**Колледж ВятГУ**

**ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ ДЛЯ ГОСТИНИЦЫ**

Курсовой проект

090207.249755.ПЗ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Выполнил обучающийся гр. ИСПк-304-52-00 |  | / | Селезенев К.А. | / | 26.12.2025 |
|  | (подпись) |  | (инициалы, фамилия) |  | (дата) |
| Руководитель курсового проекта  Преподаватель Колледжа ВятГУ |  | / | Самоделкин П.А. | / | 26.12.2025 |
|  | (подпись) |  | (инициалы, фамилия) |  | (дата) |

Киров

2025

**РЕФЕРАТ**

Селезенев К.А. Проектирование информационной системы для гостиницы 090207.249755.ПЗ курсовой проект / ВятГУ, Колледж ВятГУ; рук. П.А. Самоделкин. – Киров, 2025. – ПЗ 50 с., 24 рис., 1 табл., 7 источников, 2 прил.

ИНФОРМАЦИОННАЯ СИСТЕМА ДЛЯ ГОСТИНИЦЫ, БАЗА ДАННЫХ, ИНФОРМАЦИОННАЯ СИСТЕМА, АВТОМАТИЗАЦИЯ.

Объект исследования – процесс автоматизации учета и управления деятельностью гостиницы, включая управление номерным фондом, обработку бронирований, регистрацию гостей, хранение их персональных данных с применением шифрования и формирование отчетности.

Предмет исследования – проектирование и разработка специализированной информационной системы «ГостиТут», обеспечивающей оптимизацию рабочих процессов гостиницы за счет внедрения цифровых инструментов для централизованного управления, безопасного хранения данных и автоматизации ключевых операций.

Цель курсового проекта: разработка информационной системы «ГостиТут» для автоматизации бизнес-процессов гостиницы, которая повысит эффективность бронирования и заселения гостей, обеспечит безопасное хранение конфиденциальной информации и упростит формирование отчетов для администратора.

Задачи:

* проанализировать предметную область;
* провести обзор существующих аналогов;
* разработать техническое задание;
* спроектировать пользовательский интерфейс информационной системы;
* смоделировать систему в нотации IDEF0, DFD и на языке UML;
* спроектировать структуру базы данных;
* реализовать минимально жизнеспособный продукт;
* реализовать пользовательский интерфейс на основе библиотеки PyQt6;
* реализовать серверную часть приложения с использованием СУБД PostgreSQL для хранения и обработки данных;
* физически реализовать базу данных на основе PostgreSQL;
* подготовить выводы.

Новизна заключается в создании специализированного программного решения, ориентированного на малые и средние гостиницы, которое сочетает в себе необходимый для их работы функционал (управление номерным фондом, бронирование, учет гостей) с повышенным уровнем безопасности за счет встроенного шифрования конфиденциальных данных (AES-256). В отличие от рассмотренных сложных и дорогих аналогов, система «ГостиТут» предлагает упрощенный и интуитивно понятный интерфейс, не требующий специальной технической подготовки от персонала.

Область применения – гостиничный бизнес, в частности, малые и средние гостиницы. Система предназначена для использования администраторами гостиницы с целью автоматизации ключевых бизнес-процессов: управления бронированиями, регистрации заезда/выезда, контроля статусов номеров, ведения учета гостей и формирования отчетности.

Значимость разработки состоит в повышении эффективности и безопасности работы гостиницы за счет автоматизации рутинных операций, минимизации ошибок ручного учета и обеспечения криптографической защиты персональных данных гостей. Внедрение системы позволит администраторам оперативно управлять загрузкой номерного фонда, контролировать финансовые расчеты и формировать необходимую документацию, что в целом повысит качество обслуживания.

**СОДЕРЖАНИЕ**

[**ВВЕДЕНИЕ** 6](#_Toc217574465)

[**1** **Анализ предметной области** 7](#_Toc217574466)

[**1.1 Описание предметной области** 7](#_Toc217574467)

[**1.2 Обзор аналогов** 12](#_Toc217574468)

[**1.2.1 Аналог «OperaPMS»** 12](#_Toc217574469)

[**1.2.2 Аналог «1С: Отель»** 13](#_Toc217574470)

[**1.2.3 Аналог «Cloudbeds»** 13](#_Toc217574471)

[**2** **Техническое задание** 15](#_Toc217574472)

[**3** **Проектирование информационной системы** 16](#_Toc217574473)

[**3.1 Моделирование информационной системы** 16](#_Toc217574474)

[**3.1.1 Функциональное моделирование системы** 16](#_Toc217574475)

[**3.1.2 Моделирование потоков данных** 19](#_Toc217574476)

[**3.1.3 Варианты использования системы** 20](#_Toc217574477)

[**3.2 Проектирование базы данных** 22](#_Toc217574478)

[**3.3 Проектирование интерфейса информационной системы** 30](#_Toc217574479)

[**4** **Реализация информационной системы** 33](#_Toc217574480)

[**4.1 Пользовательский интерфейс системы** 33](#_Toc217574481)

[**4.2 Реализация программного кода** 40](#_Toc217574482)

[**ЗАКЛЮЧЕНИЕ** 46](#_Toc217574483)

[**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ** 47](#_Toc217574484)

[**Приложение А** 49](#_Toc217574485)

[**Приложение Б** 50](#_Toc217574486)

Изм.

Лист

Фамилия

Подпись

Дата

Лист

5

*090207.249755.ПЗ*

Разраб.

.

Провер.

Реценз

Н. Контр.

Утверд.

Проектирование информационной системы для гостиницы

Лит.

Листов

50

*Колледж ВятГУ*

# **ВВЕДЕНИЕ**

Курсовой проект разрабатывался на базе Колледжа ФГБОУ ВО «Вятский государственный университет» в период с 01.09.2025 г. по 26.12.2025 г.

В документе рассматриваются следующие разделы:

1. Анализ предметной области.
2. Техническое задание.
3. Реализация информационной системы.
4. Заключение.
5. Список использованных источников.
6. Приложения.

Анализ предметной области содержит описание изучаемой темы, глоссарий, концепция, обзор аналогов, их преимущества и недостатки, на основе которых были рассмотрены аспекты, которые были реализованы в проекте. В разделе Техническое задание описывается понятие ТЗ и его роль. Разделы Проектирование и Реализация информационной системы включает в себя этапы создания информационной системы, включая моделирование её работы, разработку макетов и интерфейса, а также создание базы данных, необходимой для функционирования системы. В Заключении содержится вывод о проделанной работе и обобщение полученных результатов. Список использованных источников содержит в себе перечисление всех использованных при написании курсового проекта источников информации. В Приложениях находится дополнительные материалы проекта, такие как ТЗ и программный код приложения.

Изм.

Лист

№ докум.

Подпись

Дата

Лист

6

*090207.249755.ПЗ*

# **Анализ предметной области**

## **1.1 Описание предметной области**

Изм.

Лист

№ докум.

Подпись

Дата

Лист

7

*090207.249755.ПЗ*

**Название программного обеспечения:** Проектирование и разработка информационной системы для автоматизации деятельности гостиницы.

**Название ИС:** «ГостиТут».

Современный гостиничный бизнес, особенно в сегменте малых и средних отелей, требует эффективных инструментов для управления ключевыми операционными процессами. Ручной учет номерного фонда, бронирований и данных гостей становится источником ошибок, снижает скорость обслуживания и затрудняет формирование отчетности. Автоматизация этих процессов является критически важной для повышения конкурентоспособности, качества сервиса и соблюдения требований по защите персональных данных.

Развитие информационных технологий привело к появлению множества систем управления гостиницами (PMS – Property Management System), от крупных корпоративных решений вроде Opera PMS до облачных сервисов, таких как Cloudbeds. Однако для многих небольших гостиниц такие системы оказываются избыточными по функционалу, сложными в освоении или экономически нецелесообразными из-за высокой стоимости лицензий и сопровождения.

Информационная система «ГостиТут» — это настольное программное решение, разработанное специально для малых и средних гостиниц. Оно объединяет управление номерным фондом, обработку бронирований, регистрацию гостей (с шифрованием паспортных данных), контроль статусов номеров и формирование базовой отчетности в едином простом интерфейсе.

**Глоссарий**

Гость – физическое лицо, проживающее или планирующее проживание в гостинице.

