

**Министерство образования, науки и молодежной политики Республики Коми**

**ГПОУ "Сыктывкарский политехнический техникум"**

**Курсовая работа**

**Тема: База данных гостиницы. Подсистема «Работа с клиентами»**

**Выполнил**

Студент 4 курса 414 группы Михаил Игоревич Елесов

**Проверил**

Пунгин Илья Вячеславович

**Дата проверки:**

18.12.2025

**Сыктывкар 2025**

## Оглавление

Введение .....	3
Анализ предметной области .....	5
Проектирование базы данных.....	8
Логическая структура .....	10
Физическая структура данных .....	12
Реализация системы .....	15
Безопасность .....	20
Разработка стратегии резервного копирования .....	22
Заключение.....	23
Список литературы .....	24
Приложение .....	25

# **Введение**

## **Актуальность темы**

Современные отели предлагают различные услуги: регистрация клиентов, бронирование номеров, составление расписаний и пр. Из-за этого появляется большое количество данных — о клиентах, бронях, номерах, финансах.

Вручную работать с этим объемом информации достаточно сложно, медленно, неудобно и появляется риск ошибочных данных. Поэтому сегодня важно иметь не просто файлы или таблицы, а единую базу данных.

Она нужна, чтобы:

1. Быстро находить свободные номера и оформлять клиентов.
2. Надежно и безопасно хранить все данные в одном месте.
3. Автоматизировать рутину — печать счетов, отчётов.
4. Анализировать работу отеля для принятия правильных решений.

Простыми словами, база данных упрощает структурирование, хранение больших объёмов информации, а также управление ими, поэтому тема считается довольно актуальной.

## **Цель и Задачи**

Целью работы является создание системы управления гостиницей на основе базы данных с удобным пользовательским интерфейсом.

Для достижения цели были поставлены задачи:

1. Анализ предметной области
2. Описание структуры базы данных

3. Реализация базы данных в СУБД PostgreSQL
4. Реализация пользовательского интерфейса с помощью языка программирования Python в редакторе Visual Studio Code

## **Практическая значимость**

Результаты моей курсовой работы могут быть полезны для реальной деятельности гостиницы.

1. Созданная база данных — это готовый каркас системы, который можно установить в небольшой гостинице или отеле, чтобы автоматизировать работу с бронями, клиентами и номерами.
2. Система упростит рутинную работу:
  - Ускорит процесс заселения и выписки гостей.
  - Уменьшит число ошибок в расчетах и при поиске информации.
  - Сама будет создавать счета, отчеты и напоминания.
3. С помощью базы данных руководитель сможет легко и быстро получать нужную информацию:
  - Видеть, какие номера пользуются спросом.
  - Вести историю постоянных клиентов.
4. Работа может стать хорошим примером для других студентов, которые учатся создавать базы данных, показывая весь процесс от идеи до готовой системы.

# **Анализ предметной области**

## **Описание предметной области**

База данных для гостиницы должна хранить информацию о постояльцах, работниках, номерах и их статусах и историю бронирований.

Основные компоненты системы:

1. Список должностей
2. Список сотрудников
3. Список номеров
4. Список постояльцев
5. История бронирований

## **Перечень входных данных**

1. Список должностей
2. Список сотрудников
  - ФИО
  - Должность
3. Спальное места (койки)
  - Номера, в которых койки находятся
  - Статус коек
  - Цена за ночь
4. Постояльцы (клиент)
  - ФИО
  - Контактные данные (номер телефона, адрес электронной почты)
  - Паспортные данные

## **Перечень выходных данных**

1. История бронирований
  - Дата заезда
  - Дата выселения
  - Клиент
  - Спальное место
  - Количество ночей
  - Итоговый ценник
2. Расписание уборок номеров
  - Дата
  - Номер уборки
  - Горничная

## **Требования и ограничения**

### Функциональные требования

1. Управление клиентами
  - Хранение данных гостей (ФИО, контактные и паспортные данные)
2. Управление номерным фондом
  - Учёт номеров и их статусов
3. Управление процессом бронирования
  - Создание и изменение брони
  - Контроль за датами заезда и выезда

#### 4. Управление расписаниями

##### Требования к данным

1. Целостность (гарантия, что данные не противоречат друг другу)
2. Точность (Корректное отражение всех операций)

##### Практические ограничения

1. Простота и удобство интерфейса
2. Уникальность данных (недопустимость одинаковых записей)

# Проектирование базы данных

## Концептуальная модель

Концептуальная модель – схема, отображающая основные компоненты системы и связи между ними (см. приложение рисунок 1).

Основные компоненты:

1. Должности(roles)
2. Работники(workers)
3. Номера(rooms)
4. Спальные места(beds)
5. Клиенты(clients)
6. Бронирования(reservation)
7. Расписание уборки(schedule\_of\_cleaning\_rooms)

Связи:

1. Должности – работники
  - Работник обязательно должен иметь ФИО и должность
2. Комнаты – спальные места
  - Каждое спальное место должно быть закреплено за определённой комнатой
3. Клиенты – бронирования
  - Одна бронь закрепляется за одним постояльцем
  - Один постоялец может сделать несколько броней



4. Спальные места – бронирования
  - В записи бронирования должна быть указана койка
5. Работники – бронирования
  - Бронь должен создавать администратор
6. Расписание уборки – Номера
  - Комната должна указываться в записи
7. Расписание уборки – работники
  - Должна быть указана горничная, ответственная за уборку

## Логическая структура

База данных состоит из 7 таблиц (см. приложение рисунок 2):

1. roles (должности):
  - id – Уникальный идентификатор (первичный ключ)
  - name – название должности
2. workers (работники):
  - id – Уникальный идентификатор (первичный ключ)
  - name – ФИО работника
  - role – должность (внешний ключ от roles.id)
3. rooms (комнаты):
  - number – номер комнаты, уникальный идентификатор (первичный ключ)
  - floor – этаж
  - number\_of\_places – количество мест
  - is\_active – свободен ли номер
4. beds (Спальные места):
  - id – Уникальный идентификатор (первичный ключ)
  - room\_number – номер комнаты (внешний ключ от rooms.number)
  - is\_occupied – свободна ли спальное место
  - price\_per\_night – ценник за ночь
5. clients (клиенты):
  - id – Уникальный идентификатор (первичный ключ)
  - name – Имя
  - surname – фамилия

