Содержание

[Введение 4](#_Toc169161139)

[1. Аналитическая часть 6](#_Toc169161140)

[1.1 Описание предметной области 6](#_Toc169161141)

[1.2 Анализ существующих разработок 7](#_Toc169161142)

[1.3 Требования, предъявляемые к разрабатываемой ИС 9](#_Toc169161143)

[1.4 Инфологическая модель предметной области 12](#_Toc169161144)

[1.5 Модель предметной области в стандарте IDEF0 13](#_Toc169161145)

[1.6 Диаграмма DFD предметной области 16](#_Toc169161146)

[1.7 Прототипирование интерфейса 17](#_Toc169161147)

[2. Практическая часть 21](#_Toc169161148)

[2.1. Обзор инструментальных средств для разработки ИС 21](#_Toc169161149)

[2.2 Обоснование и выбор программного обеспечения для разработки ИС 26](#_Toc169161150)

[2.3. Описание логики функционирования ИС 28](#_Toc169161151)

[2.4. Реализация интерфейса ИС 33](#_Toc169161152)

[2.5. Написание программного кода 39](#_Toc169161153)

[2.7. Тестирование ИС 41](#_Toc169161154)

[2.8. Руководство пользователя ИС 44](#_Toc169161155)

[Заключение 48](#_Toc169161156)

[Список использованных источников 49](#_Toc169161157)

[Приложение 50](#_Toc169161158)

[Приложение А 50](#_Toc169161159)

# Введение

В современном обществе информационные технологии проникают во все сферы деятельности, включая образование. Учет успеваемости студентов становится ключевым элементом образовательного процесса, оказывая существенное влияние на формирование и развитие будущих специалистов. Особенно важно это в контексте техникума, где студенты приобретают специфические профессиональные навыки, от которых зависит их будущая карьера.

При учете 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. 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 дипломного проекта – разработка автоматизированной и эффективной информационной системы для учета успеваемости студентов техникума

Актуальность дипломного проекта – 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 и автоматизации учебного процесса для учебного заведения.

Объект – автоматизация процесса введение журнала и учета успеваемости студентов.

Предметом исследования является процесс учета успеваемости студентов

В соответствии с выше поставленной целью, для ее достижения необходимо выполнить следующиезадачи:

1. Спроектировать систему;
2. Разработать инфологическую модель предметной области;
3. Разработать модель предметной области в стандарте IDEF0;
4. Разработать DFD-диаграммы;
5. Изучить средства создания информационной системы и выбрать из них более подходящие;
6. Провести разработку базы данных и информационной системы, осуществить подключение базы данных к информационной системе;
7. Обеспечить защиту информации;
8. Провести оценку эффективности разработки информационной системы.

Проект будет состоять из двух частей, аналитической и практической:

В аналитической части будет рассмотрена структура и порядок работы, проведён анализ схожих информационных системы, приведены диаграммы, спроектированы прототипы информационной системы.

В практической части будут описаны выбор и его обоснование средств создания информационной системы, процесс разработки информационной системы, описание функционала, проведено тестирование информационной системы и будет предоставлен итоговый результат

# Аналитическая часть

# Описание предметной области

Предметная область “Учет успеваемости студентов техникума” включает в себя сбор, обработку и хранение информации об успеваемости студентов, а также предоставление доступа к этой информации различным пользователям, таким как преподаватели, администрация техникума и сами студенты.

В рамках этой предметной области осуществляется регистрация и хранение данных об успеваемости студентов, включая оценки по предметам, посещаемость занятий и другую информацию. Также обеспечивается возможность формирования отчетов и статистических данных об успеваемости студентов, преподавателей и групп.

Для автоматизации процессов учета успеваемости используются различные информационные системы и программное обеспечение, которые позволяют упростить работу с данными и повысить эффективность управления учебным процессом.

Кроме того, автоматизация учета успеваемости способствует повышению прозрачности и объективности оценки знаний студентов. Это помогает избежать субъективных оценок и предвзятого отношения со стороны преподавателей, что положительно сказывается на мотивации студентов и их стремлении к саморазвитию.

Таким образом, автоматизация предметной области «Учет успеваемости студентов техникума» является необходимым условием для успешного функционирования учебного заведения и обеспечения качественного образования для всех студентов.

# Анализ существующих разработок

1. Учебный учет (https://pbprog.ru/catalog/health-and-development/411)

Учебный отчет представляет собой программу для учета посещаемости и успеваемости студентов в организациях среднего профессионального и высшего образования.

Помогает контролировать учебный процесс в образовательной организации – программа позволяет подготовить электронные журналы, сформировать отчеты посещаемости занятий и текущей успеваемости. Печать отчета по прогулам: отчет по прогулам студентов в разрезе групп, отчет по прогулам студентов в разрезе преподавателей, отчет по прогулам студентов, выбранной группы. Наглядное представление качественной успеваемости по группам и специальностям в виде графиков и диаграмм. Также предоставления прозрачности оценок учащихся. Интерфейс программы представлен на рисунке 1.1.

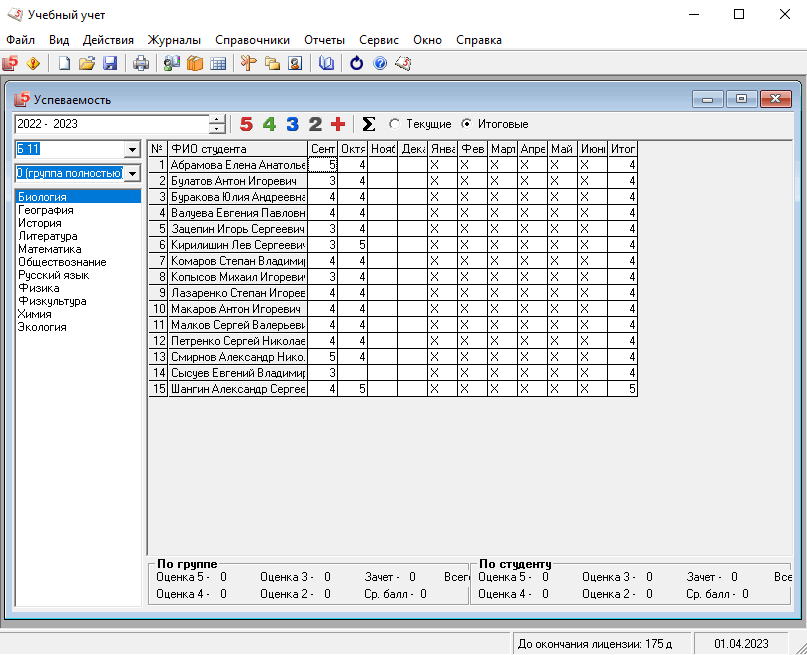


Рисунок 1.1 – Интерфейс приложения “Учебный учет”

2) Дневник.ру. (<https://dnevnik.ru/>)

Дневник.ру - это российская образовательная онлайн-платформа, предназначенная для ведения школьного дневника, взаимодействия между учителями, учениками и их родителями, а также предоставления учебных материалов.

На сайте Дневник.ру учащиеся и родители могут узнать о домашних заданиях, оценках, планах уроков, а также общаться с учителями. Учителя, в свою очередь, могут планировать и проводить уроки, выставлять оценки и вести дневники прогресса учащихся.

Сайт используется для упрощения образовательного процесса, улучшения коммуникации между учителями, учениками и их родителями, а также для обеспечения доступа к учебным материалам в любое время и из любой точки мира. Ученики могут использовать сайт Дневник.ру для доступа к домашним заданиям, расписанию уроков, просмотра своих оценок, а также общения с учителями и друзьями.

Родители могут отслеживать успехи своих детей, получать уведомления о домашних заданиях, оценках и различных событиях в школе.

Учителя используют платформу Дневник.ру для планирования и проведения уроков, выставления оценок, ведения журналов успеваемости, а также для общения с родителями и учениками. Дневник.ру представленный на рисунок 1.2

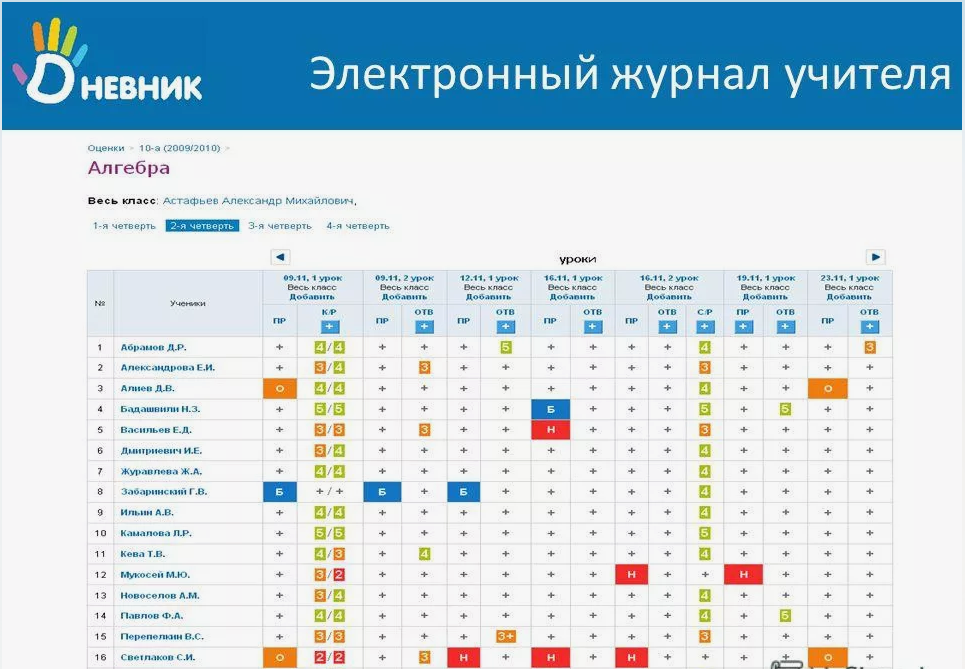


Рисунок 1.2 – Интерфейс сайта “Дневник.ру”

Таблица 1 – Cравнение

|  |  |  |
| --- | --- | --- |
| Наименование | Дневник.ру | Учебный учет |
| Интерфейс | Интуитивно понятен, но можно запутаться в датах и урока | Маленькие иконки для переходов на другие формы |
| Информация | Не всегда полная и достоверная информация. Иногда сайт публикует непроверенные или искаженные факты | Информация точная и без ошибочная |
| Вид программы | Сайт содержит рекламные блоки, которые могут отвлекать от чтения статей и создавать негативное впечатление от пользовательского опыта | Не содержит рекламных блоков, не всегда понятно какие формы к чему относятся |

Исходя из анализа, для создания своего приложения следует уделить внимание к удобству и интуитивно понятному дизайну, а также широкому информационному содержанию и прозрачности в оценках. Кроме того, важно обеспечить высокую точность оценивания и эффективную онлайн-поддержку для удовлетворения потребностей пользователей.

