

**КУРСОВОЙ ПРОЕКТ**

**по дисциплине: МДК 1.01 Разработка программных модулей**

**Тема: Разработка программного модуля**

**информационной системы «Читатели студенческой библиотеки»**

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

|  |  |  |
| --- | --- | --- |
| **Выполнил студент(ка) группы 313ИС-22** | **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** | **Е.В. Маньковская** |
| **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»**

**Студентка группы 313ИС-22 Маньковская Екатерина**

**ТЕМА: Разработка программного модуля информационной**

**системы «Читатели студенческой библиотеки»**

1. Изучить современные потребности студентов в библиотечных услугах.
2. Проанализировать существующие решения для автоматизации библиотек.
3. Разработать техническое задание.
4. Подготовить план тестирования.
5. Обосновать выбор технологий.
6. Реализовать техническое задание.
7. Провести тестирование.

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

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

**Москва 2024**

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 4](#_Toc184338170)

[1. ОПИСАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ 6](#_Toc184338171)

[1.1. Введение в предметную область 6](#_Toc184338172)

[1.2. Анализ готовых решений 8](#_Toc184338173)

[2. ПРОЕКТИРОВАНИЕ СИСТЕМЫ 10](#_Toc184338174)

[2.1. Общее назначение программы 10](#_Toc184338175)

[2.2. Требования к функциям, выполняемым системой 11](#_Toc184338176)

[2.3. План тестирования 14](#_Toc184338177)

[3. РЕАЛИЗАЦИЯ ПРОЕКТНОЙ СИСТЕМЫ 16](#_Toc184338178)

[3.1. Описание среды разработки 16](#_Toc184338179)

[3.1.1. Выбор и описание программных инструментов 16](#_Toc184338180)

[3.1.2. Обоснование выбора инструментария по разработке 17](#_Toc184338181)

[3.2. Разработка программного модуля 18](#_Toc184338182)

[3.2.1. Реализация пользовательского интерфейса программы 19](#_Toc184338183)

[3.2.2. Описание кодом функциональных узлов модуля 20](#_Toc184338184)

[3.2.3. Результат работы и тестирования 22](#_Toc184338185)

[ЗАКЛЮЧЕНИЕ 27](#_Toc184338186)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 28](#_Toc184338187)

[ПРИЛОЖЕНИЕ 1 Диаграмма прецедентности 30](#_Toc184338188)

[ПРИЛОЖЕНИЕ 2 План-тест 31](#_Toc184338189)

[ПРИЛОЖЕНИЕ 3 Функционал кнопок 41](#_Toc184338190)

[ПРИЛОЖЕНИЕ 4 Листинг кода 53](#_Toc184338191)

# **ВВЕДЕНИЕ**

Студенческие библиотеки играют важную роль в образовательном процессе, предоставляя доступ к информационным ресурсам и поддерживая учебную и научную деятельность. Однако в условиях цифровой трансформации библиотеки нуждаются в оптимизации и адаптации к новым требованиям пользователей. Автоматизация процессов учёта, управления фондами и взаимодействия с читателями через цифровые технологии является важным шагом в этом направлении.

Для повышения эффективности работы библиотек требуется создание специализированных приложений, которые облегчают доступ к ресурсам и автоматизируют обработку данных о книгах, читателях и посещениях. Цифровизация этих процессов улучшает качество обслуживания, сокращает время на рутинные операции и предоставляет студентам удобные инструменты для взаимодействия с библиотекой.

Целью разработки программного модуля системы «Читатели студенческой библиотеки» является создание приложения, которое будет способствовать упрощению работы с архивами и данными, что в итоге повысит эффективность работы библиотеки и удовлетворение информационных потребностей студентов.

Для достижения цели необходимо решить следующие задачи:

1. Изучить современные потребности студентов в библиотечных услугах: определить, какие виды информации и сервисы наиболее востребованы среди студентов, а также выделить функциональные возможности, которые могут удовлетворить эти запросы.

2. Проанализировать существующие решения для автоматизации библиотек: изучить текущие приложения и системы управления библиотеками, вы-явить их сильные и слабые стороны.

3. Разработать техническое задание: сформулировать основные требования к функционалу приложения, обеспечив его соответствие потребностям студентов и библиотек.

4. Подготовить план тестирования: создать сценарии для проверки работы приложения, включая оценку стабильности и производительности.

5. Обосновать выбор технологий: провести сравнительный анализ инструментов разработки и выбрать оптимальные решения для реализации приложения.

6. Реализовать техническое задание: разработать приложение, включая проектирование интерфейса, настройку базы данных и внедрение всех ключевых функций.

7. Провести тестирование: проверить корректность работы приложения, его удобство использования и эффективность для выполнения задач.

Объектом исследования является процесс автоматизации учета данных студенческой библиотеки. Учёт данных включает в себя обработку информации о посещениях студентов, книгах и услугах предоставляемые сотрудником библиотеки.

Предметом исследования являются учет, хранение и обработка данных о библиотеке.

С развитием цифровых технологий студенты предпочитают получать доступ к библиотечным ресурсам через удобные электронные сервисы. Современные библиотеки должны использовать цифровые инструменты для упрощения работы с архивами, книгами и данными о студентах. Разработка такого приложения оптимизирует деятельность библиотек и улучшает качество обслуживания, что делает исследование актуальным.

Таким образом, разработка приложения для читателей студенческой библиотеки станет важным шагом на пути к цифровизации и оптимизации работы библиотеки, а также улучшению взаимодействия студентов с её ресурсами. Поставленные в работе задачи, такие как анализ существующих решений, подготовка технического задания, выбор инструментов и тестирование приложения, позволят создать эффективное и функциональное решение, соответствующее современным требованиям образовательной среды и потребностям читательской аудитории.

# **ОПИСАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ**

## **Введение в предметную область**

До появления автоматизированных систем студенческие библиотеки использовали карточные каталоги для учёта литературы. Поиск книг требовал значительного времени, так как студентам приходилось вручную просматривать карточки, которые могли теряться или заполняться с ошибками. Контроль за выдачей книг также вёлся вручную: библиотекари записывали данные о книгах и читателях в журналы, что часто приводило к задержкам в учёте и путанице.

Информатизация позволила значительно упростить процессы через автоматизированные библиотечные системы. Электронные каталоги ускоряют поиск книг по ключевым данным, а автоматизированный учет позволяет библиотекарям отслеживать выданные книги и напоминать читателям о необходимости возврата. Электронные системы также поддерживают новые функции: упрощение поиска книг, аналитика и отчётность, бронирование, продление срока использования и аналитика предпочтений пользователей.

Основной принцип интеграции ИТ в библиотеки заключается в создании единой системы управления, которая позволяет осуществлять:

* Учёт и управление фондами (книгами, журналами, электронными ресурсами).
* Предоставление пользователям доступа к электронным ресурсам и оцифрованной литературе.
* Автоматизацию процессов выдачи и возврата книг.

