МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

ВЫСШЕГО ОБРАЗОВАНИЯ

**«МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ» (МОСКОВСКИЙ ПОЛИТЕХ)**

Факультет: факультет информационных технологий

Направление: прикладная информатика

Профиль: информационные технологии управления бизнесом

**Курсовая работа**

“Timurzon” – программа для пункта выдачи заказов

Исполнитель: Базгудинов Тимур Айдарович

(фамилия, имя, отчество)  (подпись)

Обучающийся группы 231-365

Проверил(а): Армаш Марина Николаевна

(ФИО, уч. степень, уч. звание) (подпись)

Дата сдачи работы:

« 29 » Мая 2024г

Дата проверки работы:

« » 2024г

Оценка:

Москва, 2024

**Содержание**

[**Введение** 2](#_Toc178865325)

[**Глава 1** 3](#_Toc178865326)

[**1. Предметная область** 3](#_Toc178865327)

[**2.Описание Системы** 4](#_Toc178865328)

[***3*. Требования к системе** 5](#_Toc178865329)

[**4. Требования к системе** 5](#_Toc178865330)

[**Глава 2** 7](#_Toc178865331)

[**1. Руководство пользователя** 7](#_Toc178865332)

[**2. Руководство модератора** 9](#_Toc178865333)

[**3. Руководство программиста** 14](#_Toc178865334)

[**4. Тестирование системы** 22](#_Toc178865335)

[**5.Анализ результатов** 24](#_Toc178865336)

[**Заключение** 25](#_Toc178865337)

# **Введение**

В современном мире электронная коммерция и онлайн-покупки стали неотъемлемой частью повседневной жизни. С ростом популярности интернет-магазинов и платформ для онлайн-шопинга, таких как Amazon, eBay и AliExpress, возникла необходимость в эффективных и удобных системах для выдачи заказов. Пункты выдачи заказов (ПВЗ) играют ключевую роль в обеспечении удобства для клиентов, позволяя им получать свои покупки в удобное время и в удобном месте.

В рамках работы были применены теоретические знания и практические навыки, полученные при изучении дисциплины "Базы данных", для разработки и реализации приложения. Основным предметом исследования стали принципы создания, проектирования, реализации и тестирования систем управления базами данных.

### **Цель**

Целью данной курсовой работы является применение теоретических знаний и практических навыков, полученных при изучении дисциплины "Базы данных", для создания приложения для пункта выдачи заказов.

### **Актуальность**

Актуальность данной работы обусловлена необходимостью применения теоретических знаний и практических навыков, полученных при изучении дисциплины "Базы данных", для разработки приложения, специализированного на выдаче заказов и посылок. Создание такого приложения позволит оптимизировать процесс выдачи заказов, улучшить пользовательский опыт и повысить эффективность работы ПВЗ.

### **Предмет исследования**

Предметом исследования в данной курсовой работе являются принципы проектирования, реализации и тестирования приложений с использованием систем управления базами данных, специализированных для решения задач по выдаче и отслеживанию посылок.

### **Задачи**

Задачи, решаемые в рамках курсовой работы:

* Исследование концепций и технологий баз данных, включая языки проектирования и манипулирования данными.
* Изучение и применение методов работы с СУБД PostgreSQL.
* Разработка структуры базы данных, содержащей информацию о посылках и пользователях.
* Реализация интерфейса приложения для осуществления основных функций управления данными.

# **Глава 1**

## **1. Предметная область**

Предметная область данной курсовой работы охватывает управление информацией, связанной с хранением и выдачей заказов и пользователями.

Основными сущностями предметной области являются:

* **Посылки**: Посылки, находящиеся в системе, с уникальными кодами, владельцами, содержимым, статусом и весом.
* **Пользователи**: Пользователи системы с их идентификаторами, логинами, паролями, электронной почтой, номером телефона, возрастом и именами.

Взаимосвязи между этими сущностями можно представить следующим образом:

* **Посылка может принадлежать только 1 пользователю.**
* **Пользователь** может иметь несколько посылок.

Основные операции, выполняемые в предметной области:

* **Добавление новых посылок и пользователей**.
* **Удаление существующих пользователей и посылок**.
* **Обновление информации о посылках и пользователях.**

### Требования к системе управления данными:

* **Система должна обеспечивать хранение, организацию и управление информацией о посылках и пользователях.**
* **Система должна поддерживать различные операции, такие как добавление, удаление, обновление и поиск данных**.
* **Система должна обеспечивать целостность и согласованность данных**.
* **Система должна быть удобной и простой в использовании**.

### Основные компоненты системы:

* **База данных**: PostgreSQL используется для хранения информации о турах, пользователях, рекомендациях и ключевых словах. Структура базы данных спроектирована таким образом, чтобы обеспечивать эффективное хранение и быстрый доступ к данным.
* **Интерфейс пользователя**: Приложение с графическим интерфейсом на основе PyQt6, которое позволяет пользователям взаимодействовать с системой, получать рекомендации и управлять своими данными.

## **2.Описание Системы**

**2.1. Наименование системы**

“Timurzon” – программа для пунктов выдачи заказов.

**2.2. Основания для проведения работ**

Курсовой проект для основной профессиональной образовательной программы по направлению 09.03.03 «Прикладная информатика», профиль «Информационные технологии управления бизнесом», 2024 год, по дисциплине «Базы данных».

**2.3. Плановые сроки начала и окончания работы**

Плановый срок для начала работ: 25.03.2024

Плановый срок окончания работ: 03.05.2024

**2.4. Источники и порядок финансирования**

Проект выполняется в рамках образовательного процесса на безвозмездной основе.