# Требования, предъявляемые к разрабатываемой ИС

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.

Требования к численности и квалификации персонала

Пользователями информационной системы 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;

• Сотрудники техникума.

Пользователи должны иметь минимальные навыки работы на компьютере.

Обслуживающий персонал должен иметь опыт работы с SQL Server, владеть языком программирования C# и уметь работать в среде Visual Studio 19 и выше.

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

Требования к надежности

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

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.

Требования к защите информации от несанкционированного доступа

Защита ИС и информации, с которой она работает, от несанкционированного доступа возлагается частично на операционную систему и специализированные программные и программно–аппаратные продукты, частично на систему аутентификации, реализованную в компонентах ИС, которая разгневает доступ к данным в зависимости от их роли. Доступ к данным предоставляется только легальным пользователям.

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.

Требования к защите от влияния внешних воздействий

Автоматизированная система должна быть устойчивой к параллельной работе нескольких пользователей.

Для уменьшения влияния перебоев с электропитанием рекомендуется питание «Сервера БД» производить через блок бесперебойного питания.

Требования к стандартизации и унификации

Информационная система должен предоставлять пользователю привычный, общепринятый в среде Microsoft Windows интерфейс. Программная документация, поставляемая с программой, должна быть оформлена в соответствии со стандартом ЕСПД.

Требования к лингвистическому обеспечению

В качестве языка программирования используются – T-SQL, XAML и С#, взаимодействие пользователя и информационной системы осуществляется через оконный интерфейс.

Требования к программному обеспечению

Система должна иметь следующее программное обеспечение: ОС MS Windows 10 или выше и SQL Server.

Требования к техническому обеспечению

Для работоспособности информационной системы необходимы следующие программные и технические средства:

• клавиатура;

• компьютерная мышь;

• процессор Intel Core i3 2200 и выше;

• оперативная память 2 Гб и больше;

• видеокарт 2 Гб и больше;

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

• дисковое пространство 5 Гб и больше;

# Инфологическая модель предметной области

Инфологическая модель представляет семантику. Она должна включать в себя емкое описание предметной области, в котором можно было бы оценить корректность разработки проекта.

В инфологической модели содержится информация сущностях системы (объектах, имеющих важность в предметной области), атрибуты сущностей (свойства) и связями (отношениями) их с другими сущностями.

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 1.3.

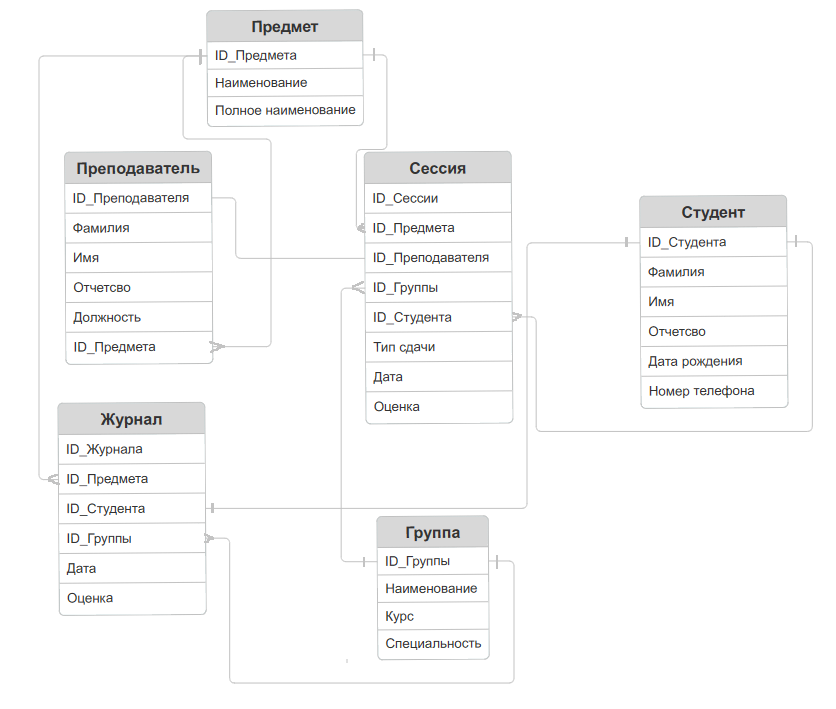


Рисунок 1.3 – Инфологическая модель “Учет успеваемости студентов техникума”

# Модель предметной области в стандарте IDEF0

На рисунках 1.4, 1.5 функциональной модели показано, что после автоматизации системы учета посещаемости и успеваемости студентов многое меняется. Данные о студенте, преподавателе вносятся в систему самостоятельно. В системе учета успеваемости и посещаемости студентов преподаватель проставляет посещаемость студентов своей дисциплины, вносит количество лабораторных работ по плану, сданные лабораторные работы студента, проставляет аттестацию студентов, ставит свои часы приема студентов (например, должников). Студент может в системе просмотреть учебную нагрузку по дисциплине, оценить свою посещаемость, количество выполненных работ на текущий момент, узнать часы приема преподавателя. Чтобы отследить статистику студента не нужно перебирать бумаги от преподавателей с посещением, достаточно зайти в систему.

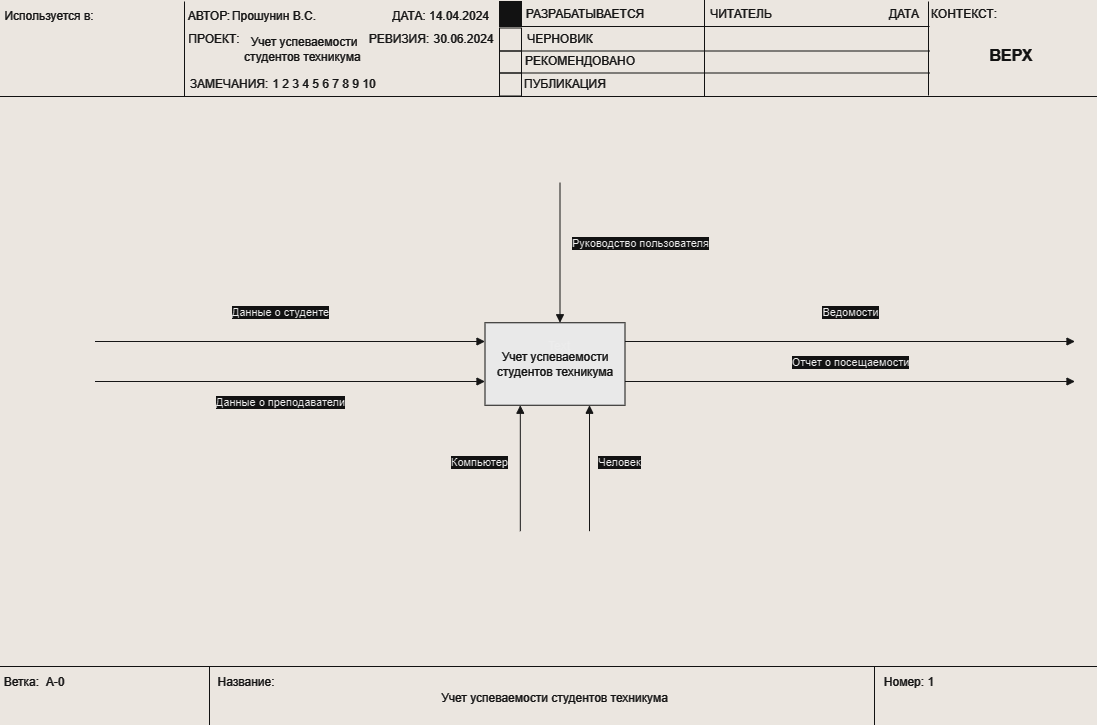


Рисунок 1.4 – Контекстная диаграмма в стандарте IDEF0.