Современные электронные системы для библиотек должны соответствовать ряду требований, таких как:

1. Удобство использования: Интерфейсы электронных систем должны быть интуитивно понятными для пользователей любого уровня подготовки.
2. Доступность и безопасность: Электронные системы обязаны обеспечивать защиту персональных данных пользователей и данных о книгах, а также иметь доступ к данным 24/7.
3. Соответствие законодательству: Для разработки информационной системы (ИС) библиотеки важно учитывать ряд законодательных и стандартных требований. Вот основные документы, регулирующие такие разработки:

Федеральный закон №152-ФЗ "О персональных данных" — регулирует обработку и защиту персональных данных, чтобы обеспечить конфиденциальность информации, предоставляемой пользователями. Согласно этому закону, библиотечные системы, которые хранят и обрабатывают личные данные, обязаны принимать меры для предотвращения их утечки и несанкционированного доступа. [[​5]](#_СПИСОК_ИСПОЛЬЗОВАННЫХ_ИСТОЧНИКОВ) Глава 1, Статья 1-2

Федеральный закон №149-ФЗ "Об информации, информационных технологиях и о защите информации" — регулирует все аспекты использования информационных технологий и защиты информации. Он предписывает установку необходимых средств защиты в ИС, что важно для защиты данных библиотеки от внешних угроз и несанкционированного доступа. [[4]](#_СПИСОК_ИСПОЛЬЗОВАННЫХ_ИСТОЧНИКОВ) Статья 1-2

Федеральный закон №187-ФЗ "О безопасности критической информационной инфраструктуры Российской Федерации" — направлен на защиту объектов, которые относятся к критической информационной инфраструктуре. Хотя библиотеки не всегда входят в перечень таких объектов, этот закон может быть полезен при разработке системы с повышенными требованиями к информационной безопасности. [[3]](#_СПИСОК_ИСПОЛЬЗОВАННЫХ_ИСТОЧНИКОВ) Статья 1-2

ГОСТ Р ИСО/МЭК 12207 — международный стандарт, определяющий процессы жизненного цикла программного обеспечения, включая разработку, эксплуатацию и поддержку. Применение этого стандарта помогает обеспечить высокий уровень качества и совместимость разрабатываемого программного обеспечения с другими системами и требованиями безопасности. [[1]](#_СПИСОК_ИСПОЛЬЗОВАННЫХ_ИСТОЧНИКОВ) Глава 1, Пункт 1.1

ЕСПД (Единая система программной документации) — это комплекс стандартов, разработанных для создания, сопровождения, оформления и эксплуатации программной документации. Сопровождение программы включает анализ функционирования, развитие и совершенствование программы, а также внесение изменений в нее с целью устранения ошибок. [[2]](#_СПИСОК_ИСПОЛЬЗОВАННЫХ_ИСТОЧНИКОВ) Пункт 1

1. Поддержка разных форматов: Важно, чтобы электронные системы могли работать с разнообразными форматами книг и документов, включая текстовые и мультимедийные файлы.

Для разработки библиотечных систем чаще всего используется каскадный (водопадный) подход, который предполагает последовательное выполнение этапов проекта:

1. Сбор и анализ требований: Изучаются потребности библиотеки и пользователей, формулируются цели системы.

2. Проектирование: Создаются проектные документы, описывающие структуру и функциональность системы.

3. Разработка: Пишется код, разрабатываются интерфейсы и интегрируются базы данных.

4. Тестирование: Проверяется система на наличие ошибок и соответствие требованиям.

5.Внедрение: Система вводится в эксплуатацию, пользователи и библиотекари проходят обучение.

6. Сопровождение: Обслуживание системы, исправление неполадок и доработка функционала.

## **Анализ готовых решений**

На современном рынке существует несколько информационных систем, которые активно используются в сфере управления библиотеками и предоставления электронных библиотечных услуг. Рассмотрим наиболее популярные решения, применяемые в образовательных учреждениях:

1. Государственная публичная научно-техническая библиотека России  
   ГПНТБ России работает над инновациями в автоматизации библиотек, включая разработку и реализацию крупных федеральных и международных проектов в области библиотечно-информационных технологий. Она является учредителем Международной ассоциации пользователей и разработчиков электронных библиотек и новых информационных технологий.[[8]](#_СПИСОК_ИСПОЛЬЗОВАННЫХ_ИСТОЧНИКОВ)
2. Ruslan  
   АБИС "Ruslan" — система для автоматизации работы с библиотечными фондами и пользователями, поддерживающая учёт книг и читателей. Она интегрируется с национальными библиотеками и учебными учреждениями, а также предоставляет инструменты для работы с каталогами и аналитическими базами данных. [[9]](#_СПИСОК_ИСПОЛЬЗОВАННЫХ_ИСТОЧНИКОВ)
3. ЭБС Znanium

Электронно-библиотечная система Znanium предоставляет доступ к большой коллекции электронных изданий и учебной литературы для студентов, преподавателей. Она поддерживает интеграцию с вузовскими информационными системами и включает инструменты для анализа научных публикаций. [[16]](#_СПИСОК_ИСПОЛЬЗОВАННЫХ_ИСТОЧНИКОВ)

Анализ существующих решений показал, что на рынке представлено множество систем для управления библиотечными фондами и доступа к электронным изданиям. Их внедрение улучшает качество обслуживания, обеспечивая быстрый доступ к ресурсам и интеграцию с внешними образовательными системами.

Для разработки нового приложения важно учитывать опыт существующих систем, чтобы создать функционал, отвечающий требованиям современной цифровой среды.

Таким образом, в первой главе был проанализирован процесс автоматизации работы библиотек, включая переход от традиционных методов учёта к современным автоматизированным системам. Рассмотрены преимущества информатизации. Описаны основы интеграции технологий в библиотечную деятельность и требования к современным системам, включая функциональность, надежность и соответствие законодательству.

# **ПРОЕКТИРОВАНИЕ СИСТЕМЫ**

## **Общее назначение программы**

Внедрение разработанного программного модуля для учёта читателей студенческой библиотеки предполагает значительные экономические и организационные выгоды. Прежде всего, автоматизация процессов управления библиотечным фондом и взаимодействия с пользователями существенно сократит трудозатраты сотрудников библиотеки на выполнение рутинных операций, таких как учёт книг, регистрация новых читателей, обработка заявок на бронирование литературы и контроль за возвратом.

Экономический эффект будет выражен в следующих аспектах:

* Снижение трудозатрат. Сокращение времени на выполнение библиотекарями ручных операций приведёт к экономии ресурсов на оплату труда и позволит уменьшить количество ошибок, связанных с человеческим фактором.
* Повышение эффективности управления. Быстрый доступ к информации о наличии литературы, истории использования и доступных ресурсах позволит библиотеке лучше управлять своим фондом, что сократит расходы на приобретение излишков литературы.
* Увеличение удовлетворённости пользователей. Улучшение взаимодействия с читателями через электронные сервисы, такие как бронирование и продление срока использования книг, обеспечит удобство для студентов и повысит популярность библиотеки.
* Снижение издержек на обслуживание фондов. Электронная система учёта снизит необходимость в физическом хранении карточных каталогов и прочих бумажных документов, что уменьшит затраты на материалы и организацию хранения данных.

Таким образом, внедрение программного модуля создаст условия для более рационального использования ресурсов библиотеки и повышения качества обслуживания читателей.

