

**КУРСОВОЙ ПРОЕКТ**

**по дисциплине: МДК 01.01 Разработка программных модулей**

**Тема: Разработка программного модуля**

**информационной системы «Расписание студента»**

**Специальность 09.02.07 «Информационные системы и программирование»**

|  |  |  |
| --- | --- | --- |
| **Выполнил студент(ка) группы 313ИС-22** | **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** | **П.А. Кулыгина** |

|  |  |  |
| --- | --- | --- |
| **Руководитель** | **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** | **Л.А. Хамрилова** |

**Москва 2024**



**УТВЕРЖДАЮ**

**Зам. директора КМПО**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_С.Ф. Гасанов**

**«\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2024 г.**

**ЗАДАНИЕ НА КУРСОВОЙ ПРОЕКТ**

**по дисциплине: МДК 01.01 Разработка программных модулей**

**Специальность 09.02.07 «Информационные системы и программирование»**

**Студент группы 313ИС-22 Кулыгина Полина**

**ТЕМА: «Разработка программного модуля информационной**

**системы «Расписание студента»**

Дата выдачи задания «\_\_\_\_\_» \_\_\_\_\_\_\_\_\_ 2024 г.

Срок сдачи проекта «\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_ 2024 г.

**Москва 2024**

**Перечень вопросов, подлежащих разработке:**

ВВЕДЕНИЕ

1. ОПИСАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ

1.1. Введение в предметную область

1.2. Анализ готовых решений

2. ПРОЕКТИРОВАНИЕ СИСТЕМЫ

2.1. Общее назначение программы

2.2. Требования к функциям, выполняемым системой

2.3. План тестирования

3. РЕАЛИЗАЦИЯ ПРОЕКТНОЙ СИСТЕМЫ

3.1. Описание среды разработки

3.1.1. Выбор и описание программных инструментов

3.1.2. Обоснование выбора инструментария по разработке

3.2. Разработка программного модуля

3.2.1. Реализация пользовательского интерфейса программы

3.2.2. Описание кодом функциональных узлов модуля

3.2.3. Результат работы и тестирование

ЗАКЛЮЧЕНИЕ

**Исходные данные:** ГОСТ 7.32-2017, ГОСТ 7.0.100-2018, ГОСТ 7.9-95, ГОСТ 7.80-2000, ГОСТ 9327-60

**Перечень разрабатываемых материалов для визуализации:** ER-диаграмма базы данных, диаграмма сценариев, экранные формы интерфейса.

Задание выдал:

Руководитель курсового проекта Л.А. Хамрилова

Задание принял к исполнению П.А. Кулыгина

*Рассмотрено*

на заседании предметно-цикловой комиссии

информационных технологий и системного

администрирования

Протокол № \_\_\_*от «\_\_» \_\_\_\_\_\_\_\_\_* 2024 г.

Председатель ПЦК \_\_\_\_\_\_\_\_\_\_\_\_\_

# СОДЕРЖАНИЕ

[**ВВЕДЕНИЕ** 3](#_Toc184327575)

[**1** **ОПИСАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ** 5](#_Toc184327576)

[1.1 Введение в предметную область 5](#_Toc184327577)

[1.2 Анализ готовых решений 7](#_Toc184327578)

[**2** **ПРОЕКТИРОВАНИЕ СИСТЕМЫ** 10](#_Toc184327579)

[2.1 Общее назначение программы 10](#_Toc184327580)

[2.2 Требования к функциям, выполняемым системой 10](#_Toc184327581)

[2.3 План тестирования 14](#_Toc184327582)

[**3** **РЕАЛИЗАЦИЯ ПРОЕКТНОЙ СИСТЕМЫ** 16](#_Toc184327583)

[3.1 Описание среды разработки 16](#_Toc184327584)

[3.1.1 Выбор и описание программных инструментов 16](#_Toc184327585)

[3.1.2 Обоснование выбора инструментария по разработке 18](#_Toc184327586)

[3.2 Разработка программного модуля 19](#_Toc184327587)

[3.2.1 Реализация пользовательского интерфейса программы 20](#_Toc184327588)

[3.2.2 Описание кодом функциональных узлов модуля 22](#_Toc184327589)

[3.2.3 Результат работы и тестирование 23](#_Toc184327590)

[**ЗАКЛЮЧЕНИЕ** 26](#_Toc184327591)

[**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ** 27](#_Toc184327592)

[**ПРИЛОЖЕНИЯ** 29](#_Toc184327593)

[**ПРИЛОЖЕНИЕ 1 Диаграммы** 29](#_Toc184327594)

[**ПРИЛОЖЕНИЕ 2 План-тест** 31](#_Toc184327595)

[**ПРИЛОЖЕНИЕ 3 Функционал кнопок** 38](#_Toc184327596)

# ВВЕДЕНИЕ

В условиях активного развития информационных технологий, сфера образования также стремится к автоматизации многих процессов, связанных с организацией учебного процесса. Одним из важнейших элементов управления в учебных заведениях является составление расписания занятий для студентов. Ручное составление расписания зачастую приводит к ошибкам, дублированию и несогласованности, что создает дополнительные трудности для всех участников образовательного процесса.

Для повышения эффективности и точности составления расписания все чаще используются информационные системы, которые автоматизируют данный процесс. Разработка программных модулей, интегрированных в такие системы, позволяет упростить задачи по организации учебного процесса, снизить вероятность ошибок и сократить временные затраты на составление расписания.

Целью данного курсового проекта является создание программного модуля для системы «Расписание занятий студента», который автоматизирует процесс отображения и изменения расписания с учетом всех необходимых факторов. Предлагаемое решение повысит удобство и сделает простым восприятие расписания для пользователей, благодаря простому графическому интерфейсу

Для достижения поставленной цели необходимо решить следующие задачи:

1. Провести анализ предметной области и потребностей студентов. Важно исследовать, какие данные и функции наиболее востребованы в расписаниях занятий.
2. Проанализировать готовые решения для автоматизации расписаний. Это позволит выявить преимущества и недостатки существующих систем, а также определить функциональные возможности, которые могут быть полезны в разработке модуля.
3. Сформировать технические требования к программному модулю. Необходимо определить набор функциональных задач, которые должен выполнять модуль, например, отображение расписания.
4. Создать план тестирования программного модуля. Следует определить сценарии тестирования, для проверки корректности работы модуля, стабильности функционирования и удобства использования интерфейса.
5. Обосновать выбор инструментов и технологий разработки. Требуется изучить возможные языки программирования, фреймворки и базы данных, которые позволят эффективно реализовать функционал модуля.
6. Реализовать программный модуль согласно техническим требованиям. На этом этапе необходимо разработать пользовательский интерфейс и интегрировать модуль с базой данных для корректного отображения расписания.
7. Провести тестирование модуля и оценить эффективность. Важно провести тестирование на реальных данных, для оценки удобства и надежности модуля

Объектом исследования выступает процесс автоматизации составления расписания с использованием программных модулей.

Предметом исследования являются процессы создания, хранения, обработки и анализа данных, необходимых для составления расписания учебных занятий в образовательных учреждениях.

Актуальность проекта обусловлена необходимостью повышения эффективности и точности процесса составления расписания, что способствует лучшей организации учебной деятельности. Современные информационные системы помогают автоматизировать этот процесс, снижая вероятность ошибок и оптимизируя временные затраты.

Таким образом, реализация программного модуля снизит количество ошибок, сократит временные затраты на составление расписания и повысит качество восприятия информации.

# ОПИСАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ

# Введение в предметную область

Исторически, процесс составления расписания в образовательных учреждениях был исключительно ручным и основывался на опыте и интуиции сотрудников. Ответственные за составление расписания анализировали доступные ресурсы (аудитории, преподавателей и учебные группы), однако этот процесс был сложным и подверженным человеческим ошибкам. Проблемы, такие как дублирование занятий, а также неэффективное использование аудиторий, были весьма распространены. В условиях увеличивающихся масштабов образовательных учреждений и роста числа студентов, ручное составление расписания становилось все более проблематичным.