Бронирование (Бронь) – предварительная фиксация номера за гостем на определенные даты заезда и выезда.

Заселение (Регистрация) – процесс оформления гостя на проживание, включающий внесение его персональных и паспортных данных в систему.

Выселение – процесс завершения проживания гостя, освобождения номера и фиксации итоговой стоимости.

Номерной фонд – совокупность всех номеров (комнат) гостиницы, классифицированных по категориям комфортности (например, Стандарт, Комфорт+, Люкс).

Изм.

Лист

№ докум.

Подпись

Дата

Лист

8

*090207.249755.ПЗ*

Статус номера – текстовое и визуальное обозначение текущего состояния номера: «Свободен», «Занят», «На уборке», «Забронирован».

Администратор – основной пользователь системы, сотрудник гостиницы, обладающий полными правами на управление бронированиями, заселением, номерным фондом и формированием отчетов.

Категория номера – группирующий признак, определяющий уровень комфорта и стоимость проживания (например, «Стандарт», «Комфорт+»).

СУБД (PostgreSQL) – система управления базами данных, используемая для создания, хранения, управления и обеспечения целостности структурированной информации в системе.

AES-256 – криптографический алгоритм симметричного шифрования, используемый в системе для защиты конфиденциальных паспортных данных гостей при хранении в базе данных.

**Концепция**

Информационная система «ГостиТут» должна предоставлять администратору гостиницы возможность:

* управлять номерным фондом (добавлять, редактировать, удалять номера и их характеристики);

Изм.

Лист

№ докум.

Подпись

Дата

Лист

9

*090207.249755.ПЗ*

* осуществлять бронирование номеров с фиксацией дат заезда и выезда;
* регистрировать заезд и выезд гостей;
* вносить и обеспечивать защищенное хранение паспортных данных гостей с использованием шифрования AES-256;
* управлять статусами номеров (свободен/занят/на уборке/бронь);
* просматривать список проживающих гостей;
* формировать отчеты о гостях;
* просматривать статистику загрузки номерного фонда;
* автоматически рассчитывать стоимость проживания.

ИС использует локальную базу данных PostgreSQL, которая хранит всю информацию о номерах, бронированиях, гостях и операциях. Система работает с этой БД, выполняя запросы на добавление, обновление, удаление и выборку данных.

Для входа в систему администратор должен пройти авторизацию по логину и паролю.

**Требования**

Требования к применению:

* интерфейс ИС должен быть простым, интуитивно понятным и не требовать специальной технической подготовки;
* ИС должна быть реализована на русском языке;
* ИС должна работать под управлением операционных систем Windows 10 (64-бит) или выше, а также macOS версии 12 или выше.

Требования к производительности:

* ИС должна обеспечивать отзывчивую работу в рамках задач одного рабочего места администратора.

Требования к реализации:

* архитектура: настольное клиент-серверное приложение;
* сервер базы данных: СУБД PostgreSQL версии 15 или выше;
* данные хранятся локально на том же компьютере, где работает приложение.

Изм.

Лист

№ докум.

Подпись

Дата

Лист

10

*090207.249755.ПЗ*

Требования к надежности:

* должна быть обеспечена стабильная работа приложения;
* необходимо настроить регулярное резервное копирование базы данных для защиты от потери информации;
* система должна обеспечивать целостность данных при выполнении операций.

Требования к интерфейсу:

* ИС должна предоставлять графический пользовательский интерфейс с основными разделами: Главная, Гости, Брони, Номера;
* размер главного окна приложения должен быть изменяемым;
* интерфейс должен визуально отображать статусы номеров с помощью цветовых маркеров.

Требования к безопасности:

* ИС должна реализовывать шифрование паспортных данных гостей по алгоритму AES-256 перед сохранением в базу данных;
* доступ к функционалу системы должен предоставляться только после успешной авторизации;
* система должна обеспечивать защиту от несанкционированного доступа к конфиденциальной информации.

Функциональные требования:

* авторизация в систему;
* управление номерным фондом;
* обработка бронирований (создание, просмотр, отмена);
* регистрация заезда и выезда гостей;
* внесение и безопасное хранение паспортных данных гостей;
* управление статусами номеров;
* просмотр списка проживающих;
* формирование отчетов о госте (в формате Word);
* просмотр статистики загрузки;
* автоматический расчет стоимости проживания.

Проектная организация: ИС использует реляционную базу данных PostgreSQL для хранения всей структурированной информации: данных о номерах, категориях, гостях, бронированиях, пользователях системы.

Действующее лицо ИС:

* администратор – единственная роль в системе, которая выполняет все перечисленные выше функции по управлению гостиницей.

Изм.

Лист

№ докум.

Подпись

Дата

Лист

11

*090207.249755.ПЗ*

## **1.2 Обзор аналогов**

В данном разделе рассмотрены аналоги информационных систем для гостиницы.

### **1.2.1 Аналог «OperaPMS»**

Изм.

Лист

№ докум.

Подпись

Дата

Лист

12

*090207.249755.ПЗ*

Opera PMS – распространенная система управления гостиницами в мире, используется в крупных сетях.

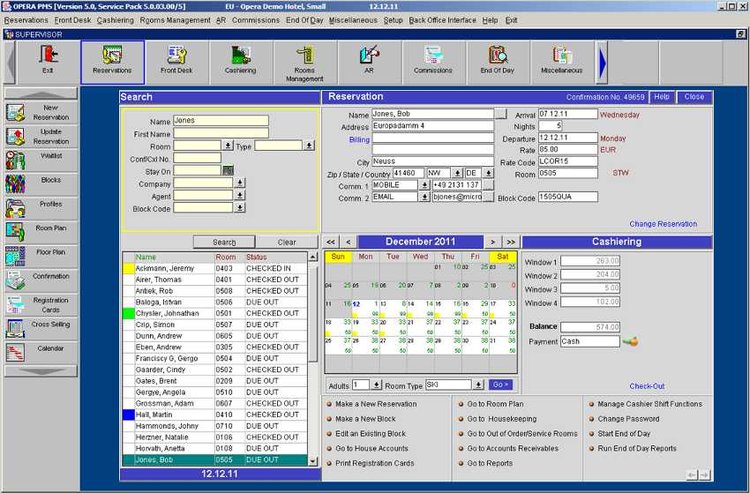


Рисунок 1 - Скриншот интерфейса программы «Opera PMS»

**Преимущества:**

* поддержка разных языков и валют;
* подходит для сетевых и крупных отелей;
* очень широкий функционал.

**Недостатки:**

* высокая стоимость внедрения и обслуживания;
* сложный интерфейс;
* избыточен для небольших гостиниц.

### **1.2.2 Аналог «1С: Отель»**

Изм.

Лист

№ докум.

Подпись

Дата

Лист

13

*090207.249755.ПЗ*

1С:Отель – технологичное решение для автоматизации управления и повышения эффективности гостиничного бизнеса.



Рисунок 2 - Скриншот интерфейса программы «1C:Отель»

**Преимущества:**

* большой функционал;
* интеграция с другими продуктами 1С;
* база специалистов и поддержка.

**Недостатки:**

* требуется обучение персонала;
* нужда в IT-специалисте для сопровождения;
* высокая стоимость лицензий и внедрения.

### **1.2.3 Аналог «Cloudbeds»**

Cloudbeds система управления отелем с акцентом на онлайн-бронирования и мультиканальные продажи.

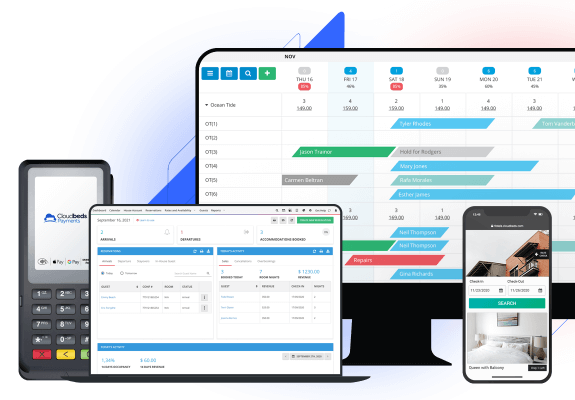


Рисунок 3 - Скриншот интерфейса программы «Cloudbeds»

**Преимущества:**

* интеграция с 300+ каналами бронирования;
* автоматическое управление тарифами;
* встроенный конструктор сайта отеля.