## **Требования к функциям, выполняемым системой**

Исходя из анализа предметной области, система для учёта читателей студенческой библиотеки должна включать следующие функциональные задачи:

* Улучшение взаимодействия с читателями. Программа должна предоставлять возможность регистрации читателей, обработки их запросов, бронирования и продления литературы через личные кабинеты.
* Автоматизация учёта библиотечного фонда. Необходимо обеспечить автоматический учёт поступления и выдачи книг, с возможностью отслеживания их местонахождения и сроков возврата.
* Ведение статистики и аналитики. Система должна собирать информацию о популярности книг, активности читателей и текущем состоянии фонда, предоставляя отчёты для оптимизации работы библиотеки.
* Интеграция с внешними системами. Возможность синхронизации с образовательными платформами и системами учёта студентов для автоматической проверки информации о пользователях.

Для реализации данного функционала требуется анализ среды разработки. Система должна быть гибкой, масштабируемой, поддерживать веб-интерфейсы и иметь интуитивно понятный интерфейс. Для создания диаграмм и разработки модуля выделены основные сущности, представленные в таблице 1.

Таблица 1 – Основные объекты

| Номер | Наименование объекта | Краткое описание |
| --- | --- | --- |
| 1 | Пользователь | Пользователи системы, которые могут просматривать доступные книги, бронировать их и возвращать. |
| 2 | Книга | Элемент фонда библиотеки, доступный для выдачи читателю. |
| 3 | Посещение | Записи о посещениях читателей в систему. |
| 4 | Выданные книги | Информация о книгах, находящихся на руках у читателей. |
| 5 | Бронирование | Список заявок на книги, которые читатели хотят временно зарезервировать. |

После определения основных объектов следует отбор атрибутов для каждой сущности. Информация приведена в таблицах 2-6.

Таблица 2 - сущность «Пользователь»

|  |  |  |
| --- | --- | --- |
| Атрибут | Тип | Описание |
| ID | Числовой | Уникальный идентификатор читателя. |
| ФИО | Короткий текст | Полное имя читателя. |
| Группа студента | Короткий текст | Группа, в которой обучается студент-читатель. |
| Имя пользователя | Короткий текст | Уникальное имя пользователя для входа в систему. |
| Пароль | Короткий текст | Ключ безопасности каждого аккаунта. |
| Роль пользователя | Короткий текст | Роль пользователя: "Читатель" или "Библиотекарь". |

Таблица 3 - сущность «Книга»

|  |  |  |
| --- | --- | --- |
| Атрибут | Тип | Описание |
| ID | Числовой | Уникальный идентификатор книги |
| Название | Короткий текст | Название книги |
| Автор | Короткий текст | Автор книги |
| Жанр | Короткий текст | Жанр книги |
| Статус | Короткий текст | Статус книги (выдана/забронирована/доступна) |

Таблица 4 - сущность «Посещение»

| Атрибут | Тип | Описание |
| --- | --- | --- |
| ID | Числовой | Уникальный идентификатор записи о посещении. |
| Имя пользователя | Короткий текст | Имя пользователя, который совершил посещение. |
| Дата посещения | Короткий текст | Дата, когда пользователь посетил библиотеку. |

Таблица 5 - сущность «Выданные книги»

| Атрибут | Тип | Описание |
| --- | --- | --- |
| ID | Числовой | Уникальный идентификатор записи о выданной книге. |
| Имя пользователя | Короткий текст | Имя пользователя, которому была выдана книга. |
| ID Книги | Числовой | Уникальный идентификатор книги, которая была выдана. |
| Дата выдачи | Короткий текст | Дата, когда книга была выдана пользователю. |
| Ожидаемая дата возврата | Короткий текст | Предполагаемая дата, к которой книга должна быть возвращена. |
| Статус | Короткий текст | "Выдано", "Доступно" или "Забронировано". |

Таблица 6 - сущность «Бронирование»

|  |  |  |
| --- | --- | --- |
| Атрибут | Тип | Описание |
| ID | Числовой | Уникальный идентификатор записи о бронировании. |
| Имя пользователя | Короткий текст | Уникальный идентификатор пользователя, который создал бронь. |
| ID Книги | Числовой | Уникальный идентификатор книги, забронированной пользователем. |
| Время бронирования | Дата/Время | Дата и время создания записи о бронировании. |
| Ожидаемая дата возврата | Короткий текст | Дата, до которой книга забронирована пользователем. (По умолчанию бронь храниться 24 часа) |
| Статус | Короткий текст | Состояние записи: "Выдано", "Доступно" или "Забронировано". |

На основании целей разработки модуля и анализа требований определены основные действия пользователей в системе «Читатели студенческой библиотеки». Сценарии использования описывают ключевые взаимодействия, обеспечивая удобство и эффективность работы. Пользователи системы могут выполнять следующие действия:

Библиотекарю доступны такие функции как: авторизация, поиск и просмотр произведений, выдача и возврат, просмотр выданных книг, просмотр и отмена всех бронирований, просмотр посещений студентов.

Читателю доступна авторизация, поиск и просмотр книг, бронирование книги, просмотр забронированных книг читателем, просмотр выданных книг.

Диаграмма прецедентности представлена в [Приложении 1, рисунок 1.1](#_ПРИЛОЖЕНИЕ_1_Диаграмма)

ER-диаграмма (Entity-Relationship Diagram) позволяет визуализировать взаимосвязи между основными объектами системы: пользователь, книга, посещение, выданные книги и бронирование. Диаграмма представлена на Рисунке 1.

Изображение выглядит как диаграмма, текст, рисунок, План

Автоматически созданное описание

Рисунок 1 - ER-диаграмма

## **План тестирования**

Тестирование модуля системы "Читатели студенческой библиотеки" направлено на подтверждение его качества и соответствия функциональным и нефункциональным требованиям.

Цель тестирования программного модуля заключается в подтверждении его качества и соответствия заранее определенным функциональным и нефункциональным требованиям. Тестирование направлено на обнаружение дефектов в функциональности, подтверждение правильности требований, проверку производительности и стабильности в разных условиях, защиту данных пользователей, оценку совместимости с различными средами.

Задачи тестирования:

* Проверка функциональности. Тестирование основных функций приложения, а также проверка работы системы с разным количеством пользователей.
* Обеспечение безопасности. Тестирование аутентификации, авторизации и конфиденциальности данных пользователей, а также соответствия требованиям по защите данных и их резервному копированию.
* Тестирование совместимости. Проверка совместимости программного модуля с Excel.
* Оценка удобства использования. Проверка интерфейса на доступность для разных пользователей, оценка эргономики и удобства

Объекты тестирования: авторизация, выдача книг, возврат книг, бронирование и отмена книги, просмотр забронированных книг, просмотр выданных книг, просмотр посещений, проверка перехода между окнами, поиск книг.

Обеспечение безопасности данных:

1. Аутентификация. Проверка корректности процесса входа в систему, надежность и стойкость системы аутентификации против попыток несанкционированного доступа.
2. Авторизация. Проверка прав пользователей на доступ к различным функциям системы.

