Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Вятский государственный университет»

Колледж ВятГУ

**ТЕХНИЧЕСКИЙ ПРОЕКТ**  
Информационная система «Личный кабинет студента»

Киров,

2024

**Содержание**

[1. Концепция предлагаемого решения 3](#_Toc182561317)

[1.1. Пользовательский интерфейс 3](#_Toc182561318)

[1.2. Организация хранения 5](#_Toc182561319)

[2. Бизнес-процессы системы 7](#_Toc182561320)

[3. Структура программы и алгоритмы 11](#_Toc182561321)

[4. Выбранная архитектура системы 12](#_Toc182561322)

# **Концепция предлагаемого решения**

## **Пользовательский интерфейс**

Пользовательский интерфейс главной страницы приложения представлен на рисунке 1.

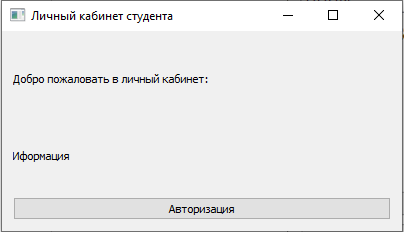


Рисунок 1 — Приветственное окно

На приветственном окне присутствуют: название ИС (сверху экрана), приветственная надпись (по центру экрана), а также кнопка «Авторизации», которая перенаправит пользователя к окну входа.

Пользовательский интерфейс главной страницы приложения представлен на рисунке 2.

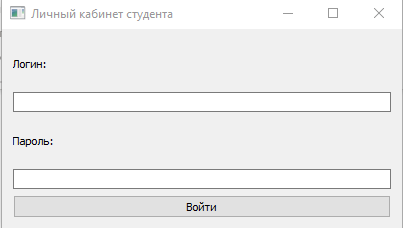


Рисунок 2 — Окно входа в личный кабинет студента

На окне входа присутствуют: название окна (сверху экрана), надписи об оповещении и информации, а также кнопки «Главная страница», «Задания», «Опросы», «Оповещения», «Выйти», «Данные», «Создать запись» и «Профиль».

Пользовательский интерфейс главной страницы приложения представлен на рисунке 3.

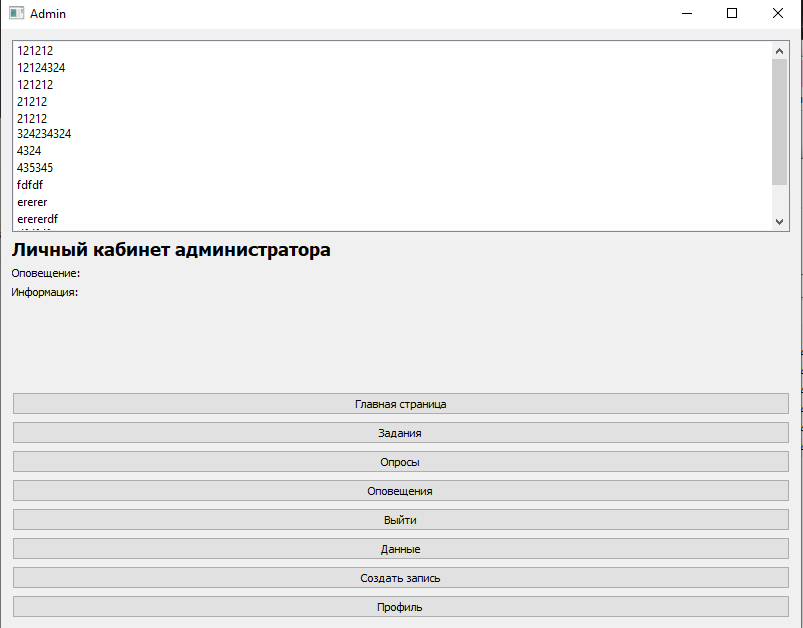


Рисунок 3 – Главная страница приложения

На главной странице приложения присутствуют: название окна (сверху экрана), надписи об оповещении и информации, а также кнопки «Главная страница», «Задания», «Опросы», «Оповещения», «Выйти», «Данные», «Создать запись» и «Профиль».

Форма управления опросами, представлена на рисунке 4.