**Недостатки:**

* высокая ежемесячная стоимость;
* зависимость от скорости интернета;
* ограниченная кастомизация отчетности.

В результате анализа аналогов сформировалось видение конечного продукта «ГостиТут», который должен содержать: централизованное управление номерным фондом, безопасное хранение персональных данных гостей с шифрованием AES-256, формирование отчетов о гостях, контроль статусов уборки и занятости номеров, а также упрощенный интерфейс для работы администратора без специальной технической подготовки.

Изм.

Лист

№ докум.

Подпись

Дата

Лист

14

*090207.249755.ПЗ*

# **Техническое задание**

Техническое задание — это документ, который формулирует требования к результату работы (проекту, продукту, услуге) и служит основой для его разработки, выполнения и контроля. ТЗ содержит информацию о целях и задачах проекта, функциональных и нефункциональных требованиях, технических характеристиках, сроках и бюджете выполнения, а также другую необходимую информацию.

ГОСТы, используемые при подготовке ТЗ:

* ГОСТ 34.602-2020 «Техническое задание на создание автоматизированной системы» – содержит рекомендации по структуре и оформлению технического задания на создание автоматизированных систем [3].
* ГОСТ 19.201-78 «Техническое задание. Требования к содержанию и оформлению» – устанавливает порядок составления технического задания на создание программных средств [2].
* ГОСТ 7.32-2017 – «Отчет о научно-исследовательской работе. Структура и правила оформления» – устанавливает требования к структуре, содержанию и оформлению отчётов о научно-исследовательских работах [1].

Техническое задание приведено в Приложении А.

Изм.

Лист

№ докум.

Подпись

Дата

Лист

15

*090207.249755.ПЗ*

# **Проектирование информационной системы**

## **3.1 Моделирование информационной системы**

Моделирование информационной системы включает в себя: создание функциональной модели в нотации IDEF0, создание диаграммы потоков данных в нотации DFD, разработка диаграммы вариантов использования и создание диаграммы прецедентов.

## **3.1.1 Функциональное моделирование системы**

На рисунке 4 представлена контекстная диаграмма в нотации IDEF0.

Изм.

Лист

№ докум.

Подпись

Дата

Лист

16

*090207.249755.ПЗ*

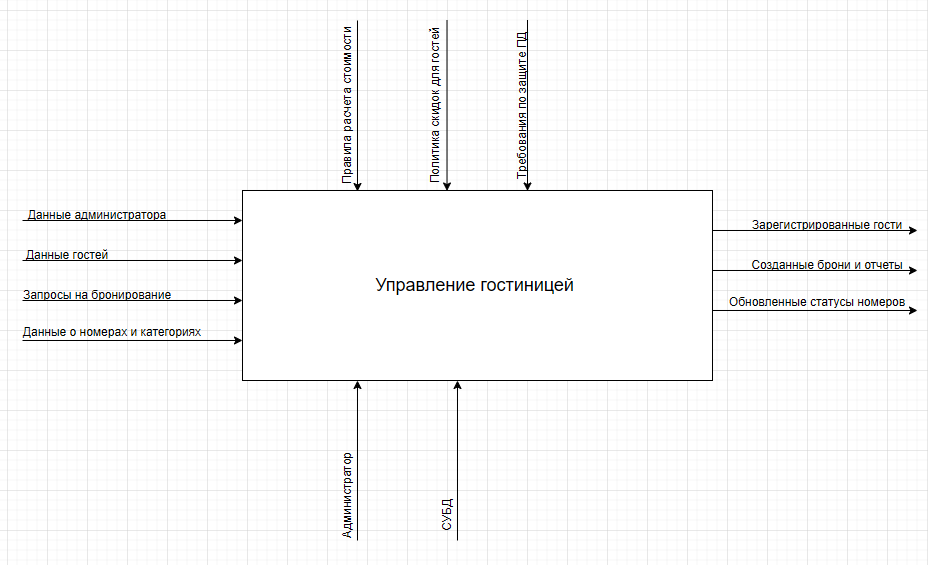


Рисунок 4 – Функциональная модель системы «ГостиТут» в нотации IDEF0 (контекстная диаграмма)

Основной процесс:

* управление гостиницей — центральный процесс, который включает в себя обработку данных о гостях, номерах, бронированиях и выполнение операций, связанных с управлением гостиницей.

Изм.

Лист

№ докум.

Подпись

Дата

Лист

17

*090207.249755.ПЗ*

Входные данные:

* данные администратора — информация о пользователе системы, который выполняет операции (например, учетные данные для авторизации);
* данные гостей — информация о клиентах гостиницы, включая их персональные данные и историю бронирований;
* запросы на бронирование — поступающие заявки на бронирование номеров от гостей;
* данные о номерах и категориях — информация о доступных номерах, их категориях, статусах и стоимости.

Управляющие воздействия:

* правила расчета стоимости — алгоритмы и параметры, определяющие стоимость проживания в зависимости от категории номера, длительности пребывания и других факторов;
* политика скидок для гостей — правила предоставления скидок, например, для постоянных клиентов или при длительном проживании;
* требования по защите персональных данных (ПД) — нормативные акты и внутренние правила, регулирующие обработку и хранение персональных данных гостей.

Выходные данные:

* зарегистрированные гости — обновленная информация о клиентах, зарегистрированных в системе;
* созданные брони и отчеты — данные о новых бронированиях, а также отчеты для анализа работы гостиницы;
* обновленные статусы номеров — информация о текущем состоянии номеров (свободен, занят, забронирован и т.д.).

Изм.

Лист

№ докум.

Подпись

Дата

Лист

18

*090207.249755.ПЗ*

Диаграмма декомпозиции в нотации IDEF0 представлена на рисунке 5.

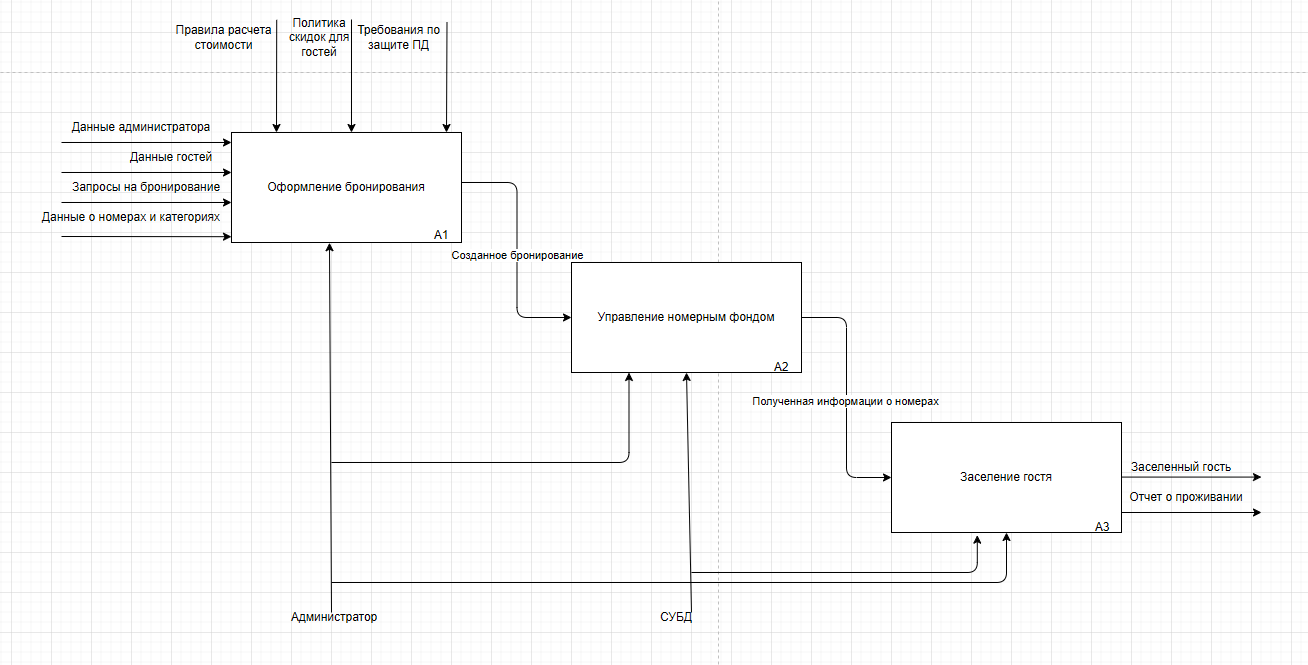


Рисунок 5 – Диаграмма декомпозиции функциональной модели системы «ГостиТут» в нотации IDEF0

Оформление бронирования (A1): на вход поступают данные администратора, гостей, запросы на бронирование, информация о номерах и категориях, а также правила расчета стоимости и политика скидок. На выходе формируется созданное бронирование.