Таким образом, программный модуль «Читатели студенческой библиотеки» удовлетворяет потребности читателей и сотрудников, автоматизируя процессы, такие как поиск, бронирование, выдача и возврат книг, а также фиксацию посещений. Реализация модуля снижает рутинные операции, повышает качество обслуживания и упрощает работу библиотеки.

# **РЕАЛИЗАЦИЯ ПРОЕКТНОЙ СИСТЕМЫ**

## **Описание среды разработки**

**PyCharm** — PyCharm - полноценная среда разработки для профессиональной разработки на Python, предоставляющая все необходимое в чистом, интуитивно понятном и настраиваемом пользовательском интерфейсе. [[12]](#_СПИСОК_ИСПОЛЬЗОВАННЫХ_ИСТОЧНИКОВ)

#### Основные возможности PyCharm:

* **Автодополнение кода** — помогает ускорить процесс написания и минимизирует ошибки.
* **Отладчик** — позволяет пошагово отслеживать выполнение программы и проверять значения переменных.
* **Поддержка библиотек** — PyCharm поддерживает библиотеки для работы с базами данных, такие как sqlite3, а также для создания графических интерфейсов через **PyQt**.
* **Тестирование** — интеграция с фреймворком **pytest** для автоматизации тестирования.
* **Виртуальные окружения** — возможность создания изолированных окружений для управления зависимостями проекта.
* **Интерфейс для работы с базами данных** — PyCharm позволяет выполнять SQL-запросы и просматривать данные базы данных.

### **Выбор и описание программных инструментов**

Для реализации программного модуля "Читатели студенческой библиотеки" были выбраны следующие программные инструменты:

1. Язык программирования Python:

Python – это высокоуровневый язык программирования, который был разработан в конце 1980-х годов. Его разработчик, Гвидо ван Россум, вложил в основу языка простоту и читабельность кода, что позволяет использовать Python для быстрой и эффективной разработки. У Python большая библиотека сторонних модулей и инструментов, что делает его мощным инструментом. [[14]](#_СПИСОК_ИСПОЛЬЗОВАННЫХ_ИСТОЧНИКОВ)

1. Библиотека PyQt5:

PyQt5 — это полный набор привязок Python для Qt v5. Он реализован в виде более чем 35 модулей расширения и позволяет использовать Python в качестве альтернативного языка разработки приложений вместо C++ на всех поддерживаемых платформах, включая iOS и Android.Библиотека Qt является одной из самых мощных библиотек GUI (графического интерфейса пользователя).[[13]](#_СПИСОК_ИСПОЛЬЗОВАННЫХ_ИСТОЧНИКОВ)

1. Библиотека SQLite:

SQLite — это библиотека, которая реализует самодостаточный, бессерверный, не требующий настройки, транзакционный механизм баз данных SQL. Код SQLite находится в общественном достоянии и, следовательно, является бесплатным для использования в любых целях, коммерческих или личных. [[15]](#_СПИСОК_ИСПОЛЬЗОВАННЫХ_ИСТОЧНИКОВ)

1. Библиотека openpyxl:

Openpyxl предназначена для обработки файлов формата Excel (.xlsx) в языке программирования Python. Она обеспечивает широкие возможности для работы с данными в электронных таблицах, включая чтение, запись, редактирование и анализ содержимого Excel файлов. [[11]](#_СПИСОК_ИСПОЛЬЗОВАННЫХ_ИСТОЧНИКОВ)

### **Обоснование выбора инструментария по разработке**

Для разработки модуля «Читатели студенческой библиотеки» был выбран следующий инструментарий, основываясь на критериях функциональности, простоты интеграции и стабильности.

1. PyCharm использовался как основная среда разработки благодаря поддержке автодополнения, отладки и интеграции с внешними библиотеками, такими как PyQt5 и SQLite, что ускорило процесс разработки.
2. Python был выбран за простоту использования, поддержку множества библиотек и активное сообщество, что делает его идеальным для приложений с графическим интерфейсом и базами данных.
3. PyQt5 применялся для создания графического интерфейса. Библиотека обеспечивает разработку кроссплатформенных приложений с современным интерфейсом и удобством использования.
4. SQLite выбрана за лёгкость и высокую производительность. Эта СУБД подходит для локальных приложений с небольшими объёмами данных.
5. Openpyxl используется для работы с Excel-файлами, что необходимо для генерации отчета о посещениях студентов в формате таблиц. Библиотека отличается простотой применения и хорошей поддержкой сообщества.

## **Разработка программного модуля**

Разработка программного модуля "Читатели студенческой библиотеки" включает несколько ключевых этапов, направленных на создание удобной и функциональной системы.

На первом этапе проектируется интерфейс с использованием PyQt5. Разрабатываются окна для авторизации, просмотра книг, бронирования и возврата. Интуитивно понятный интерфейс включает кнопки, поля ввода и списки для удобной навигации пользователей.

Следующий этап включает реализацию логики приложения: авторизация, бронирование, выдача книг, учет посещений. Взаимодействие с базой данных на SQLite позволяет эффективно хранить данные о пользователях, книгах и операциях в компактной структуре.

Для создания отчетности используется Openpyxl, что позволяет генерировать данные о посещениях и операциях с книгами в формате Excel. Это упрощает мониторинг активности и управление ресурсами библиотеки.

После завершения разработки проводится тестирование на функциональность, стабильность и безопасность системы. Проверяется корректность выполнения всех задач, удобство интерфейса и защита данных.

Финальный этап включает оптимизацию, исправление ошибок и документирование изменений. Результатом является удобная, стабильная и безопасная система, которая автоматизирует ключевые процессы работы библиотеки.

### **Реализация пользовательского интерфейса программы**

Разработка пользовательского интерфейса для модуля «Читатели студенческой библиотеки» включает создание нескольких окон, каждое из которых отвечает за выполнение определенных функций. В интерфейсе программы предусмотрено шесть основных окон, которые обеспечивают комфортное взаимодействие с пользователем.

1. Окно авторизации.

Это стартовое окно программы, где пользователи вводят свои логин и пароль для входа в систему. Окно содержит поля ввода, кнопку подтверждения входа. Представлено на рисунке 2.

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

Рисунок 2 - Окно авторизации.

2. Главное окно пользователя.

После успешной авторизации пользователь попадает на основной экран программы. Для читателей здесь отображается поиск книг. Также доступны кнопки для перехода в разделы: «Забронировать книгу», «Мои брони», «Мои выданные книги», «Выйти». Представлено на рисунке 3.

Изображение выглядит как текст, снимок экрана, дисплей, программное обеспечение

Автоматически созданное описание

Рисунок 3 - Окно читателя.

Окно библиотекаря содержит такие кнопки как: «Просмотр броней», «Выдать книгу», «Просмотр выданных книг», «Посещения студентов», «Выйти». Также сверху окна отображается поиск книг. Представлено на рисунке 4.

Изображение выглядит как текст, снимок экрана, дисплей, программное обеспечение

Автоматически созданное описание

Рисунок 4 - Окно библиотекаря.

### **Описание кодом функциональных узлов модуля**

В этом разделе мы подробно опишем функциональные узлы модуля «Читатели студенческой библиотеки» и их реализацию кодом. Основной акцент сделан на ключевых функциях, обеспечивающих корректную работу программы и взаимодействие пользователя с системой.