- surname1 – отчество
- phone\_number – номер телефона
- email – адрес электронной почты
- passport\_series – серия паспорта
- passport\_number – номер паспорта

6. reservation (бронирования):

- date\_of\_checkin – дата заезда
- date\_of\_checkout – дата выезда
- client – клиент (внешний ключ от clients.id)
- administrator – администратор (внешний ключ от workers.id)
- bed – спальное место (внешний ключ от beds.id)
- number\_of\_nights – количество ночей
- total\_price – Итоговый ценник

7. schedule\_of\_cleaning\_rooms (расписание уборок комнат):

- date\_of\_cleaning – Дата уборки
- room\_number – номер комнаты (внешний ключ от rooms.number)
- housemaid – горничная (внешний ключ от workers.id)

## Физическая структура данных

Физическая реализация включает в себя создание таблиц и их описание.

roles:

Название	Тип	Комментарий
id	INTEGER PRIMARY KEY	Ключ
name	VARCHAR NOT NULL	Название

workers:

Название	Тип	Комментарий
id	INTEGER PRIMARY KEY	Ключ
name	VARCHAR NOT NULL	ФИО
role	INTEGER	Должность

rooms:

Название	Тип	Комментарий
number	VARCHAR(3) PRIMARY KEY	Номер комнаты, ключ
floor	INTEGER	Этаж
number_of_places	INTEGER	количество мест

is_active	BOOLEAN	Свободна ли комната
-----------	---------	---------------------

beds:

Название	Тип	Комментарий
id	INTEGER PRIMARY KEY	Ключ
room_number	VARCHAR(3) NOT NULL	Номер комнаты
is_occupied	BOOLEAN	Занято ли спальное место
price_per_night	REAL	Цена за ночь

clients:

Название	Тип	Комментарий
id	INTEGER PRIMARY KEY	Ключ
name	VARCHAR NOT NULL	Имя
surname	VARCHAR NOT NULL	Фамилия
surname1	VARCHAR	Отчество
phone_number	VARCHAR(11) NOT NULL	Номер телефона
email	VARCHAR	Электронная почта

passport_series	VARCHAR(4) NOT NULL	Серия паспорта
passport_number	VARCHAR(6) NOT NULL	Номер паспорта

reservation:

Название	Тип	Комментарий
date_of_checkin	DATE NOT NULL	Дата заезда
date_of_checkout	DATE NOT NULL	Дата выезда
client	INTEGER	Постоялец
administrator	INTEGER	Администратор
number_of_nights	INTEGER	Количество ночей
total_price	REAL	Итоговый ценник

schedule\_of\_cleaning\_rooms:

Название	Тип	Комментарий
date_of_cleaning	DATE NOT NULL	Дата уборки
room_number	VARCHAR(3) NOT NULL	Номер
housemaid	INTEGER NOT NULL	Горничная

# Реализация системы

## Создание базы данных

База данных создавалась в СУБД PostgreSQL 18.

Создание таблиц:

- create table roles (id integer primary key, name varchar(20) not null); -- таблица с должностями
- create table workers(id integer primary key, name varchar not null, role integer, foreign key (role) references roles(id)); -- таблица с работниками
- create table rooms(number varchar(3) primary key, floor integer, number\_of\_places integer, is\_active boolean); -- таблица с номерами
- create table beds(id integer primary key, room\_number varchar(3) not null, is\_occupied boolean, price\_per\_night real, foreign key (room\_number) references rooms(number)); -- таблица с койками
- create table clients(id integer primary key, name varchar not null, surname varchar not null, surname1 varchar, phone\_number varchar(11) not null, email varchar, passport\_series varchar(4) not null, passport\_number varchar(6) not null); -- таблица с постояльцами
- create table reservation(date\_of\_checkin date, date\_of\_checkout date, client integer, administrator integer, bed integer, number\_of\_nights integer, total\_price real, foreign key (client) references clients(id), foreign key (administrator) references workers(id), foreign key (bed) references beds(id)); -- история брони (будет заполняться в приложении)
- create table schedule\_of\_cleaning\_rooms (date\_of\_cleaning date not null, room\_number varchar(3) not null, housemaid integer not null, foreign key (room\_number) references rooms(number), foreign key (housemaid) references workers(id)); -- таблица для расписания уборок номеров (будет заполняться в приложении)

Заполнение таблиц необходимыми данными:

- insert into rooms values('111', 1, 1, true), ('112', 1, 1, true), ('113', 1, 1, true), ('114', 1, 1, true), ('115', 1, 1, true), ('116', 1, 1, true);
- insert into rooms values('211', 2, 2, true), ('212', 2, 2, true), ('213', 2, 2, true), ('214', 2, 2, true), ('215', 2, 2, true), ('216', 2, 2, true);
- insert into rooms values('311', 3, 3, true), ('312', 3, 3, true), ('313', 3, 3, true), ('314', 3, 3, true), ('315', 3, 3, true), ('316', 3, 3, true);
- insert into rooms values('411', 4, 4, true), ('412', 4, 4, true), ('413', 4, 4, true), ('414', 4, 4, true), ('415', 4, 4, true), ('416', 4, 4, true);
- insert into beds values(1, '111', false, 100), (2, '112', false, 100), (3, '113', true, 100), (4, '114', true, 100), (5, '115', true, 100), (6, '116', true, 100);
- insert into beds values(7, '211', true, 200), (8, '211', true, 200), (9, '212', true, 200), (10, '212', true, 200), (11, '213', true, 200), (12, '213', true, 200);
- insert into beds values(13, '214', true, 200), (14, '214', true, 200), (15, '215', true, 200), (16, '215', true, 200), (17, '216', true, 200), (18, '216', true, 200);
- insert into beds values(19, '311', true, 300), (20, '311', true, 300), (21, '311', true, 300);
- insert into beds values(22, '312', true, 300), (23, '312', true, 300), (24, '312', true, 300);
- insert into beds values(25, '313', true, 300), (26, '313', true, 300), (27, '313', true, 300);
- insert into beds values(28, '314', true, 300), (29, '314', true, 300), (30, '314', true, 300);
- insert into beds values(31, '315', true, 300), (32, '315', true, 300), (33, '315', true, 300);
- insert into beds values(34, '316', true, 300), (35, '316', true, 300), (36, '316', true, 300);



