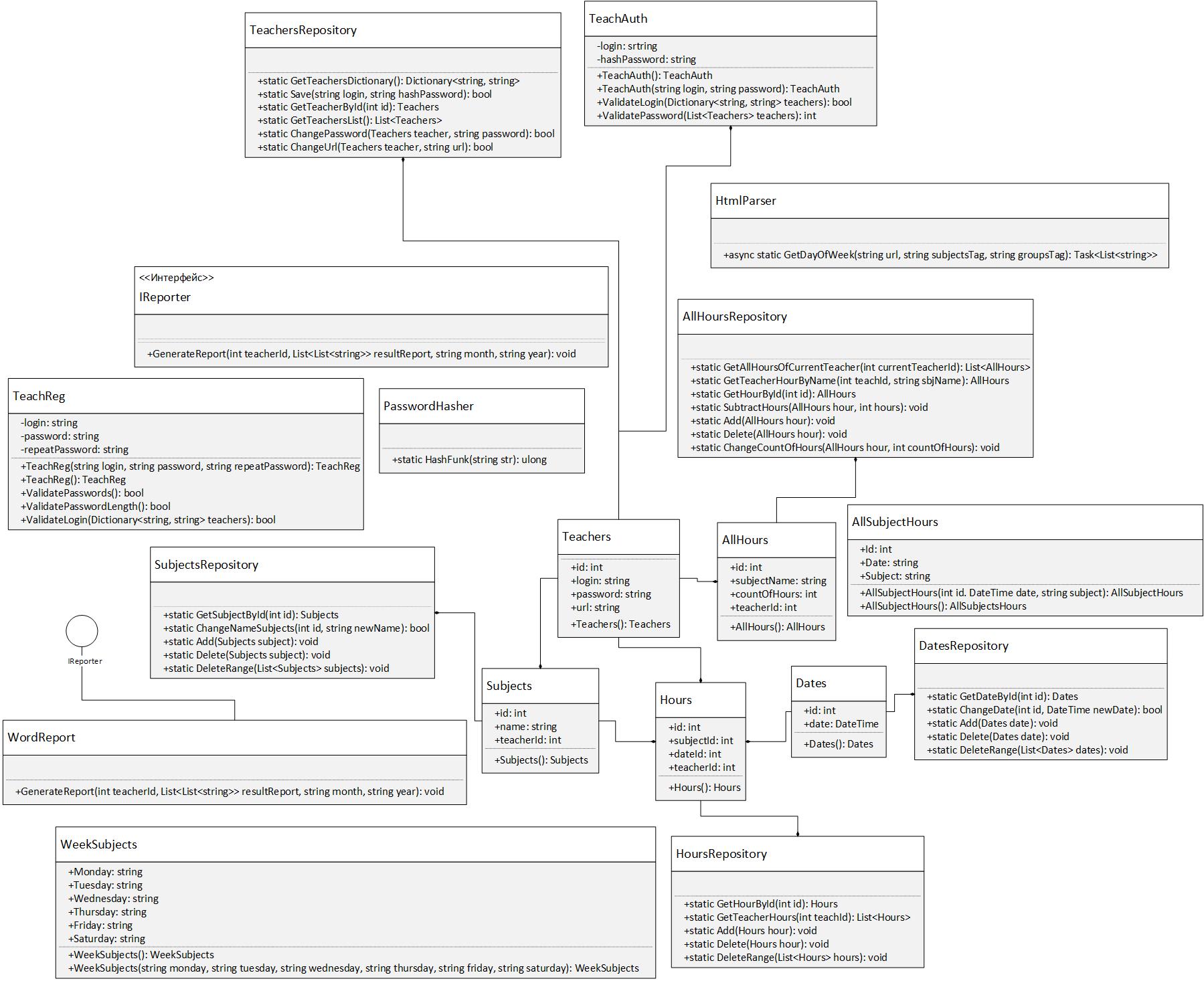
Подп. и дата

Взам.инв.№

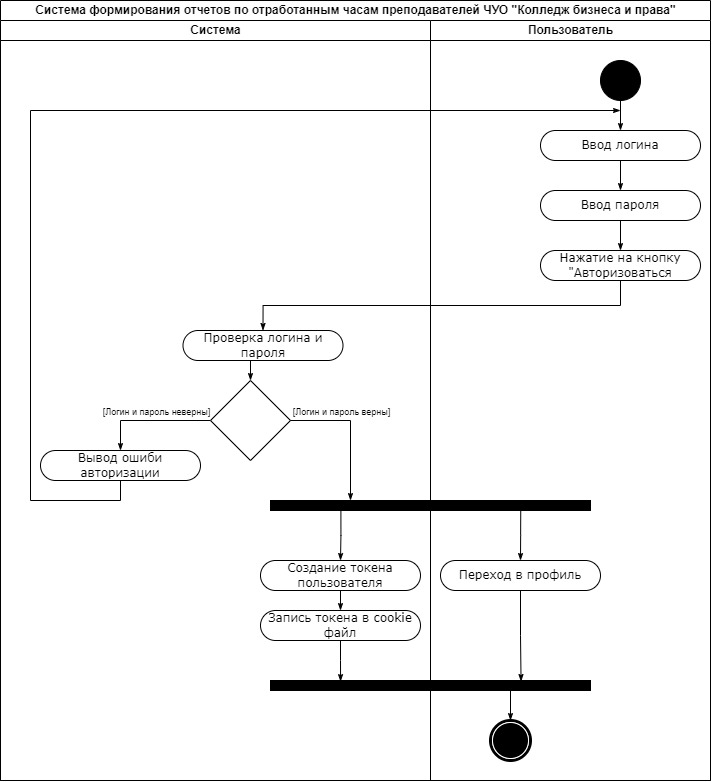
Инв.№дубл.

Подп. и дата

КП Т.292001.401 ГЧ



*Диаграмма классов*



КП Т.292001.401 ГЧ

*Диаграмма деятельности*

*(процесс авторизации преподавателя)*

КБП

У

Т. Контр.

Разраб. Разраб.

Буев Д.Р.

Провер. Провер.

Коропа Е.Н.

Реценз.

Утверд.

Н. Контр.

*РАЗРАБОТКА СИСТЕМЫ ФОРМИРОВАНИЕ ОТЧЕТОВ ПО ОТРАБОТАННЫМ ЧАСАМ ПРЕПОДАВАТЕЛЕЙ 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.292001.401 ГЧ