Управление номерных фондом (A2): созданное бронирование передается в базу данных (СУБД), проверяется доступность номерного фонда. На выходе – полученная информация о номерах.

Заселение гостя (A3): на основе выполнения операции происходит заселение гостя. На выходе - данные о заселенном госте и отчет о проживании.

## **3.1.2 Моделирование потоков данных**

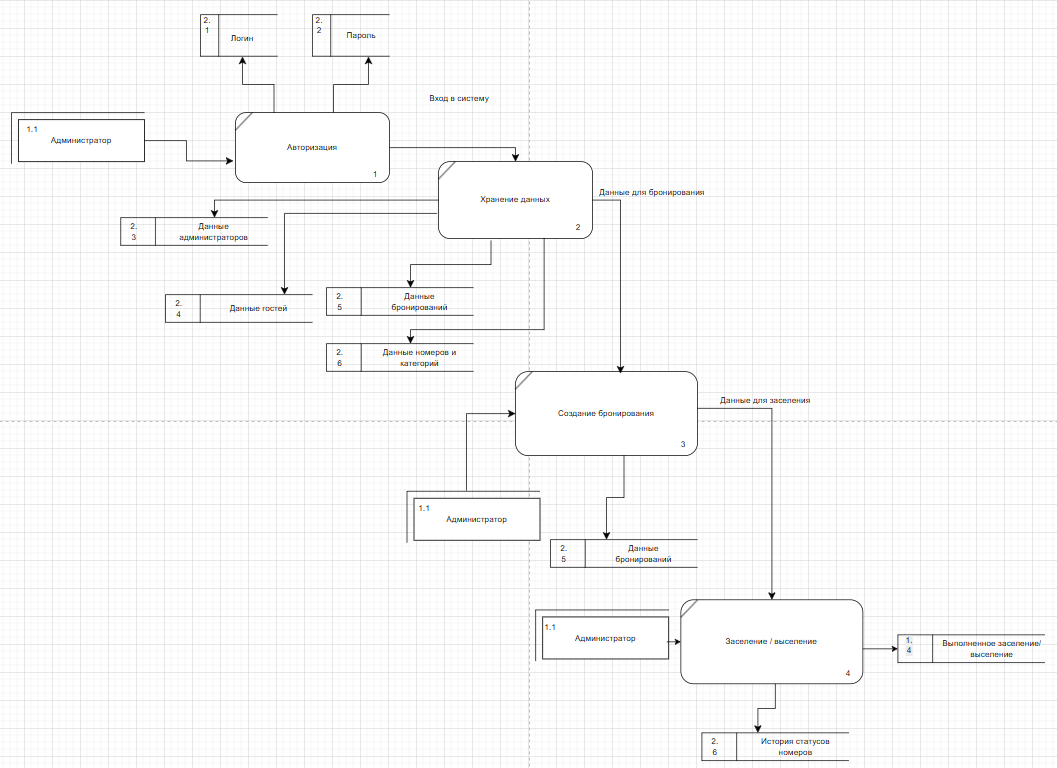


Рисунок 6 – Диаграмма потоков данных в нотации DFD

Авторизация (1):

Изм.

Лист

№ докум.

Подпись

Дата

Лист

19

*090207.249755.ПЗ*

* входные данные: логин и пароль администратора;
* администратор проходит авторизацию для доступа к системе;
* на выходе: подтверждение входа в систему.

Хранение данных (2):

* система хранит данные администраторов, гостей, номеров и категорий;
* эти данные используются для создания бронирований и заселения гостей.

Создание бронирования (3):

* входные данные: информация о гостях, номерах и категориях, а также данные администратора;
* на выходе: данные о созданном бронировании, которые передаются в систему для дальнейшей обработки.

Заселение гостей (4):

* входные данные: данные о бронировании и информация от администратора;
* на выходе: данные о заселении гостей и журнал статусов номеров.

Выполнение заселения (5):

* на основе данных о заселении выполняется операция заселения гостей.

## **3.1.3 Варианты использования системы**

Диаграмма вариантов использования (рисунок 7) представляет собой модель взаимодействия администратора с информационной системой для гостиницы. Она описывает основные функции системы и действия, которые может выполнять администратор. Ниже приведено текстовое описание элементов диаграммы.

**Основной участник:**

Администратор - основной пользователь системы, который выполняет все действия, связанные с управлением гостиницей.

**Основные варианты использования.**

Вход в систему:

* администратор авторизуется в системе для получения доступа к функционалу.

Изм.

Лист

№ докум.

Подпись

Дата

Лист

20

*090207.249755.ПЗ*

Выход из системы:

* администратор завершает работу с системой, выходя из нее.

Просмотр списка гостей:

* администратор может просматривать список всех зарегистрированных гостей.

Расширяется следующими действиями:

* добавить гостя — добавление нового гостя в систему;
* изменить данные гостя — редактирование информации о существующем госте;
* удалить гостя — удаление информации о госте из системы.

Просмотр списка номеров:

* администратор может просматривать список всех номеров гостиницы.

Расширяется следующими действиями:

* добавить номер/категорию — добавление нового номера или категории номеров;
* удалить номер/категорию — удаление существующего номера или категории;
* редактировать номер/категорию — изменение информации о номере или категории.

Просмотр бронирований:

* администратор может просматривать список всех текущих бронирований.

Изм.

Лист

№ докум.

Подпись

Дата

Лист

21

*090207.249755.ПЗ*

Расширяется следующими действиями:

* добавление брони — создание нового бронирования;
* удаление брони — удаление существующего бронирования.

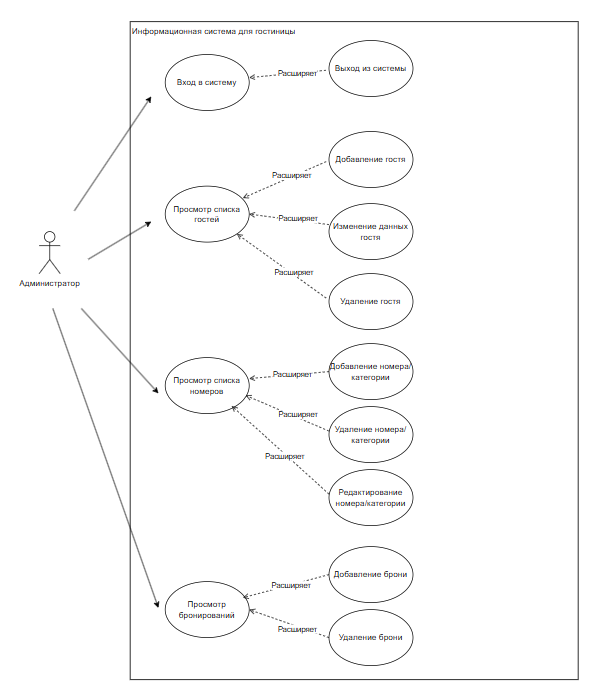


Рисунок 7 – Диаграмма прецендентов

## **3.2 Проектирование базы данных**

Проектирование базы данных включает в себя следующие этапы: составления набора сущностей и описание их свойств, описание связей между сущностями, построение логической и физической модели данных в нотации IDEF1X и описание атрибутов сущностей базы данных.

Изм.

Лист

№ докум.