## ***3*. Требования к системе**

***3*.1. Требования с технической стороны**

**3.1.1. Существующее техническое обеспечение**

Для работы системы требуется устройство с конфигурацией процессора Intel Core i3 2120 и выше, оперативной памятью 8GB DDR4 и объемом дискового хранилища минимум 5 ГБ. Операционная система — Windows 10/11.

**3.1.2. Существующее программное обеспечение**

На устройствах пользователей должно быть установлено следующее программное обеспечение:

* ОС Windows 11, ОС Windows 10 — проприетарные операционные системы;
* PostgreSQL – система для работы с базами данных
* PyCharm – система для программирования на языке python

## **4. Требования к системе**

**4.1. Требования к численности**

Для начала эксплуатации системы необходим следующий персонал:

* Программист — 1 шт. единица.
* Системный администратор — 1 шт. единица.

**4.2. Требования к функциям (задачам), выполняемым системой**

**4.2.1. Пользовательские роли и права**

В системе предусмотрены как минимум 2 пользовательские роли с разными правами:

* **Авторизованный пользователь**: доступен весь пользовательский функционал, а именно: просмотр всех сформированных заказов и редактирование личных данных
* **Администратор**: доступна возможность редактировать, просматривать и удалять записи в системе, такие как посылки и пользователи.

**4.2.2. Подсистема авторизации**

Подсистема авторизации дает возможность пользователям авторизовываться в системе для просмотра всех сформированных заказов и редактирования личных данных. Авторизация пользователя должна осуществляться с помощью единой учетной записи и происходить только с использованием действительных логина и пароля.

**4.2.3. Подсистема «Личный кабинет»**

Подсистема «Личный кабинет» должна быть доступна пользователю после авторизации и предоставлять следующие функции:

* Просмотр своих заказов.
* Обновление личной информации.

Также данная подсистема должна иметь другой интерфейс при входе в систему через аккаунт модератора и выполнять функцию просмотра, удаления и редактирования записей в системе, таких как пользователи и посылки.

# **Глава 2**

## **1. Руководство пользователя**

**1.1 Описание приложения**

Приложение предназначено отслеживания заказанных посылок и редактирования личных данных в профиле.

Целевой аудиторией приложения являются люди всех возрастов, заинтересованные в интернет-покупках и получении своих заказов в ПВЗ.

**1.2 Подготовка к использованию приложения**

Перед использованием приложения необходимо убедиться в наличии следующих компонентов на вашем компьютере:

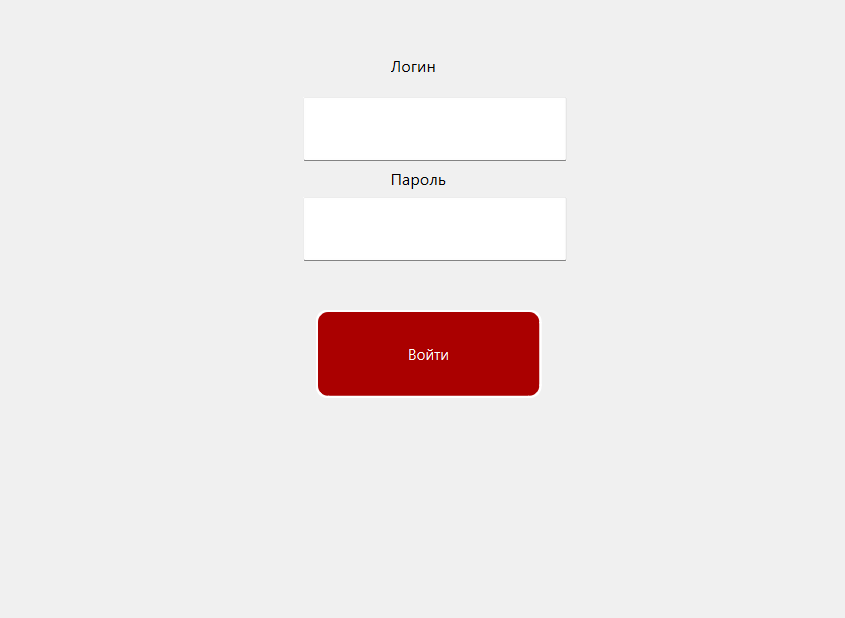
* Наличие PyCharm для запуска системы
* Наличие PostgreSQL для получения доступа к базе данных

**1.3 Функционал приложения**

Для запуска приложения его на вашем устройстве необходимо загрузить файлы с кодом и запустить их в PyCharm и базу данных sql.

