**Содержание**

[Введение 4](#_Toc159874659)

[1. Предварительное планирование и проверка осуществимости 6](#_Toc159874660)

[1.1 Требования к системе в целом 6](#_Toc159874661)

[1.2 Требования к функциям, выполняемым системой 8](#_Toc159874662)

[1.3 Требования к видам обеспечения 9](#_Toc159874663)

[1.4 Сравнение и анализ аналогов системы 10](#_Toc159874664)

[1.5 Инфологическая модель предметной области 11](#_Toc159874665)

[1.6 Прототипирование интерфейса 12](#_Toc159874666)

[2. Практическая часть 15](#_Toc159874667)

[2.1. Назначение и цели создания системы 15](#_Toc159874668)

[2.2 Характеристики объекта автоматизации 15](#_Toc159874669)

[2.3. Обзор инструментальных средств создания продукта 17](#_Toc159874670)

[2.4. Обоснование выбора средств реализации ИС 17](#_Toc159874671)

[2.5. Описание логики функционирования ПО 18](#_Toc159874672)

[2.6. Реализация интерфейса ПО 23](#_Toc159874673)

[2.7. Написание программного кода 29](#_Toc159874674)

[2.8. Защита информации 32](#_Toc159874675)

[Заключение 36](#_Toc159874676)

[Список использованных источников 37](#_Toc159874677)

[Приложение 38](#_Toc159874678)

[**Приложение А** 38](#_Toc159874679)

# Введение

В современном образовании учет успеваемости студентов играет важную роль, так как позволяет оценить знания и умения обучающихся, а также оказывает влияние на их будущую карьеру. Для техникума – это особенно важно, так как студенты обучаются специфическим профессиональным навыкам. В данном курсовом проекте мы рассмотрим методы учета успеваемости студентов техникума, а также разработаем программное обеспечение, которое позволит эффективно отслеживать успехи обучающихся и предоставлять им информацию о своем прогрессе. Такой подход позволит улучшить качество обучения и повысить академическую мотивацию студентов.

**Актуальность темы** – Потребность в постоянном контроле и оценки учебных достижений обучающихся. Разработка и внедрение специализированных программ и информационных систем для учета успеваемости студентов техникума способствует повышению эффективности учебного процесса для учебного заведения. Такая программа может обеспечить более точный анализ данных об учебных достижениях студентов, помочь выявить индивидуальные потребности в обучении и разработать персонализированные подходы к образовательному процессу. В результате, техникум сможет оперативно реагировать на изменения в обучении, повышать уровень подготовки специалистов и обеспечить высокое качество образования.

**Цель** – Разработка модуля для учета успеваемости студентов техникума

**Объект** – Введение журнала и учета успеваемости студентов.

**Предмет** – Модуль для учета успеваемости студентов техникума.

**Для достижения цели, требуется выполнить следующие задачи:**

1. Провести предварительное планирование и проверку осуществимости.
2. Определить требования к программному продукту.
3. Разработать прототип приложения.
4. Реализовать приложение в выбранной среде IDE.
5. Разместить проект на веб-сервисе GitHub или аналогичном.

# Предварительное планирование и проверка осуществимости

# Требования к системе в целом

1.1.1. Требования к структуре и функционированию системе

Система должна состоять из сервера баз данных и приложения.

Сервер БД отвечает за хранение базы данных. Приложение организует обмен данными с сервером баз данных. Приложение – основная часть системы – обрабатывает данные полученные от сервера БД и выводит их в понятном для конечного пользователя виде, а также формирует запросы от пользователя к серверу БД.

1.1.2. Требования к численности и квалификации персонала

Пользователями модуля являются в основном студенты и преподаватели. Пользователи используют систему через персональный компьютер или специально установленный терминал в здании техникума, поэтому количество людей, желающих просмотреть свою успеваемость или выставить обучающемуся оценку неограниченно.

Обслуживающий персонал:

• Администратор.

• Диспетчер расписания.

Пользователи должны иметь минимальные навыки работы на компьютере и использование сенсорных экранных клавиатур.

Обслуживающий персонал должен являться продвинутыми пользователями в компьютере и в глобальной сети Интернет, а также иметь опыт работы с SQL Server и владеть языком программирования C#.

Администратор сервера должен иметь опыт работы с MYSQL и в программе Visual Studio.

1.1.3. Требования к надежности

Программное обеспечение системы должно функционировать и не приводить к зависанию системы при любых ошибочных действиях операторов в среде данного программного модуля.

1.1.4. Требования к эргономике и технической эстетике

Программное обеспечение и технические средства должны иметь эстетически привлекательный вид и быть удобными в использовании как для обучающегося, так и для преподавателей.

1.1.5. Требования к эксплуатации и техническому обслуживанию

Для нормальной работы системы необходимо произвести ряд действий

На компьютере:

• установить лицензионную копию ОС;

• запустить программу установки приложения.

На сервере базы данных:

• установить лицензионную копию ОС;

• установить копию SQL Server;

• запустить программу установки сервера БД.

1.1.6. Требования к защите информации от несанкционированного доступа

Защита ИС и информации, с которой она работает, от несанкционированного разграничение доступа по ролям и частично возлагается на операционную систему. Доступ к данным предоставляется только легальным пользователям.

1.1.7. Требования по сохранности информации при авариях

Сохранность данных должна быть обеспеченна:

• при сбоях в работе оборудования сети в любой момент времени;

• при сбоях электропитания или программного обеспечения не ведущих к физическому уничтожению или повреждению информации.

1.1.8. Требования к защите от влияния внешних воздействий

Автоматизированная система должна быть устойчивой к наличию и параллельной с ней работе на том же ПК другого программного обеспечения. Должна быть устойчивой к одновременной работе пользователей.

1.1.9. Требования к стандартизации и унификации

Автоматизированная система должна предоставлять пользователю интуитивно понятный интерфейс. Программная документация, поставляемая с программой, должна быть оформлена в соответствии со стандартом ЕСПД.

# 1.2 Требования к функциям, выполняемым системой

1.2.1. Функции подсистемы «Сервер базы данных»:

• получение запросов от «пользователей БД»

• проверка легальности выполнения запроса на основе идентификационной информации, вложенной в запрос;

• выполнение SQL-запросов к БД;

• передача результата SQL-запроса «Модуля».

1.2.2. Функции подсистемы «Модуля»:

• передача пользователю следующей информации:

• информация о предметах;

• информация об оценках учащихся;

• визуальная информация о аттестации.

# 1.3 Требования к видам обеспечения

1.3.1. Требования к математическому обеспечению

В основе разрабатываемого ПО должна лежать реляционная модель базы данных. Все таблицы должны находиться в третьей или четвертой нормальной форме. Связи между таблицами должны быть вида: «один к одному» или «один ко многим». При проектировании используется метод «сущность – связь».

1.3.2. Требования к информационному обеспечению

Данные в системе должны быть организованны в виде связанных таблиц. Хранение и доступ к данным осуществляется с применением реляционные систем управления базами данных. Информационный обмен между компонентами системы осуществляется по технологии клиент-сервер. Желательно обеспечить возможность резервного копирования и восстановления данных.

1.3.3. Требования к лингвистическому обеспечению

В качестве языка программирования используются – MS SQL, С#, взаимодействие пользователя и ПО осуществляется через оконный интерфейс.

1.3.4. Требования к программному обеспечению

Система должна иметь следующее программное обеспечение: ОС MS Windows 7 или выше, C#, WPF, SQL Server Management.