Преимущества информатизации предметной области становятся очевидными при анализе проблем традиционного метода. Автоматизация позволяет сократить трудозатраты на составление расписания, снизить вероятность ошибок и ускорить процесс корректировки. Информационные системы, используемые в данной сфере, могут учитывать множество факторов одновременно, что делает планирование более точным и гибким и позволяют эффективно использовать ресурсы.

Интеграция информационных технологий в образовательный процесс также предоставляет доступ к онлайн-доступу к расписанию, интеграции с системами управления образовательным процессом и оперативное оповещение студентов и преподавателей о изменениях. Системы, автоматизирующие процесс, включают в себя модули для работы с базами данных, анализа свободных ресурсов и расчетов оптимальных вариантов составления расписания.

Современные требования к электронным системам в образовательной сфере основываются на соблюдении требований законодательства:

1. Федеральный закон №152-ФЗ «О персональных данных», регулирует порядок сбора, обработки и хранения персональных данных. Целью настоящего Федерального закона является обеспечение защиты прав и свобод человека и гражданина при обработке его персональных данных [5] глава 1, статья 1, 2.
2. Федеральный закон №149-ФЗ «Об информации, информационных технологиях и о защите информации» направлен на обеспечение информационной безопасности в процессе создания, передачи и использования информации. Регулирует отношения возникающие при осуществлении права на поиск, получение, передачу, производство и распространение информации, обеспечение защиты информации [4] статья 1, 2.
3. Федеральный закон №187-ФЗ «О безопасности критической информационной инфраструктуры Российской Федерации» устанавливает правила защиты критически важной инфраструктуры, к которой может быть отнесена система с персональными данными студентов [3] статья 1, 2.

Также современные требования основываются на международном стандарте ГОСТ Р ИСО/МЭК 12207, определяющем жизненный цикл программного обеспечения, устанавливающем процесс, который может использоваться при определении, управлении и совершенствовании процессов жизненного цикла программных средств [1] глава 1, статья 1, 2. Ещё одним стандартом является Единая система программной документации - комплекс государственных стандартов, устанавливающих взаимоувязанные правила разработки, оформления и обращения программ и программной документации. В стандартах ЕСПД устанавливают требования, регламентирующие разработку, сопровождение, изготовление и эксплуатацию программ [2] глава 1, статья 1, 2.

А также электронные системы должны быть адаптированы под нужды конкретных учебных заведений, обеспечивая удобство, гибкость и масштабируемость, интеграцию с другими системами для создания единого информационного пространства учебного заведения.

Методология разработки таких систем также играет важную роль. Одним из популярных подходов к созданию информационных систем является водопадная модель (или каскадный подход). Водопадная модель включает в себя несколько ключевых этапов, которые выполняются последовательно.

Каждый этап завершает определенный набор задач, после чего переход осуществляется к следующему этапу: анализ требований (сбор и документация всех требований к программному обеспечению), проектирование системы (разрабатывается архитектура программного обеспечения, то есть определяются основные компоненты системы и их взаимодействие), реализация (написание кода), тестирование (проверка всех функций системы на соответствие требованиям и выявление возможных ошибок), внедрение (по завершении тестирования программное обеспечение передается пользователям), поддержка (после внедрения системы осуществляется сопровождение и устранение возникших в ходе эксплуатации ошибок).

Водопадная модель предполагает, что этот этап включает лишь минимальные изменения, так как основное проектирование и тестирование проводились на начальных этапах.

Главное преимущество водопадной модели заключается в ее четкой структуре и понятности для всех участников процесса. Она особенно удобна в случаях, когда требования к системе известны заранее и маловероятно, что они будут изменены в процессе разработки. Однако недостаток этого подхода состоит в его низкой гибкости.

В случае разработки программного модуля для автоматизации составления расписания, водопадная модель подходит благодаря ее последовательному подходу и возможности тщательно проработать все ключевые аспекты системы на ранних этапах.

# Анализ готовых решений

Для автоматизации составления расписания в образовательных учреждениях существует несколько эффективных информационных систем. Рассмотрим наиболее актуальные и популярные решения:

«Галактика: Расписание учебных занятий» - флагманский продукт корпорации «Галактика» для высших учебных заведений. Это уникальный российский программный продукт, предназначенный для составления и публикации расписаний всевозможных занятий, анализа занятости аудиторного фонда, фактической загруженности профессорско-преподавательского состава, контроля качества образовательного процесса.

Возможности: масштабируемость и способность обработки больших потоков информации, эффективный конструктор расписания, формирование расписания в автоматическом, ручном или комбинированном режиме, регулировка приоритетности использования ресурсов в расписании через приоритеты дисциплин, преподавателей, показатели сложности дисциплин и видов работ, прямой экспорт данных в специальные формы-шаблоны Excel, встроенный дизайнер отчетных форм, подсистема импорта [15]

1С: Университет – программа для составления расписания предназначена для решения задач автоматизированного составления учебных расписаний и оперативного управления помещениями в ВУЗах. С помощью этой программы составлять расписание можно в автоматическом, ручном и смешанном режимах с учетом многих ограничений и условий. При этом можно построить как допустимое расписание, так и оптимизированное, в котором сокращено количество окон или количество используемых помещений.

Возможности: составление расписаний в ручном, автоматическом, смешанном режимах, встроенные средства интеграции с "1С: Университет", "1С: Университет ПРОФ", удобная форма "шахматка" для быстрой ручной модификации расписания перетаскиванием "drap&drop", оптимизация учебных расписаний по критериям, составление расписания сессии, учет параллельных занятий, разбиения на подгруппы и потоковых лекций при составлении расписания, разграничение доступа подокументно, учет графика учебно-производственного процесса; [9]

ИС «Планы» позволяет создать в рамках высшего учебного заведения единую систему автоматизированного планирования учебного процесса. Учебные планы полностью совместимы с форматом, используемым в процедуре государственной аккредитации. Система также включает в себя комплект формализованных ФГОС, которые можно использовать для создания учебных планов и проверки качества УП.

Возможности: автоматически отслеживать обеспеченность кафедры рабочими программами дисциплин на основе УП и данных о закрепленном за ними контингенте, создавать РПД на базе учебного плана любой формы обучения, предоставляя удобный интерфейс для заполнения содержательной части рабочей программы, автоматически переносить данные из учебных планов, импортировать в систему рабочие программы дисциплин, перечень литературы и материально-технического обеспечения из файлов XML; [13]

Таким образом, анализ готовых решений показывает, что на рынке существует множество мощных информационных систем, которые эффективно решают задачи автоматизации составления расписания. Каждая из них предоставляет пользователям широкий набор функциональных возможностей

# ПРОЕКТИРОВАНИЕ СИСТЕМЫ

## **Общее назначение программы**

Программный модуль «Расписание студента» предназначен для автоматизации процесса составления расписания занятий для студентов на основе данных, хранящихся в базе данных учебного заведения. Этот модуль ориентирован на упрощение процесса планирования на основе уже существующих данных. Основная задача модуля — автоматизировать формирование расписания, что позволит сократить время на выполнение этой работы и минимизировать вероятность ошибок, возникающих при ручном составлении.