Таблица 2 – Характеристики объектов первого уровня функциональной модели учет успеваемости студентов.

|  |  |  |
| --- | --- | --- |
| Объект | Название | Описание |
| Функция А0 | Учет успеваемости студентов техникума | система оценки успеваемости и качества знаний студентов является комплексная оценка качества учебной работы студентов в процессе обучения. |
| Вход А0 | Данные о студенте | Информация, содержащая данные о студенте, его идентификатор, ФИО и различные данные. |
| Вход А1 | Данные об оборудовани | Информация, содержащая данные о преподавателе, его идентификатор, ФИО и наименование предмета который преподает. |
| Управление А0 | Руководство пользователя | Установленные правила и нормы работы с приложением. |
| Механизм А0 | Компьютер | Содержит, анализирует и обрабатывает значения которые вводит пользователь |
| Механизм А1 | Человек | Вводит значения оценивания и посещяемости |
| Выход А0 | Ведомость | Результат обучающихся за семестр. |
| Вход А0 | Данные о студенте | Информация, содержащая данные о студенте, его идентификатор, ФИО и различные данные. |

Декомпозиция функциональной модели – это декомпозиция, выполненная в соответствии с функциями, выполняющимися организациями или людьми. Декомпозицию применяют при разборе процесса на функции, которые его составляют. Модель реаизована в среде программы MS Visio. Диаграммы декомпозиции представлена на рсиунке 1.5.

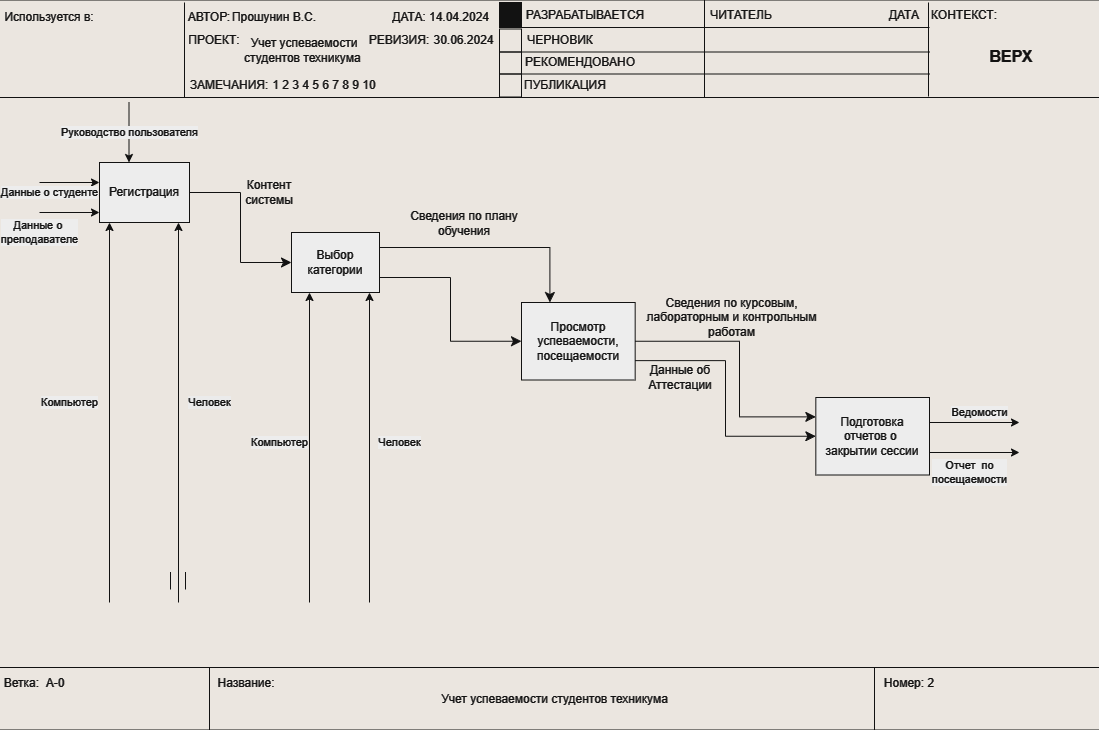


Рисунок 1.5 - Диаграмма декомпозиции в стандарте IDEF0

Таблица 3 – Характеристики объектов второго уровня функциональной модели учета успеваемости студентов техникума.

|  |  |  |
| --- | --- | --- |
| Объект | Название | Описание |
| Функция А1 | Регистрация | Заполнение заявки пользователем и последующие отправление. |
| Функция А2 | Выбор категории | Выбор формы отчетности, оценочная форма или форма посещаемости |
| Функция А3 | Просмотр успеваемости, посещаемости | Оценка деятельности учащихся в техникуме |
| Функция А4 | Подготовка отчета о закрытии сессии | Анализ и подвидения итогов обучающихся |

Продолжение таблицы 3

|  |  |  |
| --- | --- | --- |
| Вход А0 | Данные о студенте | Информация, содержащая данные о студенте, его идентификатор, ФИО и различные данные. |
| Вход А1 | Данные об оборудовани | Информация, содержащая данные о преподавателе, его идентификатор, ФИО и наименование предмета который преподает. |
| Управление А0 | Руководство пользователя | Установленные правила и нормы работы с приложением. |
| Механизм А0 | Компьютер | Содержит, анализирует и обрабатывает значения которые вводит пользователь |
| Механизм А1 | Человек | Вводит значения оценивания и посещяемости |
| Выход А0 | Ведомость | Результат обучающихся за семестр. |
| Выход А1 | Отчет о посещаемости | Результат посещаемости студентов. |

# Диаграмма DFD предметной области

Диаграмма потоков данных DFD – это нотация, которая является основным средством моделирования функциональных требований проектируемой системы. Для моделирования информационный систем с точки зрения хранения, обработки и передачи данных.

В этой нотации отображены данные, напрямую связанные с обработкой заявки. Пользователь получает техническую поддержку, указывает необходимые данные в заявке, после этого они передаются специалисту для дальнейшего анализа и оказания технической поддержки. Контекстная диаграмма потоков данных (DFD) изображена на рисунке 1.6.

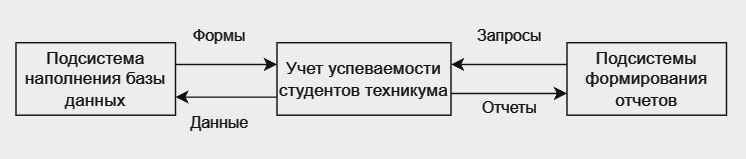


Рисунок 1.6 - Декомпозиция DFD 1-го уровня.

На рисунке 1.7 представлена диаграмма декомпозиции в нотации DFD, описывающая деятельность учета успеваемости студентов техникума.

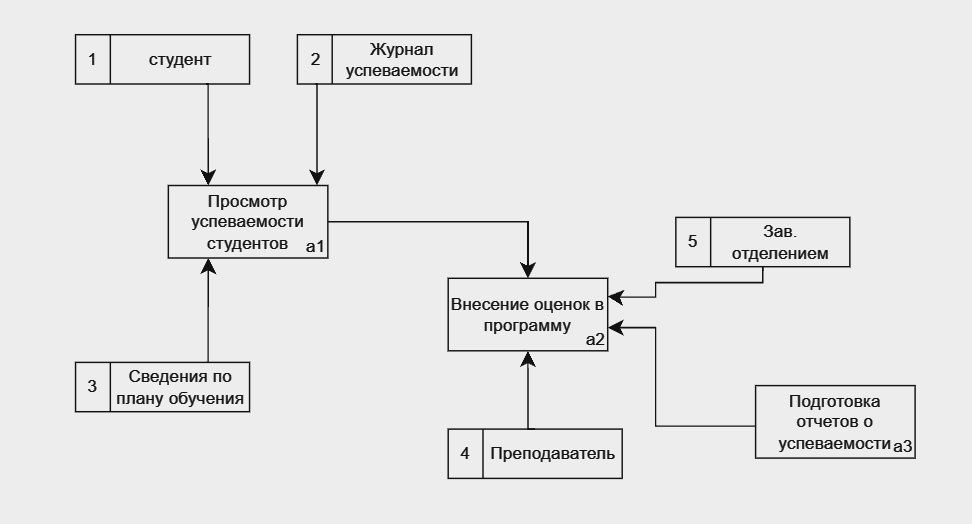


Рисунок 1.8 - Декомпозиция DFD 2-го уровня.

# 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. Авторизация используется для идентификации пользователя и предоставления ему доступа к ресурсам или функциями системы, которые требуют определенных прав или разрешений. Например, при входе на сайт или приложение, пользователь должен будет авторизоваться, введя свой логин и пароль. После успешной авторизации он получит доступ к личному кабинету или другим функциям, которые доступны только зарегистрированным пользователям. Без авторизации возможны утечки данных, несанкционированный доступ к личным информации и другие угрозы для безопасности. Поэтому авторизация является важным элементом в процессе работы с информацией и системами. Представлена на рисунке 1.6.