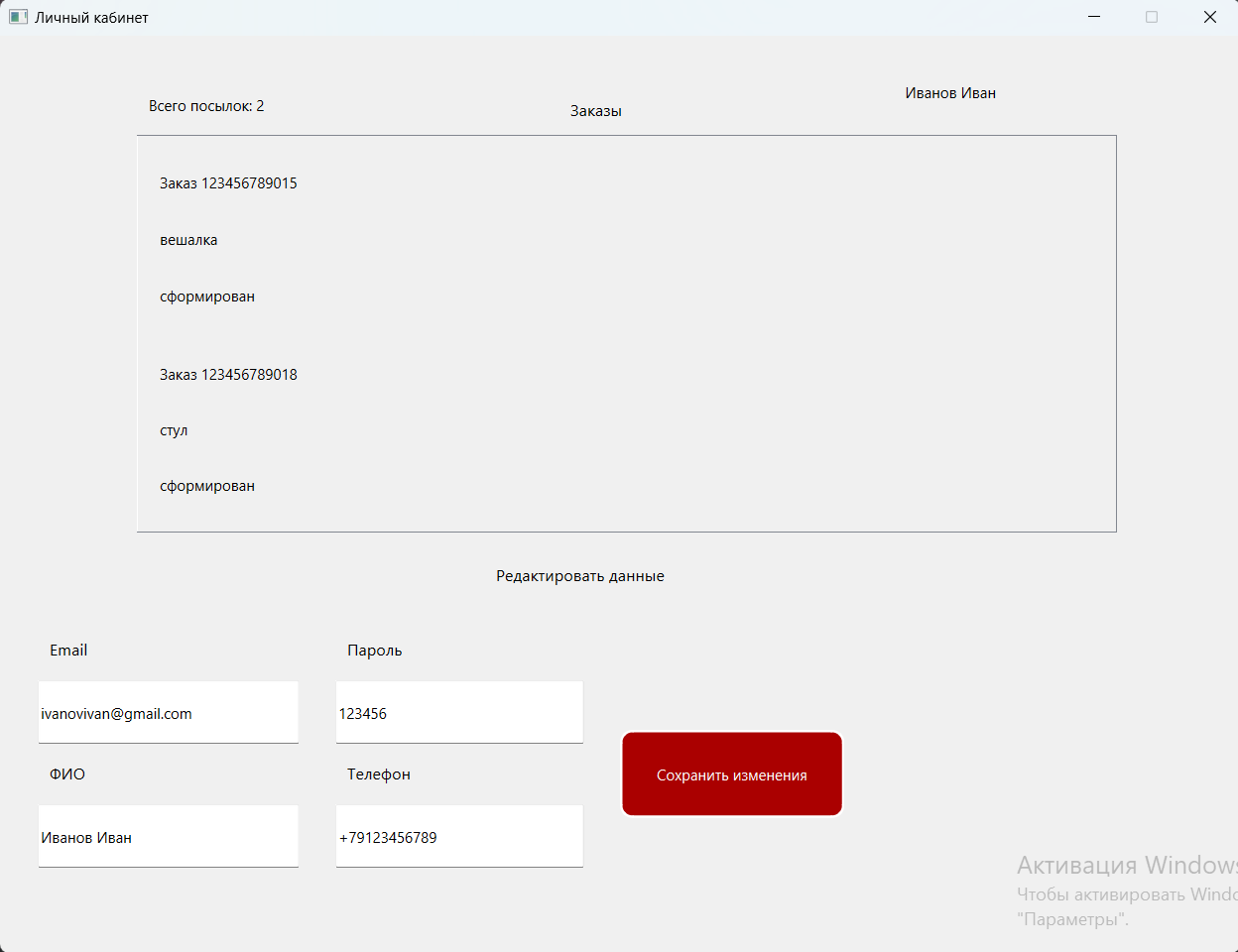
**1.3.1 Вход и авторизация пользователя**

При запуске приложения пользователь попадает на страницу авторизации. Здесь он вводит свои логин и пароль и входит в систему. (Рис. 1).