1.3.5. Требования к техническому обеспечению

Для работоспособности модуля необходимы следующие программные и технические средства:

• клавиатура, компьютерная мышь;

• процессор Intel Core Pentium и выше;

• оперативная память 2 Гб и больше;

• видеокарта и монитор, поддерживающего разрешение 1280х720 точек и больше (рекомендуется 1920х1080 точек) при глубине цвета 16 бит и более;

• дисковое пространство 1Г и больше;

# 1.4 Сравнение и анализ аналогов системы

<https://dnevnik.ru/> - Дневник.ру (Рисунок 1.1)

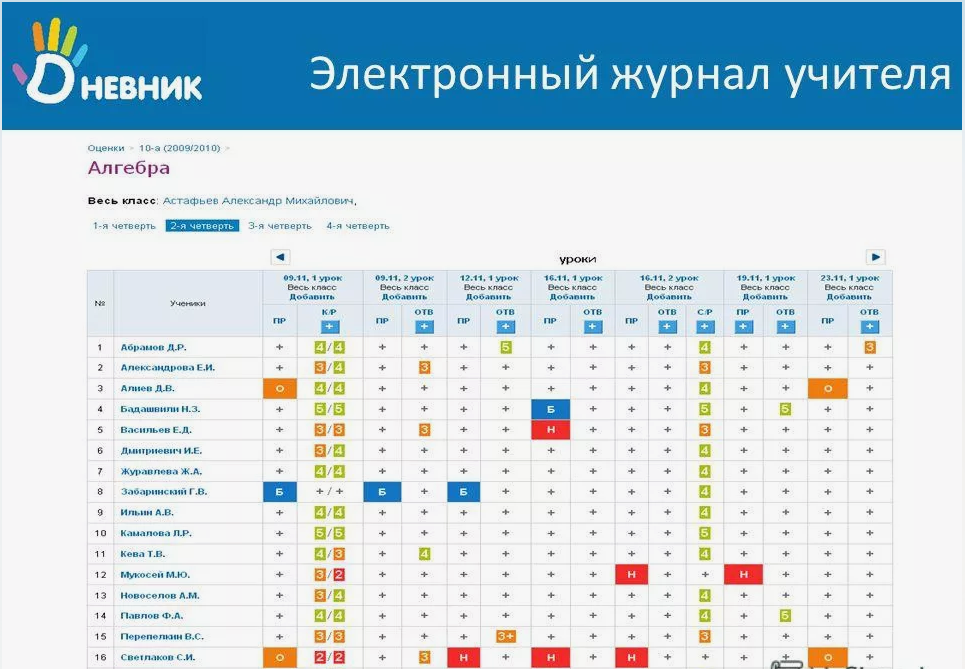


Рисунок 1.1 – «Дневник.ру».

Таблица 1 - Cравнения.

|  |  |  |
| --- | --- | --- |
| **Наименование** | **Дневник.ру** | **Моя программа** |
| Интерфейс | Интуитивно понятен, но можно запутаться в датах и урока | Понятный интерфейс, в котором трудно запутаться и правильные даты, которые выставляет сам преподаватель |
| Информация | Не всегда полная и достоверная информация. Иногда сайт публикует непроверенные или искаженные факты | Информация всегда достоверна, и нет публикаций. |
| Вид программы | Сайт содержит рекламные блоки, которые могут отвлекать от чтения статей и создавать негативное впечатление от пользовательского опыта | Не содержит рекламы и различных блоков, чтобы не отвлекать пользователей моего модуля |

Вывод по сравнению: Самое главное нужно увидеть достоверную информацию об оценках студентов техникума, ясный интерфейс для преподавателей, а также учащихся.

# 1.5 Инфологическая модель предметной области

1.5.1. Описание инфологической модели.

Сущность группа и студент имеют связь «1:М», так как студентов в одной группе множество. Между сущностями студент и журнал имеется связь «М:М» так как у каждого студента, множество оценок. Между сущностью дисциплина и журнал имеется связь «1:М» так как по каждой дисциплины ставится множество оценок. У сущностей студент и сессия идёт связь «1:1». У сущности сессия и дисциплины имеется связь «М:М» так как у в техникуме множество дисциплин и они входят к отчётам по успеваемости. У сущности преподаватель и дисциплины есть связь «1:М» так как один преподаватель может преподавать несколько дисциплин. Инфологическая модель представлена на рисунке 1.2.



Рисунок 1.2 – Инфологическая модель «Учет успеваемости студентов техникума»

# 1.6 Прототипирование интерфейса

Прототипирование является важным этапом в разработке программы, так как позволяет проверить идеи и концепции до того, как будет написан код. Это экономит время и ресурсы, так как можно быстро определить, что работает и что не работает, и внести необходимые изменения. С помощью прототипа можно взаимодействовать с продуктом и проводить пользовательские тестирования.

С помощью прототипирования можно создавать интерактивные модели, которые могут быть использованы для сбора обратной связи от пользователей. Это позволяет дизайнерам и разработчикам получить ценную информацию о том, как пользователи будут взаимодействовать с продуктом.

Прототипирование также помогает улучшить коммуникацию между дизайнерами и разработчиками, так как оно позволяет им видеть продукт из одного и того же ракурса. Это уменьшает количество ошибок и несоответствий между дизайном и кодом.

Прототипирование экономит денежные затраты на разработку того, что может не сработать. Качество прототипирования влияет удобство использование и на качество связанного с разметкой кода.

Высококачественная разработка прототипа — это хорошие вложения в программную часть. Простота важный вклад, чем меньше строк будет содержать прототип, тем меньше мест, где может скрыться ошибка, и тем проще будет ориентироваться в разметке.

Форма авторизации. Эта форма создана для того чтобы взаимодействовать с программой, и правильно расставить ролями и права пользования программой. Представлена на рисунке 1.3.

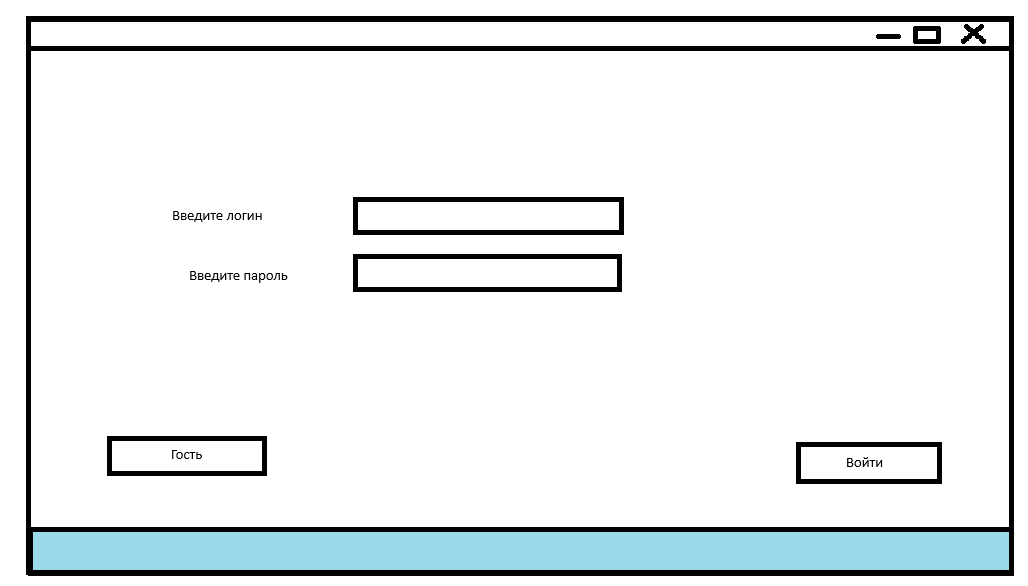


Рисунок 1.3 – Форма «Авторизация в систему»

Главная форма, создана для удобства в пользовании программы и упрощения в перемещении между формами. Представлен на рисунке 1.4.