Внедрение программного модуля «Расписание студента» в учебное заведение имеет ряд экономических преимуществ:

1. Сокращение времени на составление расписания, благодаря автоматизации, что позволяет сократить трудозатраты сотрудников, которые вручную выполняли эту работу.
2. Устранение ошибок, возникающих при ручной работе с большими объёмами данных (например, пересечение расписаний или неверная занятость аудиторий), способствует повышению точности и снижает вероятность конфликтов в расписании.
3. Быстрое составление и изменение расписания упрощает управление учебным процессом, что ведёт к лучшей организации и повышению общей продуктивности учебного заведения.
4. Автоматизация части административной работы позволяет перераспределить ресурсы и снизить необходимость в дополнительных сотрудниках для составления расписания.

# Требования к функциям, выполняемым системой

Программный модуль «Расписание студента» выполняет ряд функциональных задач, которые обеспечивают автоматизацию процесса составления расписания на основе данных, содержащихся в базе данных учебного заведения. Основные функции модуля: формирование расписание на основе данных о студентах, преподавателях, предметах и аудиториях, интерфейс для просмотра расписания, внесение изменений в расписание, а также экспорт данных

Основные объекты предметной области, с которыми работает модуль, приведены в таблице 1.

Таблица 1 – Основные объекты предметной области

| № | Наименование объекта | Краткое описание |
| --- | --- | --- |
| 1 | Студенты | Список студентов, для которых формируется расписание |
| 2 | Преподаватели | Список преподавателей, участвующих в учебном процессе |
| 3 | Предметы | Учебные дисциплины, включенные в учебные планы |
| 4 | Аудитории | Учебные помещения, где проходят занятия |
| 5 | Группы студентов | Учебные группы, объединяющие студентов по курсу или факультету |
| 6 | Расписание звонков | Время начала и окончания занятий, расписанных по парам |
| 7 | Занятия | Свод данных, определяющий время, место и участников учебного занятия |
| 8 | Пользователи | Информация о пользователях, их логины, пароли и роли |

На основе анализа предметной области выделены ключевые атрибуты для каждого объекта и приведены в таблицах 2–9.

Таблица 2 – Сущность: Студенты

|  |  |  |
| --- | --- | --- |
| Атрибут | Тип | Описание |
| ID\_Студента | INTEGER PRIMARY KEY AUTOINCREMENT | Уникальный идентификатор студента |
| ФИО\_Студента | TEXT NOT NULL | Полное имя студента |
| Группа\_Студента | INTEGER | Группа, к которой относится студент |

Таблица 3 – Сущность: Преподаватели

| Атрибут | Тип | Описание |
| --- | --- | --- |
| ID\_Преподавателя | INTEGER PRIMARY KEY AUTOINCREMENT | Уникальный идентификатор преподавателя |
| ФИО\_Преподавателя | TEXT NOT NULL | Полное имя преподавателя |

Продолжение Таблицы 3

|  |  |  |
| --- | --- | --- |
| Атрибут | Тип | Описание |
| Должность | TEXT NOT NULL | Должность преподавателя |

Таблица 4 – Сущность: Предметы

|  |  |  |
| --- | --- | --- |
| Атрибут | Тип | Описание |
| ID\_Предмета | INTEGER PRIMARY KEY AUTOINCREMENT | Уникальный идентификатор предмета |
| Название\_Предмета | TEXT NOT NULL | Полное название предмета |

Таблица 5 – Сущность: Аудитории

|  |  |  |
| --- | --- | --- |
| Атрибут | Тип | Описание |
| ID\_Аудитории | INTEGER PRIMARY KEY AUTOINCREMENT | Уникальный идентификатор аудитории |
| Название\_Аудитории | TEXT NOT NULL | Номер или название аудитории |

Таблица 6 – Сущность: Группы студентов

|  |  |  |
| --- | --- | --- |
| Атрибут | Тип | Описание |
| ID\_Группы | INTEGER PRIMARY KEY AUTOINCREMENT | Уникальный идентификатор группы |
| Название\_Группы | TEXT NOT NULL | Название группы |
| Курс\_Группы | TEXT NOT NULL | Курс обучения группы |

Таблица 7 – Сущность: Расписание звонков

|  |  |  |
| --- | --- | --- |
| Атрибут | Тип | Описание |
| Номер\_Пары | INTEGER PRIMARY KEY | Номер учебной пары |
| Время\_Начала | TEXT NOT NULL | Время начала пары |
| Время\_Окончания | TEXT NOT NULL | Время окончания пары |

Таблица 8 – Сущность: Пользователи

|  |  |  |
| --- | --- | --- |
| Атрибут | Тип | Описание |
| ID\_Пользователя | INTEGER PRIMARY KEY | Уникальный идентификатор пользователя |
| Логин | TEXT NOT NULL | Имя пользователя для входа в систему |
| Пароль | TEXT NOT NULL | Пароль пользователя для входа в систему |
| ID\_Студента | FOREIGN KEY, INTEGER | Студент, которому принадлежит аккаунт |
| ID\_Преподавателя | FOREIGN KEY, INTEGER | Преподаватель, которому принадлежит аккаунт |

Таблица 9 – Сущность: Занятия

| Атрибут | Тип | Описание |
| --- | --- | --- |
| ID\_Занятий | INTEGER PRIMARY KEY AUTOINCREMENT | Уникальный идентификатор записи расписания |
| ID\_Преподавателя | FOREIGN KEY, INTEGER NOT NULL | Преподаватель, ведущий занятие |
| ID\_Группы | FOREIGN KEY, INTEGER NOT NULL | Группа студентов на занятии |
| ID\_Предмета | FOREIGN KEY, INTEGER NOT NULL | Предмет, который проводится на занятии |
| ID\_Аудитории | FOREIGN KEY, INTEGER NOT NULL | Аудитория, где проходит занятие |
| Номер\_Пары | FOREIGN KEY, INTEGER NOT NULL | Номер пары согласно расписанию звонков |
| День\_Недели | TEXT | День недели |
| Дата | TEXT | Дата проведения занятия |

Визуализация структуры базы данных модуля и взаимосвязи между сущностями показана с помощью ER-диаграммы (ПРИЛОЖЕНИЕ 1, рисунок 1.3).

Для программного модуля "Расписание студента" выделим пользователей и их взаимодействия с системой. Диаграмма прецедентов описана в приложении (ПРИЛОЖЕНИЕ 1, рисунок 1.1). Диаграмма показывает роли пользователей и функции, которые они могут выполнить. Основные пользователи модуля: администратор, преподаватель и студент.

Диаграмма IDEF1X используется для создания графической модели информации, которая представляет собой структуру базы данных, отображает связи между сущностями, а также визуализирует организацию данных. (ПРИЛОЖЕНИЕ 1, рисунок 1.2)

Администратору доступны функции управления расписанием (изменение, удаление данных из расписания), создание аккаунтов, назначение ролей.

Преподавателю и студенту доступны функции просмотра расписания, поиск расписания по дате и экспорт своего расписания в Excel.

# План тестирования

Цель тестирования программного модуля «Расписание студента» заключается в обеспечении подтверждения его качества и соответствия заранее определённым функциональным и нефункциональным требованиям. Тестирование направлено на выявление ошибок, проверку корректности выполнения основных задач модуля и обеспечение стабильности его работы. Это поможет гарантировать, что конечный продукт будет удобен для пользователей и надёжен в использовании в реальных условиях.

Задачи тестирования включают:

1. Проверка функциональности – тестирование базовых функций модуля: просмотр, поиск и редактирование расписания, авторизация и создание пользователей.
2. Проверка безопасности – обеспечение конфиденциальности данных и защита от несанкционированного доступа.
3. Тестирование совместимости – проверка работы модуля в совместимости с Excel.

Для тестирования выделено 7 объектов тестирования:

1. Создание пользователя – проверка возможности корректного создания учетных записей для студентов и преподавателей.
2. Авторизация пользователя – проверка правильности процесса авторизации, валидации введённых данных, управления ролями.
3. Просмотр расписания – проверка корректного отображения расписания в зависимости от пользователя (преподаватель, студент).
4. Поиск расписания по дате – проверка корректного вывода результата расписания для выбранной даты
5. Проверка корректности сохранения и экспорта данных в Excel
6. Проверка перехода между окнами – корректность возвращения в главное меню или окно авторизации
7. Добавление, изменение, удаление занятий

План тестирования описан в Приложении 2 таблица 2.1.