**Рис. 1: Форма входа**

**1.3.2 Личный кабинет пользователя**

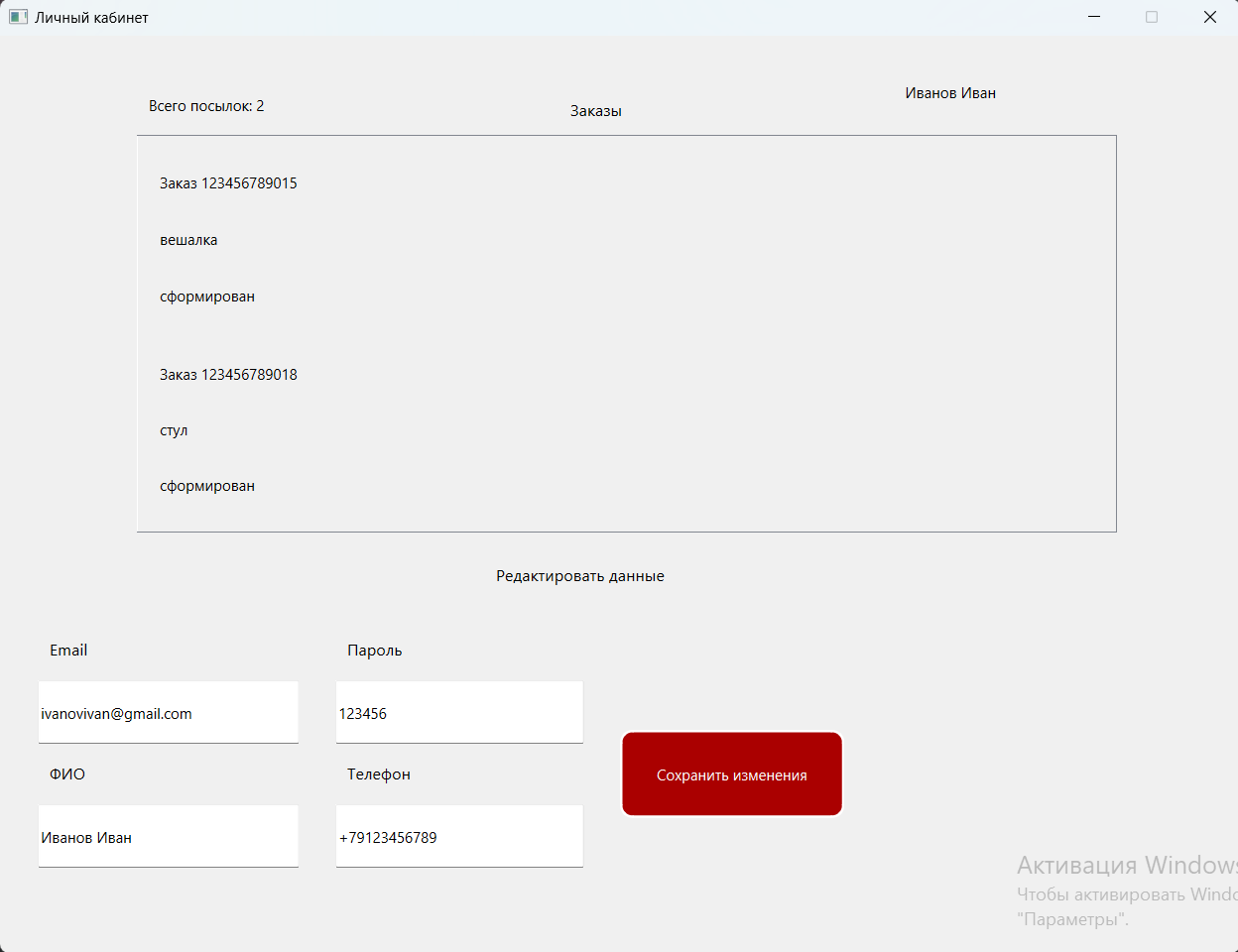


**Рис. 2: Главная страница приложения**

**В личном кабинете пользователю доступно окно с сформированными посылками и окно редактирования личных данных** (Рис. 2).

**1.3.3 Функционал главной страницы**

В ЛК пользователя стрелочками указаны ФИО клиента, окно с посылками и окно редактирования личных данных. (Рис. 3).



**Рис. 3: Главная страница приложения**

## **2. Руководство модератора**

**2.1 Общие данные о программе**

Для получения функций модератора определенному пользователю выдаются определенная электронная почта и пароль. Также окно авторизации пользователя и администратора – это 2 разных окна. (Рис.4)

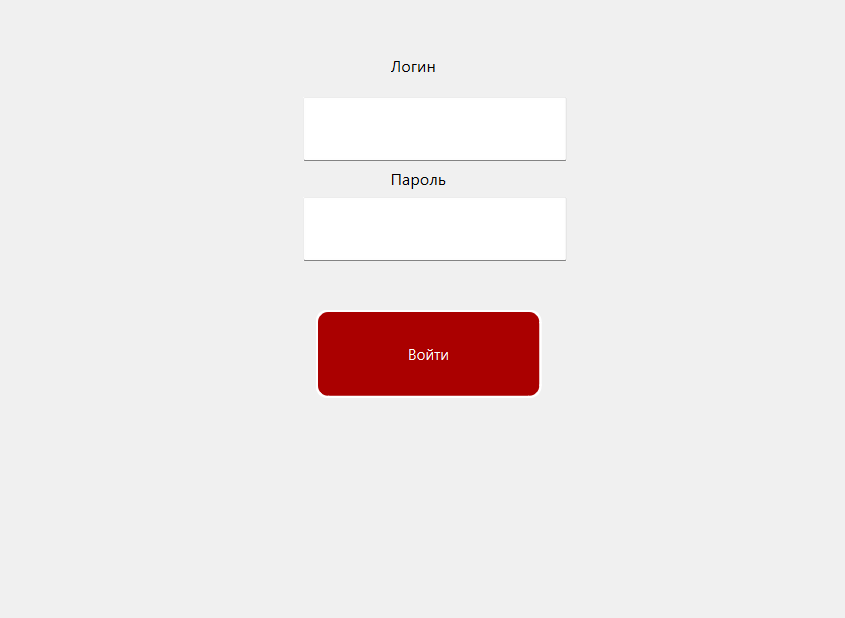
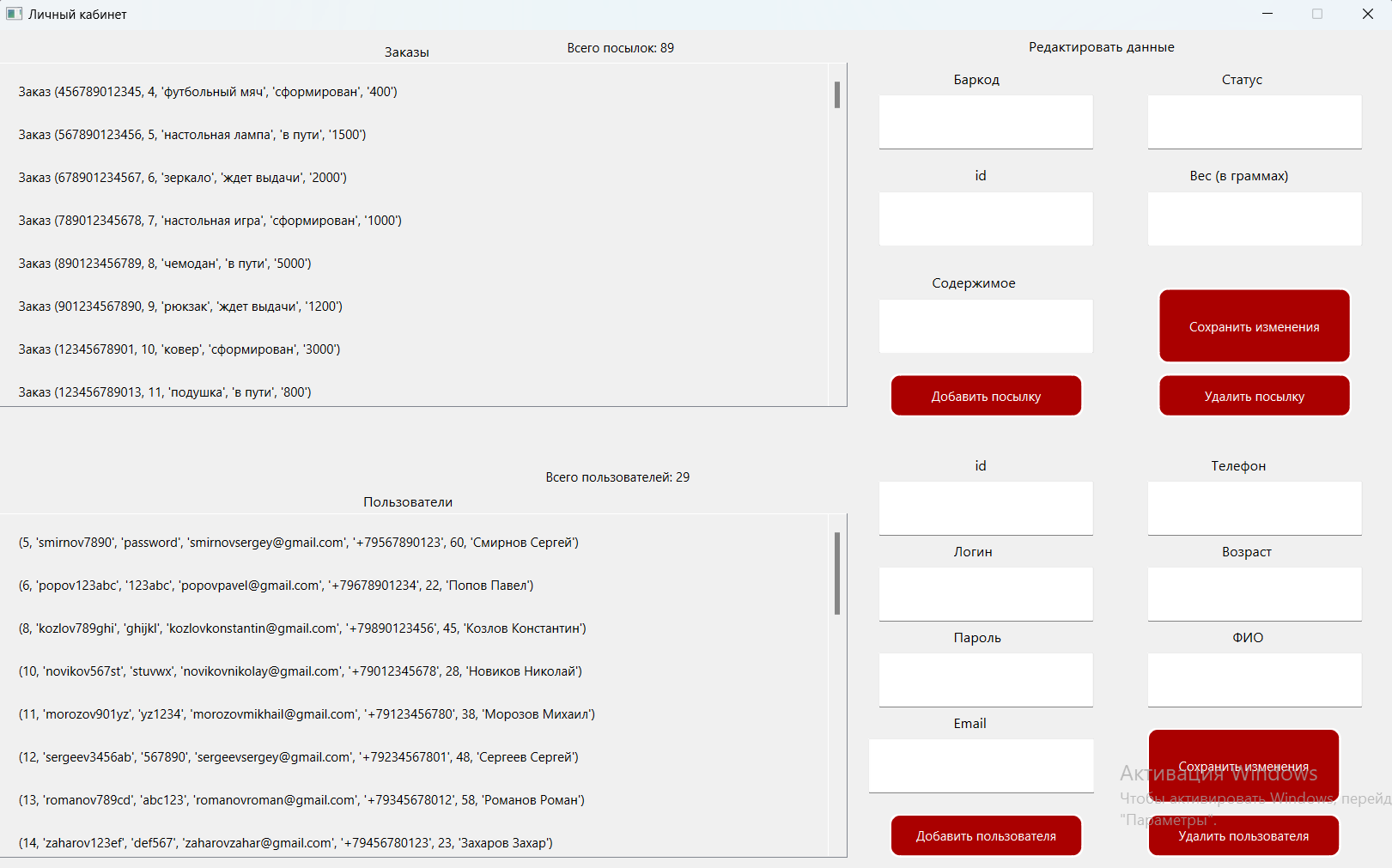