- insert into beds values(37, '411', true, 400), (38, '411', true, 400), (39, '411', true, 400), (40, '411', true, 400);
- insert into beds values(41, '412', true, 400), (42, '412', true, 400), (43, '412', true, 400), (44, '412', true, 400);
- insert into beds values(45, '413', true, 400), (46, '413', true, 400), (47, '413', true, 400), (48, '413', true, 400);
- insert into beds values(49, '414', true, 400), (50, '414', true, 400), (51, '414', true, 400), (52, '414', true, 400);
- insert into beds values(53, '415', true, 400), (54, '415', true, 400), (55, '415', true, 400), (56, '415', true, 400);
- insert into beds values(57, '416', true, 400), (58, '416', true, 400), (59, '416', true, 400), (60, '416', true, 400);

Создание процедур:

- create or replace procedure extension(client\_id integer, new\_date date) as \$\$ begin update reservation set date\_of\_checkout=new\_date where client=client\_id; end; \$\$ language plpgsql; -- процедура, обновляющая дату выселения
- create or replace procedure checkin(date\_checkin date, date\_checkout date, client\_id integer, admin integer, bed integer, number\_of\_nights integer) as \$\$ declare t\_price real; b\_price real; begin select price\_per\_night into b\_price from beds where id = bed; t\_price := b\_price \* number\_of\_nights; insert into reservation values(date\_checkin, date\_checkout, client\_id, admin, bed, number\_of\_nights, t\_price); end; \$\$ language plpgsql; -- процедура заселения

## **Разработка пользовательского интерфейса**

Для пользовательского интерфейса для базы данных использовался язык программирования Python и редактор кода Visual Studio Code.

Для реализации нужного результата использовались следующие библиотеки:

- tkinter
- psycpg2
- datetime
- tkalendar

Архитектура программы и его компоненты (см. приложение рисунки 3-10):

1. Главное окно

- Вкладки
- Вкладка для бронирований
  - Поля для заполнений информации о клиентах, администраторах и датах
  - Таблица бронирований
  - Кнопки создания, удаления бронирований и обновления данных
- Вкладка постояльцев
  - Таблица с клиентами
  - Кнопки добавления, удаления постояльцев и обновления данных
  - Фильтр постояльцев
- Вкладка номеров и коек
  - Таблица со спальными местами
  - Кнопка для изменения статуса определённой койки
  - Фильтр спальных мест
- Вкладка для расписаний уборок
  - Таблица с расписанием
  - Поля для заполнения информации о дате, номере и горничной
  - Кнопка для добавления записи в базу
  - Фильтр записей об уборке
- Вкладка работников

- Таблица с работниками
  - Кнопки добавления, удаления работников и обновления данных
  - Фильтр работников
- Вкладка отчётов
2. Диалоговые окна (Окна для заполнения)

## Тестирование системы

Функциональное тестирование:

3. Проверка корректного добавления данных
4. Поиск брони, постояльца, работника по определённым параметрам

Тестирование пользовательского интерфейса:

1. Проверка работоспособности всех кнопок
2. Проверка фильтров
3. Проверка добавления в базу данных различных записей
4. Попытка внести в базу некорректную запись

Обнаруженные ошибки:

**Проблема:** можно сделать несколько броней на одно и то же спальное место.

**Решение:** Обновление виджета, отвечающего за выбор койки, который будет принимать в себя номера только тех коек, которые не заняты.

## Безопасность

Для защиты данных были созданы представления:

- create VIEW reservation\_history as SELECT r.date\_of\_checkin, r.date\_of\_checkout, r.number\_of\_nights, r.total\_price, c.id as client\_id, c.name as client\_name, c.surname as client\_surname, c.phone\_number as client\_phone, w.id as admin\_id, w.name as admin\_name, b.id as bed\_id, b.room\_number, b.price\_per\_night as bed\_price, rm.floor, rm.number\_of\_places FROM reservation r LEFT JOIN clients c ON r.client = c.id LEFT JOIN workers w ON r.administrator = w.id LEFT JOIN beds b ON r.bed = b.id LEFT JOIN rooms rm ON b.room\_number = rm.number ORDER BY r.date\_of\_checkin DESC;
- create VIEW current\_reservations as select rh.\*, CasE when rh.date\_of\_checkout > CURRENT\_DATE THEN rh.date\_of\_checkout - CURRENT\_DATE ELSE 0 END as nights\_remaining FROM reservation\_history rh where CURRENT\_DATE BETWEEN rh.date\_of\_checkin and rh.date\_of\_checkout ORDER BY rh.room\_number;
- CREATE VIEW cleaning\_schedule\_view AS SELECT sc.date\_of\_cleaning, sc.room\_number, r.floor, r.number\_of\_places, w.id as housemaid\_id, w.name as housemaid\_name, rol.name as housemaid\_role, CASE WHEN sc.date\_of\_cleaning < CURRENT\_DATE THEN 'Выполнена' WHEN sc.date\_of\_cleaning = CURRENT\_DATE THEN 'На сегодня' ELSE 'Запланирована' END as cleaning\_status FROM schedule\_of\_cleaning\_rooms sc LEFT JOIN rooms r ON sc.room\_number = r.number LEFT JOIN workers w ON sc.housemaid = w.id LEFT JOIN roles rol ON w.role = rol.id ORDER BY sc.date\_of\_cleaning, r.floor, sc.room\_number;
- CREATE VIEW workers\_info AS SELECT w.id, w.name as full\_name, split\_part(w.name, ' ', 1) as first\_name, split\_part(w.name, ' ', 2) as middle\_name, split\_part(w.name, ' ', 3) as last\_name, r.id as role\_id, r.name as

```
role_name, FROM workers w LEFT JOIN roles r ON w.role = r.id ORDER BY r.id,  
w.name;
```

## Разработка стратегии резервного копирования

Резервное копирование – процесс, целью которого является обеспечить возможность восстановления данных при сбое (аппаратном, человеческой ошибке и пр.) с минимальными потерями.