Подпись

Дата

Лист

22

*090207.249755.ПЗ*

В таблице 1 представлен набор сущностей и описание их свойств.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Сущность | Свойство | Тип | Уникальность | Обязательность | Ограничения/примечания  Изм.  Лист  № докум.  Подпись  Дата  Лист  23  *090207.249755.ПЗ* |
| admins | id | Serial | Да | Да | Первичный ключ (PK), автоинкремент |
| username | Text | Да | Да | Уникальный логин для входа в систему |
| password\_hash | Text | Нет | Да | Хэш пароля, вычисленный по алгоритму SHA-256 |
| first\_name | Text | Нет | Нет | Имя администратора |
| last\_name | Text | Нет | Нет | Фамилия администратора |
| created\_at | Timestamptz | Нет | Да | Метка времени создания записи (по умолчанию now()) |
| room\_types | id | Serial | Да | Да | Первичный ключ (PK), автоинкремент |
| name | Text | Да | Да | Название типа (например, "Стандарт", "Комфорт+", "Люкс") |
| description | Text | Нет | Нет | Текстовое описание категории |
| base\_price | Numeric(10,2) | Нет | Да | Базовая стоимость проживания за сутки для данного типа номера. Значение по умолчанию: 0 |
| rooms | id | Serial | Да | Да | Первичный ключ (PK), автоинкремент |
| number | Text | Да | Да | Уникальный номер комнаты (например, "2-101") |
| type\_id | Integer | Нет | Нет | Внешний ключ (FK) на room\_types(id). Тип номера. При удалении типа устанавливается в NULL |
| floor | Integer | Нет | Нет | Этаж, на котором расположен номер |
| max\_guests | Integer | Нет | Да | Максимальное количество гостей. Значение по умолчанию: 2 |
| status | Text | Нет | Да | Текущий статус номера. Допустимые значения: 'свободен', 'занят', 'уборка', 'бронь'. Значение по умолчанию: 'свободен' |
| created\_at | Timestamptz | Нет | Да | Метка времени создания записи |
| guests | id | Serial | Да | Да | Первичный ключ (PK), автоинкремент |
| first\_name | Text | Нет | Да | Имя гостя |
| last\_name | Text | Нет | Да | Фамилия гостя |
| phone | Text | Нет | Нет | Контактный телефон |
| email | Text | Нет | Нет | Адрес электронной почты |
| passport\_encrypted | Bytea | Нет | Нет | Зашифрованные паспортные данные с использованием AES-256. Хранятся в бинарном виде |
| passport\_iv | Bytea | Нет | Нет | Вектор инициализации (IV) для алгоритма AES-256. Необходим для расшифровки |
| discount | Numeric(5,2) | Нет | Да | Персональная скидка гостя (в процентах или фиксированной сумме). Значение по умолчанию: 0 |
| created\_at | Timestamptz | Нет | Да | Метка времени создания записи  Изм.  Лист  № докум.  Подпись  Дата  Лист  24  *090207.249755.ПЗ* |
| bookings | id | Serial | Да | Да | Первичный ключ (PK), автоинкремент |
| room\_id | Integer | Нет | Нет | Внешний ключ (FK) на rooms(id). Забронированный номер. При удалении номера удаляется и бронь |
| guest\_id | Integer | Нет | Нет | Внешний ключ (FK) на guests(id). Гость, сделавший бронь. При удалении гостя удаляется и бронь |
| created\_by | Integer | Нет | Нет | Внешний ключ (FK) на admins(id). Администратор, оформивший бронирование |
| date\_from | Date | Нет | Да | Дата начала проживания (заезда) |
| date\_to | Date | Нет | Да | Дата окончания проживания (выезда) |
| status | Text | Нет | Да | Статус бронирования. Значение по умолчанию: «active» |
| total\_price | Numeric(12,2) | Нет | Да | Общая расчетная стоимость проживания. Значение по умолчанию: 0 |
| created\_at | Timestamptz | Нет | Да | Метка времени создания брони. |
| room\_status\_history | id | Serial | Да | Да | Первичный ключ (PK), автоинкремент |
| room\_id | Integer | Нет | Нет | Внешний ключ (FK) на rooms(id). Номер, для которого изменился статус |
| old\_status | Text | Нет | Нет | Статус номера до изменения |
| new\_status | Text | Нет | Нет | Новый статус номера |
| changed\_by | Integer | Нет | Нет | Внешний ключ (FK) на admins(id). Администратор, изменивший статус |
| changed\_at | Timestamptz | Нет | Да | Метка времени изменения статуса |

База данных включает в себя следующие связи между сущностями:

* admins – bookings: Один ко многим (1:M). Один администратор может оформить много бронирований, но каждое бронирование оформлено одним администратором (created\_by);

Изм.

Лист

№ докум.

Подпись

Дата

Лист

25

*090207.249755.ПЗ*

* admins – room\_status\_history: Один ко многим (1:M). Один администратор может инициировать много изменений статусов, но каждая запись в истории относится к одному администратору (changed\_by);

Изм.

Лист

№ докум.

Подпись

Дата

Лист

26

*090207.249755.ПЗ*

* room\_types – rooms: Один ко многим (1:M). Один тип номера может быть присвоен многим номерам, но каждый номер имеет один тип (type\_id). Связь с ON DELETE SET NULL;
* rooms – bookings: Один ко многим (1:M). Один номер может быть забронирован много раз (в разные даты), но каждое бронирование относится к одному номеру (room\_id). Связь с ON DELETE CASCADE;
* rooms – room\_status\_history: Один ко многим (1:M). Для одного номера может быть много записей в истории изменений статуса, но каждая запись относится к одному номеру (room\_id);
* guests – bookings: Один ко многим (1:M). Один гость может иметь несколько бронирований, но каждое бронирование сделано одним гостем (guest\_id). Связь с ON DELETE CASCADE.

На рисунке 8 представлена логическая модель данных в нотации IDEF1X.

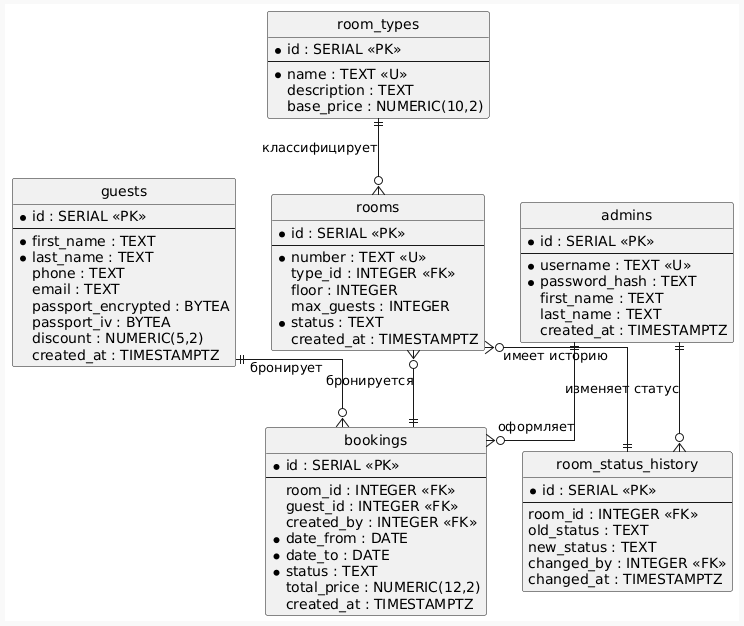


Рисунок 8 – Логическая модель данных в нотации IDEF1X

На рисунке 9 представлена физическая модель данных в нотации IDEF1X.

Изм.

Лист

№ докум.