Рис.4 Окно авторизации модератора

**2.1 Личный кабинет модератора**

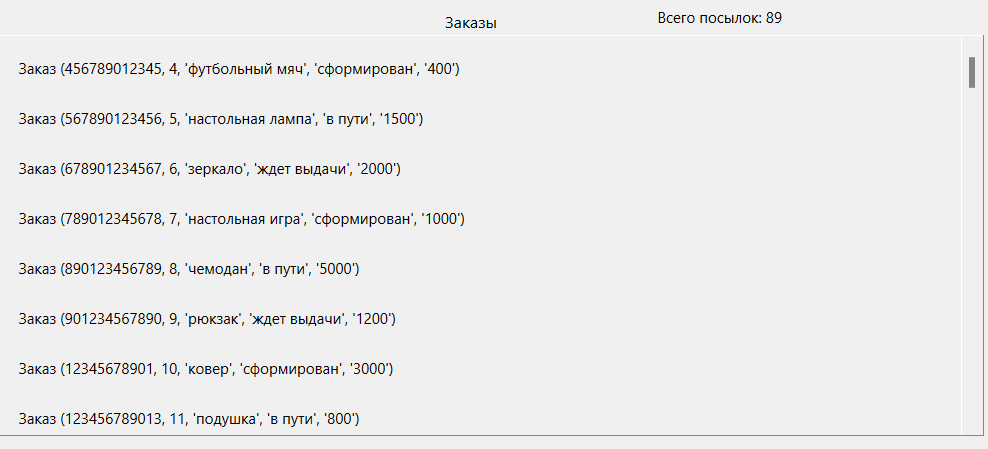
От интерфейса пользователя интерфейс модератора отличается личным кабинетом и его функционалом (Рис. 5).



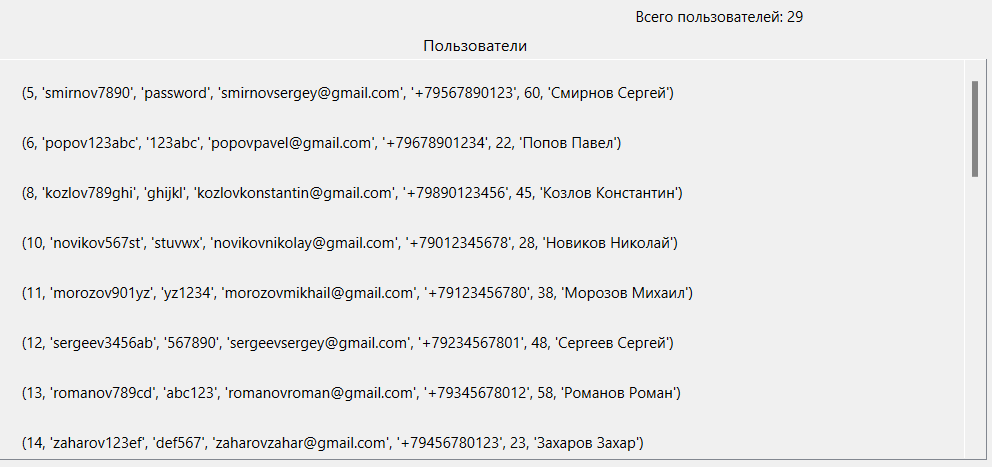
**Рис. 5: Интерфейс личного кабинета модератора**

В данном интерфейсе модератор может:

* Просматривать таблицы баз данных соответствующих названий (Рис. 6-7).