Сначала был воспроизведён полный бэкап всей базы данных: `pg_dump -U elesov elesov > elesovhostel.db_backup.sql`

Затем были созданы бэкапы отдельных таблиц: `pg_dump -t reservation elesov > /tmp/reservationhostel.dump; pg_dump -t schedule_of_cleaning_rooms elesov > /tmp/schedule_of_cleaning_rooms hostel.dump;`

Также была создана копия python-файла, отвечавшего за графический пользовательский интерфейс.

## Заключение

В ходе выполнения курсовой работы поставленная цель была достигнута. Создана система для гостиничной деятельности: реализованы простая и удобная база данных для управления гостиницей и подсистема для работы с клиентами в виде понятного пользовательского интерфейса.

### 1. Аналитическая часть:

- Проведён анализ предметной области
- Определены все требования и ограничения

### 2. Проектная часть

- Разработана концептуальная модель базы данных
- На основе этой концептуальной модели была разработана логическая структура
- На основе логической структуры была разработана физическая структура базы данных

### 3. Программная реализация:

- Созданы база данных и запросы к ней в СУБД PostgreSQL
- Разработан пользовательский интерфейс для упрощённого управления этой самой базой данных с помощью языка программирования Python
- Были созданы резервные копии всех компонентов системы в случае сбоев ?

Из ключевых характеристик системы можно выделить возможность добавления неограниченного количества записей, фильтрацию информации, а также отображение простой и краткой статистики.

Система полностью функционирует и может использоваться для управления гостиницей и её составляющими, в том числе и для работы с клиентами.

## Список литературы

1. PostgreSQL: Documentation: 18 – <https://www.postgresql.org/docs/current/>
2. Руководство по работе с базами данных PostgreSQL – <https://metanit.com/sql/postgresql/?ysclid=mj5wn2eii2901306410>
3. tkinter — Python interface to Tcl/Tk — Python 3.14.2 documentation – <https://docs.python.org/3/library/tkinter.html>
4. Руководство по программированию на Tkinter и Python – <https://metanit.com/python/tkinter/?ysclid=mj5wsjm2gu718044868>
5. psycopg2 · PyPI – <https://pypi.org/project/psycopg2/>
6. Python и базы данных | Модуль psycopg. Подключение к серверу PostgreSQL – <https://metanit.com/python/database/2.1.php?ysclid=mj5wuhzuvv34999340>