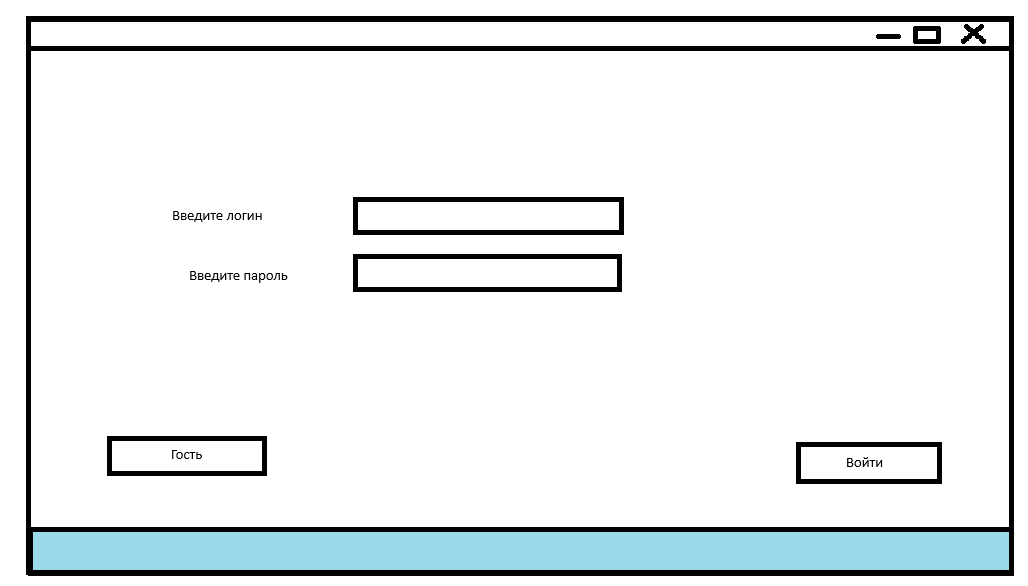


Рисунок 1.6 – Форма “Авторизация в систему”

Главная форма, создана для удобства в пользовании программы и упрощения в перемещении между формами. Главная форма в приложении является визуальным интерфейсом, через который пользователь может взаимодействовать с функциональностью приложения. Она обычно содержит основное меню, элементы управления, информационные блоки и другие элементы, необходимые для работы с приложением. Представлен на рисунке 1.7.

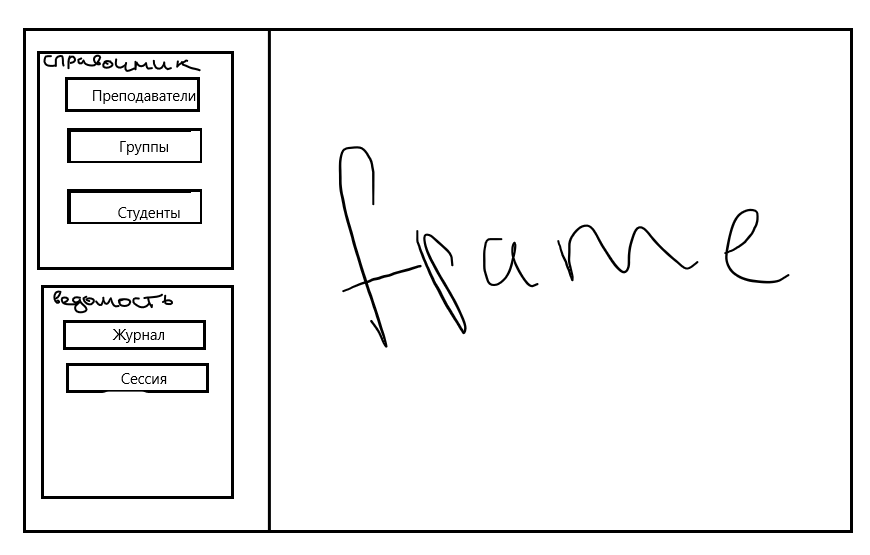


Рисунок 1.7 – Прототип “Главная форма”

Форма редактирования и добавления. Данная форма создана для работы с данными, их редактирование удаление и добавления. Форма редактирования и добавления играет важную роль в процессе работы с приложением, так как она предоставляет пользователю удобное средство для управления данными и информацией. С ее помощью пользователь может быстро и эффективно вносить изменения, обновлять данные или создавать новые записи, что делает работу с приложением более удобной и продуктивной. Кроме того, форма редактирования и добавления помогает снизить вероятность ошибок и повышает точность данных, так как пользователям предоставляется структурированное и интуитивно понятное средство для взаимодействия с информацией. Это также способствует повышению удовлетворенности пользователей и общему уровню удобства использования приложения. Представлена на рисунке 1.8.

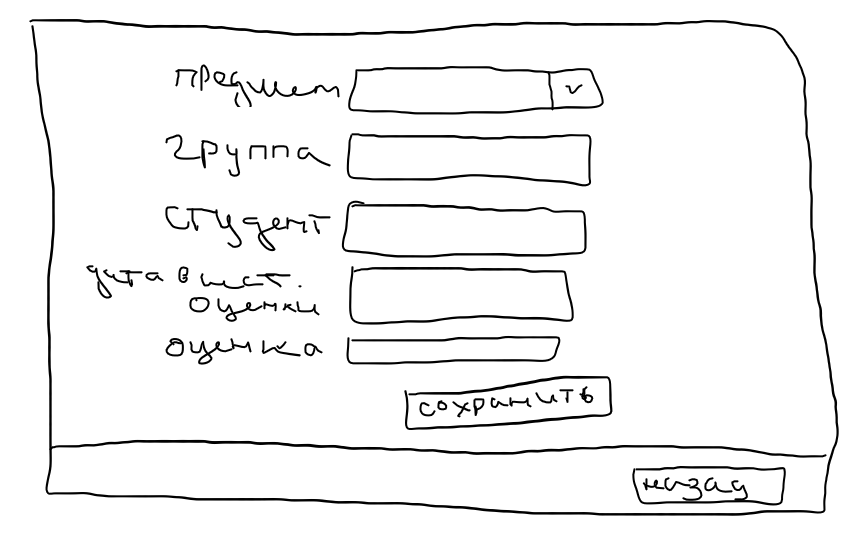


Рисунок 1.8 – Форма “Редактирования и добавления”

# 2. Практическая часть

# 2.1. Обзор инструментальных средств для разработки ИС

В настоящее для создания приложений используется множество средств программирования. Это языки программирования, различные среды разработки, используются шаблоны и инструменты для базы данных и многое другое.

Наибольшей популярностью при написании приложений пользуются такие языки программирования, как:

• C# – объектно-ориентированный язык программирования. C# относится к семье языков с C-подобным синтаксисом, из них его синтаксис наиболее близок к C++ и Java. Язык имеет статическую типизацию, поддерживает полиморфизм, перегрузку операторов (в том числе операторов явного и неявного приведения типа), делегаты, атрибуты, события, переменные, свойства, обобщённые типы и методы, итераторы, анонимные функции с поддержкой замыканий, LINQ, исключения, комментарии в формате XML.

• C++ – компилируемый, статически типизированный язык программирования общего назначения. Поддерживает такие парадигмы программирования, как процедурное программирование, объектно-ориентированное программирование, обобщённое программирование. Язык имеет богатую стандартную библиотеку, которая включает в себя распространённые контейнеры и алгоритмы, ввод-вывод, регулярные выражения, поддержку многопоточности и другие возможности. C++ сочетает свойства как высокоуровневых, так и низкоуровневых языков.

• Java – кроссплатформенный, строго типизированный объектно-ориентированный язык программирования общего назначения. Зачастую используется для написания серверного программного обеспечения, драйверов, мобильных и мультимедийных приложений. Программы на Java транслируются в байт-код, который затем выполняется виртуальной машиной Java (JVM). JVM — это программа, которая обрабатывает байтовый код и передает инструкции оборудованию как интерпретатор. Достоинством подобной реализации является независимость байт-кода от операционной системы и оборудования, что позволяет выполнять Java-приложения на любом устройстве, для которого существует JVM.

• Python – высокоуровневый язык программирования общего назначения, который используется в том числе и для разработки веб-приложений. Язык ориентирован на повышение производительности разработчика и читаемости кода. Поддерживает несколько парадигм программирования: структурное, объектно-ориентированное, функциональное, императивное и аспектно-ориентированное. В языке присутствует динамическая типизация, автоматическое управление памятью, полная интроспекция, механизм обработки исключений, поддержка многопоточных вычислений и удобные высокоуровневые структуры данных. Программный код на Python организовывается в функции и классы, которые могут объединяться в модули, а они в свою очередь могут быть объединены в пакеты.

• Delphi – императивный, структурированный, объектно-ориентированный, высокоуровневый язык программирования со строгой статической типизацией переменных. Основная область использования — написание прикладного программного обеспечения. Этот язык программирования является диалектом языка Object Pascal.

Управление базами данных:

• Oracle – система, которая функционирует на подавляющем большинстве платформ, даже таких как масштабные сервера и большие ЭВМ. Oracle поддерживает все возможные архитектуры. Эта системы будет полезна на больших предприятиях. Для чтения и записи данных требуется использовать Oracle NoSQL Database с Java/C API, что не является удобным для дальнейшего ее использования при разработке небольшой web-системы. Полностью поддерживает разработку схематичных приложений с помощью модели данных JSON для повышения производительности разработчиков.

• MySQL – СУБД, позволяющая упростить работу с реляционными базами данных. Одним из преимуществ является то, что это свободное программное обеспечение. Это одна из самых популярных СУБД, которой может воспользоваться любой человек, которого она заинтересует. Главной особенность MySQL считается её гибкость, которая обеспечивается поддержкой большого количества типов таблиц. Она включает в себя огромный набор инструментов, который будет полезен при разработке любого проекта. Зачастую является решением для малых и средних приложений.

• Microsoft SQL Server – система управления реляционными базами данных, разработанная корпорацией Microsoft. Также является одной из популярной СУБД. Эта СУБД поддерживает работу с другими программными продуктами от Microsoft: MS Excel, MS Access. Использует язык запросов – Transact-SQL, создан совместно Microsoft и Sybase, который является реализацией стандарта ANSI/ISO по структурированному языку запросов с расширениями. Имеет множество утилит и компонент для работы, например таких как SQL Server Management Studio (SSMS).

• MongoDB – СУБД, которая написана на C++. Она используется для хранения той информации, которую невозможно нормально структурировать в обычных реляционных базах данных. Это документно-ориентированная база данных. Может быть удобной при наличии на предприятии крупного документооборота. MongoDB кроссплатформенна, что позволяет портировать ее на различные платформы. Имеется возможность и загрузить исходный код, самостоятельно скомпилировав его, однако, потребуется использовать библиотеки с официального сайта.