Оценка производительности и стабильности системы организуется при помощи проверки удобства интерфейса, его интуитивности и логичности. Это тестирование направлено на оценку опыта конечного пользователя (студента, преподавателя). Важные критерии: удобство навигации по расписанию, легкость поиска информации.

Обеспечение безопасности данных обеспечивается с помощью:

1. Аутентификация: проверка системы на правильную идентификацию пользователей по логину и паролю.
2. Авторизация: проверка корректного распределения ролей в системе и доступа к функциям на основе ролей (студент, преподаватель, администратор).

Программный модуль должен быть совместим с различными программными и аппаратными средствами и должен корректно работать на разных версиях операционных систем.

Таким образом, тщательное тестирование всех компонентов и аспектов модуля «Расписание студента» позволит убедиться, что система работает стабильно, безопасно и удовлетворяет всем требованиям.

# РЕАЛИЗАЦИЯ ПРОЕКТНОЙ СИСТЕМЫ

# Описание среды разработки

Средой разработки создания программного модуля системы «Расписание занятий студента» является PyCharm. PyCharm — это специализированная интегрированная среда разработки (IDE) для Python, предоставляющая широкий спектр необходимых инструментов для разработчиков Python, тесно интегрированных для создания удобной среды для продуктивной разработки на Python, веб-разработки и науке о данных. PyCharm — это кроссплатформенная IDE, которая работает на Windows, macOS и Linux. [14]

PyCharm поддерживает работу с разными версиями Python, предоставляет возможности для разработки веб-приложений и анализирует код, помогая разработчикам выявлять ошибки и улучшать производительность кода. Поддерживаются функции автозаполнения кода, статического анализа и рефакторинга, что делает процесс программирования более эффективным и менее трудоемким

# Выбор и описание программных инструментов

Для разработки модуля был выбран язык программирования Python. Python — это язык программирования общего назначения, который часто используется для написания скриптов. Обычно его определяют, как объектно-ориентированный язык сценариев — определение, которое сочетает в себе поддержку ООП и общую ориентацию на написание скриптов – из книги [7], глава 1, страница 5. Помимо того, что Python — это хорошо продуманный язык программирования, он также полезен для решения реальных задач, с которыми разработчики сталкиваются изо дня в день. Он широко используется в различных областях в качестве инструмента для написания сценариев для других компонентов и реализации автономных программ – из книги [7], глава 1, страница 9.

Этот язык программирования также отлично взаимодействует с базами данных. Для традиционных требований к базам данных существуют интерфейсы Python для всех широко используемых систем реляционных баз данных — Sybase, Oracle, Informix, ODBC, MySQL, PostgreSQL, SQLite и других. В мире Python также существует переносимый API для работы с базами данных для доступа к системам баз данных SQL из скриптов Python, который работает одинаково с различными базовыми системами баз данных – из книги [7], глава 1, страница 11.

В качестве базы данных была выбрана SQLite — это встроенная база данных SQL. В отличие от большинства других баз данных SQL, SQLite не имеет отдельного серверного процесса. SQLite выполняет чтение и запись непосредственно в обычные файлы на диске. Полная база данных SQL с несколькими таблицами, индексами, триггерами и представлениями содержится в одном файле на диске. Формат файла базы данных является кроссплатформенным — вы можете свободно копировать базу данных между 32-разрядными и 64-разрядными системами или между архитектурами с прямым порядком байтов и с обратным порядком байтов. [12]

В качестве инструмента для создания графического интерфейса был выбран PyQt — это набор привязок Python для фреймворка Qt от компании Qt и работает на всех платформах, поддерживаемых Qt, включая Windows, macOS, Linux, iOS и Android. Привязки реализованы в виде набора модулей Python и содержат более 1000 классов. Qt также включает в себя Qt Designer — конструктор графического пользовательского интерфейса. PyQt может генерировать код Python из Qt Designer. В Qt Designer также можно добавлять новые элементы управления, написанные на Python. [11]

Дополнительно были использованы следующие библиотеки:

OpenPyXL — openpyxl — это библиотека Python для чтения и записи файлов Excel 2010 xlsx/xlsm/xltx/xltm. Он появился из-за отсутствия подходящей библиотеки для чтения/записи формата Office Open XML на Python [10]. Она позволяет создавать, изменять и сохранять файлы .xlsx без необходимости установки программы Microsoft Excel.

Datetime — стандартная библиотека Python для работы с датами и временем. Она обеспечивает функционал для вычисления временных интервалов, форматирования и преобразования дат.

# Обоснование выбора инструментария по разработке

Язык программирования Python был выбран из-за его простоты и удобства модификации. Python обладает огромной экосистемой библиотек и фреймворков, которые поддерживают широкий спектр функционала, необходимого для создания учебных и образовательных приложений. Python хорошо подходит для работы с данными, что важно для модуля расписания. Кроме того, Python активно поддерживается сообществом, что гарантирует доступ к обширной документации и ресурсам, которые могут облегчить решение задач, возникающих в процессе разработки. Также был выбран PyCharm как удобный и функциональный инструмент для разработки Python-проектов, позволяющий ускорить процесс программирования и управления проектом.

В качестве базы данных была выбрана SQLite, так как она идеально подходит для небольших приложений, которые требуют локального хранения данных без необходимости подключения к удалённому серверу. В нашем модуле «Расписание студента» база данных используется для хранения информации о занятиях, студентах, преподавателях и аудиториях, и SQLite обеспечивает высокую производительность и надёжность для обработки таких данных. Она также поддерживает язык SQL, что облегчает работу с данными и выполнение различных запросов.

Библиотека PyQt выбрана для создания графического интерфейса модуля. Эта библиотека предоставляет все необходимые элементы интерфейса, такие как окна, кнопки, текстовые поля и таблицы, которые можно использовать для создания удобного и интуитивно понятного интерфейса для пользователя. С помощью PyQt можно создать профессионально выглядящее приложение с минимальными затратами времени на проектирование интерфейса. Библиотеки OpenPyXL и Datetime в данном курсовом проекте были выбраны для выгрузки необходимой информации в Excel и удобства работы с датами

# Разработка программного модуля

В процессе разработки программного модуля «Расписание студента» планируется реализовать последовательность действий, направленных на создание функционального и рабочего программного модуля. Основная цель разработки заключается в создании инструмента для формирования расписания занятий на основе готовых данных из базы.

Первоначально необходимо разработать архитектуру программного модуля. Это включает в себя описание структуры модуля, взаимосвязей между его компонентами и способов взаимодействия с базой данных. Архитектура должна быть удобной для понимания и поддержки, что обеспечит простоту её дальнейшей доработки.

Важной частью разработки является реализация взаимодействия с базой данных. Для хранения данных будет использоваться SQLite, как легковесное и встроенное решение. В процессе разработки будет создано и настроено соединение с базой, а также разработаны SQL-запросы для получения необходимых данных о студентах, преподавателях, предметах, аудиториях, времени занятий и расписании звонков. Особое внимание будет уделено проверке корректности выполнения запросов и обеспечению их высокой производительности.

Следующим этапом разработки станет создание графического интерфейса на основе библиотеки PyQt. Графический интерфейс позволит с лёгкостью взаимодействовать с модулем. Будут добавлены формы для входа в систему, визуализации данных расписания, а также функциональность для управления просмотром расписания в разрезе дней недели, студентов или преподавателей. Особое внимание будет уделено удобству интерфейса и интуитивности навигации.