Рисунок 4 – Управление опросами

На форме управления опросами присутствуют: название окна (сверху экрана), текстовое поле «Найти» в области, область в котором присутствует раскрывающийся список «Выбрать группу», также список студентов, выбранной группы и область, в котором присутствует список «Выбрать опрос», при выбранном студенте можно посмотреть ответы опроса, а также кнопки «Создать опрос», «Назад».

Форма управления оповещения, представлен на рисунке 5.

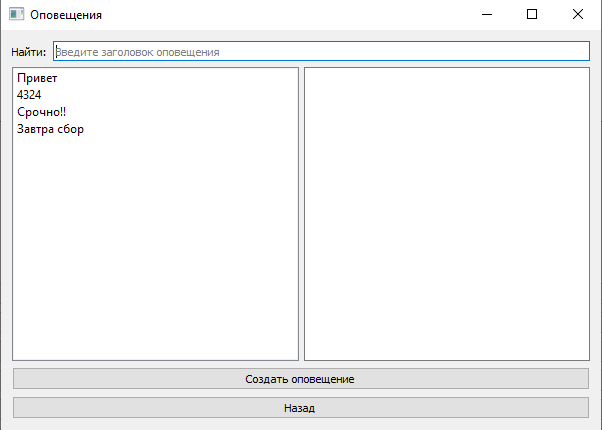


Рисунок 5 – Управление оповещениями

На окне управление оповещениями присутствуют: название окна (сверху экрана), текстовое поле «Найти» в области, область в котором представлены все оповещения и область, где отображено выбранное оповещение, а также кнопки «Назад» и «Создать оповещение».

Форма создания оповещения, представлена на рисунке 6.

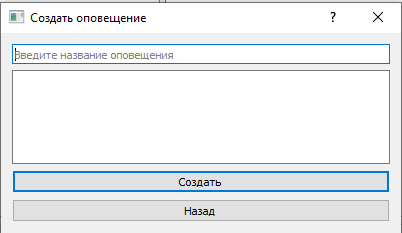


Рисунок 6 – Создание оповещения

На окне создания оповещения присутствуют: название окна (сверху экрана), текстовое поле «Текст оповещения», а также кнопки «Назад» и «Создать оповещение».

Форма управления данными, представлена на рисунке 7.

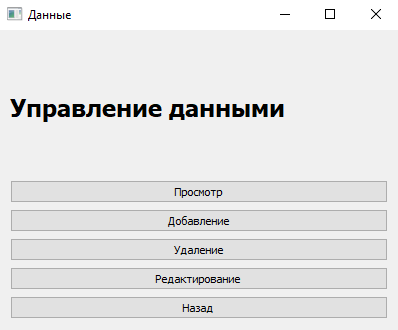


Рисунок 7 – Управление данными

На окне управление данными присутствуют: название окна (сверху экрана), а также кнопки «Просмотр», «Добавление», «Удаление», «Редактирование» и «Назад», которая вернет пользователя на главную страницу.

Форма создания оповещения, представлена на рисунке 8.

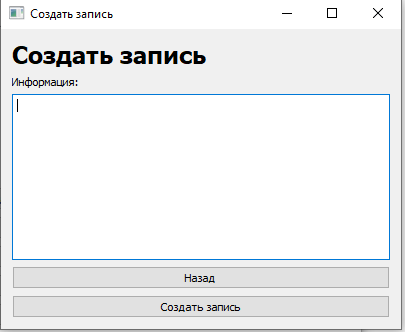


Рисунок 8 – Создание оповещения

На окне создания оповещения присутствуют: название окна (сверху экрана), текстовое поле «Текст оповещения», а также кнопки «Назад» и «Создать запись».

Форма управлением профиля пользователя, представлена на рисунке 9.