• PostgreSQL – Масштабируемая объектно-реляционная база данных, работающая на Linux, Windows, OSX и некоторых других системах. В PostgreSQL 10 есть такие функции, как логическая репликация, декларативное разбиение таблиц, улучшенные параллельные запросы, более безопасная аутентификация по паролю на основе SCRAM-SHA-256 с поддержкой табличных пространств, а также хранимых процедур, объединений, представлений и триггеров.

Среды разработки:

• Visual Studio – представляет собой полностью интегрированную среду разработки. Она спроектирована таким образом, чтобы делать процесс написания кода, его отладки и компиляции в сборку для поставки конечным потребителям как можно более простым. Является наиболее функционально насыщенной IDE-средой, в которой можно разрабатывать приложения на C#. Данный продукт поставляется с конструкторами графических пользовательских интерфейсов, с возможностями работы с базами данных, с поддержкой фрагментов кода, с возможностями для просмотра всего проекта в целом, с просмотром свойств объектов.

• Standard Widget Toolkit – библиотека графических пользовательских элементов с открытым исходным кодом для разработки интерфейсов пользователя на языке Java. Предоставляет из себя переносимый API для разработчиков Java. Использует собственные библиотеки операционной системы через собственный интерфейс Java (Java Native Interface) для визуализации элементов графического интерфейса, аналогично программам, написанным с использованием собственного API операционной системы.

• PyCharm – это интегрированная среда разработки на языке Python, которая была разработана международной компанией JetBrains (да, и снова эти ребята). Эта IDE распространяется под несколькими лицензиями, в том числе как Community Edition, где чуть урезан функционал. Сами разработчики характеризуют свой продукт как «самую интеллектуальную Python IDE с полным набором средств для эффективной разработки на языке Python», но она не ограничивается на Python. PyCharm поддерживает Google App Engine; IronPython, Jython, Cython, PyPy wxPython, PyQt, PyGTK и др.

• Qt Creator – кроссплатформенная свободная IDE для разработки на С, С++ и QML с помощью фреймворка Qt. Разработана Trolltech для работы с фреймворком Qt. Включает в себя графический интерфейс отладчика и визуальные средства разработки интерфейса как с использованием QtWidgets, так и QML. Зачастую используется при написания программного обеспечения для роботов, автомобилей и кораблей, которые используются на производствах, в перевозке грузов и других похожих отраслях. Qt Creator позволяет добиться высокой производительности при прототипировании интерфейса в приложениях любой сложности.

• CLion – продукт JetBrains. Многофункциональная IDE. В которой поддерживается разработка кода на C и C++, а также web-разработка на HTML(5), CSS, JavaScript, XML. А также на некоторых других языках доступен в виде плагинов (например, Lua). Основным преимуществом является статический и динамический анализ с безопасный рефакторингом кода. Позволяет кастомизировать редактор так, как удобно пользователю.

• IntelliJ IDEA – ещё один продукт от JetBrains. Комплекс программных средств, который используется для написания, исполнения, отладки и оптимизации кода для многих языков программирования, в частности Java, JavaScript, Python и других языков программирования от компании JetBrains. Отличается обширным набором инструментов для рефакторинга (перепроектирования) и оптимизации кода. Интеграция с серверами приложений, включая Tomcat, TomEE, GlassFish и другие. Имеет набор инструментов для работы с базами данных, SQL файлами и инструменты для запуска тестов и анализа покрытия кода, включая поддержку всех популярных фреймворков для тестирования.

• Eclipse – то бесплатная опенсорсная среда разработки, которая хорошо подойдет как новичкам, так и опытным разработчикам. Помимо инструментов отладки и поддержки Git/CVS, Eclipse поставляется с Java и инструментом для создания плагинов. Изначально Eclipse использовалась только для Java, но сейчас, благодаря плагинам и расширениям, ее функции значительно расширились. Именно из-за возможности расширить Eclipse своими модулями эта платформа и завоевала свою популярность среди разработчиков. Функционал Eclipse не такой большой, как у IntelliJ IDEA, зато эта среда разработки распространяется с открытым исходным кодом. Основными преимуществами является возможность программировать сразу на нескольких языках и возможность интеграции JUnit.

# 2.2 Обоснование и выбор программного обеспечения для разработки ИС

Для разработки программного обеспечения мной было принято решение реализовывать информационную системы с нуля в среде Visual Studio на платформе Windows Presentation Foundation (WPF) используя язык C#. Ниже описано каждое средство реализации, вместе с их преимуществами среди других.

В качестве среды разработки была выбрана «Visual Studio», потому что:

• Поддержка контроля версий GIT;

• Технология автодополнения IntelliSense, которая умеет исправлять ошибки разработчика, предлагая ряд вариантов, но, в отдельных случаях, может и сгенерировать фрагменты кода;

• Расширенный функционал, который можно дополнятся плагинами;

• Интегрированная среда разработки (IDE);

• Удобный и интуитивно понятный интерфейс, который можно кастомизировать под каждого пользователя индивидуально;

• Присутствует встроенный отладчик;

• Наличие встроенного средства для быстрого рефакторинга;

• Множество функций, которые позволяют изменять редактор в соответствии с использованием, что означает, что пользователь может загружать библиотеки из Интернета и интегрировать его с кодом в соответствии со своими требованиями.

• Автоматизированный процесс публикации и автоматизации сборки проекта и приложения;

• Наличие множество утилит, который упрощают работу с сторонними компонентами и приложениями, например таким как SQL Server.

В качестве СУБД был выбран «SQL Server», потому что:

• Обеспечивает практически неограниченный рост объемов хранения за счет увеличения надежности и масштабируемости системы, используя все преимущества мультипроцессорной обработки файлов;

• Возможность взаимодействия с публичным облаком Windows Azure;

• Облачное резервирование локальных систем и аварийное восстановление из облака повышает отказоустойчивость;

• Возможность интеграция структурированных и неструктурированных данных;

• Наличие интегрированная среды для управления любой инфраструктурой SQL.

В качестве языка программирования была выбран «C#», потому что:

• Объектно-ориентированный язык программирования;

• Работает на базе платформы .NET Framework за счёт чего платформа поддерживает несколько языков: наряду с C# это VB.NET, C++, F#, а также различные диалекты других языков, привязанные к .NET, например, Delphi.NET;

• Наличие большого количества библиотек и шаблонов, которые позволяют не тратить время на изобретение “велосипеда”;

• Из-за большого разнообразия синтаксических конструкций и возможности работать с платформой .Net, C# позволяет быстрее, чем любой другой язык.

# 2.3. Описание логики функционирования ИС

При разработке информационной системы учета успеваемости студентов техникума, была создана база данных “StudentProgress”, в которой хранится вся необходимая информация для корректной работы информационной системы. Это информация о оценках, обучающихся в учебном учреждении, преподавателей, ведение журнала и выставления оценки за экзамен или же зачет за сессию. Также была спроектирована   
схема базы данных, которая представлена на рисунке 2.1.



Рисунок 2.1 – Схема базы данных “StudentProgress”

Таблица “Users” - пользователи программы их пароли и логины, создана для того, чтобы правильно распределить роли. Представлена на рисунке 2.2. и на таблице 4.

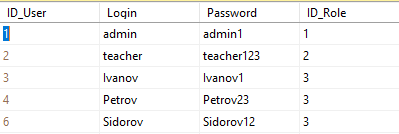


Рисунок 2.2 – Сущность “Пользователи”

Таблица 4 – Описание сущности “Пользователи”.

|  |  |  |
| --- | --- | --- |
| Название | Тип | Описание |
| ID\_User | int | Первичный ключ |
| Login | Nchar(10) | Логины пользователей |
| Password | Nchar(10) | Пароли пользователей |
| ID\_Role | int | Внешний ключ, идентификатор сущности Роли |

Таблица “Role” - описаны роли для пользователей в программе. Представлена на рисунке 2.3 и на таблице 5.

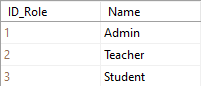


Рисунок 2.3 – Сущность “Роли”

Таблица 5 – Описание сущности “Роли”

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

Таблица “Student” - содержит все нужные данные студента. Представлена на рисунке 2.4 и на таблице 6.

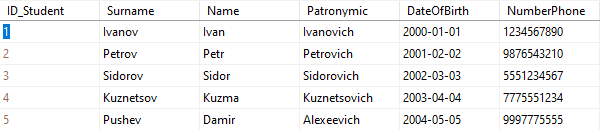


Рисунок 2.4 – Сущность “Студент”

Таблица 6 – Описание сущности “Студент”

|  |  |  |
| --- | --- | --- |
| Название | Тип | Описание |
| ID\_Student | Int | Первичный ключ |
| Surname | Nvarchar(MAX) | Фамилию студента |
| Name | Nvarchar(50) | Имя студента |
| Patronymic | Nvarchar(50) | 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 2.5 на таблице 7.

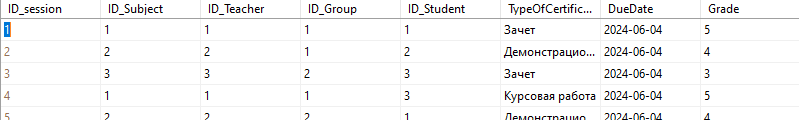


Рисунок 2.5 – Сущность “Сессия”

Таблица 7 – Описание сущности “Сессия”

|  |  |  |
| --- | --- | --- |
| Название | Тип | Описание |
| ID\_Session | Int | Первичный ключ сессии |
| ID\_Subject | Int | Внешний ключ, ссылающийся на идентификатор предмета |
| ID\_Teacher | Int | Внешний ключ идентификатора сущности преподавателя |
| ID\_Group | Int | Внешний ключ идентификатора группы |
| ID\_Student | Int | Внешний ключ идентификатора студента |

Продолжение таблицы 7

|  |  |  |
| --- | --- | --- |
| TypeOfCertification | Nvarchar(50) | Вид зачета Зачет/Экзамен |
| DueDate | Nvarchar(50) | Дата выставления оценки |
| Grade | Nchar(10) | Выставление |

Таблица “Subject” - Содержит в себе данные о сдачи зачетов или же экзаменов. Таблица представлена на рисунке 2.6 и на таблице 8.