В процессе реализации программного кода на Python необходимо будет настроить обработку данных, обеспечить валидацию вводимой информации и обработку возможных ошибок. Это позволит пользователям работать с модулем без риска сбоев и некорректной работы.

На завершающем этапе разработки модуля будет проведено тестирование, направленное на проверку корректности его работы и соответствие заданным функциональным и нефункциональным требованиям. В рамках тестирования будут использованы различные сценарии использования, чтобы убедиться в работоспособности ключевых функций.

# Реализация пользовательского интерфейса программы

Разрабатываемое приложение представляет собой десктопную программу, интерфейс которой выполнен с использованием библиотеки PyQt5. Приложение включает несколько окон, каждое из которых выполняет свою функцию, обеспечивая удобное взаимодействие пользователя с системой. В частности, система включает окно авторизации, главное меню для студентов и преподавателей, а также главное меню для администратора.

Первым окном, с которым сталкивается пользователь, является окно авторизации. В данном окне пользователю предоставляется возможность войти в систему, указав свои данные — логин и пароль. В интерфейсе авторизации расположены два поля для ввода: одно для логина, другое — для пароля. Над каждым из полей ввода расположены поясняющие подписи. Также в окне есть кнопка «Войти», которая запускает функцию проверки введённых данных. Кроме того, окно авторизации содержит заголовок «Авторизация» и стандартные кнопки управления окнами (свернуть, развернуть, закрыть). Окно авторизации представлено на рисунке 2.

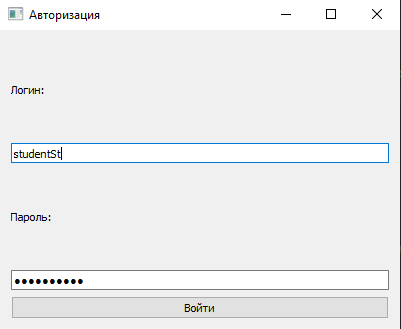


Рисунок 2 – Окно авторизации

После успешного входа в систему, пользователю открывается главное меню, которое зависит от его роли. В этом окне находится несколько ключевых элементов управления. В верхней части окна расположено приветственное сообщение для пользователя, а ниже — кнопки для взаимодействия с расписанием: кнопка «Расписание на неделю», кнопка «Экспорт расписания в Excel» и кнопка «Найти расписание по дате». Для выхода из системы предусмотрена кнопка «Выйти из аккаунта». Это окно также содержит заголовок и стандартные кнопки управления окнами (свернуть, развернуть, закрыть). Главное меню изображено на рисунке 3.

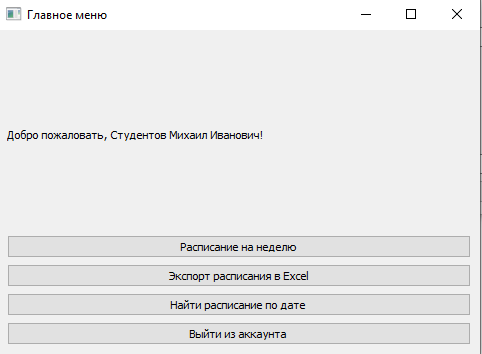


Рисунок 3 – Окно главного меню

Для пользователей с ролью администратора доступно специальное окно, предназначенное для управления пользователями и расписанием. В верхней части окна размещено приветственное сообщение для администратора. Также предусмотрены кнопки для управления расписанием и пользователями: кнопка «Просмотреть расписание», кнопка «Добавить занятие», кнопка «Удалить/заменить занятия». В нижней части окна находится кнопка «Выйти из аккаунта», которая возвращает администратора в окно авторизации. Как и в других окнах программы, здесь также присутствуют заголовок и кнопки для управления окнами. Окно администратора изображено на рисунке 4.

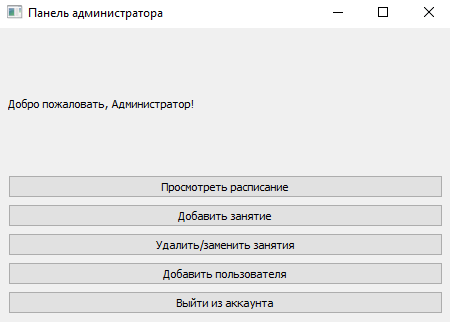


Рисунок 4 – Окно администратора

# Описание кодом функциональных узлов модуля

Кнопка «Войти» предназначена для авторизации пользователя в приложении. Нажатие на кнопку инициирует процесс проверки введенных данных (логина и пароля) с целью подтверждения их достоверности и последующего предоставления доступа к соответствующему функционалу. Затем открывается главное окно. (ПРИЛОЖЕНИЕ 3, рисунок 3.1)

Кнопка «Расписание на неделю» выполняет отображение расписания на неделю для текущего пользователя (преподавателя или студента). (ПРИЛОЖЕНИЕ 3, рисунок 3.2). При нажатии на эту кнопку осуществляется подключение к базе данных, после чего выполняется запрос на получение данных о расписании и открывается интерфейс где можно посмотреть расписание на неделю. Также имеется кнопка "Вернуться в главное меню", которая возвращает пользователя в главное меню. (ПРИЛОЖЕНИЕ 3, рисунок 3.6)

Кнопка «Экспорт расписания в Excel» экспортирует расписание в формат Excel. При нажатии открывается диалоговое окно для выбора места сохранения файла. После выбора файла, данные расписания сохраняются в Excel, и выводится сообщение о успешном сохранении. (ПРИЛОЖЕНИЕ 3, рисунок 3.4)

Кнопка "Найти расписание" позволяет искать расписание по указанной дате. При нажатии открывается окно для ввода даты. (ПРИЛОЖЕНИЕ 3, рисунок 3.3). Так же в открывшемся окне имеется кнопка «Найти» при нажатии на которую открывается интерфейс с расписанием по выбранной дате. (ПРИЛОЖЕНИЕ 3, рисунок 3.5)

Кнопка "Выйти из аккаунта" располагается на главном окне всех пользователей и выполняет выход из текущего аккаунта и возвращает пользователя в окно авторизации. При нажатии очищаются поля ввода логина и пароля, закрывается текущее окно, и открывается окно авторизации. (ПРИЛОЖЕНИЕ 3, рисунок 3.7)

Кнопка "Добавить пользователя" в главном меню администратора открывает диалоговое окно для ввода данных нового пользователя. В этом окне администратор выбирает роль и вводит все необходимые данные. После нажатия на кнопку "Добавить" данные сохраняются в базе данных. Кнопка "Отмена" закрывает окно без внесения изменений. (ПРИЛОЖЕНИЕ 3, рисунок 3.8)

Кнопка "Удалить/заменить занятия" открывает окно для редактирования занятий, где администратор может выбрать группу и внести изменения в расписание. Функционал кнопки представлен в приложении (ПРИЛОЖЕНИЕ 3, рисунок 3.9). Кнопка "Сохранить изменения" сохраняет изменения в расписании, а "Отмена" закрывает окно без изменений. "Удалить пару" удаляет выбранную пару. (ПРИЛОЖЕНИЕ 3, рисунок 3.10)

Кнопка "Добавить занятие" открывает диалоговое окно для создания нового занятия в расписании. В этом окне администратор заполняет все данные для добавления занятия. После чего данные сохраняются в базе данных. (ПРИЛОЖЕНИЕ 3, рисунок 3.11)

Для разработки данного функционала применялись книги [11], [12].

# Результат работы и тестирование

Для проверки корректности функционирования модуля «Расписание студента» были проведены тестирования всех основных функций приложения. Тесты включали как стандартные сценарии использования, так и случаи с ошибками ввода. Ниже приведены результаты тестирования с описанием тестовых случаев.