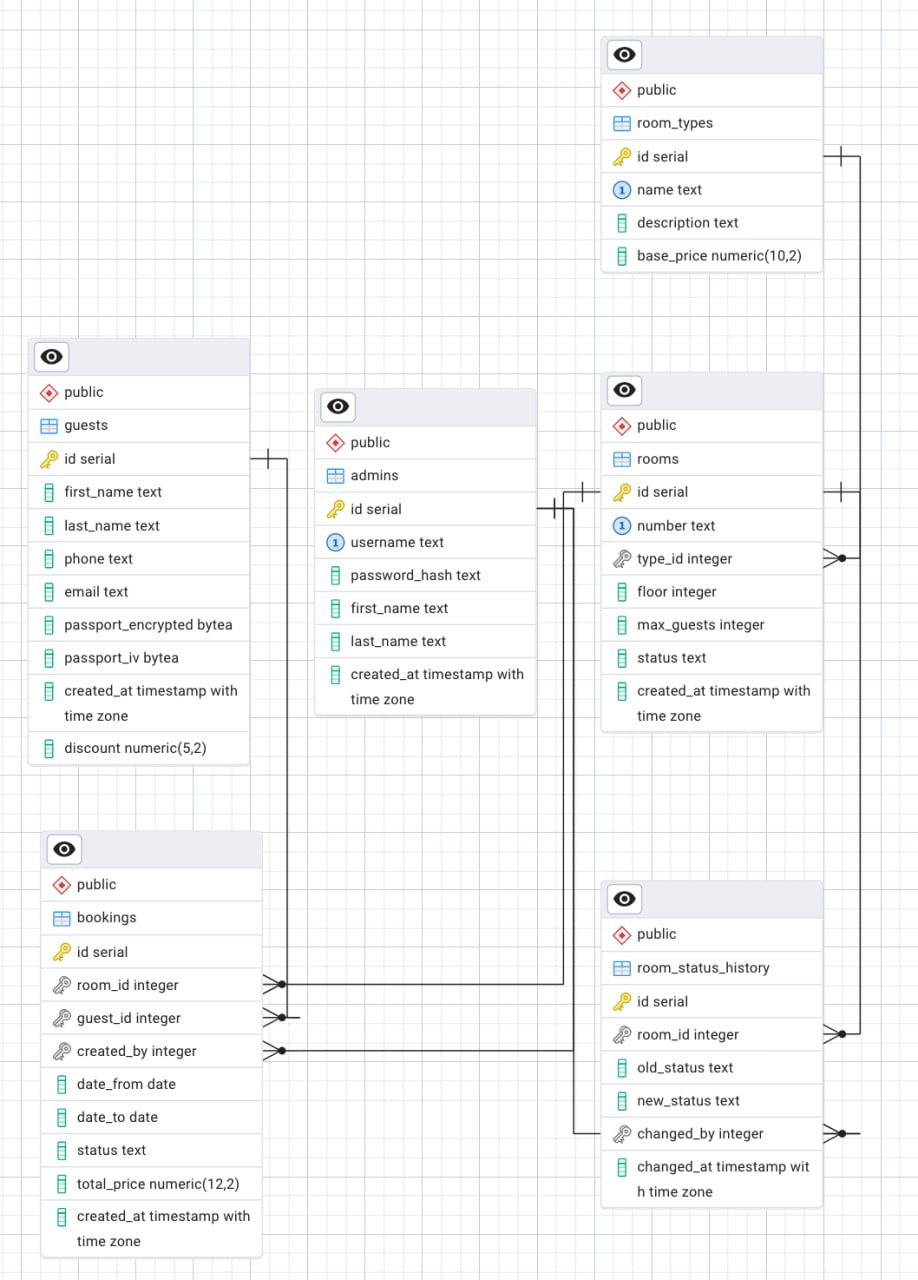
Подпись

Дата

Лист

27

*090207.249755.ПЗ*



Изм.

Лист

№ докум.

Подпись

Дата

Лист

28

*090207.249755.ПЗ*

Рисунок 9 – Физическая модель данных в нотации IDEF1X

Целостность сущностей (Entity Integrity):

* каждая таблица имеет первичный ключ id с типом SERIAL PRIMARY KEY, который гарантирует уникальность каждой записи и не допускает значений NULL;
* admins.username — логин администратора должен быть уникальным в системе;
* rooms.number — номер комнаты должен быть уникальным в гостинице;
* room\_types.name — название категории номера должно быть уникальным.

Доменная целостность (Domain Integrity).

Для критичных атрибутов, представляющих перечислимые значения, применяются проверочные ограничения (CHECK constraints), реализованные на уровне логики приложения или явно задаваемые при вставке/обновлении. Например:

Изм.

Лист

№ докум.

Подпись

Дата

Лист

29

*090207.249755.ПЗ*

* поле rooms.status предполагает строго определённый набор значений: 'свободен', 'занят', 'уборка', 'бронь';
* поле bookings.status также управляется конечным набором статусов (например, 'active', 'cancelled', 'completed').

Использование типов данных с явным указанием точности (NUMERIC(10,2), NUMERIC(5,2)) гарантирует корректное хранение финансовых данных.

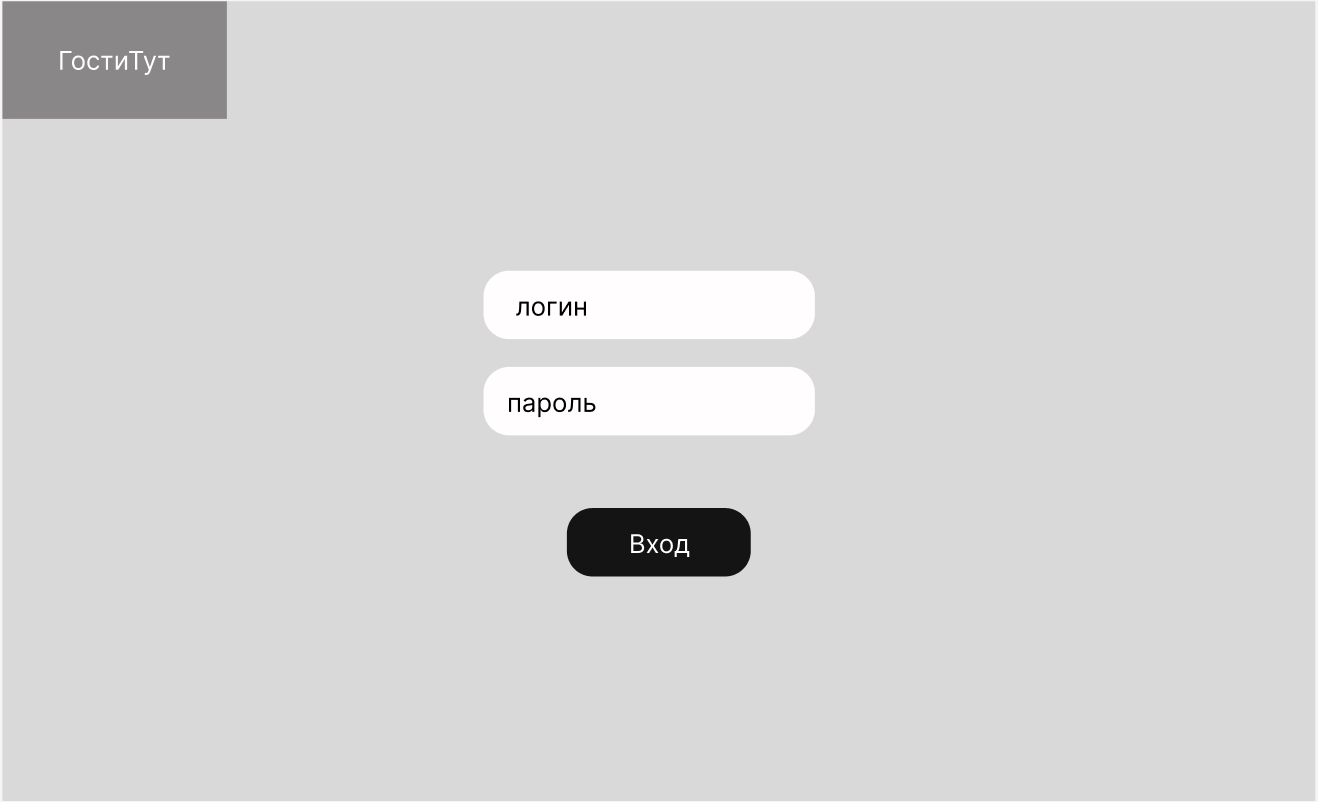
Целостность ссылок (Referential Integrity):

* все внешние ключи (FOREIGN KEY) явно объявлены в определениях таблиц, что гарантирует согласованность данных между связанными сущностями;
* в таблице bookings: при удалении гостя (guest\_id) или номера (room\_id) связанные с ними бронирования также удаляются, что предотвращает появление «висячих» ссылок;
* для связи, где удаление родительской сущности не должно приводить к удалению дочерней, используется правило ON DELETE SET NULL (например, rooms.type\_id при удалении категории room\_types).

Для реализации базы данных информационной системы «ГостиТут» была выбрана реляционная СУБД PostgreSQL. Данный выбор обусловлен её надежностью, соответствием промышленным стандартам, поддержкой сложных типов данных.

## **3.3 Проектирование интерфейса информационной системы**

На рисунках 10 – 14 представлены макеты основных экранных форм информационной системы.