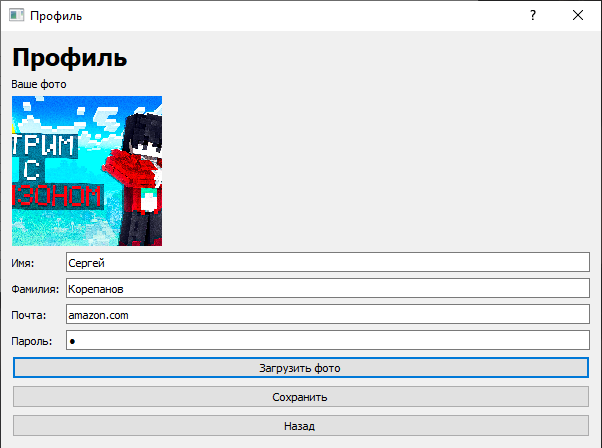


Рисунок 9 – Управление профилем

На окне управление профилем присутствуют: название окна (сверху экрана), изображение, надпись под изображением, поля для ввода (Фамилия, Имя, Почта, Пароль), а также кнопки «Назад», которая вернет пользователя на главную страницу, и кнопка «Сохранить», которая изменит информацию о пользователе.

## **Организация хранения**

Для организации хранения данных используется СУБД MySQL.

На рисунке 10 представлена физическая ER-диаграмма базы данных.

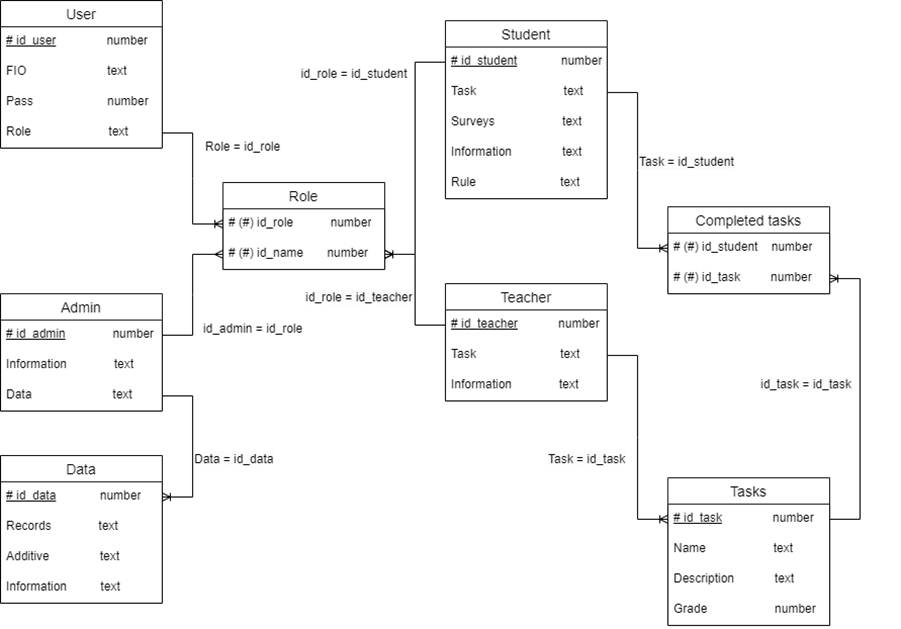


Рисунок 10 – Физическая ER-диаграмма

Исходя из диаграммы можно выделить 6 сущностей:

* Пользователь.
* Роли.
* Студент.
* Преподаватель.
* Администратор.
* Данные.
* Задания.
* Выполненные заадания.

В таблице 5 выделен набор сущностей с их атрибутами.

Таблица 5 – Набор сущностей и их атрибутов

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Сущность** | **Свойство** | **Тип** | **Уникальность** | **Обязательность заполнения** | **Ограничения** |
| **Пользователь** | id\_user | Number | Да | Да | PK |
| fio | Text |  | Да |  |
| pass | Number |  | Да |  |
| role | Text |  | Да | FK |
| **Роли** | id\_role | Number |  | Да | FK |
| id\_name | Number |  | Да | FK |
| **Студент** | id\_student | Number | Да | Да | PK |
| task | Text |  | Да | FK |
| surveys | Text |  | Да |  |
| information | Text |  | Да |  |
| rule | Text |  | Да |  |
| **Преподаватель** | id\_teacher | Number | Да | Да | PK |
| task | Text |  | Да |  |
| information | Text |  | Да |  |
| **Администратор** | id\_admin | Number | Да | Да | PK |
| information | Text |  | Да |  |
| data | Text |  | Да | FK |
| **Данные** | id\_data | Number | Да | Да | PK |
| records | Text |  | Да |  |
| additive | Text |  | Да |  |
| information | Text |  | Да |  |
| **Задание** | id\_task | Number | Да | Да | PK |
| name | Text |  | Да |  |
| description | Text |  | Да |  |
| grade | Number |  | Да |  |
| **Выполненные задания** | id\_student | Number |  | Да | FK |
| id\_task | Number |  | Да | FK |