Кейс 1. Авторизация. Тестируемая функциональность: Авторизация

Тестовый набор: Проверка полей логин и пароль, ввод корректного логина и пароля.

Ожидаемый результат: успешный вход в систему, открытие главного меню. Результат: Вход выполнен корректно. Открылось главное окно. (ПРИЛОЖЕНИЕ 2 рисунок 2.1)

Кейс 2. Просмотр расписания. Тестируемая функциональность: Просмотр расписания

Тестовый набор: Проверка кнопки «Расписание на неделю»

Ожидаемый результат: Открывается расписание на неделю в виде таблицы. Результат: Таблица отображается корректно. (ПРИЛОЖЕНИЕ 2 рисунок 2.3)

Кейс 3. Экспорт расписания в Excel. Тестируемая функциональность: Экспортирование расписания в Excel

Тестовый набор: проверка кнопки «Экспорт расписания в Excel»

Ожидаемый результат: Появляется окно выбора пути файла. Файл сохраняется. Результат: Окно выбора пути файла корректно открылось. Файл сохранился по выбранному пути. (ПРИЛОЖЕНИЕ 2 рисунок 2.4)

Кейс 4. Поиск расписания по дате. Тестируемая функциональность: Поиск расписания по дате. Ввод существующей даты

Тестовый набор: Проверка поля ввода даты

Ожидаемый результат: Корректно отображается расписания на выбранную дату. Результат: Расписание корректно отобразилось. (ПРИЛОЖЕНИЕ 2 рисунок 2.5)

Кейс 5. Изменение расписания. Тестируемая функциональность: Изменение расписания

Тестовый набор: Поля выбора группы, даты, пары, преподавателя и аудитории. Ввод корректных значений.

Ожидаемый результат: Вывод сообщения «Изменения успешно сохранены». Результат: Сообщение корректно выводится. (ПРИЛОЖЕНИЕ 2 рисунок 2.7)

Кейс 6. Добавление нового пользователя. Тестируемая функциональность: Добавление нового пользователя

Тестовый набор: Поля выбора роли пользователя, логин, пароль, ФИО, идентификатор группы. Ввод корректных значений

Ожидаемый результат: Выводится сообщение «Пользователь успешно добавлен». Результат: Выводится сообщение «Пользователь Фёдоров Андрей Павлович успешно добавлен». (ПРИЛОЖЕНИЕ 2 рисунок 2.9)

Кейс 7. Добавление занятия в расписание. Тестируемая функциональность: Добавление нового пользователя

Тестовый набор: Поля выбора группы, преподавателя, предмета, аудитории, номера пары, дня недели, даты занятия. Ввод корректных значений

Ожидаемый результат: Выводится оповещение о том, что расписание успешно добавлено. Результат: Оповещение успешно выводится. (ПРИЛОЖЕНИЕ 2 рисунок 2.12)

Кейс 8. Выход в главное меню. Тестируемая функциональность: Возвращение в главное меню

Тестовый набор: Проверка кнопки «Выход в главное меню»

Ожидаемый результат: открывается главное меню пользователя. Результат: главное меню открылось корректно. (ПРИЛОЖЕНИЕ 2 рисунок 2.13)

Кейс 8. Выход из аккаунта. Тестируемая функциональность: Выход из аккаунта

Тестовый набор: Проверка кнопки «Выйти из аккаунта»

Ожидаемый результат: открывается окно авторизации. Результат: окно авторизации открылось корректно. (ПРИЛОЖЕНИЕ 2 рисунок 2.14)

Основываясь на тестировании, можно сказать, что разработанная система полностью отвечает функциональным требованиям. Все выявленные недочёты были исправлены. Структура проекта представлена в (ПРИЛОЖЕНИЕ 3, рисунок 3.12). Готовый продукт представлен на сайте github [10].

# ЗАКЛЮЧЕНИЕ

В рамках курсового проекта была разработана программная система «Расписание студента», ориентированная на автоматизацию процессов управления учебным расписанием. Главной целью разработки стало создание инструмента, который упрощает организацию данных, связанных с учебным процессом, и предоставляет пользователям различных ролей доступ к функциональности, необходимой для их работы.

Одним из ключевых результатов стало создание удобного и функционального пользовательского интерфейса, разработанного с использованием библиотеки PyQt5. Интерфейс отличается интуитивной понятностью, обеспечивая лёгкость навигации для каждого типа пользователя. Была разработана функциональность для управления расписанием. Пользователи могут просматривать актуальное расписание, экспортировать его в формат Excel, а также производить изменения, добавлять и удалять занятия. Проведено тестирование всех функций программы, включая проверку обработки ошибок, что подтвердило стабильность и корректность её работы.

Разработанный программный продукт может быть внедрён в учебных заведениях. Он может быть использован как на уровне отдельных групп или курсов, так и в масштабах факультета или учреждения в целом. Для успешного внедрения рекомендуется провести обучение пользователей основным принципам работы с системой, а также предоставить подробные инструкции по эксплуатации.

Возможным направлением дальнейшего развития является интеграция приложения с существующими системами управления образовательными процессами. Также перспективным является создание сетевой версии продукта, которая обеспечит доступ к данным из любой точки, можно добавить возможность работы в многопользовательском режиме через сеть.

Таким образом, разработанный модуль представляет собой практическое решение для автоматизации управления учебным расписанием, полностью отвечающее поставленным целям и задачам курсового проекта.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

Стандарты и законодательные материалы

1. ГОСТ Р ИСО/МЭК 12207-2010. Информационная технология. Системная и программная инженерия. Процессы жизненного цикла программных средств.
2. Единая система программной документации. Общие положения.
3. Федеральный закон от 26.07.2017 N 187-ФЗ (ред. от 10.07.2023) "О безопасности критической информационной инфраструктуры Российской Федерации".
4. Федеральный закон от 27.07.2006 N 149-ФЗ (ред. от 08.08.2024) "Об информации, информационных технологиях и о защите информации" (с изм. и доп., вступ. в силу с 19.08.2024).
5. Федеральный закон от 27.07.2006 N 152-ФЗ (ред. от 08.08.2024) "О персональных данных".
6. ГОСТ 7.32-2017 Система стандартов по информации, библиотечному и издательскому делу. Отчет о научно-исследовательской работе. Структура и правила оформления
7. ГОСТ 7.0.100-2018 Система стандартов по информации, библиотечному и издательскому делу. Библиографическая запись. Библиографическое описание. Общие требования и правила составления.
8. ГОСТ 7.80-2000 Система стандартов по информации, библиотечному и издательскому делу. Библиографическая запись. Заголовок. Общие требования и правила составления.
9. ГОСТ 9327-60 Бумага и изделия из бумаги. Потребительские форматы.

Монографии

1. Подробный код программного модуля «Расписание студента»: <https://github.com/polinkkq/Kylugina_KursovoyProect>

Иностранная литература

1. Марк Лутц. Изучаем Python, том 1, 5-е изд.: Пер. с англ. — СПб.: ООО “Диалектика”, 2019.
2. Уолтер Шилдс. SQL: быстрое погружение. — СПб.: Питер, 2022. — 224 с.: ил. — (Серия «Библиотека программиста»).