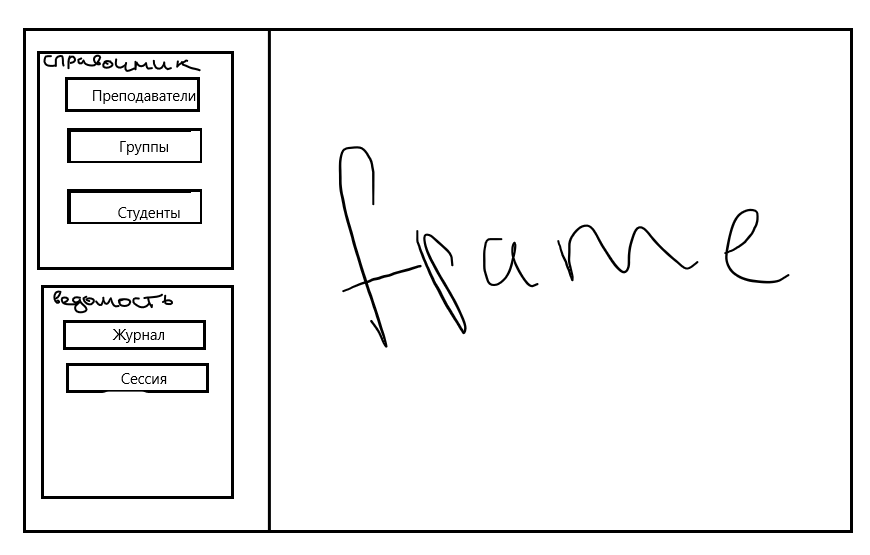


Рисунок 1.4 – Прототип главной формы

Форма редактирования и добавления. Данная форма создана для работы с данными, их редактирование удаление и добавления. Представлена на рисунке 1.5

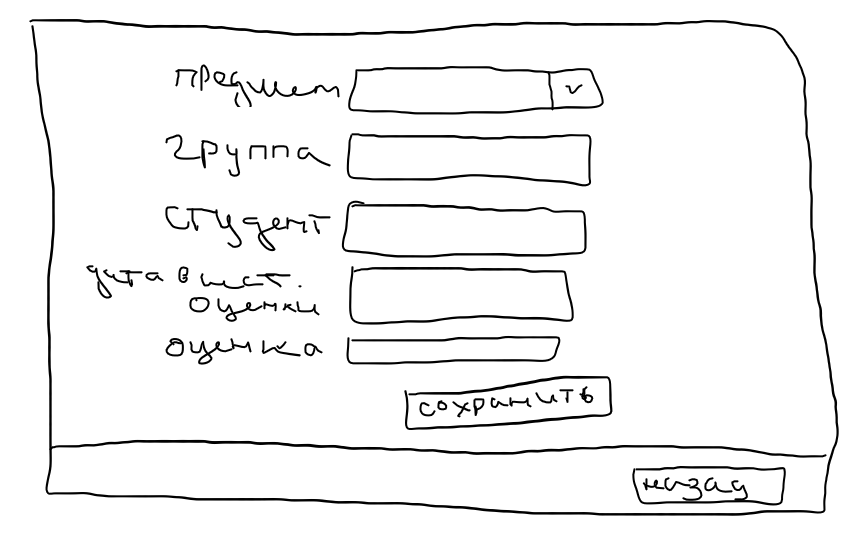


Рисунок 1.5 – Форма редактирования и добавления.

# 2. Практическая часть

# 2.1. Назначение и цели создания системы

2.1.1. Полное наименование программного обеспечения и его условное обозначение.

Полное наименование – Разработка модуля «Учет успеваемости студентов техникума».

2.1.2. Назначение системы

Автоматизация процесса введения учета, необходимой для студентов техникума.

2.1.3. Цели создания системы

Цели создания модуля:

• помощь преподавателям и студентам в проведении и подготовке к учебному процессу и исправлении своих оценок;

• повышение доступности информации к оценкам учащихся;

• повышение уровня доступности к предоставленной информации.

# 2.2 Характеристики объекта автоматизации

2.2.1. Краткие сведения об объекте автоматизации

Объекты автоматизации:

• процесс получения информации о успеваемости студентов;

• процесс получения информации об оценках учащихся;

• процесс получения информации о студентах;

• процесс получения информации об ведомости;

• процесс получения информации о преподавателях.

2.2.2. В процессе выдачи информации разделах должны отображаться следующие данные:

• название раздела;

• полное описание раздела.

2.2.3. В процессе выдачи информации о студентах, имеющих отношение к техникуму, должны отображаться следующие данные:

• Ф.И.О;

• Дата рождения;

• телефон;

2.2.4. Сведения об условиях эксплуатации объекта автоматизации и характеристик окружающей среды

Условия эксплуатации объекта автоматизации (на стороне сервера БД):

• операционная система MS Windows 10 Professional;

• MS SQL Server;

Условия эксплуатации объекта автоматизации (на стороне приложения):

• операционная система MS Windows 10 Professional;

• Visual Studio 19.

Физические условия эксплуатации и характеристики окружающей среды определяются соответствующими характеристиками окружающей среды используемого аппаратного обеспечения, указанными в документации на него.

Условия эксплуатации объекта автоматизации (на стороне клиента):

• любая операционная система;

2.2.5. Физические условия эксплуатации и характеристики окружающей среды определяются соответствующими характеристиками окружающей среды используемого аппаратного обеспечения, указанными в документации.

# 2.3. Обзор инструментальных средств создания продукта

С# - VS, WPF

SQL- SQL Server Management Studio, основная база данных.

# 2.4. Обоснование выбора средств реализации ИС

Выбор языка программирования C# для реализации информационной системы (ИС) может быть обоснован следующими причинами:

1. Поддержка платформы .NET: C# является основным языком программирования для разработки приложений на платформе .NET. Платформа .NET обеспечивает мощные инструменты и библиотеки для разработки ИС, включая возможности для работы с базами данных, сетевыми протоколами и другими важными компонентами.

2. Широкая поддержка и сообщество разработчиков: C# является одним из наиболее популярных языков программирования, и имеет огромное сообщество разработчиков. Это означает, что Вы сможете найти множество ресурсов, учебных материалов и помощи в случае возникновения проблем.

3. Простота и эффективность: C# предлагает простой и понятный синтаксис, который облегчает разработку и поддержку кода. Он также обладает мощными возможностями для работы с объектно-ориентированным программированием, что позволяет создавать модульные и расширяемые ИС.

4. Интеграция с другими технологиями Microsoft: C# тесно интегрирован с другими технологиями Microsoft, такими как SQL Server, Azure и SharePoint. Это облегчает разработку ИС, которые взаимодействуют с другими системами и сервисами Microsoft.

5. Безопасность: C# предоставляет механизмы безопасности, такие как типизация и контроль доступа, которые помогают предотвратить ошибки и обеспечить безопасность данных в ИС.

6. Поддержка многопоточности: C# предоставляет удобные инструменты для работы с многопоточностью, что особенно полезно при разработке ИС, которые должны обрабатывать большие объемы данных или выполнять множество задач одновременно.

7. Переносимость: Язык C# и платформа .NET поддерживаются на различных операционных системах, включая Windows, macOS и Linux. Это позволяет разрабатывать ИС, которые могут быть запущены на различных платформах.

В целом, выбор языка программирования C# для реализации ИС может обеспечить эффективную разработку, хорошую поддержку и интеграцию с другими технологиями, а также безопасность и переносимость.

# 2.5. Описание логики функционирования ПО

Таблица «Users» - пользователи программы их пароли и логины, создана для того чтобы правильно распределить роли. Представлена на рисунке 4.

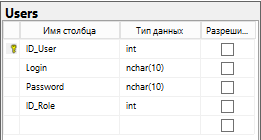


Рисунок 2.1 – Сущность «Пользователи».