Изм.

Лист

№ докум.

Подпись

Дата

Лист

30

*090207.249755.ПЗ*

Рисунок 10 – Макет меню входа

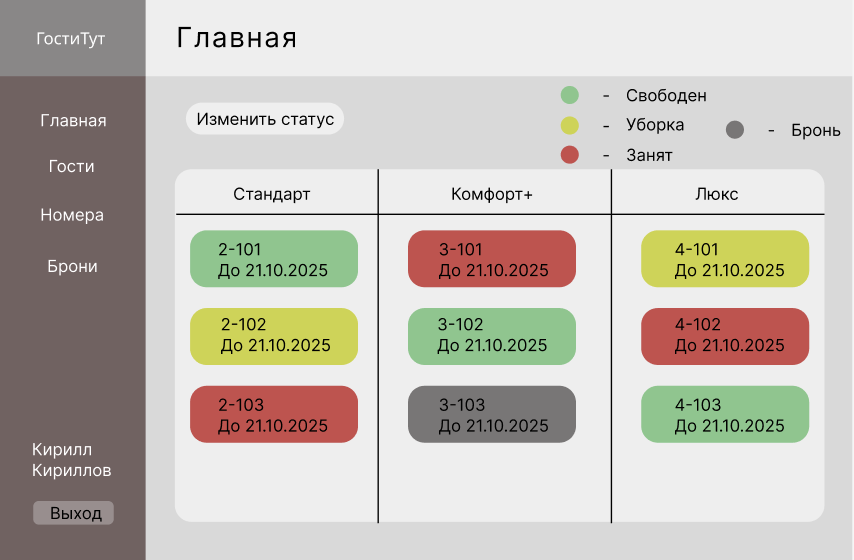
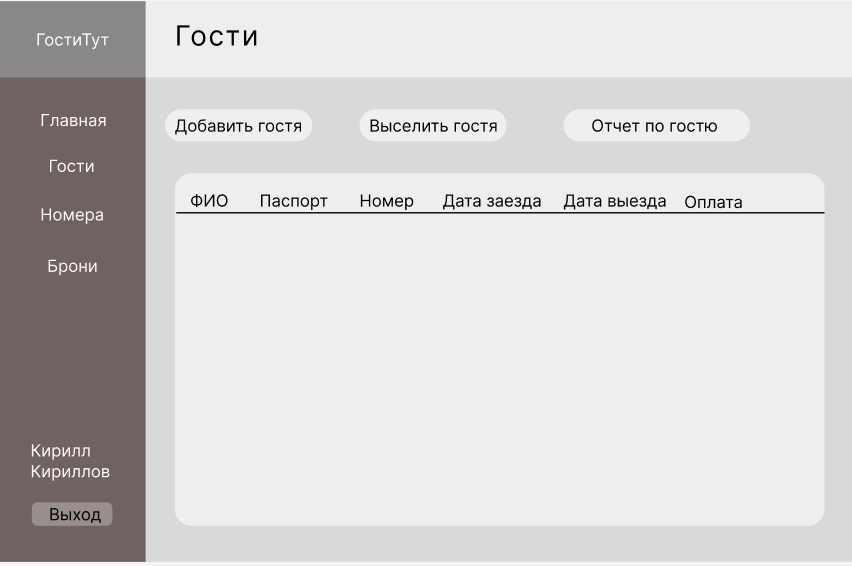


Рисунок 11 – Макет раздела «Главная»



Изм.

Лист

№ докум.

Подпись

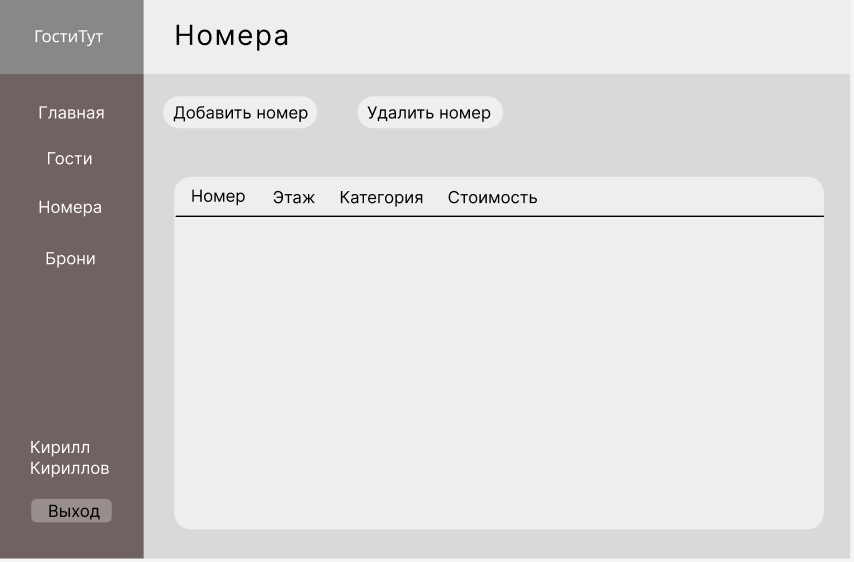
Дата

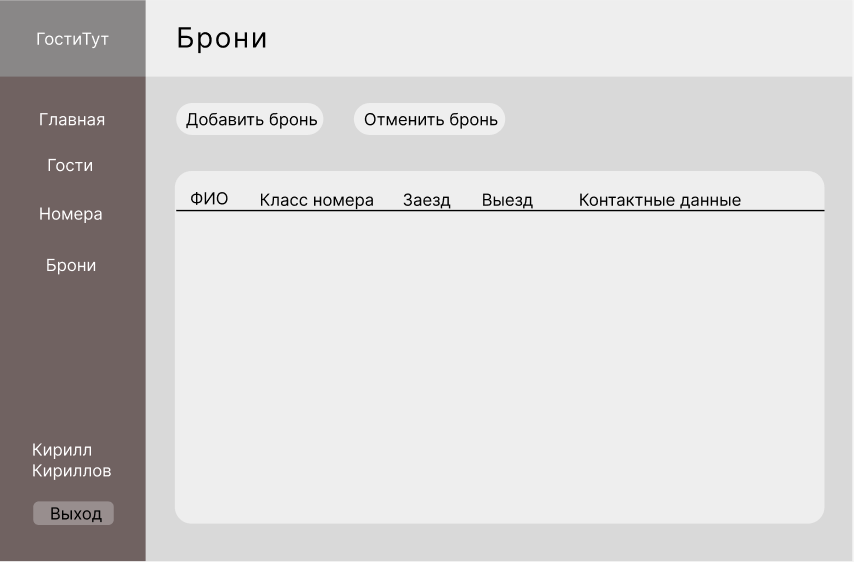
Лист

31

*090207.249755.ПЗ*

Рисунок 12 – Макет раздела «Гости»

  
Рисунок 13 – Макет раздела «Номера»



Изм.

Лист

№ докум.

Подпись

Дата

Лист

32

*090207.249755.ПЗ*

Рисунок 14 – Макет раздела «Брони»

В результате подраздела 3.3 были разработаны макеты для ИС при помощи Figma.

# **Реализация информационной системы**

Изм.

Лист

№ докум.

Подпись

Дата

Лист

33

*090207.249755.ПЗ*

В данном разделе содержится информация о реализации информационной системы.

## **4.1 Пользовательский интерфейс системы**

Интерфейс информационной системы «ГостиТут» спроектирован в соответствии с требованиями технического задания и макетами, представленными в разделе 3.3. Реализация выполнена с использованием библиотеки PyQt6.

На рисунках 15-19 представлены основные экранные формы системы.