Связи между сущностями:

«Роль» – «Пользователь»: один ко многим, каждый пользователь может иметь только одну роль, но одной роли может быть присвоено несколько пользователей.

«Роль» – «Студент», «Преподаватель», «Администратор»: один ко многим, одна роль может быть связана с несколькими студентами, преподавателями или администраторами, но каждая сущность относится только к одной роли.

**«**Задание» – «Выполненное задание»: один ко многим, одно задание может быть выполнено несколькими студентами, но каждое выполненное задание привязано к конкретному заданию.

«Студент» – «Выполненное задание»: один ко многим, один студент может выполнить несколько заданий, но каждое выполненное задание привязано к одному студенту.

«Администратор» – «Данные»: один ко многим, один администратор может управлять несколькими наборами данных, но каждая запись данных привязана к одному администратору.

# **Бизнес-процессы системы**

На рисунке 11 представлена контекстная диаграмма IDF0 для ИС.

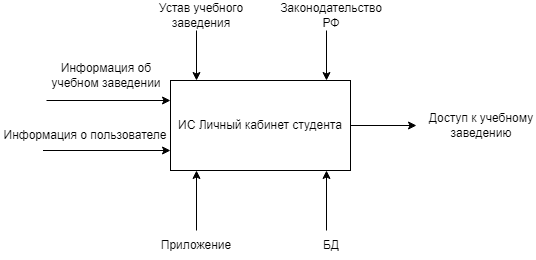


Рисунок 11 – Контекстная диаграмма IDF0

На диаграмме отображена работа информационной системы для реализации доступа к учебному заведению. Для начала есть запросы пользователей на проверку определенных данных и информации об учебном заведении, для входа в систему.

В работе ИС участвует приложение, а все данные хранятся в БД. Результатом работы является доступ к учебному заведению.

На рисунке 12 представлена декомпозиция контекстной диаграммы.