Таблица 2 – Описание сущности «Пользователи».

|  |  |  |
| --- | --- | --- |
| **Название** | **Тип** | **Описание** |
| ID\_User | int | Первичный ключ |
| Login | Nchar(10) | Содержит логины пользователей |
| Password | Nchar(10) | Содержит пароли |
| ID\_Role | int | Внешний ключ, идентификатор сущности Role |

Таблица «Role» - описаны роли для пользователей в программе. Представлена на рисунке 2.2

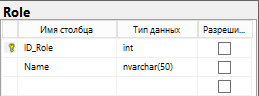


Рисунок 2.2 – Сущность «Роли».

Таблица 3 – Описание сущности «Роли».

|  |  |  |
| --- | --- | --- |
| **Название** | **Тип** | **Описание** |
| ID\_Role | int | Первичный ключ |
| Name | Nvarchar(50) | Содержит наименование ролей. |

Таблица «Student» - содержит все нужные данные студента. Представлена на рисунке 2.3

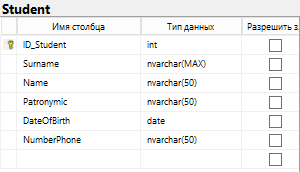


Рисунок 2.3 – Сущность «Студент».

Таблица 4 – Описание сущности «Студент».

|  |  |  |
| --- | --- | --- |
| **Название** | **Тип** | **Описание** |
| ID\_Student | Int | Первичный ключ |
| Surname | Nvarchar(MAX) | Содержит фамилию студента |
| Name | Nvarchar(50) | Содержит имя студента |
| Patronymic | Nvarchar(50) | Содержит отчество студента |
| DateOfBirth | Date | Дата рождения |
| NumberPhone | Nvarchar(50) | Номер телефона студента |

Таблица «Session» - Содержит в себе данные о сдачи зачетов и экзаменов. Таблица представлена на рисунке 2.4

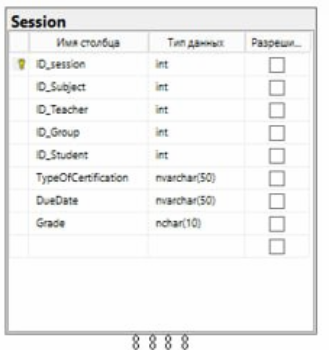


Рисунок 2.4 – Сущность «Сессия».

Таблица 5 – Описание сущности «Сессия».

|  |  |  |
| --- | --- | --- |
| **Название** | **Тип** | **Описание** |
| ID\_Session | Int | Содержит первичный ключ |
| ID\_Subject | Int | Содержит внешний ключ, ссылающийся на инденфикатор |
| ID\_Teacher | Int | Содержит внешний ключ идентификатор сущности Teacher |
| ID\_Group | Int | Содержит внешний ключ идентификатор сущности Group |
| ID\_Student | Int | Содержит внешний ключ идентификатор сущности Student |
| TypeOfCertification | Nvarchar(50) | Содержит вид зачета Зачет/Экзамен |
| DueDate | Nvarchar(50) | Содержит дату выставления оценки |
| Grade | Nchar(10) | Содержит поставленную оценку в зачет |

Таблица «Subject» - Содержит в себе данные о сдачи зачетов или же экзаменов. Таблица представлена на рисунке 2.5

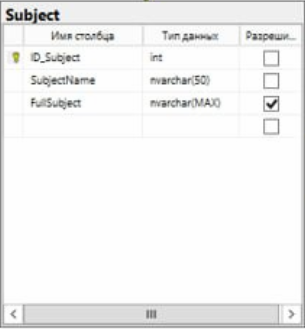


Рисунок 2.5 – Сущность «Предмет».

Таблица 6 – Описание сущности «Предмет»

|  |  |  |
| --- | --- | --- |
| **Название** | **Тип** | **Описание** |
| ID\_Subject | Int | Содержит первичный ключ, номер предмета |
| SubjectName | Nvarchar(50) | Содержит наименование предмета. |
| FullSubject | Nvarchar(MAX) | Содержит полную расшифровку предмета. |

Таблица «Teacher» - содержит все нужные данные студента. Таблица представлена на рисунке 2.6

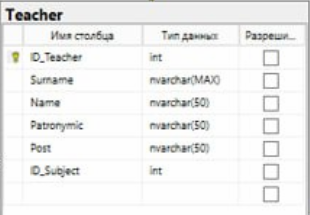


Рисунок 2.6 – Сущность «Преподаватель».

Таблица 7 – Описание сущности «Преподаватель»

|  |  |  |
| --- | --- | --- |
| **Название** | **Тип** | **Описание** |
| ID\_Student | Int | Первичный ключ, номер студента |
| Surname | Nvarchar(MAX) | Содержит фамилию преподавателя |
| Name | Nvarchar(50) | Содержит имя преподавателя |
| Patronymic | Nvarchar(50) | Содержит отчество преподавателя |
| Post | Nvarchar(50) | Содержит должность преподавателя |
| ID\_Subject | Int | Содержит внешний ключ идентификатор сущности |

Таблица «Journal» - содержит все нужные данные студента. Таблица представлена на рисунке 2.7

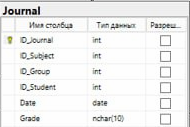


Рисунок 2.7 – Сущность «Журнал».

Таблица 8 – Описание сущности «Журнал»

|  |  |  |
| --- | --- | --- |
| **Название** | **Тип** | **Описание** |
| ID\_Journal | Int | Содержит первичный ключ |
| ID\_Subject | Int | Содержит внешний ключ |
| ID\_Group | Int | Содержит внешний ключ |
| ID\_Student | Int | Содержит внешний ключ |
| Date | Date | Содержит дату выставления оценки |
| Grade | Nchar(10) | Содержит поставленную оценку за сутки |

Таблица «Group» - содержит все нужные данные студента. Таблица представлена на рисунке 2.8

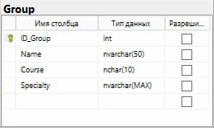


Рисунок 2.8 – Сущность «Группа».

Таблица 9 – Описание сущности «Группа»

|  |  |  |
| --- | --- | --- |
| **Название** | **Тип** | **Описание** |
| ID\_Group | int | Первичный ключ |
| Name | Nvarchar(50) | Содержит логины пользователей |
| Course | Nchar(10) | Содержит пароли |
| Speciality | Nvarchar(MAX) | Описание специальности группы |

# 2.6. Реализация интерфейса ПО

2.6.1 Разработка логотипа - входе разработки курсового проекта, был разработан логотип для моей информационной модели, логотип представлен на рисунке 2.9.

****

Рисунок 2.9 – Логотип программы «Учета успеваемости студентов техникума».

2.6.2. Выбор цветовой модели (выбор палитры/цветового круга/треугольника)

Таблица 2. Цветовая модель приложения

|  |  |  |
| --- | --- | --- |
| Основной фон | Дополнительный фон | Акцентирование  внимания |
| RGB  #FFFFFFFF | RGB  #FF55FFDD | RGB  #FF5FD1DD |
| (255, 255, 255) | (85, 255, 221) | (95, 209,211) |
|  |  |  |

При запуске приложения пользователь встречает форма «Авторизации», где пользователь может ввести данные: логин и пароль, тем самым войдя в систему. Если пользователь не имеет данные, то он может перейти через кнопку Гость в форму для пользователей без данных, форма представленный на рисунке 2.10

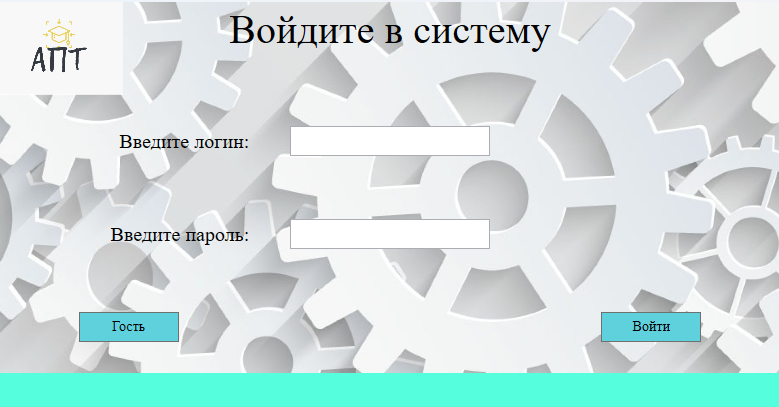


Рисунок 2.10 - Форма «Авторизации».