Интернет источники

1. 1С: Автоматизированное составление расписания. Университет: [*https://solutions.1c.ru/catalog/asp\_univer/features*](https://solutions.1c.ru/catalog/asp_univer/features)
2. Документация openpyxl. Автор Эрик Газони, Чарли Кларк: <https://openpyxl.readthedocs.io/en/stable/>
3. Официальная документация PyQt: <https://www.riverbankcomputing.com/software/pyqt/intro>
4. Официальная документация SQLite, раздел «О SQLite»: <https://www.sqlite.org/about.html>
5. Программный комплекс "ПЛАНЫ": [*https://www.mmis.ru/programs/plany*](https://www.mmis.ru/programs/plany)
6. Статья «PyCharm» материал из Википедии: <https://ru.wikipedia.org/wiki/PyCharm>
7. Экспертный центр корпорации «Галактика»: [*https://galaktika-it.ru/spb-2/ruz*](https://galaktika-it.ru/spb-2/ruz)

# ПРИЛОЖЕНИЯ

# ПРИЛОЖЕНИЕ 1 Диаграммы

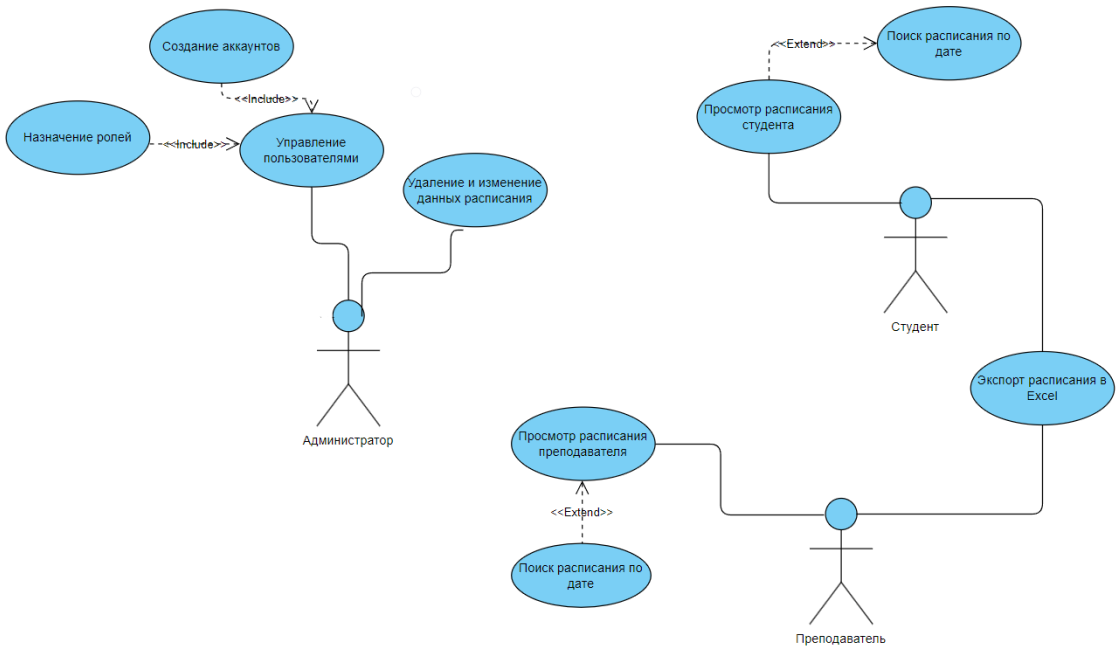


Рисунок 1.1 - Диаграмма сценариев

Изображение выглядит как текст, диаграмма, Параллельный, План

Автоматически созданное описание

Рисунок 1.2 – Диаграмма IDEF1X

Изображение выглядит как диаграмма, План, линия, текст

Автоматически созданное описание

Рисунок 1.3 – ER-диаграмма

# ПРИЛОЖЕНИЕ 2 План-тест

Таблица 2.1 - План-тест

| № | Наименование функциональности | Наименование поля | Тестовый набор | Ожидаемый результат | Результат тестирования |
| --- | --- | --- | --- | --- | --- |
| 1 | Авторизация | Поле ввода логина | StudentSt | Успешный вход в систему, открытие главного меню | Вход выполнен корректно. Открылось главное окно |
|  |  | Поле ввода пароля | Student123 |
| Рисунок 2.1 | | | | | |
|  |  | Поле ввода логина | StudentSt | Вывод сообщения об ошибке: 'Имя пользователя и пароль не могут быть пустыми.' | Вывелось сообщение об ошибке |
|  |  | Поле ввода пароля | Пустое значение |
| Рисунок 2.2 | | | | | |
| 2 | Просмотр расписания | Кнопка «Расписание на неделю» | Нажатие на кнопку «Расписание на неделю» | Открывается расписание на неделю в виде таблицы | Таблица отображается корректно |

Продолжение Таблицы 2.1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Наименование функциональности | Наименование поля | Тестовый набор | Ожидаемый результат | Результат тестирования |
| Рисунок 2.3 | | | | | |
| 3 | Экспорт расписания в Excel | Кнопка «Экспорт расписания в Excel» | Нажатие кнопки «Экспорт расписания в Excel». Выбор корректного пути сохранения файла | Появляется окно выбора пути файла. Файл сохраняется | Окно выбора пути файла корректно открылось. Файл сохранился по выбранному пути |
| Рисунок 2.4 | | | | | |
| 4 | Поиск расписания по дате | Поле ввода даты | Дата: 2024-11-05 | Корректно отображается расписания на выбранную дату | Расписание корректно отобразилось |

Продолжение Таблицы 2.1

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| № | Наименование функциональности | Наименование поля | Тестовый набор | | Ожидаемый результат | Результат тестирования |
| Рисунок 2.5 | | | | | | |
| 4 | Поиск расписания по дате | Поле ввода даты | Дата: 2025-08-16 | Появится сообщение: «Расписание отсутствует на указанную дату.» | | Сообщение корректно отобразилось |
| Рисунок 2.6 | | | | | | |
| 5 | Изменение расписания | Выбор группы, даты, пары, преподавателя, аудитории | Группа: 201ИС, Дата: 2024-11-04  Пара: 3  Преподаватель: Петров Пётр Петрович  Аудитория: 104 | Вывод сообщения «Аудитория занята в указанное время» | | Сообщение корректно выводится |
| Рисунок 2.7 | | | | | | |

Продолжение Таблицы 2.1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Наименование функциональности | Наименование поля | Тестовый набор | Ожидаемый результат | Результат тестирования |
|  | Изменение расписания | Выбор группы, даты, пары, преподавателя, аудитории | Группа: 202ИС,  Дата: 2024-11-04  Пара: 3  Преподаватель: Николаев Вячеслав Андреевич  Аудитория: 105 | Выводится сообщение о том, что изменения успешно сохранены | Сообщение корректно выводится |
| Рисунок 2.8 | | | | | |
| 6 | Добавление нового пользователя | Роль пользователя, поле логин, поле пароль, поле ФИО, идентификатор группы | Роль пользователя: студент  Логин: StudentAndrey  Пароль: student123  ФИО: Фёдоров Андрей Павлович  Идентификатор группы: 301ИС | Выводится сообщение о том, что пользователь успешно добавлен | Сообщение успешно выводится |
| Рисунок 2.9 | | | | | |

Продолжение Таблицы 2.1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Наименование функциональности | Наименование поля | Тестовый набор | Ожидаемый результат | Результат тестирования |
|  |  | Роль пользователя, поле логин, поле пароль, поле ФИО, идентификатор группы | Роль пользователя: студент  Логин: пустое поле  Пароль: пустое поле  ФИО: пустое поле  Идентификатор группы: 101ИС | Выводится ошибка о том, что все поля должны быть заполнены | Ошибка успешно выводится |
| Рисунок 2.10 | | | | | |
| 7 | Добавление занятия в расписание | Поле группа, преподаватель, предмет, аудитория, номер пары, день недели, дата занятия | Поле группа: 101ИС  Преподаватель: Васильев Владимир Львович  Предмет: Обществознание  Аудитория: 101  Номер пары: 1 – 09:00 – 10:30  День недели: понедельник  Дата занятия: 06.11.2024 | Выводится ошибка о том, что указанная аудитория уже занята на указанное время | Ошибка успешно выводится |
| Рисунок 2.11 | | | | | |