* 1. «Войти». Кнопка авторизации пользователя. После ввода логина и пароля пользователь нажимает её для проверки введенных данных с базой. При успешной аутентификации открывается соответствующий интерфейс: для читателя или библиотекаря. Смотреть Приложение 3, рисунок 3.1.
  2. «Выйти». Завершает текущую сессию пользователя, закрывает все активные окна и возвращает интерфейс к стартовому окну авторизации. Смотреть Приложение 3, рисунок 3.2
  3. «Забронировать книгу». Кнопка позволяет читателю оформить резервирование выбранной книги. При её нажатии данные о брони (название книги, автор, жанр, срок брони) записываются в базу, а статус книги изменяется на «Забронирована». Смотреть Приложение 3, рисунок 3.3
  4. «Мои брони». Открывает окно, где пользователь видит свои текущие бронирования. Для каждой записи отображается полное название книги, автор, жанр и дата окончания брони, что позволяет планировать визит в библиотеку. Смотреть Приложение 3, рисунок 3.4
  5. «Назад». Кнопка для перехода в предыдущее окно или возвращения к главному меню. Она обеспечивает удобную навигацию по интерфейсу. Смотреть Приложение 3, рисунок 3.5
  6. «Мои выданные книги». Предоставляет пользователю список книг, которые находятся у него на руках. Для каждой книги указаны дата выдачи и срок возврата, чтобы вовремя напомнить о необходимости вернуть литературу. Смотреть Приложение 3, рисунок 3.6
  7. «Просмотр броней». Доступна библиотекарю. С её помощью отображается полный список забронированных книг с указанием данных читателей (имя, ID) и сроков бронирования. Смотреть Приложение 3, рисунок 3.7
  8. «Выдать книгу по брони». Используется библиотекарем для оформления выдачи книги, которая была ранее забронирована. При нажатии открывается окно, где вводится ID читателя, дата выдачи и срок возврата. Смотреть Приложение 3, рисунок 3.8
  9. «Выдать книгу». Кнопка для подтверждения выдачи произведения по бронированию. После подтверждения данные записываются в базу, а статус книги обновляется на «Выдана». Смотреть Приложение 3, рисунок 3.9
  10. «Отменить бронь». Используется для удаления брони на книгу. Пользователь выбирает нужную запись из списка бронированных книг, после чего система снимает бронь, а статус книги обновляется на «Доступна». Смотреть Приложение 3, рисунок 3.10
  11. «Выдать книгу». Кнопка позволяет библиотекарю оформить процесс выдачи книги читателю, который не сделал предварительное бронирование. Когда библиотекарь выбирает нужную книгу, система открывает окно с подробной информацией о ней (название, автор, жанр). Библиотекарь вводит ID читателя, дату выдачи и предполагаемую дату возврата книги. После этого система обновляет статус книги на «Выдана», и информация о выдаче фиксируется в базе данных. Смотреть Приложение 3, рисунок 3.11
  12. «Выдать». Кнопка принимает введённые данные, записывает их в базу данных и меняет статут на «Выдана». Смотреть Приложение 3, рисунок 3.12
  13. «Выданные книги». Отображает библиотекарю список всех книг, находящихся на руках у читателей. Отдельно указывается имя пользователя, дата выдачи и срок возврата. Смотреть Приложение 3, рисунок 3.13
  14. «Принять книгу на возврат». Применяется для регистрации возврата книги. Библиотекарь вводит данные о книге, и она становится доступной для других пользователей. Смотреть Приложение 3, рисунок 3.14
  15. «Посещения студентов». Позволяет библиотекарю выгрузить отчет в формате Excel, содержащий данные о входах пользователей в систему. Отчет включает дату, время и ID аккаунта, что упрощает анализ посещаемости. Смотреть Приложение 3, рисунок 3.15

### **Результат работы и тестирования**

Кейс 1. Авторизация. Тестируемая функция: Проверка корректности входа в систему с разными наборами данных. Тестовый набор: Проверка поля логина и пароля.

* Ввод корректных данных. Ожидаемый результат: успешный вход в систему и переход в главное меню. Смотреть Приложение 2 , рисунок 2.1
* Ввод пустых полей. Ожидаемый результат: ошибка «Имя пользователя и пароль не могут быть пустыми». Смотреть Приложение 2,рисунок 2.2
* Ввод некорректных данных. Ожидаемый результат: ошибка «Неверное имя пользователя или пароль». Смотреть Приложение 2, рисунок 2.3

Результат: Все тесты прошли успешно, система корректно обрабатывает разные сценарии.

Кейс 2. Поиск книги. Тестируемая функция: Проверка функциональности поиска книг. Тестовый набор: Проверка поиска по названию.

* Ввод корректного названия книги. Ожидаемый результат: система отображает книгу с таким названием. Смотреть Приложение 2, рисунок 2.4

Результат: Поиск работает корректно, система возвращает данные в зависимости от введённых критериев.

Кейс 3. Бронирование книги. Тестируемая функция: Проверка возможности бронирования книги. Тестовый набор: кнопка «Забронировать книгу».

* Выбор доступной книги. Ожидаемый результат: система сообщает «Книга успешно забронирована!». Смотреть Приложение 2, рисунок 2.6
* Выбор забронированной книги. Ожидаемый результат: система выводит сообщение «Эта книга уже забронирована». Смотреть Приложение 2, рисунок 2.7

Результат: Все тесты прошли успешно, функция бронирования работает корректно.

Кейс 4. Отображение забронированных книг. Тестируемая функция: Проверка корректности отображения забронированных книг у пользователя. Тестовый набор: кнопка «Мои брони».

* Есть забронированные книги. Ожидаемый результат: система отображает список забронированных книг с полными данными. Смотреть Приложение 2, рисунок 2.10

Результат: Система корректно отображает список броней или уведомление о его отсутствии.

Кейс 5. Удаление забронированных книг. Тестируемая функция: Проверка возможности удаления брони. Тестовый набор: кнопка «Отменить бронь».

* Выбор книги для удаления брони. Ожидаемый результат: Сообщение: «Выбранные брони успешно удалены». Смотреть Приложение 2, рисунок 2.11
* Нажатие кнопки без выбора книги. Ожидаемый результат: ошибка «Выберите хотя бы одну бронь для отмены». Смотреть Приложение 2, рисунок 2.12

Результат: Все тесты прошли успешно, удаление брони работает корректно.

Кейс 6. Отображение выданных книг. Тестируемая функция: Проверка отображения списка выданных книг. Тестовый набор: кнопка «Мои выданные книги».

* Нет выданных книг. Ожидаемый результат: система выводит сообщение «Нет выданных книг». Смотреть Приложение 2, рисунок 2.13
* Есть выданные книги. Ожидаемый результат: система отображает список выданных книг с датами выдачи и возврата. Смотреть Приложение 2, рисунок 2.14

Результат: Система корректно отображает выданные книги.

Кейс 7. Просмотр всех броней читателей. Тестируемая функция: Проверка того, что библиотекарь видит все брони. Тестовый набор: Проверка функциональности кнопки «Просмотр броней»

* Есть активные бронирования. Ожидаемый результат: открытие окна со списком всех забронированных книг. Смотреть Приложение 2, рисунок 2.15

Результат: Все активные бронирования отображаются корректно.