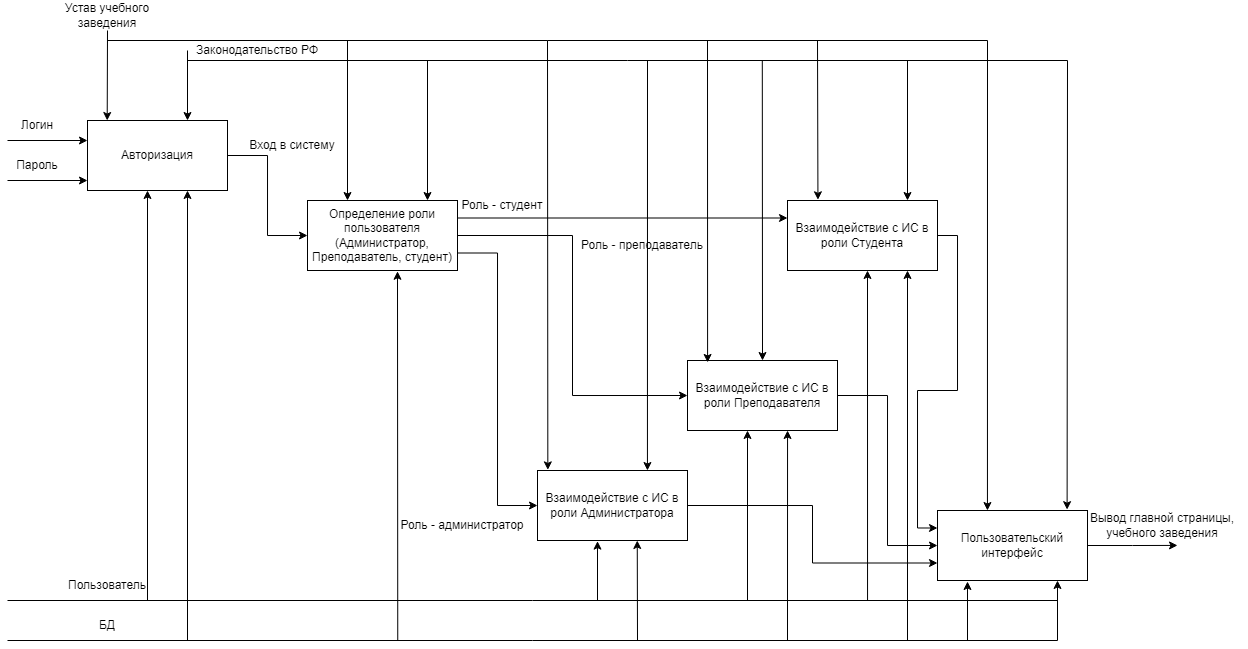


Рисунок 12 – Декомпозиция контекстной диаграммы

На данной диаграмме более детально показана работа ИС. Диаграмма отображает процесс входа пользователя в систему. К пользователю относятся следующие роли: «Администратор», «Преподаватель», «Студент». После авторизации происходит определение роли пользователя, от этого зависит дальнейшее взаимодействие с системой.

На рисунке 5 представлена декомпозиция функциональных возможностей студента. Данная диаграмма отображает взаимодействие с информационной системой в роли студента. У студента есть возможности получать информацию, задания, опросы, а также профиль, который он может редактировать.

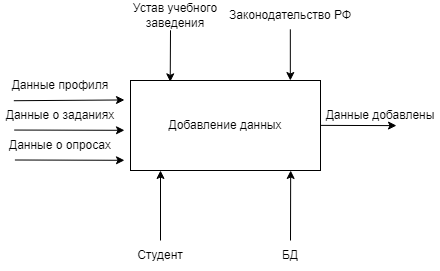


Рисунок 13 – Декомпозиция функциональных возможностей студента

На рисунке 14 представлена Декомпозиция функциональных возможностей преподавателя. Диаграмма отображает взаимодействие с информационной системой в роли преподавателя. У преподавателя есть возможности студента, а также он может создавать задания и оценивать работы студентов.

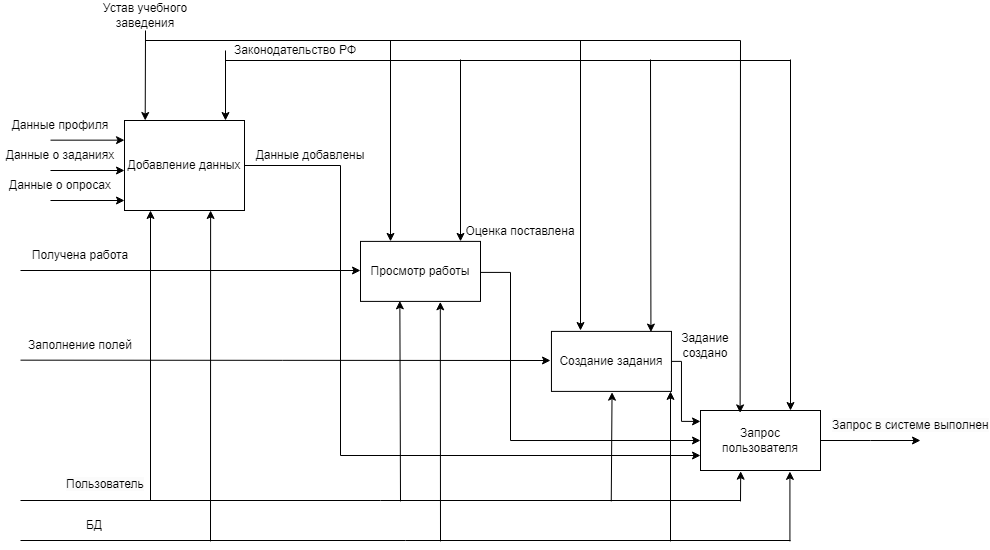


Рисунок 14 – Декомпозиция функциональных возможностей преподавателя

На рисунке 15 представлена функциональных возможностей администратора. Диаграмма отображает взаимодействие с информационной системой в роли администратора. У администратора есть все возможности других пользователей, а также он может создавать опросы, оповещения, информацию и работать с данными.