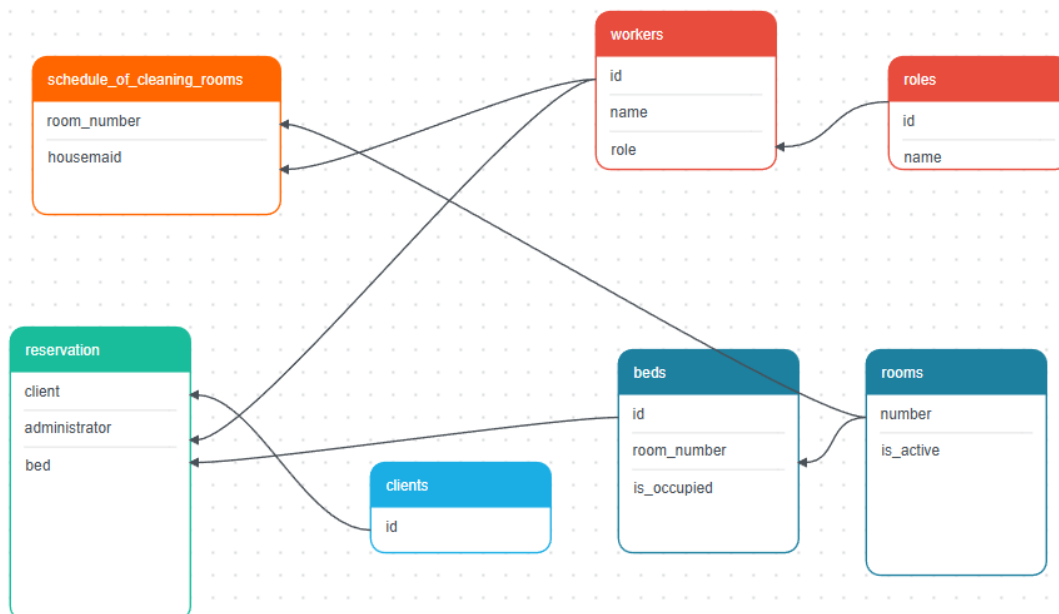


Рисунок 1. Концептуальная модель

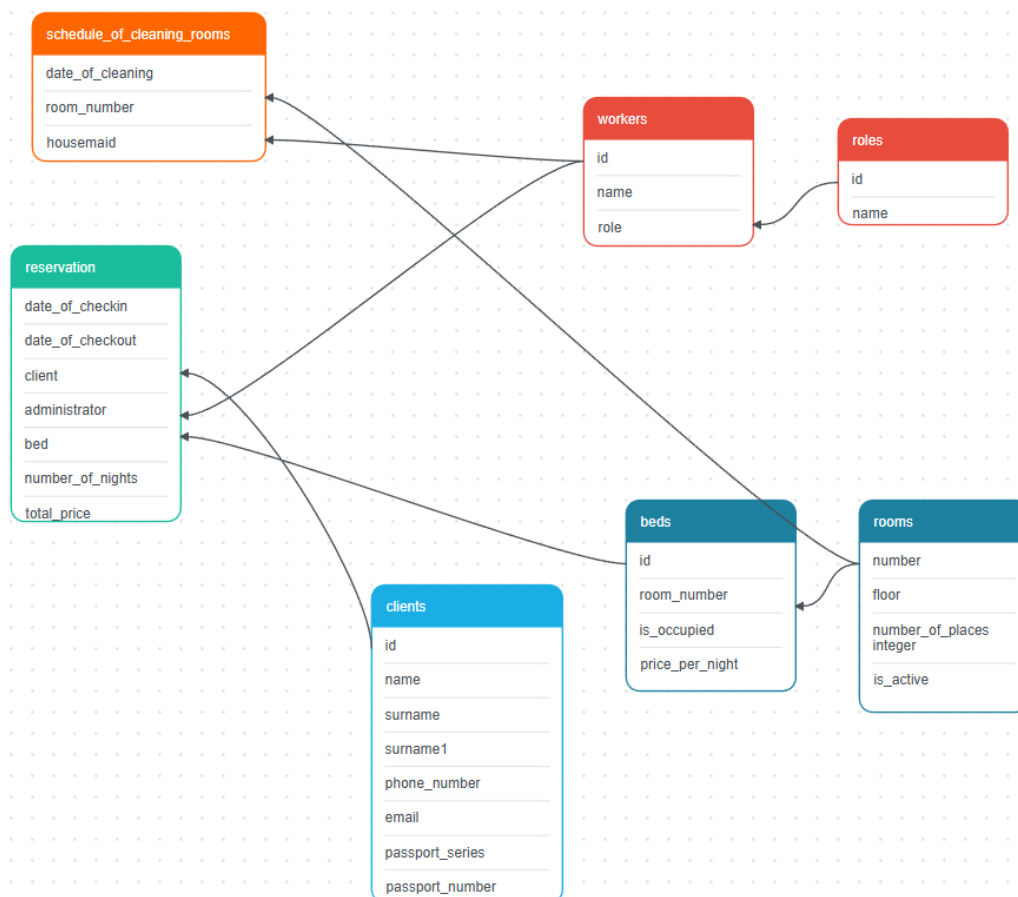


Рисунок 2. Логическая структура



Рисунок 3. Структура приложения

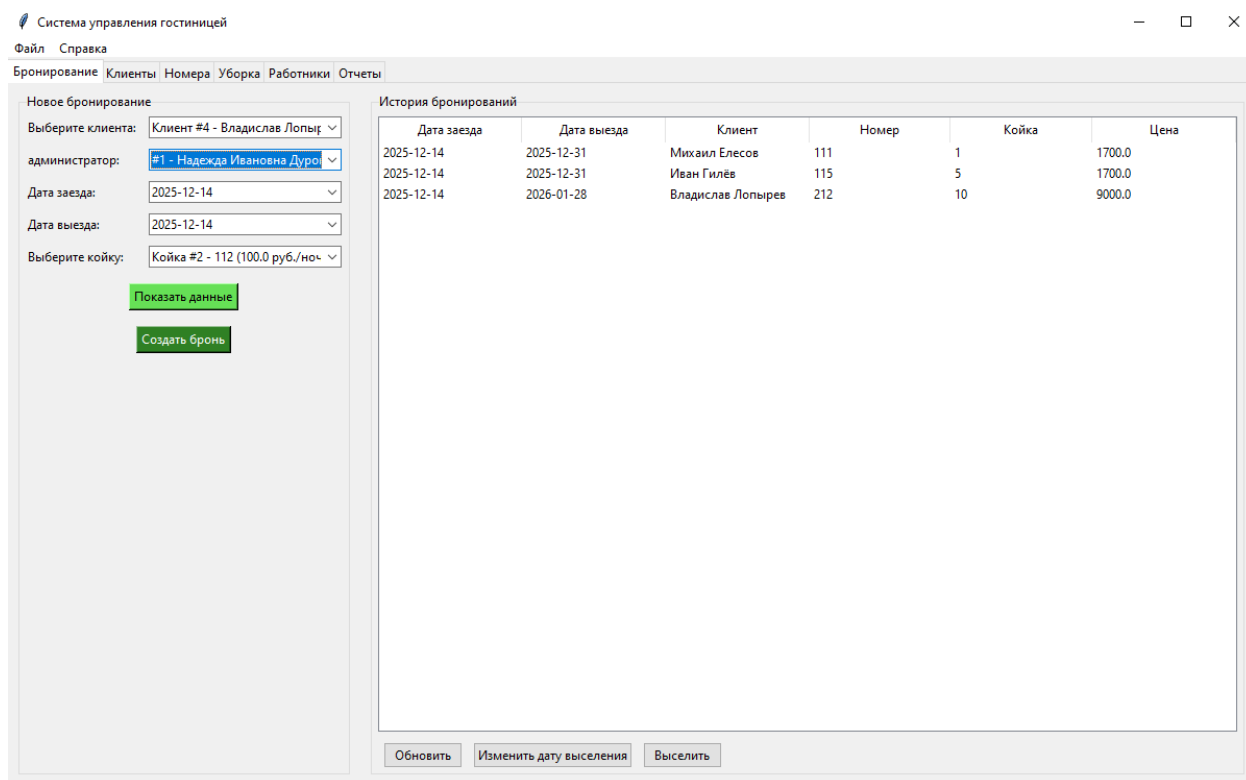