Кейс 8. Выдача книги по бронированию. Тестируемая функция: Проверка выдачи книги, забронированной заранее. Тестовый набор: кнопка «Выдача книги по бронированию».

* Выбор книги для выдачи по брони. Ожидаемый результат: открытие окна с указанием даты выдачи и ожидаемой датой возврата книги. Смотреть Приложение 2, рисунок 2.16
* Выбор нужной даты, нажатие на кнопку «Выдать книгу». Ожидаемый результат: сообщение «Книга успешна выдана». Смотреть Приложение 2, рисунок 2.17

Результат: Книга выдается корректно, статус обновляется.

Кейс 9. Выдача книг. Тестируемая функция: Проверка процесса выдачи книги пользователю. Тестовый набор: кнопка «Выдать книгу».

* Выбор книги для выдачи. Ожидаемый результат: открытие окна для заполнения данных о выдачи книги. Смотреть Приложение 2, рисунок 2.18
* Ввод всех необходимых данных, нажатие на кнопку «Выдать». Ожидаемый результат: Сообщение: «Книга успешно выдана!». Смотреть Приложение 2, рисунок 2.19
* Нажатие кнопки «Выдать». Ожидаемый результат: сообщение: «Введите данные для выдачи». Результат: ошибка при выдачи книги: list index out of range». Смотреть Приложение 2, рисунок 2.20

Результат: Книга успешно выдается, статус обновляется в базе данных. Ошибка обработана.

Кейс 10. Просмотр всех выданных книг. Тестируемая функция: Проверка того, что библиотекарь может просматривать все выданные книги. Тестовый набор: кнопка «Выданные книги».

* Есть выданные книги. Ожидаемый результат: открытие окна с корректным списком всех выданных книг. Смотреть Приложение 2, рисунок 2.22

Результат: Все выданные книги отображаются корректно.

Кейс 11. Принятие книги на возврат. Тестируемая функция: Проверка корректности обработки возврата книги. Тестовый набор: Проверка функциональности кнопки «Принятие книги на возврат».

* Выбор книги для возврата. Ожидаемый результат: открытие окна для заполнения данных о возврате книги. Смотреть Приложение 2, рисунок 2.24
* Выбор даты возврата не позднее сегодняшней. Ожидаемый результат: сообщение «Книга успешно возвращена!». Смотреть Приложение 2, рисунок 2.25
* Выбор даты позднее сегодняшней. Ожидаемый результат: сообщение «Фактическая дата возврата не может быть позже сегодняшней даты». Смотреть Приложение 2, рисунок 2.26

Результат: Книга успешно возвращена, статус обновляется.

Кейс 12. Просмотр посещений студентов. Тестируемая функция: Проверка, просмотра информацию о посещениях студентов и её экспорт в Excel. Тестовый набор: кнопка «Посещения студентов»

* Наличие посещений. Ожидаемый результат: выбор пути загрузки, отчет о посещении с данными генерируется в формате Excel. Смотреть Приложение 2, рисунок 2.27

Результат: Отчёт успешно генерируется и сохраняется.

Кейс 13. Возвращение в главное меню. Тестируемая функция: Проверка функциональности кнопки «Назад». Тестовый набор: Наличие активной сессии, проверка кнопки «Назад»

* Пользователь нажимает кнопку «Назад». Ожидаемый результат: переход в главное меню. Смотреть Приложение 2, рисунок 2.28

Результат: Переход в главное меню осуществляется корректно.

Кейс 14. Выход из аккаунта. Тестируемая функция: Проверка функциональности кнопки «Выход». Тестовый набор: кнопка «Выйти»

* Пользователь нажимает кнопку «Выйти». Ожидаемый результат: Возвращение к окну авторизации. Смотреть Приложение 2, рисунок 2.29

Результат: Переход в главное меню осуществляется корректно.

Таким образом, в третьей главе курсовой работы был описан процесс разработки программного модуля системы «Читатели студенческой библиотеки». Проведён выбор инструментов, реализован пользовательский интерфейс и ключевые функции модуля, такие как авторизация, поиск, бронирование, выдача и возврат книг. Тестирование показало, что система работает стабильно и соответствует требованиям, обеспечивая удобство для пользователей и библиотекарей.

# **ЗАКЛЮЧЕНИЕ**

В ходе выполнения курсового проекта была проведена работа по анализу, проектированию, разработке и тестированию программного модуля системы «Читатели студенческой библиотеки». Целью работы было создание функционального приложения, которое упрощает управление библиотечными фондами, взаимодействие с читателями и учет посещений.

В первой главе были рассмотрены теоретические аспекты предметной области, включая историческую справку о развитии библиотек и преимуществ информатизации. Анализ готовых решений позволил выделить ключевые требования к разрабатываемой системе и определить актуальность внедрения такого модуля в современных студенческих библиотеках.

Во второй главе были сформулированы требования к функциональности программы, включающей авторизацию, поиск, бронирование, выдачу и возврат книг, а также управление данными о пользователях. Проектирование модуля включало создание ER-диаграммы базы данных, разработку пользовательского интерфейса и определение тестовых сценариев.

В третьей главе был реализован программный модуль с использованием языка Python и библиотек PyQt5, SQLite и Openpyxl. Модуль был протестирован по ключевым функциональным узлам, включая авторизацию, поиск книг, работу с бронированиями, выдачу и возврат книг. Тестирование показало, что система работает корректно, стабильно и полностью соответствует заявленным требованиям.

Разработанный модуль предоставляет удобный интерфейс для читателей и библиотекарей, автоматизирует ключевые процессы, такие как бронирование и выдача книг, а также обеспечивает фиксирование данных о посещениях в формате Excel. Итоги работы демонстрируют, что созданное приложение не только отвечает актуальным потребностям библиотек, но и способствует оптимизации их работы, снижая время на выполнение рутинных операций и повышая качество обслуживания пользователей

# **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. Общие положения.
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".

Монография

1. PyQt. Создание оконных приложений на Python 3. Прохоренок Н.А. 2011
2. Программный модуль «Читатели студенческой библиотеки»: <https://github.com/mnkatrin/Kursovaia_rabota_Readers_students_of_library_Mankovskaia_Ekaterina_313IS>

Иностранная литература

1. Язык программирования Python. Г. Россум, Ф.Л.Дж. Дрейк, Д.С. Откидач. 2001

Интернет-источники

1. АБИС "Ruslan": <https://www.ruslan.ru/>
2. ГПНТБ России: <https://www.gpntb.ru/>
3. Описание Openpyxl: <https://pydocs.ru/openpyxl-python/>
4. Описание PyCharm: <https://www.jetbrains.com/pycharm/features/>
5. Описание PyQt: <https://pypi.org/project/PyQt5/>
6. Описание Python: <https://www.nic.ru/help/osnovy-yazyka-programmirovaniya-python_11662.html>
7. Описание SQLite: <https://www.sqlite.org/about.html>
8. ЭБС Znanium : <https://znanium.ru/>