Продолжение Таблицы 2.1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Наименование функциональности | Наименование поля | Тестовый набор | Ожидаемый результат | Результат тестирования |
|  | Добавление занятия в расписание | Поле группа, преподаватель, предмет, аудитория, номер пары, день недели, дата занятия | Поле группа: 301ИС  Преподаватель: Клеменко Владимир Николаевич  Предмет: География  Аудитория: 103  Номер пары: 4 – 14:40 – 16:10  День недели: четверг  Дата занятия: 07.11.2024 | Выводится оповещение о том, что расписание успешно добавлено | Оповещение успешно выводится |
| Рисунок 2.12 | | | | | |
| 8 | Возвращение в главное меню | Кнопка «Вернуться в главное меню» | Нажатие кнопки «Вернуться в главное меню» | Текущее окно закрывается и открывается окно главное меню пользователя | Текущее окно закрылось, а главное меню открылось корректно |

Продолжение Таблицы 2.1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Рисунок 2.13 | | | | | |
| 9 | Выход из аккаунта | Кнопка «Выйти из аккаунта» | Нажатие кнопки «Выйти из аккаунта» | Текущее окно закрывается и открывается окно авторизации | Текущее окно закрылось, а окно авторизации открылось корректно |
| Рисунок 2.14 | | | | | |

# ПРИЛОЖЕНИЕ 3 Функциональные узлы

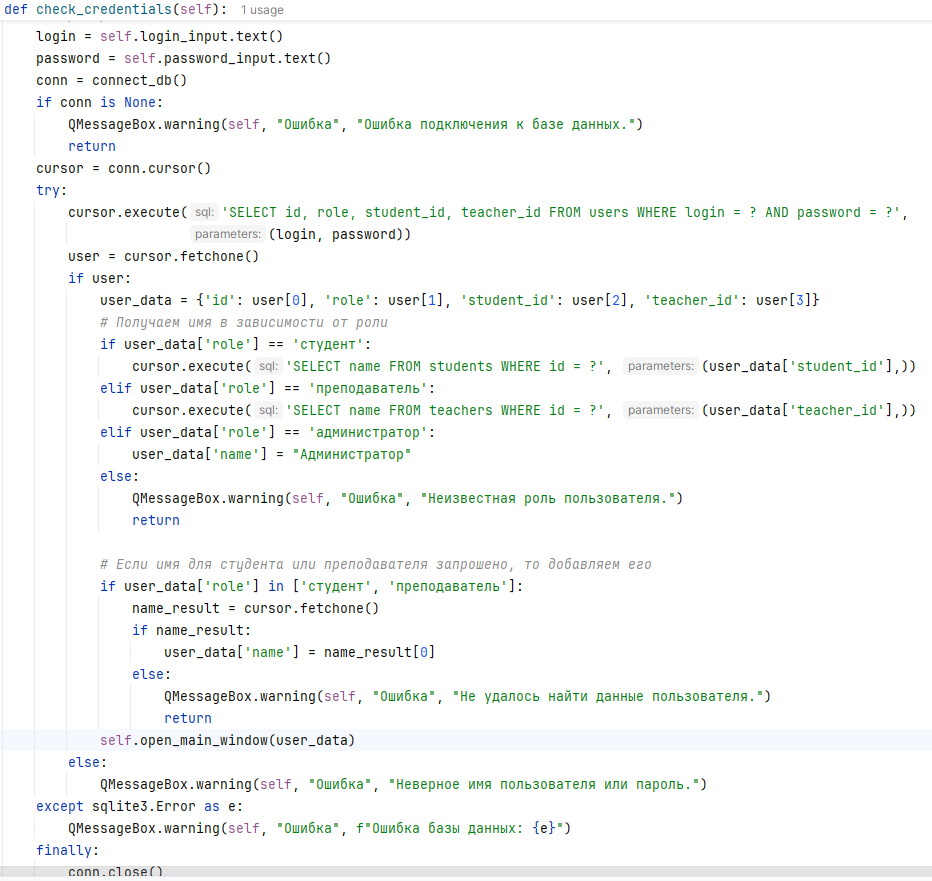


Рисунок 3.1 – Функционал кнопки «Войти»



Рисунок 3.2 – Функционал кнопки «Показать расписание»

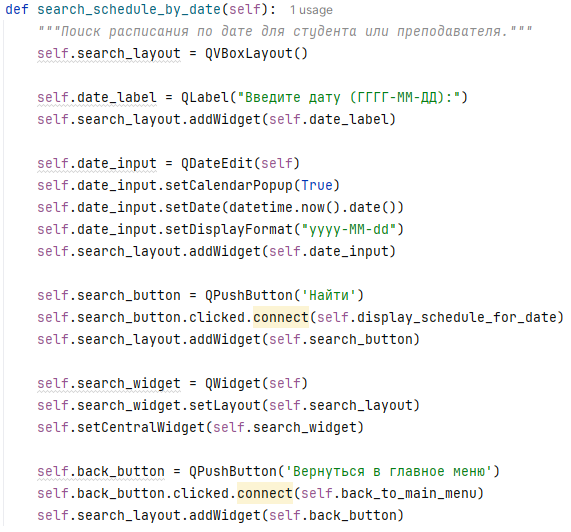


Рисунок 3.3 – Функционал кнопки «Найти расписание по дате»



Рисунок 3.4 – Функционал кнопки «Экспорт в Excel»



Рисунок 3.5 – Функционал кнопки «Найти» в окне поиска расписания по дате

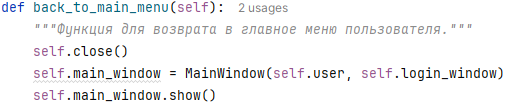


Рисунок 3.6 – функция кнопки «

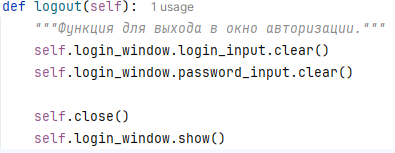


Рисунок 3.7 - Функционал кнопки «Выйти»



Рисунок 3.8 – Функционал кнопки «Добавить пользователя»



Рисунок 3.9 – Функционал кнопки «Изменить/удалить занятие»

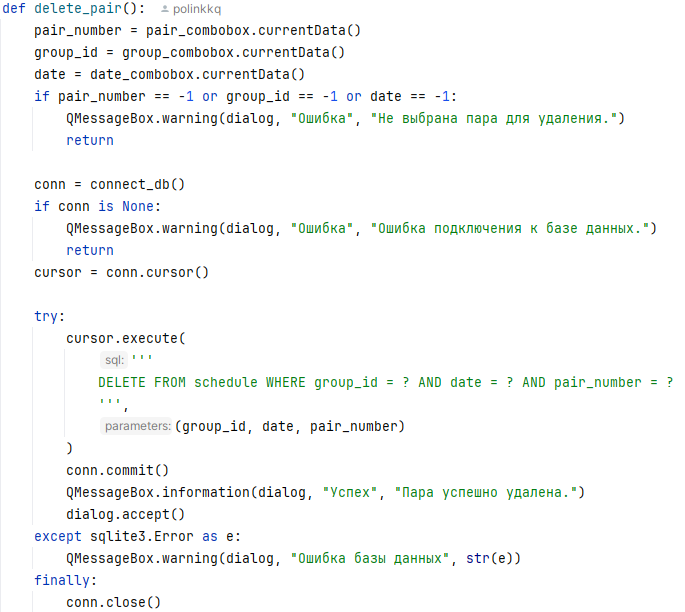


Рисунок 3.10 – Функционал кнопки «Удалить» в окне изменения занятия



Рисунок 3.11 – Функционал кнопки «Добавить занятие»

Изображение выглядит как текст, снимок экрана, программное обеспечение, веб-страница

Автоматически созданное описание

Рисунок 3.12 – Структура проекта