При входе в систему, пользователя встречает «Главная форма» приложения, на этой форме реализованы кнопки для перехода по формам программы, форма представленная на рисунке 2.11

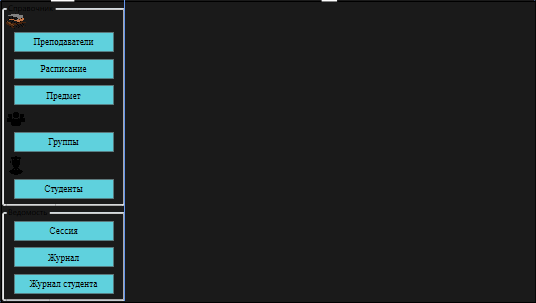


Рисунок 2.11 – «Главная форма» программы.

Форма «Гость» представляет собой страницу для ознакомления программой, а также рекомендации о поступлении в данное учебное учреждение, форма представленная на рисунке 2.12.



Рисунок 2.12. – Форма «Гость».

При нажатии на кнопку «Преподаватели», открывается форма с данными преподавателей, в котором содержится: ID преподавателя, ФИО, его должность и предмет, форма представленная на рисунке 2.13.

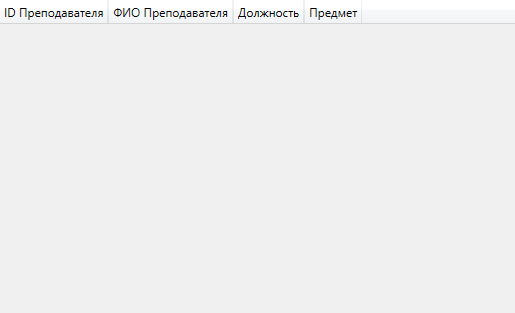


Рисунок 2.13 – Форма «Преподаватели».

При нажатии на кнопку «Расписание», открывается форма, которая выводится из excel и в ней вы можете посмотреть актуальное расписания занятий, форма представлена на рисунке 2.14.

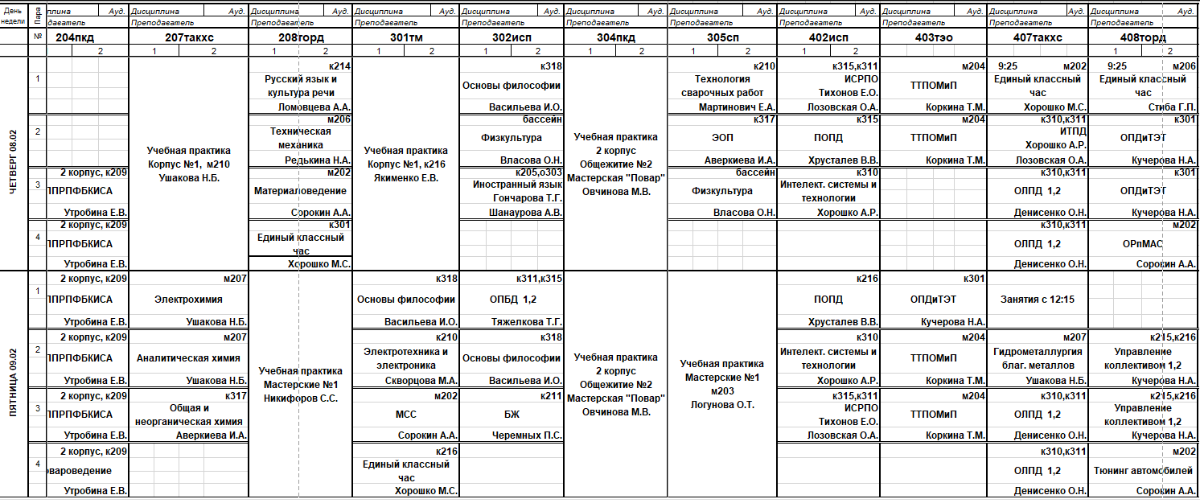


Рисунок 2.14 – Форма «Расписание».

При нажатии на кнопку «Предмет», открывается форма, в которой содержится информация о предметах, форма представлена на рисунке 2.15.

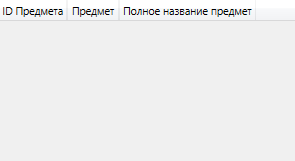


Рисунок 2.15 – форма «Предмет».

На форме «Группы» можно узнаться информацию о группах в текущий момент времени, а также увидеть их курсы, наименование и специальность, форма представлена на рисунке 2.16.



Рисунок 2.16 – Форма «Группы».

На форме «Студенты» можно увидеть информацию о студентах, обучающихся в этом учебном учреждении, увидеть номер студента, ФИО студента, его дату рождения и его номер телефона для связи с ним, форма представлена на рисунке 2.17

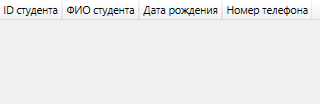


Рисунок 2.17 – Форма «Студенты».

Форма «Сессия» на ней причисляется важная информация о студентах и о их оценках в сдаче экзаменов или зачетов, форма представлена на рисунке 2.18.

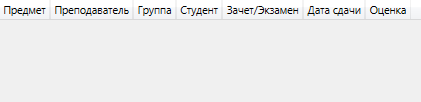


Рисунок 2.18 – Форма «Сессия»

На форме «Журнал» можно увидеть оценки учащихся по дням проведения занятий, что упрощает ведения учета успеваемости студентов техникума и обеспечивает более удобную работу, так же в этой форме можно добавлять, удалять или редактировать данные, это сделано для того чтобы предотвратить ошибки в оценках учащихся в учебном учреждении, форма представлена на рисунке 2.19.

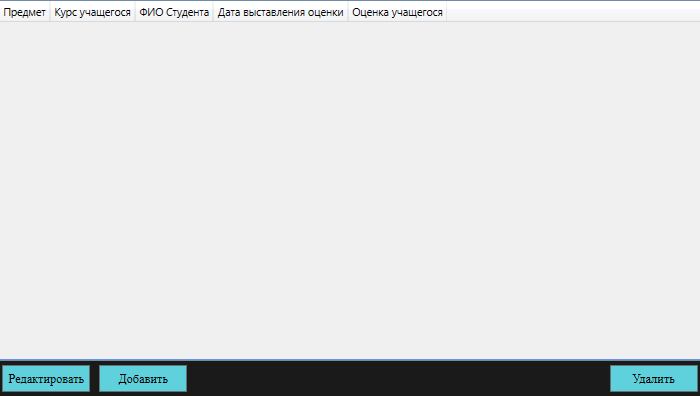


Рисунок 2.19 – Форма «Журнал».

На форме «Журнал студента» студент может ознакомится со своими текущими оценками и понять какие оценки по предметам ему нужно исправить, что дает преимущество в успеваемости студентов, форма представлена на рисунке 2.20.