Рисунок 4. Главное окно. Вкладка для бронирований

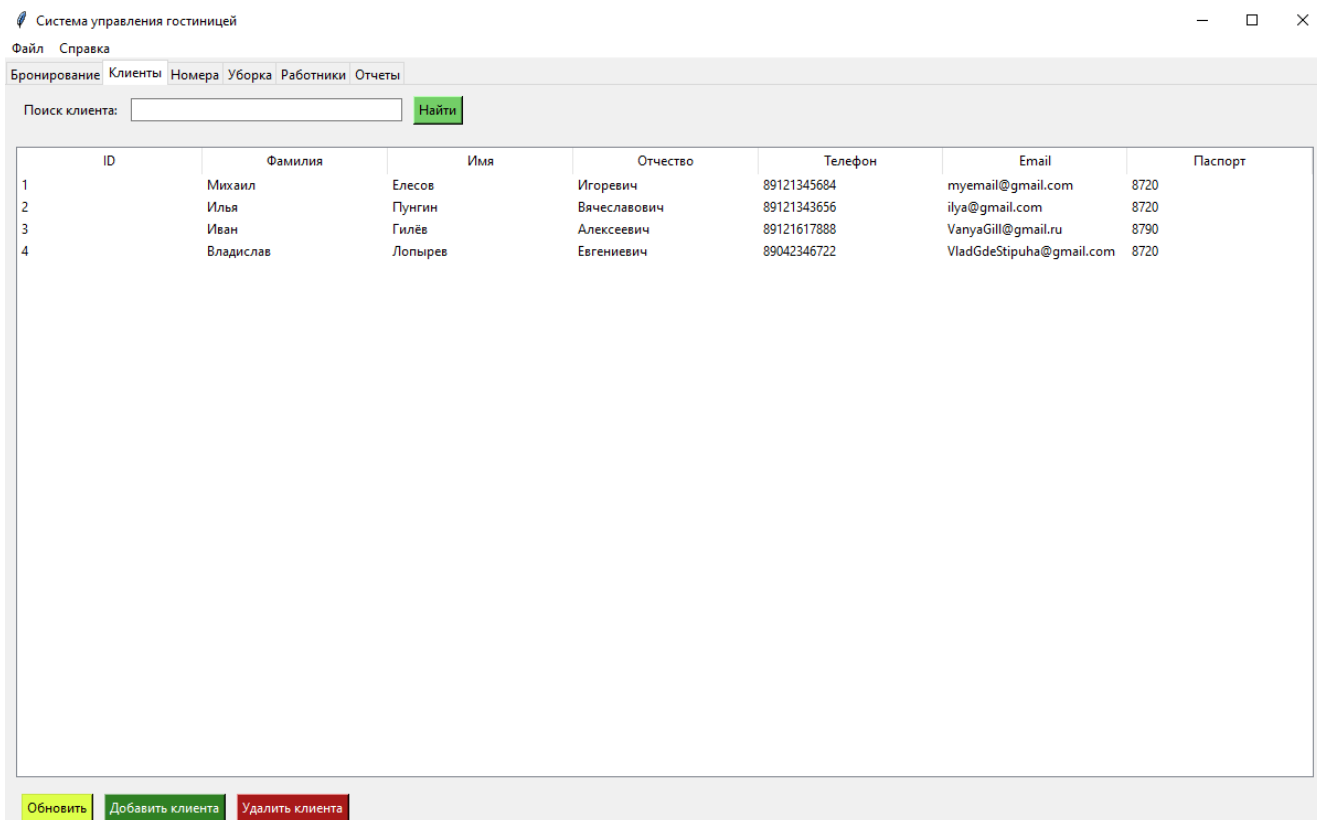


Рисунок 5. Главное окно. Вкладка "Клиенты"

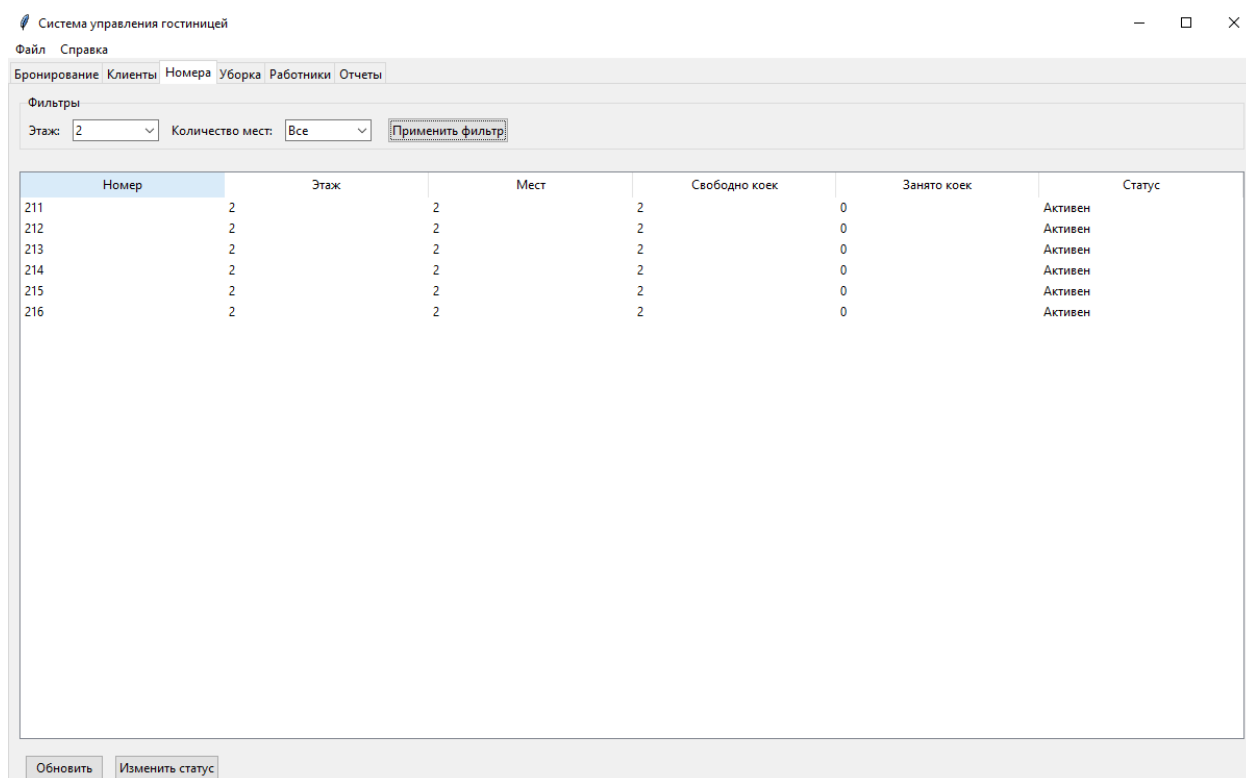


Рисунок 6. Главное окно. Вкладка "Номера"