# **ПРИЛОЖЕНИЕ 1 Диаграмма прецедентности**

Изображение выглядит как текст, диаграмма, круг, линия

Автоматически созданное описание

Рисунок 1.1 - Диаграмма прецедентности

# **ПРИЛОЖЕНИЕ 2 План-тест**

Таблица 2.1 – План-тест

| №  Тест-кейса | Наименование функциональности | Наименования поля | Тестовый набор | Ожидаемый результат | Результат тестирования |
| --- | --- | --- | --- | --- | --- |
| 1 | Авторизация | Имя пользователя | VoronovaVera | Успешный вход в систему, переход в главное меню. | Успешный вход в систему, переход в главное меню. |
| Пароль | Pass123 |
| Рисунок 2.1 | | | | | |
|  | Авторизация | Имя пользователя | Пустое поле | Всплывает окно с уведомлением ошибки «Имя пользователя и пароль не могут быть пустыми» | Всплывает окно с уведомлением ошибки «Имя пользователя и пароль не могут быть пустыми» |
| Поле логин |
| Рисунок 2.2 | | | | | |
|  | Авторизация | Имя пользователя | 3456 | Всплывает окно с уведомлением ошибки «Неверное имя пользователя или пароль» | Всплывает окно с уведомлением ошибки «Неверное имя пользователя или пароль» |
| Поле логин | password |
| Рисунок 2.3 | | | | | |
| 2 | Поиск книги | Поиск книг | Ввод названия книги: Преступление и наказание | Показаны результаты поиска | Показаны результаты поиска |
| Рисунок 2.4 | | | | | |
|  | Поиск книги | Поиск книг | Ввод некорректных данных | Сообщение: «Нет книг, соответствующих запросу» | Сообщение: «Нет книг, соответствующих запросу» |
| Рисунок 2.5 | | | | | |
| 3 | Бронирование книги | Кнопка «Забронировать книгу» | Выбор доступной книги. | Сообщение: «Книга успешно забронирована!» | Сообщение: «Книга успешно забронирована!» |
| Рисунок 2.6 | | | | | |
|  | Бронирование книги | Кнопка «Забронировать книгу» | Выбор уже забронированной книги. | Ошибка: «Эта книга уже забронирована» | Ошибка: «Эта книга уже забронирована» |
| Рисунок 2.7 | | | | | |
|  | Бронирование книги | Кнопка «Забронировать книгу» | Нажатие на кнопку. | Ошибка: «Выберите книгу для бронирования» | Ошибка: «Выберите книгу для бронирования». |
| Рисунок 2.8 | | | | | |
| 4 | Отображение забронированных книг читателя | Кнопка «Мои брони» | Отсутствие забронированных книг. | Система отображает список забронированных книг с полными данными. | Система отображает список забронированных книг с полными данными. |
| Рисунок 2.9 | | | | | |
|  | Отображение забронированных книг читателя | Кнопка «Мои брони» | Нажатие на кнопку | Система выводит сообщение «У вас нет забронированных книг». | Система выводит сообщение «У вас нет забронированных книг». |
| Рисунок 2.10 | | | | | |
| 5 | Удаление забронированных книг | Кнопка «Отменить бронь» | Выбор нужной брони для отмены, нажатие на кнопку | Сообщение: «Выбранные брони успешно удалены» | Сообщение: «Выбранные брони успешно удалены» |
| Рисунок 2.11 | | | | | |
|  | Удаление забронированных книг | Кнопка «Отменить бронь» | Нажать на кнопку | Ошибка: «Выберите хотя бы одну бронь для отмены» | Ошибка: «Выберите хотя бы одну бронь для отмены» |
| Рисунок 2.12 | | | | | |
| 6 | Отображение выданных книг читателя | Кнопка «Мои выданные книги» | Отсутствие выданных книг | Система выводит сообщение «Нет выданных книг». | Система выводит сообщение «Нет выданных книг». |
| Рисунок 2.13 | | | | | |
|  | Отображение выданных книг читателя | Кнопка «Мои выданные книги» | Нажатие на кнопку | Система отображает список выданных книг с датами выдачи и воз-врата | Система отображает список выданных книг с датами выдачи и воз-врата |
| Рисунок 2.14 | | | | | |
| 7 | Просмотр всех броней читателей | Кнопка «Просмотр броней» | Нажатие на кнопку | Открытие окна со списком всех забронированных книг | Открытие окна со списком всех забронированных книг |
| Рисунок 2.15 | | | | | |
| 8 | Выдача книги по бронированию | Кнопка «Выдать книгу по брони» | Выбор даты возврата 04.12.2024. Нажатие на кнопку. | Открытие окна с указанием даты выдачи и ожидаемой датой возврата книги. | Открытие окна с указанием даты выдачи и ожидаемой датой возврата книги. |
| Рисунок 2.16 | | | | | |
|  | Выдача книги по бронированию | Кнопка «Выдать книгу» | Нажатие на кнопку | Сообщение «Книга успешна выдана». | Сообщение «Книга успешна выдана». |
| Рисунок 2.17 | | | | | |
| 9 | Выдача книг | Кнопка «Выдать книгу» | Название книги: Шинель Имя пользователя: VoronovaVera Дата выдачи:04.12.2024 Дата возврата: 18.12.2024 Нажатие на кнопку «Выдать» | Открытие окна для заполнения данных для выдачи книги. | Открытие окна для заполнения данных для выдачи книги. |
| Рисунок 2.18 | | | | | |
|  | Выдача книг | Кнопка «Выдать» | Заполнение данных, нажатие на кнопку. | Сообщение: «Книга успешно выдана!» | Сообщение: «Книга успешно выдана!» |
| Рисунок 2.19 | | | | | |
|  | Выдача книг | Кнопка «Выдать» | Нажатие на кнопку. | Сообщение: «Введите данные для выдачи» | Ошибка при выдаче книги: list index out of range |
| Рисунок 2.20 | | | | | |
|  | Выдача книг –  Обработка и исправление ошибки | Кнопка «Выдать» | Нажатие на кнопку. | Ошибка: «Введите данные для выдачи» | Ошибка: «Введите данные для выдачи» |
| Рисунок 2.21 | | | | | |
| 10 | Просмотр всех выданных книг | Кнопка «Выданные книги» | Нажатие на кнопку | Открытие окна со списком всех выданных книг | Открытие окна со списком всех выданных книг |
| Рисунок 2.22 | | | | | |
|  | Просмотр всех выданных книг | Кнопка «Выданные книги» | Нажатие на кнопку | Сообщение: «Нет выданных книг» | Сообщение: «Нет выданных книг» |
| Рисунок 2.23 | | | | | |
| 11 | Принятие книги на возврат | Кнопка «Принять книгу на возврат» | Выбор нужной записи, нажатие на кнопку. Указание даты возврата -сегодня. 04.12.2024 | Открытие окна для заполнения данных о возврате книги | Открытие окна для заполнения данных о возврате книги |
| Рисунок 2.24 | | | | | |
|  | Принятие книги на возврат | Кнопка «Принять» | Выбор даты не позднее сегодняшней. | Сообщение «Книга успешно возвращена!» | Сообщение «Книга успешно возвращена!» |
| Рисунок 2.25 | | | | | |
|  | Принятие книги на возврат | Кнопка «Принять» | Выбор нужной записи, нажатие на кнопку. Указание даты возврата – дата позже сегодняшней. 12.12.2024 | Ошибка: «Фактическая дата возврата не может быть позже сегодняшней даты» | Ошибка: «Фактическая дата возврата не может быть позже сегодняшней даты». |
| Рисунок 2.26 | | | | | |
| 12 | Просмотр посещений студентов | Кнопка «Посещения студентов» | Нажатие на кнопку | Выбор пути загрузки. Выгрузка данных в Excel. | Выбор пути загрузки. Выгрузка данных в Excel. |
| Рисунок 2.27 | | | | | |
| 13 | Возвращение в главное меню | Кнопка «Назад» | Нажатие на кнопку | Возвращение в главное окно | Возвращение в главное окно |
| Рисунок 2.28 | | | | | |
| 13 | Выход из аккаунта | Кнопка «Выйти» | Нажатие на кнопку | Возвращение к окну авторизации | Возвращение к окну авторизации |
| Рисунок 2.29 | | | | | |