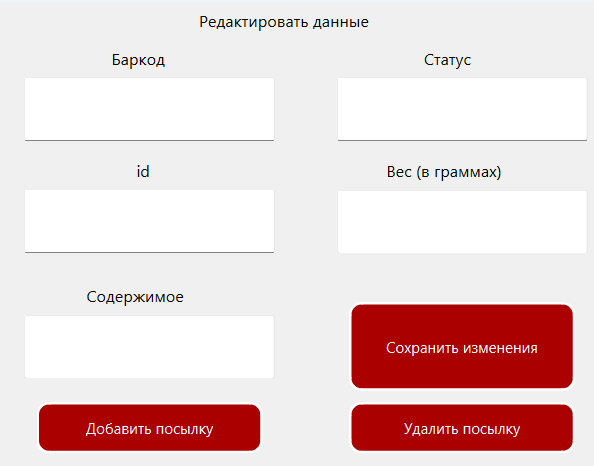


**Рис. 6: База данных посылок**

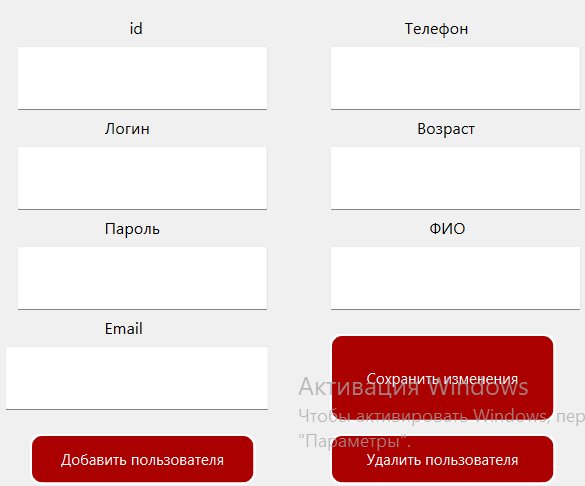


**Рис. 7: База данных клиентов**

**Присутствует полный функционал по созданию, редактированию и удалению записей. (Рис. 8-9)**



**Рис. 8: Манипуляции с базой данных посылок**



**Рис. 9: Манипуляции с базой данных клиентов**

**2.2 Редактирование записей**

* Для добавления записи необходимо заполнить все окна и нажать Добавить пользователя/посылку. (Рис.8-9)
* Для удаления записи необходимо в строке id (клиенты) или Баркод(посылки) ввести id/баркод и после нажать удалить пользователя/посылку. (Рис.8-9)
* Для изменения записи необходимо в строке id (клиенты) или Баркод(посылки) ввести id/баркод и после ввести измененные/неизмененные данные в остальные строки и нажать Сохранить изменения. (Рис.8-9)

**2.3 Контактная информация**

Для сообщения об ошибке или вопросах по пользованию приложения можно обратиться через данные интернет-ресурсы:

* Электронная почта: 123456790a495@gmail.com

## **3. Руководство программиста**

**3.1 Необходимые программы и библиотеки**

Для работы приложения необходимы следующие программы и библиотеки:

* Python версии 3.10 и выше.
* PyCharm версии 2023.1 и выше.
* PostgreSQL версии 16.3 и выше.

Библиотеки для Python:

* Psycopg2 – версия 2.9.9 и выше.
* PyQt6 – версия 6.2 и выше.

**3.2 Требования к оборудованию**

* Тип: ноутбук или ПК
* Модель процессора: Intel Core i5-12500H и выше.
* Общее количество ядер: 12 и больше.
* Объем оперативной памяти: 16 ГБ и больше.
* Модель встроенной видеокарты: Intel Iris Xe Graphics.

**3.3 Установка**

1. Распакуйте zip-файл с приложением в папку PyCharm.
2. Установите базу данных в СУБД PostgreSQL.
3. Откройте PyCharm и установите все необходимые библиотеки.

**3.4 Основные файлы и их функции**

Программа реализована с помощью нескольких основных файлов и модулей:

* **authorization.py** – файл авторизации для пользователя.
* **user\_interface.py** – файл, содержащий функции для работы с базой данных, такие как подключение, получение списка посылок и редактирования профиля пользователя.
* **admin\_access.py** – файл авторизации для администратора
* **admin\_interface.py** – файл, содержащий функции для работы с базой данных, такие как подключение, получение списка посылок и пользователей и их дальнейшее редактирование (удаление, добавление, изменение).

**Описание основных функций:**

**Файл authorization.py:**

Описание класса **Authorization**:

Класс **Authorization** наследует от design\_authorization.Ui\_Form и Qt.QtWidgets.QMainWindow, создавая окно регистрации.

**Переменные:**

- self.setupUi: Qt дизайн.

- self.connect\_to\_db: подключение к базе данных.

- self.boxA.setValidator(validator) валидатор для логина

- self.boxB.setValidator(validator) валидатор для пароля.

- self.buttonResultA.clicked.connect(self.entrance) Кнопка «Войти» для перехода в главное окно

Функции:

1. \_\_init\_\_():

\* Вызывает конструктор родительского класса.

\* Инициализирует self.setupUi(self) для настройки интерфейса.

\* Подключает обработчики событий к кнопкам и виджетам:

2. def connect\_to\_db

\* Осуществляет подключение к базе данных

3. def entrance(self):

\*Осуществляет верификацию пользователя по введенным логину и паролю и активирует главное окно

4. def go\_to\_lk(self):

\* Модуль для непосредственного перехода из одного окна в другое

**Описание работы:**

Класс authorization позволяет пользователю ввести свои логин и пароль и после направляет его в главное окно.