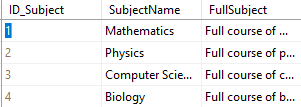


Рисунок 2.6 – Сущность “Предмет”

Таблица 8 – Описание сущности “Предмет”

|  |  |  |
| --- | --- | --- |
| Название | Тип | Описание |
| ID\_Subject | Int | Первичный ключ, номер предмета |
| SubjectName | Nvarchar(50) | Наименование предмета. |
| FullSubject | Nvarchar(MAX) | Полная расшифровка предмета. |

Таблица “Teacher” - содержит все нужные данные студента. Таблица представлена на рисунке 2.7 и на таблице 9.



Рисунок 2.7 – Сущность “Преподаватель”

Таблица 9 – Описание сущности “Преподаватель”

|  |  |  |
| --- | --- | --- |
| Название | Тип | Описание |
| ID\_Student | Int | Первичный ключ, номер студента |
| Surname | Nvarchar(MAX) | Фамилию преподавателя |
| Name | Nvarchar(50) | Имя преподавателя |
| Patronymic | Nvarchar(50) | Отчество преподавателя |
| Post | Nvarchar(50) | Должность преподавателя |
| ID\_Subject | Int | Внешний ключ идентификатор сущности предмета |

Таблица “Journal” - содержит все нужные данные студента. Таблица представлена на рисунке 2.8 и на таблице 10.

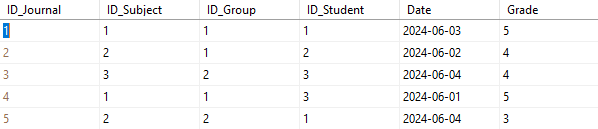


Рисунок 2.8 – Сущность “Журнал”

Таблица 10 – Описание сущности “Журнал”

|  |  |  |
| --- | --- | --- |
| Название | Тип | Описание |
| ID\_Journal | Int | Содержит первичный ключ |
| ID\_Subject | Int | Внешний ключ предмета |
| ID\_Group | Int | Внешний ключ группы |
| ID\_Student | Int | Внешний ключ студента |
| Date | Date | Дата выставления оценки |
| Grade | Nchar(10) | Поставленную оценку за сутки |

Таблица “Group” - содержит все нужные данные студента. Таблица представлена на рисунке 2.9 и на таблице 11.

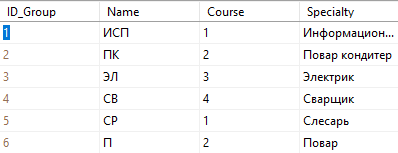


Рисунок 2.9 – Сущность “Группа”

Таблица 11 – Описание сущности “Группа”

|  |  |  |
| --- | --- | --- |
| Название | Тип | Описание |
| ID\_Group | int | Первичный ключ |
| Name | Nvarchar(50) | Наименование группы |
| Course | Nchar(10) | Курс группы |
| Speciality | Nvarchar(MAX) | Специальность группы |

# 2.4. Реализация интерфейса ИС

В ходе разработки информационной системы был разработан логотип. Логотип был реализован в программе Adobe Photoshop CC. Логотип представлен на рисунке 2.10.

****

Рисунок 2.10 – Логотип программы «Учета успеваемости студентов техникума»

Была выбрана цветовая палитра для информационной системы. Палитра представлена в таблице 11.

Таблица 12 - Цветовая модель приложения

|  |  |  |
| --- | --- | --- |
| Основной фон | Дополнительный фон | Акцентирование  внимания |
| RGB  #FFFFFFFF | RGB  #3333333 | RGB  #FF5FD1DD |
| (255, 255, 255) | (51, 51, 51) | (72, 229,79) |
|  |  |  |

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 и ознакомится с информацией о учебном учреждении, увидеть фотографии, форма представленный на рисунке 2.11.

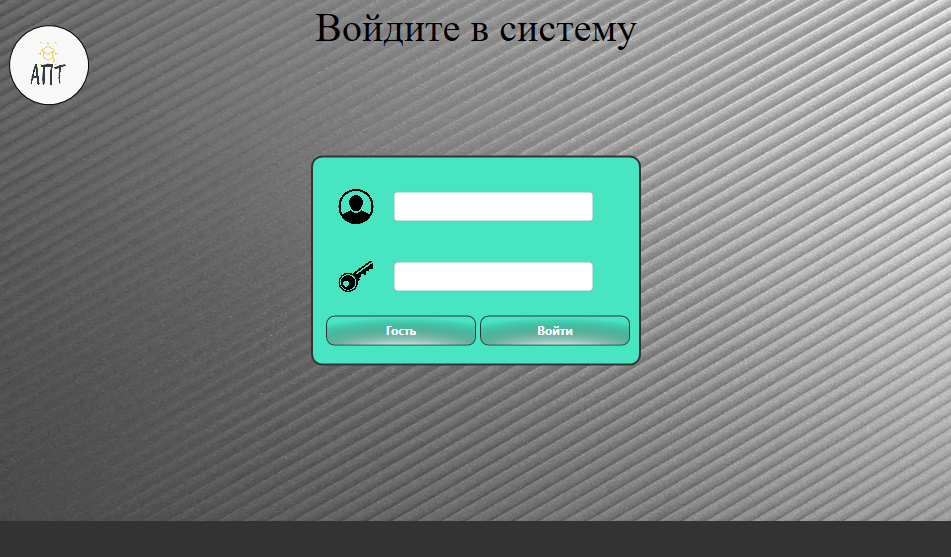


Рисунок 2.11 - Форма “Авторизации”

При входе в систему, пользователя встречает “Главная форма” приложения, на этой форме реализованы кнопки для перехода по формам программы, форма, представленная на рисунке 2.12

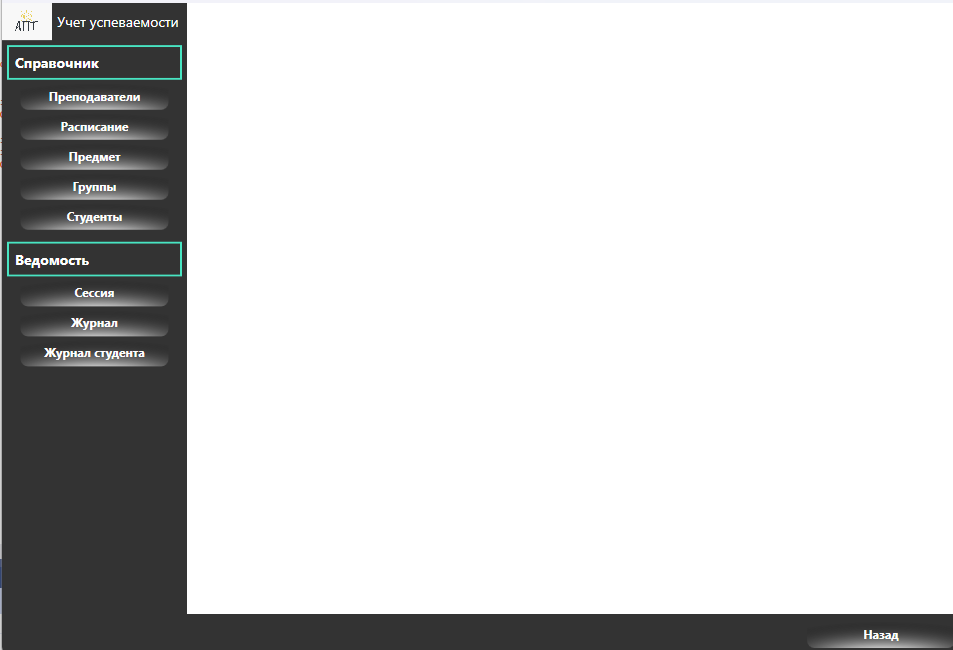


Рисунок 2.12 – “Главная форма” программы

Форма “Гость” представляет собой страницу для ознакомления программой, а также рекомендации о поступлении в данное учебное учреждение, Форма ознакомления с приложением может использоваться для помощи новым пользователям ориентироваться в функционале и возможностях приложения.

Через эту форму пользователи могут получить краткое введение в основные функции и возможности приложения, ознакомиться с тем, как пользоваться интерфейсом, настроить предпочтения и узнать об основных правилах использования, представленная на рисунке 2.13.



Рисунок 2.13 – Форма “Гость”

При нажатии на кнопку “Преподаватели”, открывается форма с данными преподавателей, в котором содержится: ID преподавателя, ФИО, его должность и предмет, представлена на рисунке 2.14.



Рисунок 2.14 – 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 2.15.