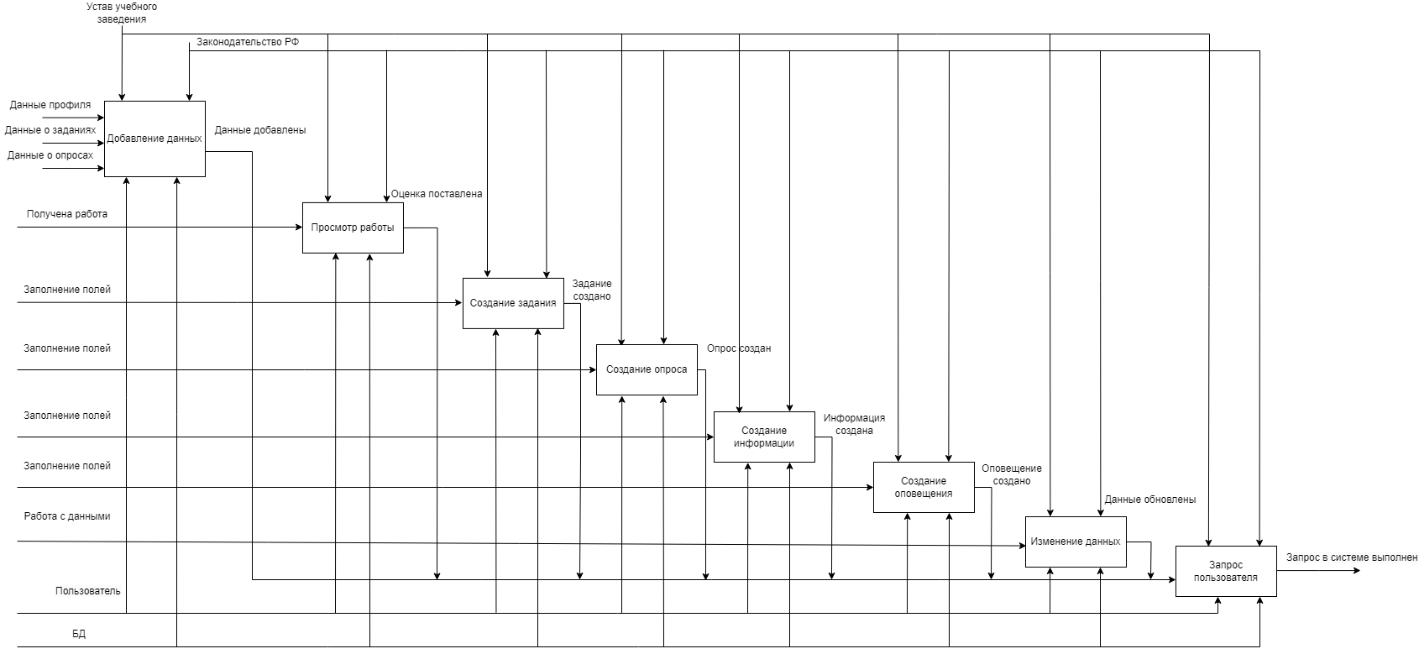


Рисунок 15 – Декомпозиция функциональных возможностей администратора

# **Структура программы и алгоритмы**

* 1. **Структура программы**

Для разработки информационной системы «Личный кабинет студента» используется Python с библиотекой PyQt5 для создания графического пользовательского интерфейса (GUI) и mysql-connector для взаимодействия с базой данных MySQL. PyQt5 предоставляет средства для построения гибких и мощных интерфейсов, а mysql-connector используется для безопасного подключения и работы с базой данных MySQL.

Преимущества использования PyQt5 и mysql-connector включают:

Кроссплатформенность: PyQt5 и mysql-connector позволяют создавать приложения, которые могут работать на различных операционных системах, таких как Windows, macOS и Linux, что дает возможность разрабатывать и развертывать приложение на разных платформах.