**Файл user.interface:**

Класс **SecondWindow** наследует от user.interface.Ui\_Form и Qt.QtWidgets.QMainWindow, создавая главное окно пользователя.

Переменные:

self.boxName.setValidator(validator4)  
self.boxEmail.setValidator(validator2)  
self.boxPassword.setValidator(validator1)  
self.boxPhone.setValidator(validator3)

Валидаторы для окон ввода личных данных

self.show\_package() Вывод таблицы с посылками  
self.buttonEdit.clicked.connect(self.set\_changes) Кнопка применить изменения для личной информации пользователя

- self.setupUi: Qt дизайн.

- self.connect\_to\_db: подключение к базе данных.

Функции:

1. \_\_init\_\_():

\* Вызывает конструктор родительского класса.

\* Инициализирует self.setupUi(self) для настройки интерфейса.

\* Подключает обработчики событий к кнопкам и виджетам:

2. def connect\_to\_db

\* Осуществляет подключение к базе данных

3. def show\_package(self):

\* Выводит данные sql таблицы в виде таблицы с посылками пользователя

4. def set\_changes(self):

\* Применяет изменения личных данных пользователя и записывает их в БД

**Файл admin\_access:**

Описание класса **Authorization**:

Класс **Authorization** наследует от design\_authorization.Ui\_Form и Qt.QtWidgets.QMainWindow, создавая окно регистрации.

**Переменные:**

- self.setupUi: Qt дизайн.

- self.connect\_to\_db: подключение к базе данных.

- self.boxA.setValidator(validator) валидатор для логина

- self.boxB.setValidator(validator) валидатор для пароля.

- self.buttonResultA.clicked.connect(self.entrance) Кнопка «Войти» для перехода в главное окно

Функции:

1. \_\_init\_\_():

\* Вызывает конструктор родительского класса.

\* Инициализирует self.setupUi(self) для настройки интерфейса.

\* Подключает обработчики событий к кнопкам и виджетам:

2. def entrance(self):

\*Осуществляет верификацию пользователя по введенным логину и паролю и активирует главное окно

3. def go\_to\_lk(self):

\* Модуль для непосредственного перехода из одного окна в другое

**Файл admin\_interface:**

Класс **SecondWindow** наследует от user.interface.Ui\_Form и Qt.QtWidgets.QMainWindow, создавая главное окно пользователя.

Переменные:

self.boxID.setValidator(validator5)  
self.boxLogin.setValidator(validator1)  
self.boxPassword.setValidator(validator1)  
self.boxEmail.setValidator(validator2)  
self.boxPhone.setValidator(validator3)  
self.boxAge.setValidator(validator6)  
self.boxName.setValidator(validator4)  
self.boxBarCode.setValidator(validator7)  
self.boxIdPac.setValidator(validator5)  
self.boxWeight.setValidator(validator5)

Валидаторы для окон ввода данных посылок и пользователей

self.show\_package() Отображение таблицы посылок  
self.show\_users() Отображение таблицы пользователей  
self.buttonEditUs.clicked.connect(self.set\_changes\_us) Кнопка для изменения данных в списке пользователей  
self.buttonEditPac.clicked.connect(self.set\_changes\_pac) Кнопка для изменения данных в списке посылок  
  
self.buttonDelUs.clicked.connect(self.del\_us) Кнопка для удаления записей в списке пользователей  
self.buttonDelPac.clicked.connect(self.del\_pac) Кнопка для удаления записей в списке посылок  
self.buttonAddUs.clicked.connect(self.add\_us) Кнопка для добавления записей в списке пользователей  
self.buttonAddPac.clicked.connect(self.add\_pac) Кнопка для добавления записей в списке посылок

- self.setupUi: Qt дизайн.

- self.connect\_to\_db: подключение к базе данных.

Функции:

1. \_\_init\_\_():

\* Вызывает конструктор родительского класса.

\* Инициализирует self.setupUi(self) для настройки интерфейса.

\* Подключает обработчики событий к кнопкам и виджетам:

2. def connect\_to\_db

\* Осуществляет подключение к базе данных

3. def show\_package(self):

\* Выводит данные sql таблицы в виде таблицы с посылками пользователей

4. def show\_users(self):

\* Выводит данные sql таблицы в виде таблицы с данными пользователей

5. def set\_changes\_us(self):

\* Изменяет данные пользователей

6. def set\_changes\_pac(self):

\* Изменяет данные посылок

7. def del\_us(self):

\* Удаляет данные пользователей

8. def del\_pac(self):

\* Удаляет данные посылок

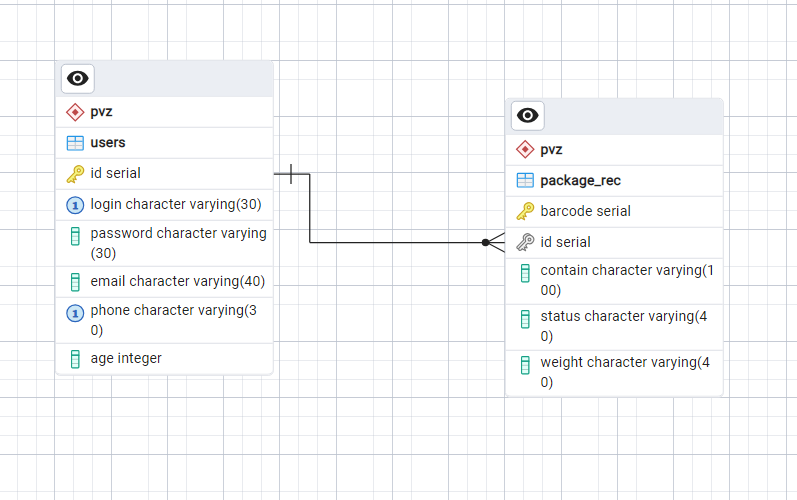
9. def add\_us(self):

\* Добавляет данные пользователей

10. def add\_pac(self):

\* Добавляет данные посылок

**3.5 База данных**

****

База данных выполнена в СУБД PostgreSQL Рис.10.