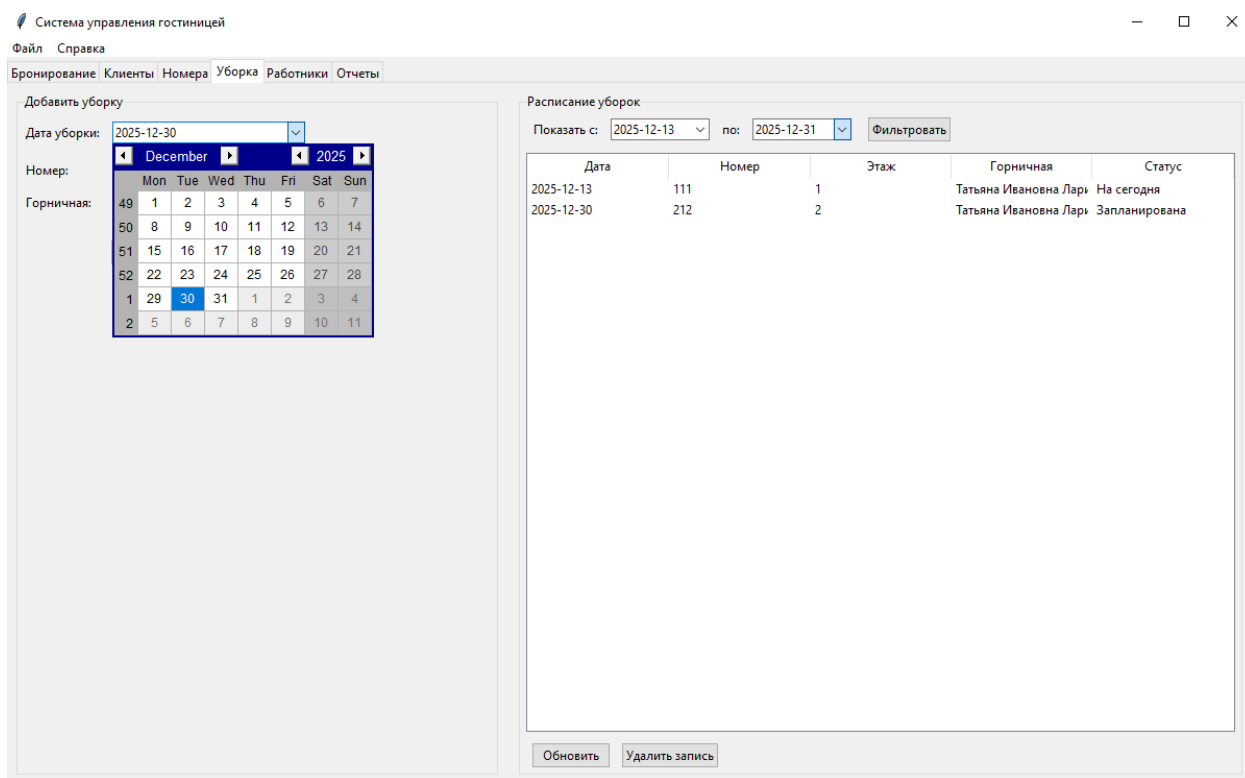


Рисунок 7. Главное окно. Вкладка с расписанием уборок

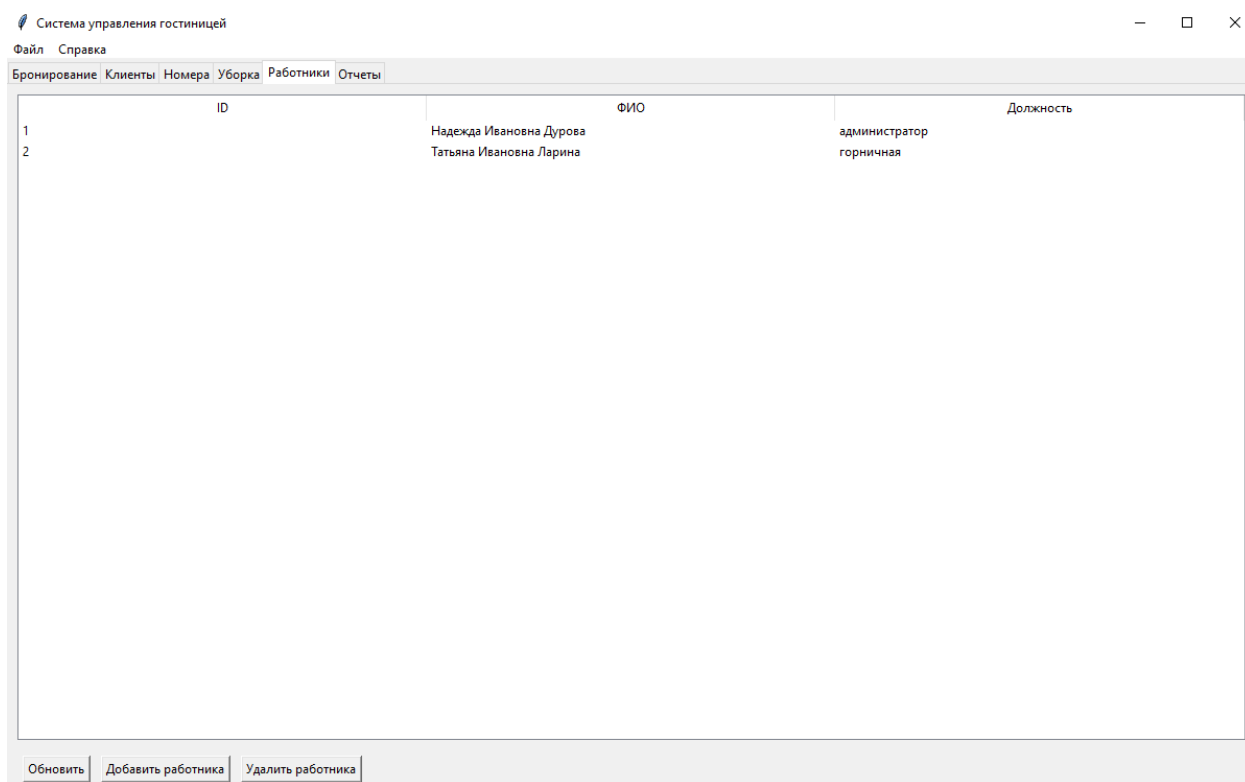


Рисунок 8. Главное окно. Вкладка "Работники"