# **ПРИЛОЖЕНИЕ 3 Функционал кнопок**

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 3.1 – Функция кнопки «Войти»

Изображение выглядит как текст, Шрифт, снимок экрана, белый

Автоматически созданное описание

Рисунок 3.2 – Функция кнопки «Выйти»

Изображение выглядит как текст, снимок экрана, документ, Шрифт

Автоматически созданное описание

Рисунок 3.3 – Функция кнопки «Забронировать книгу»

Изображение выглядит как текст, снимок экрана, документ, Шрифт

Автоматически созданное описание

Рисунок 3.3 – Функция кнопки «Забронировать книгу»

Изображение выглядит как текст, снимок экрана, документ, Шрифт

Автоматически созданное описание

Рисунок 3.4 – Функция кнопки «Мои брони»

Изображение выглядит как текст, снимок экрана, документ, Шрифт

Автоматически созданное описание

Рисунок 3.4 – Функция кнопки «Мои брони»

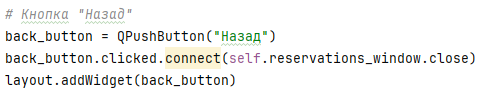


Рисунок 3.5 – Функция кнопки «Назад»

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 3.6 – Функция кнопки «Мои выданные книги»

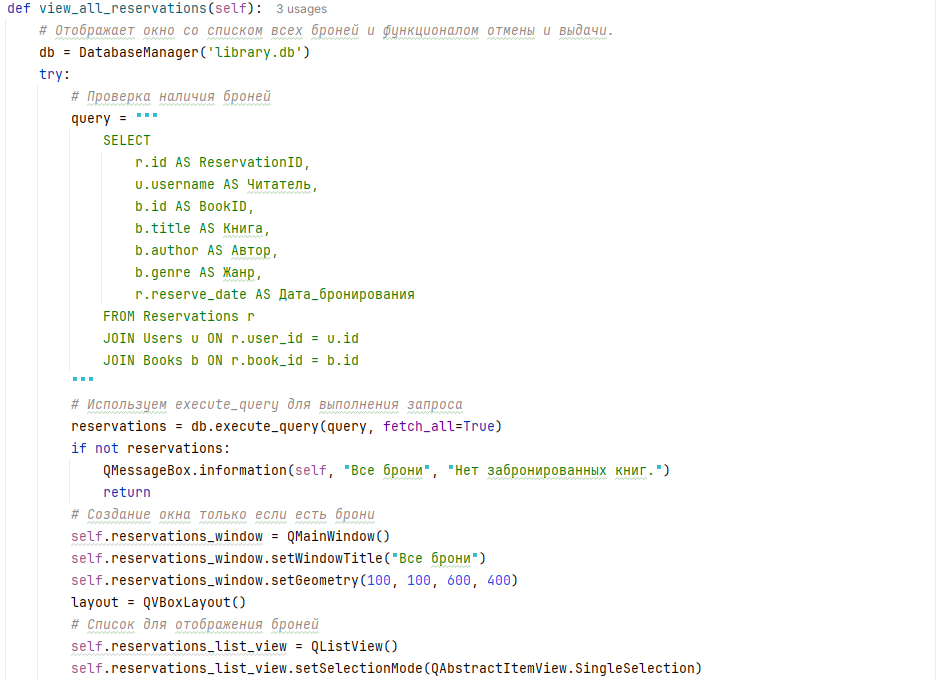


Рисунок 3.7 – Функция кнопки «Просмотр броней»

Изображение выглядит как текст, снимок экрана, Шрифт, документ

Автоматически созданное описание

Рисунок 3.7 – Функция кнопки «Просмотр броней»

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 3.8 – Функция кнопки «Выдать книгу по брони»

Изображение выглядит как текст, снимок экрана, документ, Шрифт

Автоматически созданное описание

Рисунок 3.8 – Функция кнопки «Выдать книгу по брони»

Изображение выглядит как текст, снимок экрана, документ, Шрифт

Автоматически созданное описание

Рисунок 3.9 – Функция кнопки «Выдать книгу»

Изображение выглядит как текст, снимок экрана, документ, Шрифт

Автоматически созданное описание

Рисунок 3.9 – Функция кнопки «Выдать книгу»

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 3.10 – Функция кнопки «Отменить бронь»

Изображение выглядит как текст, снимок экрана, Шрифт, документ

Автоматически созданное описание

Рисунок 3.10 – Функция кнопки «Отменить бронь»

Изображение выглядит как текст, снимок экрана, Шрифт, документ

Автоматически созданное описание

Рисунок 3.10 – Функция кнопки «Отменить бронь»

Изображение выглядит как текст, снимок экрана, Шрифт, документ

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 3.11 – Функция кнопки «Выдать книгу»

Изображение выглядит как текст, снимок экрана, Шрифт, программное обеспечение

Автоматически созданное описание

Рисунок 3.11 – Функция кнопки «Выдать книгу»

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 3.15 – Функция кнопки «Выдать»

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 3.15 – Функция кнопки «Выдать»

Изображение выглядит как текст, снимок экрана, документ, Шрифт

Автоматически созданное описание

Рисунок 3.13 – Функция кнопки «Выданные книги»

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 3.13 – Функция кнопки «Выданные книги»

Изображение выглядит как текст, снимок экрана, Шрифт, документ

Автоматически созданное описание

Рисунок 3.14 – Функция кнопки «Принять книгу на возврат»

Изображение выглядит как текст, снимок экрана

Автоматически созданное описание

Изображение выглядит как текст, Шрифт, снимок экрана

Автоматически созданное описаниеРисунок 3.14 – Функция кнопки «Принять книгу на возврат»

Изображение выглядит как текст, снимок экрана, документ, Шрифт

Автоматически созданное описание

Рисунок 3.15 – Функция кнопки «Посещения студентов»

# **ПРИЛОЖЕНИЕ 4 Листинг кода**

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

Полный листинг кода представлен на сайте «GitHub»[[7]](#_СПИСОК_ИСПОЛЬЗОВАННЫХ_ИСТОЧНИКОВ)