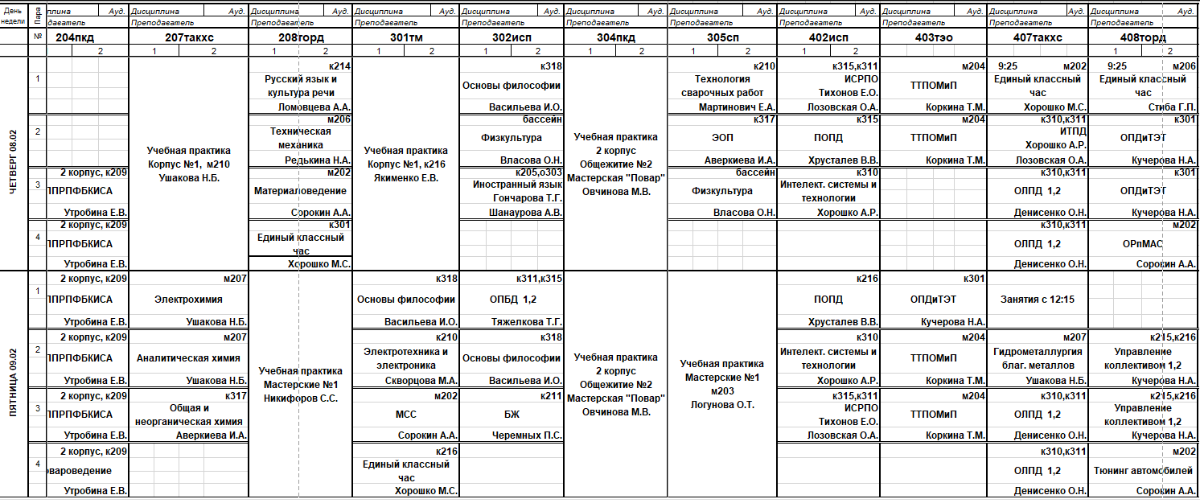


Рисунок 2.15 – Форма “Расписание”

При нажатии на кнопку “Предмет”, открывается форма, в которой содержится информация о предметах, их полное название, форма представлена на рисунке 2.16.

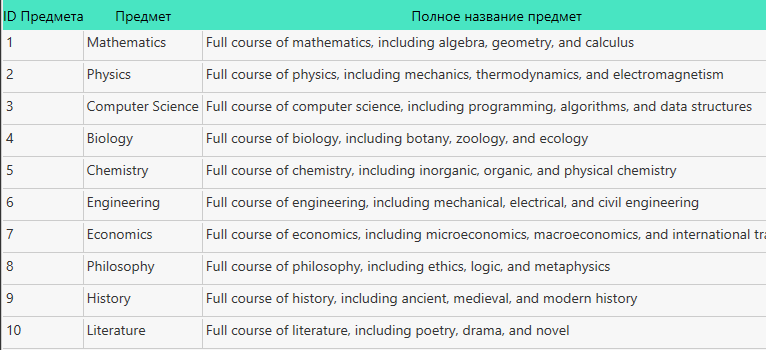


Рисунок 2.16 – форма “Предмет”

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.

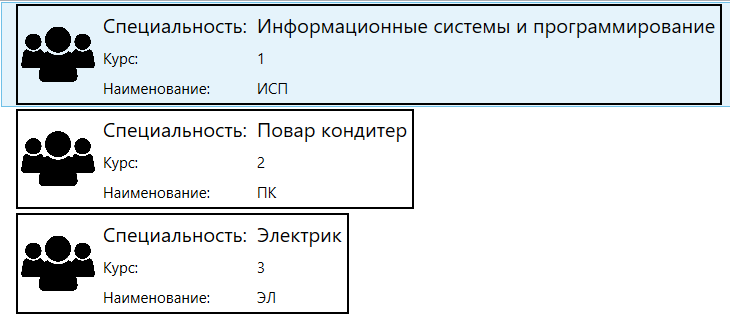


Рисунок 2.16 – Форма “Группы”

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.

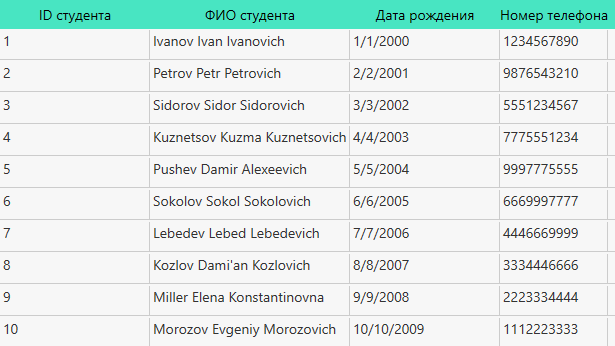


Рисунок 2.17 – Форма “Студенты”

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

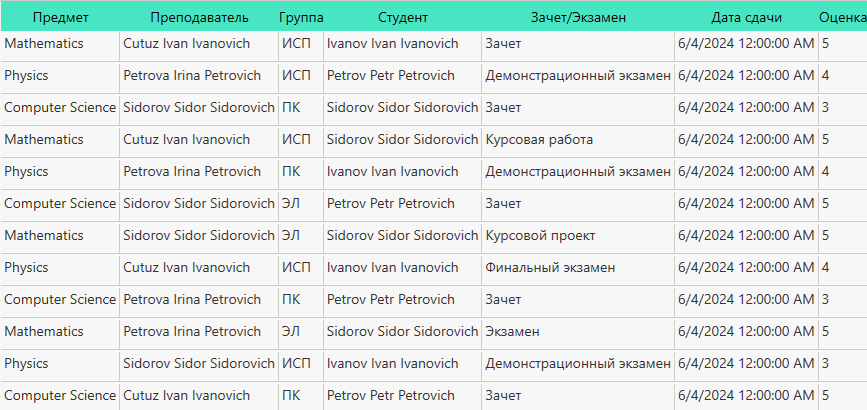


Рисунок 2.18 – Форма “Сессия”

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.

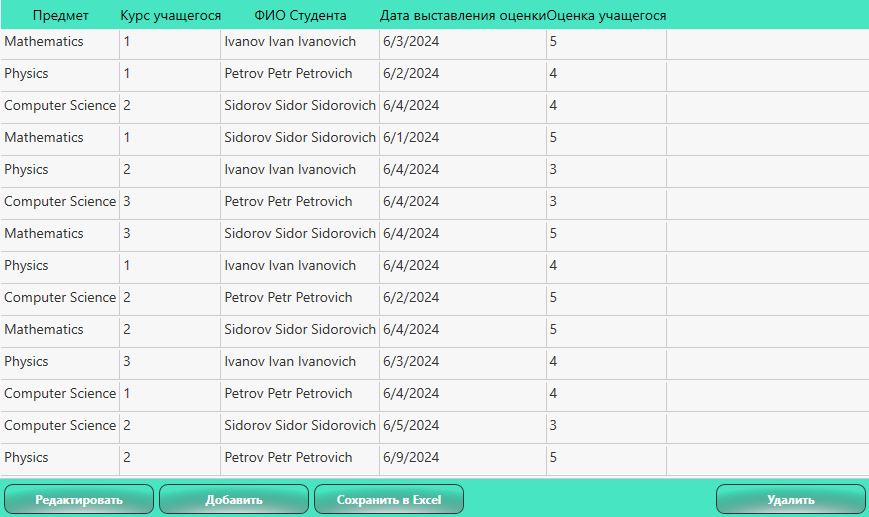


Рисунок 2.19 – Форма “Журнал”

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.

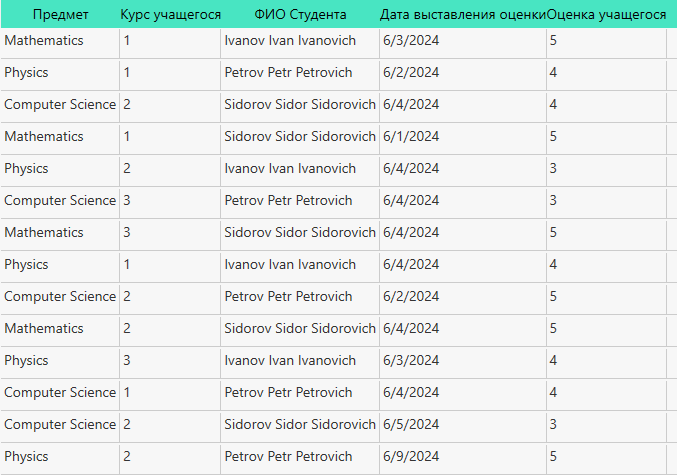


Рисунок 2.20 – Форма “Журнал студента”

# 2.5. 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); } } |

Код для удаления данных был разработан и описывает, что программа сделает, когда пользователь выберет строки для удаления и нажмёт кнопку "Удалить", представлен на листинге 2.2.

Программа собирает в список все выбранные пользователем строки. создает диалоговое окно с запросом на подтверждение удаления. Если пользователь нажмёт "Да", программа удалит выбранные строки из базы данных. После чего программа покажет пользователю сообщение о том, что данные успешно удалены и обновит данные в списке, отобразив новые данные без удаленных строк. 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()); } } } |

# 2.7. Тестирование ИС

Тестирование – процесс выполнения программы с целью обнаружения ошибок. Тестирование обеспечивает:

– обнаружение ошибок;

– демонстрацию соответствия функций программы ее назначению;

– демонстрацию реализации требований к характеристикам программы;

– отображение надежности как индикатора качества программы.

Цель этапа тестирования – снижение риска, возникающего при вводе решения в промышленную эксплуатацию.

Пилотная эксплуатация – это тестирование решения в промышленной среде. Основная задача пилотной эксплуатации – продемонстрировать, что решение способно стабильно работать условиях промышленной эксплуатации и удовлетворяет требованиям бизнеса. В процессе пилотной эксплуатации решение испытывается в реальных условиях. Пилотная эксплуатация дает возможность пользователям высказать свое мнение о работе продукта. Руководствуясь этим мнением разработчиком, устраняются все возможные неполадки или создается план действий на случай непредвиденных обстоятельств. В конечном итоге, пилотная эксплуатация позволяет принять решение, стоит ли начинать полномасштабное развертывание или отложить до устранения неполадок, способных сорвать развертывание.