**Описание базы данных**

Данная база данных представляет собой систему для управления информацией о посылках и пользователях.

**Таблицы базы данных**

Главной сущностью является таблица users. В ней находится вся информация о пользователях.

| **Столбец** | **Тип данных** | **Описание** |
| --- | --- | --- |
| id | serial | Идентификатор пользоват. |
| login | character varying(30) | Логин |
| password | character varying(30) | Пароль |
| email | character varying(40) | Почта |
| phone | character varying(30)) | Телефон |
| age | integer | Возраст |
| name | character varying(50) | ФИО |

В таблице package\_rec хранится информация о пользователях.

package\_rec

| **Столбец** | **Тип данных** | **Описание** |
| --- | --- | --- |
| barcode | bigint | Идентификатор посылки |
| id | integer | Идентификатор пользоват. |
| contain | character varying(100) | Содержимое посылки |
| status | character varying(40) | Статус(готов к выдаче) |
| weight | character varying(40) | Вес |

**3.6 Контактная информация**

Для сообщения об ошибке или вопросах по пользованию приложения можно обратиться через данные интернет-ресурсы:

* Электронная почта: 123456790a495@gmail.com

## **4. Тестирование системы**

**Авторизация пользователя/администратора**

**1.Входные данные:**

* Имя пользователя: ivanov1234
* Пароль: 123456

**Выходные данные:**

* Переход на главную страницу приложения.

**2.Входные данные:**

* Имя пользователя: kozlov789ghi
* Пароль: ghijkl

**Выходные данные:**

* Переход на главную страницу приложения.

**3. Входные данные:**

* Имя пользователя: admin
* Пароль: admin123

**Выходные данные:**

* Переход на главную страницу приложения.

Удаление пользователя

**4. Входные данные:**

* id: 4

**Выходные данные:**

* удаление пользователя с id “4” из базы данных

Добавление посылки

5. **3. Входные данные:**

* Баркод: 1122334455667
* Id: 7
* Статус: ожидает выдачи
* Содержимое: книга
* Вес: 300

**Выходные данные:**

* Добавление данной посылки в базу данных

### Описание тестирования

1. **Авторизация пользователя:**
   * Ввод данных существующего пользователя в форму авторизации.
   * Проверка перенаправления на главную страницу приложения.
2. **Авторизация пользователя:**
   * Ввод данных существующего пользователя в форму авторизации.
   * Проверка перенаправления на главную страницу приложения.
3. **Авторизация администратора:**
   * Ввод данных администратора в форму авторизации.
   * Проверка перенаправления на главную страницу приложения.
4. **Удаление пользователя:**
   * Выбор пользователя и его удаление
   * Проверка удаления записи из таблицы users
5. **Добавление посылки:**
   * Ввод данных новой посылки
   * Проверка добавления записи в таблицу package\_rec

Все тесты были успешно выполнены, и система показала корректную работу в соответствии с требованиями.

## **5.Анализ результатов**

В ходе работы над проектом были успешно реализованы следующие функции:

* **Создание системы управления базами данных, содержащей информацию о посылках и пользователях**
  + Спроектирована и реализована база данных с использованием PostgreSQL.
  + Обеспечена целостность данных с помощью связей и ограничений.
* **Разработка интерфейса для просмотра информации о посылках и пользователях, а также возможности редактирования записей**
  + Создан пользовательский интерфейс с использованием PyQt6.
  + Реализована форма авторизации пользователей.
  + Внедрены функции для отображения текущих заказов и личной информации.
* **Проведение всестороннего тестирования системы для обеспечения ее надежности и удобства использования.**
  + Тестирование функционала регистрации пользователей.
  + Проверка корректности отображение данных.
  + Тестирование функций добавления, удаления и редактирования данных в базе.

Тестирование системы показало следующие результаты:

* **Надежность работы системы.**
  + Система корректно обрабатывает запросы пользователей.
  + Все функции выполняются без сбоев и ошибок.
* **Удобство использования.**
  + Интерфейс приложения интуитивно понятен и прост в использовании.
  + Пользователи могут легко находить и получать рекомендации по турам.
* **Соответствие требованиям.**
  + Система удовлетворяет все заявленные требования к функционалу.
  + Все функции, предусмотренные в проекте, успешно реализованы.

# **Заключение**

В данной курсовой работе была успешно разработана и реализована система управления данными и интерфейс для управления информацией о посылках и пользователях. Система прошла всестороннее тестирование и продемонстрировала свою надежность, производительность и удобство использования.

Созданная система предоставляет пользователям следующие возможности:

* Просмотр информации о посылках конкретного пользователя
* Редактирования личных данных пользователя
* Редактирование и просмотр всех данных таблицы администратором

Выполнение данной курсовой работы позволило углубить и систематизировать знания в области проектирования, реализации и тестирования баз данных. Также были приобретены практические навыки в работе с различными инструментами и технологиями, используемыми в разработке современных систем управления данными.

Результаты данной работы могут быть использованы в дальнейшей профессиональной деятельности, связанной с проектированием, разработкой и эксплуатацией информационных систем.