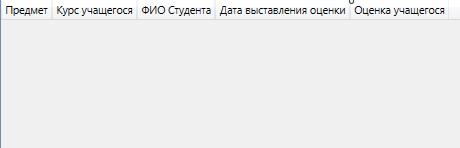


Рисунок 2.20 – Форма «Журнал студента».

# 2.7. Написание программного кода

Данный код представляет собой блок обработки авторизации пользователей в системе. Сначала происходит попытка найти пользователя в базе данных по введенному логину и паролю. Если пользователь не найден (UserObj равен null), выводится сообщение об ошибке авторизации. В противном случае, пользователь успешно аутентифицирован. Далее в зависимости от роли пользователя (хранится в поле ID\_Role) система отображает соответствующее информационное сообщение и производит навигацию на основную форму (MainForm). Если при выполнении кода возникает исключение (например, ошибка во время обращения к базе данных), оно будет обработано, и выводится уведомление с информацией об ошибке. В общем, код пытается авторизовать пользователя и в случае успеха перенаправить его на соответствующий интерфейс в системе, в противном случае сообщает об ошибке. Код для авторизации, представлен на листинге 2.1.

Листинг 2.1 – Код для авторизации в систему.

public partial class Authorization : Page { public Authorization() { InitializeComponent(); } private void BtnJoin\_Click(object sender, RoutedEventArgs e) { try { var UserObj = StudentProgressEntities.GetContext().Users.FirstOrDefault(x => x.Login == TbLogin.Text && x.Password == TbPassword.Password); if (UserObj == null) { MessageBox.Show("Такого пользователя не существует!", "Ошибка авторизации", MessageBoxButton.OK, MessageBoxImage.Error); } else { Manager.AuthUser = UserObj; switch (UserObj.ID\_Role) { case 1: MessageBox.Show("Здравствуйте, Преподаватель ", "Уведомление", MessageBoxButton.OK, MessageBoxImage.Information); Manager.MainFrame.Navigate(new MainForm()); break; case 2: MessageBox.Show("Здравствуйте, Студент ", "Уведомление", MessageBoxButton.OK, MessageBoxImage.Information); Manager.MainFrame.Navigate(new MainForm()); break; case 4: MessageBox.Show("Здравствуйте, Администратор ", "Уведомление", MessageBoxButton.OK, MessageBoxImage.Information); Manager.MainFrame.Navigate(new MainForm()); break; } } } catch (Exception Ex) { MessageBox.Show("Ошибка" + Ex.Message.ToString()+ "Критическая ошибка!", "Уведомление", MessageBoxButton.OK, MessageBoxImage.Warning); } }

Этот код для удаления данных описывает, что программа сделает, когда пользователь выберет строки для удаления и нажмёт кнопку "Удалить" и представлен на листинге 2.2:

1) Программа соберет в список все выбранные пользователем строки.

2) Программа создаст диалоговое окно с запросом на подтверждение удаления.

3) Если пользователь нажмёт "Да", программа удалит выбранные строки из базы данных.

4) Программа покажет пользователю сообщение о том, что данные удалены.

5) Программа обновит данные в списке, отобразив новые данные без удаленных строк.

6) Если во время удаления возникает какая-то ошибка, программа покажет пользователю сообщение об ошибке.

Листинг 2.2 – Код для удаления выбранных данных.

private void BtnDelete\_Click(object sender, RoutedEventArgs e) { var ObjectForDelete = DgJournal.SelectedItems.Cast<Journal>().ToList(); if(MessageBox.Show($"Вы действительно хотите удалить {ObjectForDelete.Count()} элемент", "Внимание",MessageBoxButton.YesNo, MessageBoxImage.Question)== MessageBoxResult.Yes) { try { StudentProgressEntities.GetContext().Journal.RemoveRange(ObjectForDelete); StudentProgressEntities.GetContext().SaveChanges(); MessageBox.Show("Данные удалены"); DgJournal.ItemsSource = StudentProgressEntities.GetContext().Journal.ToList(); } catch (Exception ex) { MessageBox.Show(ex.Message.ToString()); } } }

Код описывает окно в программе, которое позволяет пользователю добавлять или редактировать записи в таблице журнала оценок. Когда пользователь открывает окно, программа загружает данные из базы данных и отображает их в полях формы. Пользователь может изменить эти данные и нажать кнопку "Сохранить", чтобы сохранить изменения. Перед сохранением программа проверяет, заполнены ли все обязательные поля. Если какое-либо поле не заполнено, программа отображает пользователю сообщение об ошибке. Если все обязательные поля заполнены, программа сохраняет изменения в базе данных. Если сохранение прошло успешно, программа отображает пользователю сообщение об успешном сохранении и закрывает окно. Если при сохранении возникла ошибка, программа отображает пользователю сообщение об ошибке. Представлен на листинге 2.3.

Листинг 2.3. – Код для редактирования и добавления данных.

public partial class AddEditJournal { private Journal \_currentJournal = new Journal(); public AddEditJournal(Journal selectedJournal) { InitializeComponent(); if (selectedJournal != null) \_currentJournal = selectedJournal; DataContext = \_currentJournal; cmbGroup.ItemsSource = StudentProgressEntities.GetContext().Group.ToList(); cmbStudent.ItemsSource = StudentProgressEntities.GetContext().Student.ToList(); } private void Save\_Click(object sender, RoutedEventArgs e) { StringBuilder errors = new StringBuilder(); if (\_currentJournal.ID\_Subject <= 0) errors.AppendLine("Укажите предмет"); if (string.IsNullOrWhiteSpace(\_currentJournal.Group.Course)) errors = errors.AppendLine("Укажите курс"); if (string.IsNullOrWhiteSpace(\_currentJournal.Student.FullName)) errors = errors.AppendLine("Укажите ФИО студента"); if (clndDateOfScore.SelectedDate == null) errors = errors.AppendLine("Укажите дату"); if (string.IsNullOrWhiteSpace(\_currentJournal.Grade)) errors = errors.AppendLine("Укажите оценку"); if (\_currentJournal.ID\_Journal == 0) StudentProgressEntities.GetContext().Journal.Add(\_currentJournal); try { StudentProgressEntities.GetContext().SaveChanges(); MessageBox.Show("Информация сохранена"); Manager.SubForm.GoBack(); } catch (Exception ex) { MessageBox.Show(ex.Message.ToString()); } } }

# 2.8. Защита информации

Защита информации – важный и обязательный пункт при создании программного обеспечения. С этой задачей рано или поздно придется столкнуться каждому разработчику приложений. Каждая составляющая должна быть надежно защищена.

Чтобы понять, как защищать, важно знать, откуда берутся вирусы. Основные пути заражения:

• скачивание и установка непроверенных программ с внедренными вредоносными элементами, цель которых — перехват доступа к протоколам CMS и/или FTP с их дальнейшей обратной отправкой;

• посещение зараженных порталов;

• автоматический или ручной подбор логина и пароля злоумышленниками, используемые для входа на сервер или CMS, и заражение изнутри с помощью прописанного кода;

• использование шаблонов, плагинов и других компонентов с уязвимыми местами и дырами, через которые можно управлять ресурсом;

• действия пользователей, оставляющих вредоносный контент (ссылки, файлы);

• размещение рекламы от непроверенных источников и партнеров;

• доступ через специальные файлы внутри сайта, например, adminer.php. Это утилита для быстрого доступа к базе данных, позволяющая при подборе пароля подключиться и заполучить доступ.