Производительность: Python и его библиотеки обладают высокой производительностью при работе с графическим интерфейсом и базами данных, а также обеспечивают низкое время отклика за счет эффективной работы с асинхронными запросами.

Гибкость: PyQt5 позволяет использовать различные подходы к разработке пользовательского интерфейса, что дает гибкость в проектировании приложения.

Безопасность: Применение библиотек для работы с базами данных, таких как mysql-connector, позволяет обеспечивать безопасное соединение и защиту от SQL-инъекций.

Интеграция с современными инструментами разработки: Легкая интеграция с популярными инструментами, такими как Visual Studio Code, Docker и другие, упрощает процесс разработки и развертывания.

Эти преимущества делают использование Python и PyQt5 оптимальным решением для разработки информационной системы «Личный кабинет студента», так как позволяет эффективно управлять данными студентов, преподавателей и заданиями.

* 1. **Алгоритмы решения задач**

Форма для отправки задания от студента проверяет наличие заполненных обязательных полей, таких как «название задания» и «файл с выполненным заданием». В случае незаполненных полей выводится сообщение об ошибке, и отправка задания становится невозможной. Также проверяется срок сдачи задания, который не должен быть в прошлом. Если дата сдачи некорректна, система выводит сообщение об ошибке.

При входе в личный кабинет система проверяет, существует ли пользователь с введенным логином и паролем в базе данных MySQL. Для этого выполняется SQL-запрос через mysql-connector, и, если данные не совпадают, выводится сообщение, что такой аккаунт не найден. В случае успешной авторизации происходит переход в личный кабинет. В зависимости от роли пользователя (студент, преподаватель, администратор) доступны различные функции, такие как просмотр заданий, их выполнение, выставление оценок и управление данными.

Для работы с базой данных используется SQL-запрос SELECT, после чего результаты помещаются в переменную и выводятся в интерфейсе с помощью элементов управления, таких как QTableWidget. Редактирование и удаление записей осуществляется через интерфейс, при этом все изменения происходят по уникальному идентификатору записи.

Каждое выполненное задание хранится в таблице «Выполненные задания». Преподаватели могут просматривать задания студентов и оценивать их. Алгоритм включает фильтрацию заданий по статусу и дате сдачи, с уведомлением преподавателя о несоответствиях. Студенты могут отправлять несколько заданий, и каждое задание может быть выполнено разными студентами.

Для управления заданиями и студентами предусмотрены формы для добавления, редактирования и удаления записей. Каждое изменение данных подтверждается проверкой на корректность ввода. Например, при добавлении задания проверяется правильность формата даты и оценок. Распределение заданий между преподавателями осуществляется через таблицы «Задания» и «Преподаватели», где установлена связь многие ко многим. Однако одно задание не может быть назначено одному преподавателю несколько раз.

# **Выбранная архитектура системы**

Для созданного решения была выбрана двухзвенная архитектура «Клиент-сервер» с базой данных. Архитектура MySQL включает в себя две основные компоненты: клиентскую и серверную. Серверная часть представляет собой основу, которая управляет базой данных и обработкой запросов. Клиентская часть выполняет запросы и получает ответы от серверной части.

MySQL работает в режиме клиент-сервер, где клиент отправляет запросы на сервер, который в свою очередь получает эти запросы, обрабатывает их и возвращает результаты. Клиент в данном случае реализован на языке Python с использованием библиотеки PyQt5 для графического интерфейса и mysql-connector для взаимодействия с базой данных.

Архитектура приложения представлена на рисунке 16.

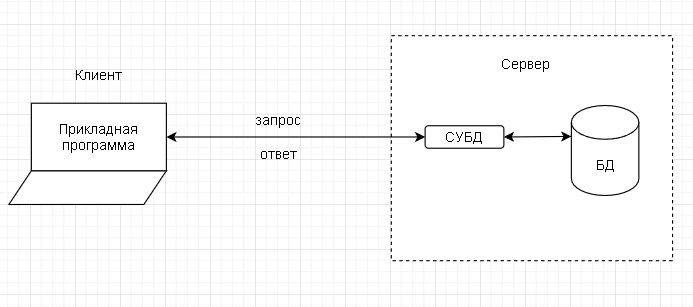


Рисунок 16 – Архитектура приложения