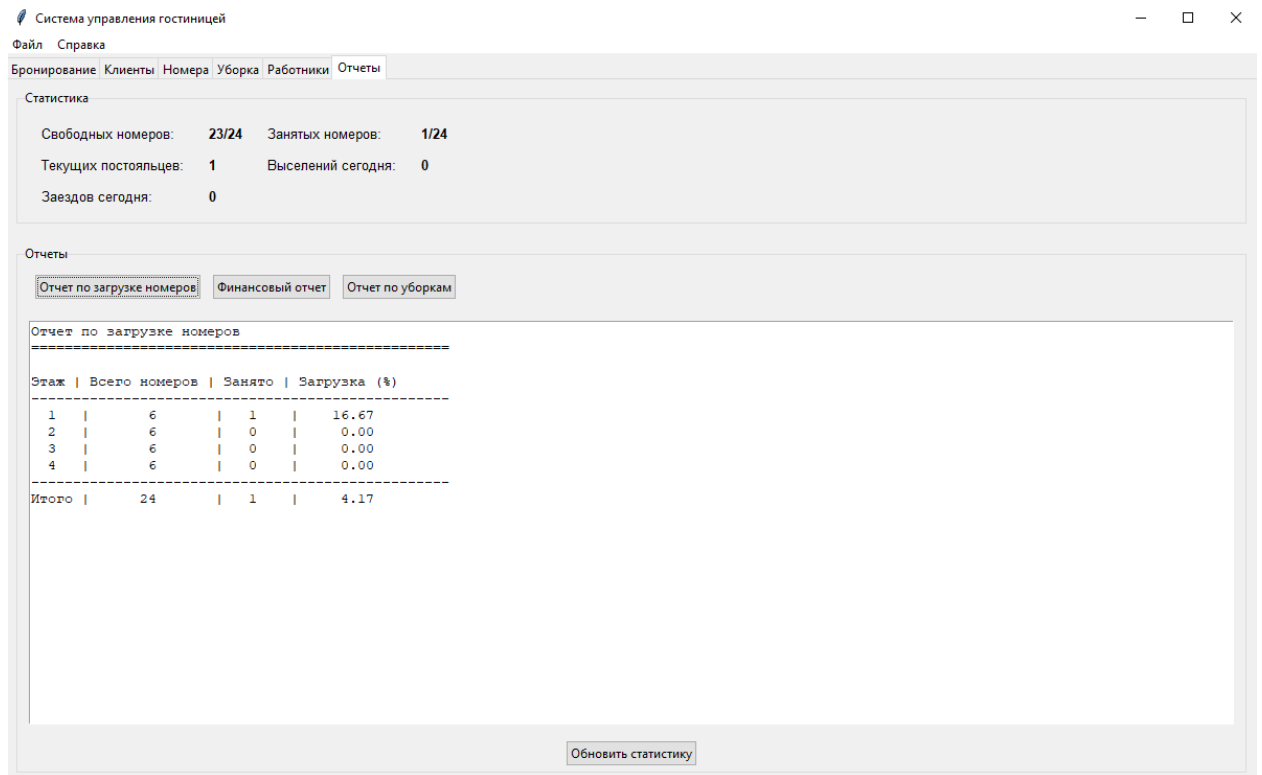


Рисунок 9. Главное окно. Вкладка Отчёты

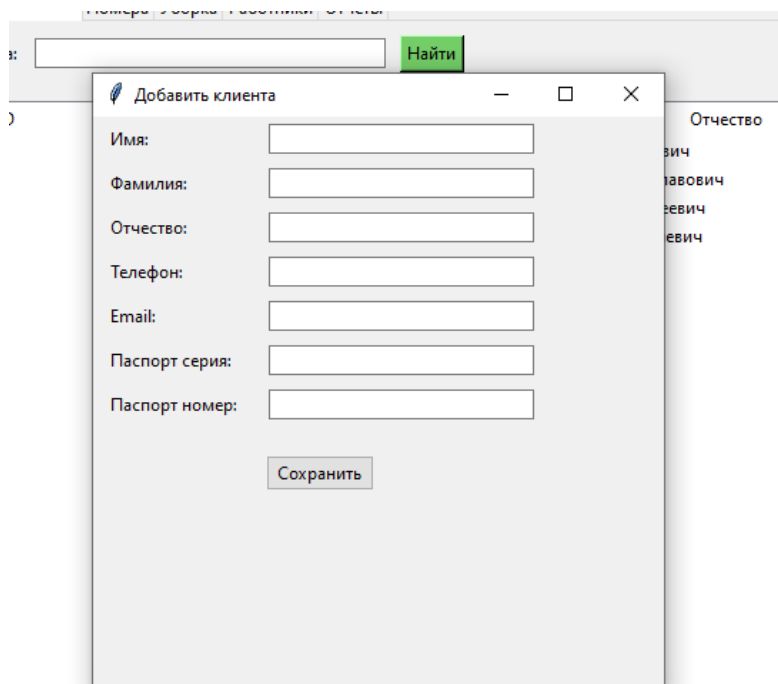


Рисунок 10. Пример диалогового окна

```
Hostel.py x
1 import tkinter as tk
2 from tkinter import ttk, messagebox
3 from tkinter import *
4 import psycopg2
5 from datetime import date, datetime
6 from tkcalendar import DateEntry # pip(pip3) install tkcalendar
7
8
9 class HotelManagementApp:
10     def __init__(self, root):
11         self.root = root
12         self.root.title("Система управления гостиницей")
13         self.root.geometry("1200x700")
14
15         # Параметры подключения к базе данных
16         self.db_params = {
17             'host': 'localhost', #127.0.0.1
18             'database': 'test', #elesov
19             'user': 'postgres', #elesov
20             'password': '1224', #1224
21             'port': '5432' #5432
22         }
23
24         self.setup_ui()
25         self.connect_to_db()
26
27     def setup_ui(self):
28         # Создаем меню
29         self.create_menu()
30
31         # Создаем вкладки
32         self.tab_control = ttk.Notebook(self.root)
33
34         # Вкладка бронирования
35         self.tab_reservation = ttk.Frame(self.tab_control)
36         self.tab_control.add(self.tab_reservation, text='Бронирование')
37         self.setup_reservation_tab()
38
39         # Вкладка клиенты
```

Рисунок 11. Фрагмент кода