**2.8.1 Безопасность Windows**

Ядро Windows предоставляет разные функции безопасности, которые формируют основу безопасности для всех приложений Windows, включая приложения, созданные с помощью WPF. Windows обеспечивает защиту путем перекомпиляции многих системных библиотек ядра, включая все зависимости WPF, такие как CLR, чтобы снизить вероятность переполнения буфера. В общих чертах, флаг компилятора -GS защищает от некоторых потенциальных переполнений буфера путем добавления специального файла cookie безопасности для защиты адреса функции, которая содержит буферы локальных строк. После возврата функции cookie безопасности сравнивается со своим предыдущим значением.

**2.8.2 Контроль учетных записей и разграничение доступа.**

Пользователи Windows предпочитают работать с правами администратора, так как многие приложения требуют такие права для установки или выполнения. Одним из примеров является возможность записи в реестр параметров приложения по умолчанию.

Работа с правами администратора означает, что приложения выполняются из процессов, которым предоставлены права администратора. Влияние этого на безопасность заключается в том, что любой вредоносный код, перехвативший процесс с правами администратора, автоматически наследует эти права, включая доступ к критически важным системным ресурсам.

Один из способов защиты от этой угрозы безопасности состоит в запуске приложений с наименьшим количеством требуемых им привилегий. В данном случае программа не требует специфических прав. За исключением доступа к панели администратора, которая требует права администратора. Эти права необходимы для разграничения доступа среди пользователей, чтобы обычные пользователи не имели доступ. А также для настроек информационной системы, таких как автозапуск, подключение к SQL серверу и т.п., всё это требует прав администратора.

**2.8.3 Развертывание ClickOnce**

ClickOnce — это комплексная технология развертывания, которая входит в состав .NET Framework и интегрируется с Visual Studio.

Приложения, развернутые с помощью ClickOnce, приобретают дополнительный уровень безопасности посредством управления доступом для кода. Приложения, развернутые с помощью ClickOnce, запрашивают разрешения, которые им необходимы. Им предоставляются только разрешения, не расширяющие набор разрешений для зоны, в которой развертывается приложения. Таким образом, уменьшается вероятность повреждений на клиентском компьютере в случае перехвата приложения.

**2.8.4 Резервное копирование базы данных (BackUp)**

Регулярно создаются резервные копии всех данных сайта, включая базы данных, файлы конфигурации и контент. Это обеспечивает возможность восстановления информации в случае ее случайного удаления, атаки злоумышленников или других инцидентов, которые могут привести к потере данных. Резервные копии хранятся на отдельных защищенных серверах или в облачном хранилище с ограниченным доступом только для авторизованных сотрудников. Проводится регулярная проверка работоспособности процесса создания резервных копий и их целостности, чтобы гарантировать их доступность в случае необходимости восстановления данных.

Обработчик ошибки авторизации, если пользователь приложения неправильно указал свой логин или пароль, то специально написанный код ошибки выдаст ему сообщение «Такого пользователя не существует!». Представлен на листинге 2.4

Листинг 2.4 – Показ ошибки «Ошибка авторизации».

if (UserObj == null) { MessageBox.Show("Такого пользователя не существует!", "Ошибка авторизации", MessageBoxButton.OK, MessageBoxImage.Error); }

Если у пользователя не получилось авторизоваться в приложении, то написанный код представленный на листинге 2.5, выдаст ему сообщение о критические ошибки и не впускает пользователя дальше окна авторизации.

Листинг 2.5 – Показ «Критической ошибки».

catch (Exception Ex) { MessageBox.Show("Ошибка" + Ex.Message.ToString()+ "Критическая ошибка!", "Уведомление", MessageBoxButton.OK, MessageBoxImage.Warning); }

# Заключение

Разработанный модуль призван решить ряд важных задач, связанных с учебным процессом:

* Повышение эффективности взаимодействия преподавателей и студентов: Модуль предоставляет удобную платформу для проведения и подготовки к учебным занятиям. Это позволяет преподавателям и студентам более эффективно взаимодействовать друг с другом и повышает качество образовательного процесса.
* Повышение доступности информации об оценках учащихся: Модуль предоставляет студентам и преподавателям круглосуточный доступ к информации об оценках, что повышает прозрачность и объективность процесса оценивания.
* Повышение уровня доступности к предоставленной информации: Модуль разработан с учетом принципов инклюзивности и доступности, что обеспечивает равный доступ к информации для всех пользователей, независимо от их способностей или ограничений.

Таким образом, разработанный модуль является ценным инструментом для повышения эффективности и доступности учебного процесса. Он помогает преподавателям и студентам более эффективно взаимодействовать друг с другом, повышает прозрачность процесса оценивания и обеспечивает равный доступ к информации для всех пользователей.

Цели и поставленные задачи выполнены:

1. Провел предварительное планирование и проверку осуществимости.
2. Определил требования к программному продукту.
3. Разработал прототип приложения.

4. Реализовал приложение в выбранной среде IDE.

5. Разместить проект на веб-сервисе GitHub или аналогичном.

6.Разместил проект на веб-сервисе GitHub - https://github.com/whit3full/StudentProgress.git.

# Список использованных источников

1. Документация по Windows Presentation Foundation [Электронный ресурс]. URL: <https://docs.microsoft.com/ru-ru/dotnet/desktop/wpf/>
2. Документация по C# [Электронный ресурс]. URL: <https://docs.microsoft.com/ru-ru/dotnet/csharp/>
3. Техническая документация по SQL Server [Электронный ресурс]. URL: <https://docs.microsoft.com/ru-ru/sql/sql-server/>
4. **Литвиненко Н.А.** Декларативное программирование на языке XAML + C#. WPF проекты. М.: Горячая Линия – Телеком, 2022. – 320 с.
5. **Рихтер Джеффри** CLR via C#. Программирование на платформе Microsoft .NET Framework 4.5 на языке C#. 4-е изд. М.: Прогресс книга,   
   2022. – 896 с.

**Диалектика-Вильямс.** Язык T-SQL для Microsoft SQL Server за 10 минут. WPF проекты., 2020. – 384 с.**Интеренет-источники:**

1. Волобуева, Т. В. Информатика. Основы алгоритмизации: учебное пособие / Т. В. Волобуева. — Воронеж: Воронежский государственный технический университет, ЭБС АСВ, 2019. — 73 c. — ISBN 978-5-7731-0740-8. — Текст: электронный // Электронный ресурс цифровой образовательной среды СПО PROFобразование: [сайт]. — URL: <https://profspo.ru/books/93316> (дата обращения: 28.02.2022). — Режим доступа: для авторизир. пользователей

# Приложение

**Приложение А**

**Листинги программного кода**

**MainWindow.xaml**

<Window x:Class="StudentProgress.MainWindow" xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation" xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml" xmlns:d="http://schemas.microsoft.com/expression/blend/2008" xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006" xmlns:local="clr-namespace:StudentProgress" mc:Ignorable="d" Title="Учет успеваемости" Height="450" Width="800" MinHeight="450" MinWidth="800" Icon="/Resources/APT.png"> <Grid> <Grid.RowDefinitions> <RowDefinition Height="1\*"/> <RowDefinition Height="40"/> </Grid.RowDefinitions> <Frame ContentRendered="FrameMain\_ContentRendered" NavigationUIVisibility="Hidden" x:Name="FrameMain" Grid.Row="0"/> <Grid Grid.Row="1" Background="#5FD"/> <Button x:Name="ButBack" Content="Назад" Click="ButBack\_Click" Grid.Row="1" HorizontalAlignment="Right"/> </Grid> </Window>

**MainWindow.xaml.cs**

public partial class MainWindow { public MainWindow() { InitializeComponent(); Manager.MainFrame = FrameMain; FrameMain.Navigate(new Authorization()); } private void ButBack\_Click(object sender, RoutedEventArgs e) { Manager.MainFrame.GoBack(); } private void FrameMain\_ContentRendered(object sender, EventArgs e) { if (FrameMain.CanGoBack) { ButBack.Visibility = Visibility.Visible; } else { ButBack.Visibility = Visibility.Hidden; } } }