Тест–кейс – это документация действий тестировщика, последовательность которых направленная на проверку какого–либо функционала, описывающая как прийти к фактическому результату.

Различают 3 результата тест–кейса:

Положительный – подтверждает, что ПО соответствует требованиям. Показывает, что при корректных входных данных и действиях пользователя ПО выполняет свои функции.

Отрицательный – показывает, что ПО способно обрабатывать некорректные входные данные или неверные действия пользователя. Например, выводить соответствующие сообщения, подсказывать, как исправить ситуацию.

Деструктивный – демонстрирует, что никакие внешние воздействия или высокие нагрузки не приводят к потере данных пользователя, ПО можно использовать. Условие: нагрузки не разрушают аппаратную часть.

Для тестирования разработанной информационной системы были разработаны следующие unit-test’ы, который тестируют надежность пароля и логина веденных пользователем. Тесты проверяют минимальное и максимально количество символов, наличие специальных символов, верхний и нижний регистр. Листинг тестов представлен на рисунках 2.21-2.23.



Рисунок 2.21 – Листинг теста логин.

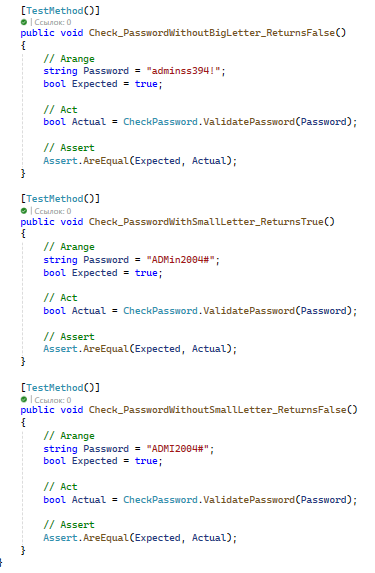


Рисунок 2.22 – Листинг тестов паролей.

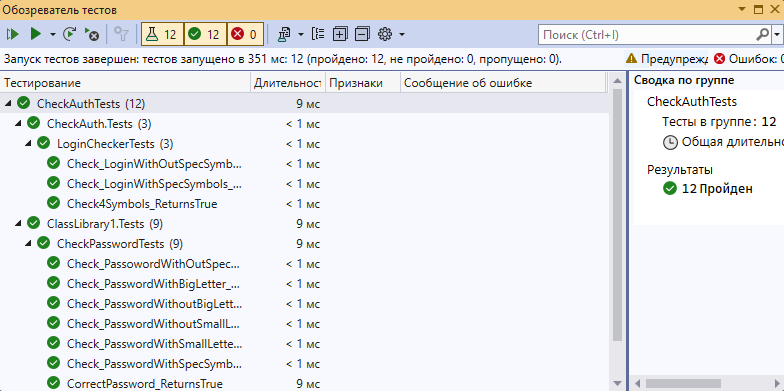


Рисунок 2.23 – Результат проведения тестов.

# 2.8. Руководство пользователя ИС

Добро пожаловать в приложение "StudentProgress". Это приложение предназначен для учета успеваемости студентов техникума. В данном руководстве объясняются основные функции приложения и инструкции по использованию.

1.Главная форма

Представляет собой навигацию по окнам, благодаря кнопкам навигации вы можете переходить в интересующие вас окна, представлено на рисунке 2.24



Рисунок 2.24 – Главная форма приложения

2.Просмотр Преподавателей

Для перехода в раздел “Преподаватели” – нажмите на кнопку "Преподаватели" в левом меню, представлена на рисунке 2.25

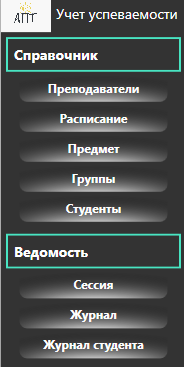


Рисунок 2.25 – Кнопка “Преподаватели”

На странице отображаются все преподаватели.

Вы можете видеть краткую информацию о преподавателях: ID, ФИО, должность и предмет, который они преподают.

3.Расписание

Для перехода в раздел “Расписание” – нажмите на кнопку "Расписание" в левом меню, представлена на рисунке 2.26

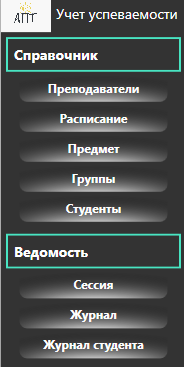


Рисунок 2.26 – Кнопка “Расписание”

На странице “Расписание” можно ознакомиться с расписание занятий для студентов.

4.Предмет

Для перехода в раздел “Предмет” – нажмите на кнопку “Предмет” в левом меню, представлена на рисунке 2.27

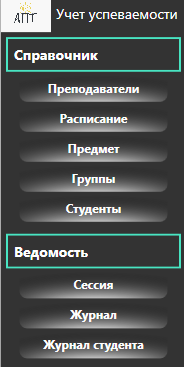


Рисунок 2.27 – Кнопка “Предмет”

На данной странице можно узнать больше о предметах, и их расшифровку.

5.Группы

Для перехода в раздел “Группы” – нажмите на кнопку “Группы” в левом меню, представлена на рисунке 2.28

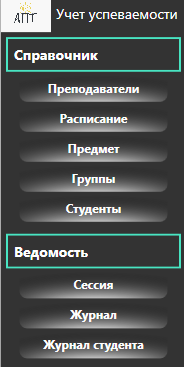


Рисунок 2.28 – Кнопка “Группы”

На странице представлены информация о группах в техникуме, их наименование, курс и специальность.

6.Редактирование и добавления

• Перейдите на страницу журнал, нажав на кнопку “Журнал” в левом меню, представлена на рисунке 2.29

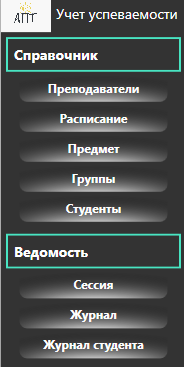


Рисунок 2.29 – Кнопка “Журнал”

• Нажать на кнопку “Редактировать” в нижнем меню.

• Заполните форму редактирования или добавления, указав предмет, группу, студента, выбрать дату и указать оценку.

• Нажмите кнопку “Сохранить”

• При успешном сохранении появиться сообщение “Информация сохранена”.

7.Авторизация

• Введите логин и пароль в пустые поля на странице авторизации, представлена на рисунке 2.30



Рисунок 2.30 – Окно авторизации в приложение.

• Нажмите кнопку “Войти”.

• Если введены правильные данные, вы будете перенаправлены на главную форму приложения.

Это руководство предоставляет базовую информацию о том, как использовать приложение “StudentProgress”.

# Заключение

В ходе выполнения дипломного проекта разработки информационной системы «АПТ» для учета успеваемости студентов техникума, также была спроектирована база данных и спроектирована схема базы данных. Разработанная информационная системы позволит решить ряд важных задач, связанных с учебным процессом:

Повышение эффективности взаимодействия преподавателей и студентов: Информационная система предоставляет удобную платформу для проведения и подготовки к учебным занятиям. Это повышает качество образовательного процесса.

Таким образом, разработанная информационная система является ценным инструментом для повышения эффективности и доступности учебного процесса. Он помогает преподавателям и студентам более эффективно взаимодействовать друг с другом.

В рамках курсового проекта были выполнены следующие цели:

• Разработано приложение для помощи преподавателям в учете и контроле успеваемости учащихся;

• повышена доступность информации об учебных предметах и преподавателях;

• повышен уровень доступности информации об успеваемости студентов.

И задачи:

• Определил требования к программному продукту.

• Разработал прототип приложения.

• Реализовал приложение в выбранной среде IDE.

• Создал сборку для установки.

• Разместил проект на веб-сервисе GitHub или аналогичном –https://github.com/whit3full/StudentProgress.git.

# Список использованных источников

1. Казанский, А. А. Программирование на Visual C# : учебное пособие для среднего профессионального образования / А. А. Казанский. — 2-е изд., переработан и дополнен — Москва: Издательство Юрайт, 2024. — 192 с. — (Профессиональное образование). — ISBN 978-5-534-14130-6. — Текст: электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/538155

2. Тузовский, А. Ф. Объектно-ориентированное программирование: учебное пособие для вузов / А. Ф. Тузовский. — Москва: Издательство Юрайт, 2024. — 213 с. — (Высшее образование). — ISBN 978-5-534-16316-2. — Текст: электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/537332

3. Казанский, А. А. Программирование на Visual C#: учебное пособие для вузов / А. А. Казанский. — 2-е изд., перераб. и доп. — Москва: Издательство Юрайт, 2024. — 192 с. — (Высшее образование). — ISBN 978-5-534-12338-8. — Текст: электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/537364

4. Демин, А. Ю. Информатика. Лабораторный практикум: учебное пособие для вузов / А. Ю. Демин, В. А. Дорофеев. — Москва: Издательство Юрайт, 2024. — 147 с. — (Высшее образование). — ISBN 978-5-534-18259-0. — Текст: электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/534628

5. Сысолетин, Е. Г. Разработка интернет-приложений: учебное пособие для вузов / Е. Г. Сысолетин, С. Д. Ростунцев; под научной редакцией Л. Г. Доронинского. — Москва: Издательство Юрайт, 2022. — 90 с. — (Высшее образование). — ISBN 978-5-9916-9975-4. — Текст: электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/492224

# 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="600" Width="970" MinHeight="600" MinWidth="970" 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()); } } |