**Authorization.xaml**

<Page x:Class="StudentProgress.Pages.Authorization" xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation" xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml" xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006" xmlns:d="http://schemas.microsoft.com/expression/blend/2008" xmlns:local="clr-namespace:StudentProgress.Pages" mc:Ignorable="d" d:DesignHeight="450" d:DesignWidth="800" Title="Authorization"> <Grid> <Grid.RowDefinitions> <RowDefinition/> <RowDefinition/> <RowDefinition/> <RowDefinition/> </Grid.RowDefinitions> <Grid.ColumnDefinitions> <ColumnDefinition/> <ColumnDefinition/> <ColumnDefinition/> </Grid.ColumnDefinitions> <Image Source="/Resources/фон.jpg" Grid.ColumnSpan="3" Margin="-107,-257,-472,-303" Grid.RowSpan="4"/> <Image Source="/Resources/APT.png" HorizontalAlignment="Left"/> <Label Content="Войдите в систему" Grid.ColumnSpan="3" HorizontalAlignment="Center" FontSize="40" Style="{StaticResource Lab}"/> <Label Content="Введите логин: " Grid.Row="1" Style="{StaticResource Lab}" HorizontalAlignment="Right" VerticalAlignment="Center"/> <Label Content="Введите пароль: " Grid.Row="2" Style="{StaticResource Lab}" HorizontalAlignment="Right" VerticalAlignment="Center"/> <Button Content="Войти" Grid.Row="3" Grid.Column="2" x:Name="BtnJoin" Click="BtnJoin\_Click"/> <Button x:Name="BtnGuest" Content="Гость" Grid.Row="3" Click="BtnGuest\_Click"/> <TextBox x:Name="TbLogin" HorizontalContentAlignment="Left" VerticalContentAlignment="Center" Grid.Column="1" Grid.Row="1" Style="{StaticResource TxB}"/> <PasswordBox x:Name="TbPassword" HorizontalContentAlignment="Left" VerticalContentAlignment="Center" Grid.Column="1" Grid.Row="2" Style="{StaticResource PwB}" Visibility="Visible"/> </Grid> </Page>

**Authorization.xaml.cs**

public Authorization() { InitializeComponent(); } private void BtnJoin\_Click(object sender, RoutedEventArgs e) { try { var UserObj = StudentProgressEntities.GetContext().Users.FirstOrDefault(x => x.Login == TbLogin.Text && x.Password == TbPassword.Password); if (UserObj == null) { MessageBox.Show("Такого пользователя не существует!", "Ошибка авторизации", MessageBoxButton.OK, MessageBoxImage.Error); } else { Manager.AuthUser = UserObj; switch (UserObj.ID\_Role) { case 1: MessageBox.Show("Здравствуйте, Преподаватель ", "Уведомление", MessageBoxButton.OK, MessageBoxImage.Information); Manager.MainFrame.Navigate(new MainForm()); break; case 2: MessageBox.Show("Здравствуйте, Студент ", "Уведомление", MessageBoxButton.OK, MessageBoxImage.Information); Manager.MainFrame.Navigate(new MainForm()); break; case 4: MessageBox.Show("Здравствуйте, Администратор ", "Уведомление", MessageBoxButton.OK, MessageBoxImage.Information); Manager.MainFrame.Navigate(new MainForm()); break; } } } catch (Exception Ex) { MessageBox.Show("Ошибка" + Ex.Message.ToString()+ "Критическая ошибка!", "Уведомление", MessageBoxButton.OK, MessageBoxImage.Warning); } } private void BtnGuest\_Click(object sender, RoutedEventArgs e) { Manager.MainFrame.Navigate(new GuestPage()); } }

**MainForm.xaml**

<Page x:Class="StudentProgress.Pages.MainForm" xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation" xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml" xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006" xmlns:d="http://schemas.microsoft.com/expression/blend/2008" xmlns:local="clr-namespace:StudentProgress.Pages" mc:Ignorable="d" d:DesignHeight="450" d:DesignWidth="800" Title="MainForm"> <Grid> <Grid.ColumnDefinitions> <ColumnDefinition Width="185"/> <ColumnDefinition Width="3\*"/> </Grid.ColumnDefinitions> <StackPanel> <GroupBox Grid.Column="0" Header="Справочник"> <StackPanel> <Image Source="/Resources/Book.png" Width="30" HorizontalAlignment="Left"/> <Button Content="Преподаватели" Width="150" Visibility="Collapsed" x:Name="btnTeachers" Click="Items\_Click"/> <Button Content="Расписание" Width="150" Visibility="Collapsed" x:Name="btnSchedule" Click="btnSchedule\_Click"/> <Button Content="Предмет" Width="150" Visibility="Collapsed" x:Name="btnSubject" Click="btnSubject\_Click"/> <Image Source="/Resources/Group.png" Width="30" HorizontalAlignment="Left"/> <Button Content="Группы" Width="150" Visibility="Collapsed" x:Name="btnGroup" Click="Group\_Click"/> <Image Source="/Resources/Student.png" Width="30" HorizontalAlignment="Left"/> <Button Content="Студенты" Width="150" Visibility="Collapsed" x:Name="btnStudent" Click="Student\_Click"/> </StackPanel> </GroupBox> <GroupBox Grid.Column="0" Header="Ведомость"> <StackPanel> <Button Content="Сессия" Width="150" Visibility="Collapsed" x:Name="btnSession" Click="Session\_Click"/> <Button Content="Журнал" Width="150" Visibility="Collapsed" x:Name="btnJournal" Click="Journal\_Click"/> <Button Content="Журнал студента" Width="150" Visibility="Collapsed" x:Name="btnJournalStudent" Click="btnJournalStudent\_Click"/> </StackPanel> </GroupBox> </StackPanel> <Frame x:Name="SubForm" Grid.Column="1"/> </Grid> </Page>

**MainForm.xaml.cs**

public MainForm() { InitializeComponent(); Manager.SubForm = SubForm; if(Manager.AuthUser.ID\_Role == 1) { btnStudent.Visibility = Visibility.Visible; btnGroup.Visibility = Visibility.Visible; btnJournal.Visibility = Visibility.Visible; btnSession.Visibility = Visibility.Visible; } else if (Manager.AuthUser.ID\_Role == 2) { btnSchedule.Visibility = Visibility.Visible; btnJournalStudent.Visibility= Visibility.Visible; } else { btnStudent.Visibility = Visibility.Visible; btnGroup.Visibility = Visibility.Visible; btnSchedule.Visibility = Visibility.Visible; btnJournal.Visibility = Visibility.Visible; btnSession.Visibility = Visibility.Visible; } } private void Items\_Click(object sender, RoutedEventArgs e) { Manager.SubForm.Navigate(new TeacherPage()); } private void Group\_Click(object sender, RoutedEventArgs e) { Manager.SubForm.Navigate(new GroupPage()); } private void Student\_Click(object sender, RoutedEventArgs e) { Manager.SubForm.Navigate(new Student()); } private void Session\_Click(object sender, RoutedEventArgs e) { Manager.SubForm.Navigate(new Session()); } private void Journal\_Click(object sender, RoutedEventArgs e) { Manager.SubForm.Navigate(new PageJournal()); } private void btnSchedule\_Click(object sender, RoutedEventArgs e) { Manager.SubForm.Navigate(new PageSchedule()); } private void btnSubject\_Click(object sender, RoutedEventArgs e) { Manager.SubForm.Navigate(new ItemsPage()); } private void btnJournalStudent\_Click(object sender, RoutedEventArgs e) { Manager.SubForm.Navigate(new PageStudentJournal()); } }