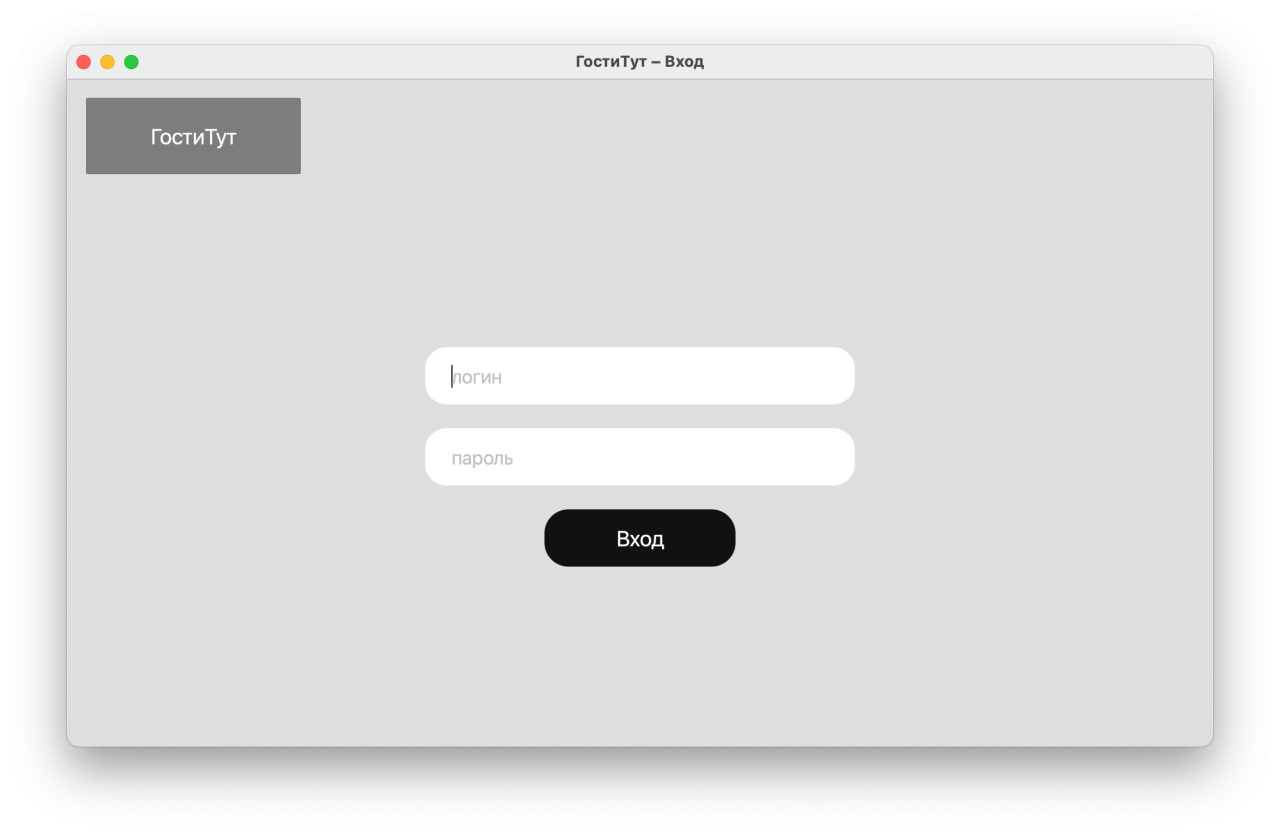


Рисунок 15 – Окно входа в системы

Окно авторизации (рисунок 15) - обеспечивает безопасный вход администратора в систему с проверкой логина и хешированного пароля.

Листинг 1 – Фрагмент кода окна авторизации

class LoginWindow(QWidget):

def \_\_init\_\_(self):

super().\_\_init\_\_()

self.setWindowTitle("ГостиТут – Вход")

self.setGeometry(200, 120, 960, 560)

Изм.

Лист

№ докум.

Подпись

Дата

Лист

34

*090207.249755.ПЗ*

# Поля логина и пароля

self.login\_input = QLineEdit()

self.login\_input.setPlaceholderText("логин")

self.password\_input = QLineEdit()

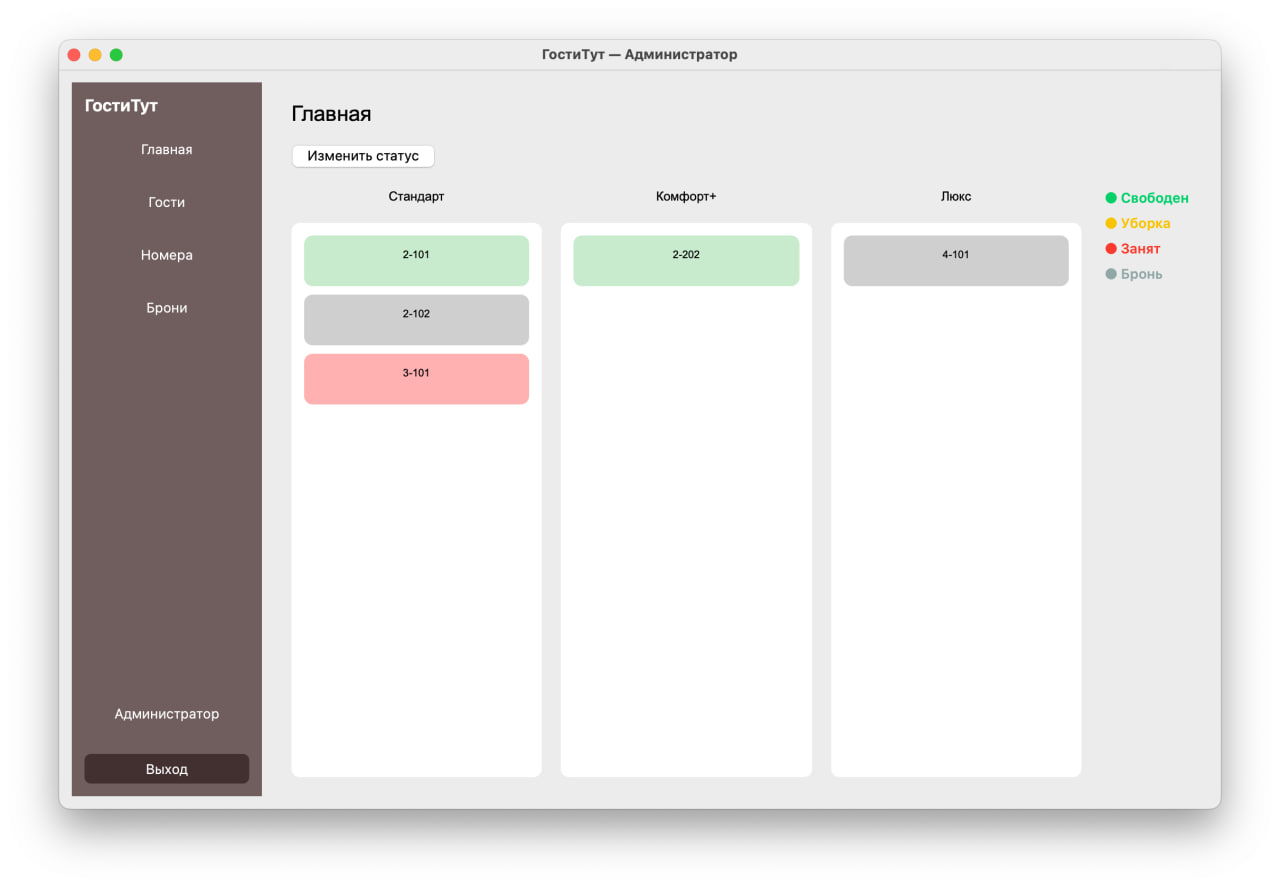
self.password\_input.setPlaceholderText("пароль")

self.password\_input.setEchoMode(QLineEdit.EchoMode.Password)

# Кнопка входа

self.btn = QPushButton("Вход")

self.btn.clicked.connect(self.check\_login)

Рисунок 16 – Окно раздела «Главная»

Главная страница (рисунок 16) - представляет визуальное отображение номерного фонда с цветовой индикацией статусов и возможностью управления статусами номеров.

Листинг 2 – Фрагмент кода раздела «Главная»

# Создание плиток для каждого номера

for r in rooms:

number, status, rid = r

tile = RoomTile(str(number) + "\n", status, rid)

tile.clicked.connect(self.on\_tile\_clicked)

self.room\_tiles.append(tile)

card.addWidget(tile)

card.addStretch()

container = QWidget()

container.setLayout(card)

container.setMinimumHeight(320)

container.setStyleSheet("background:white; border-radius:8px; padding:8px;")

box.addWidget(container)

columns\_h.addLayout(box)

Изм.

Лист

№ докум.

Подпись

Дата

Лист

35

*090207.249755.ПЗ*

# Легенда цветовых статусов

legend\_col = QVBoxLayout()

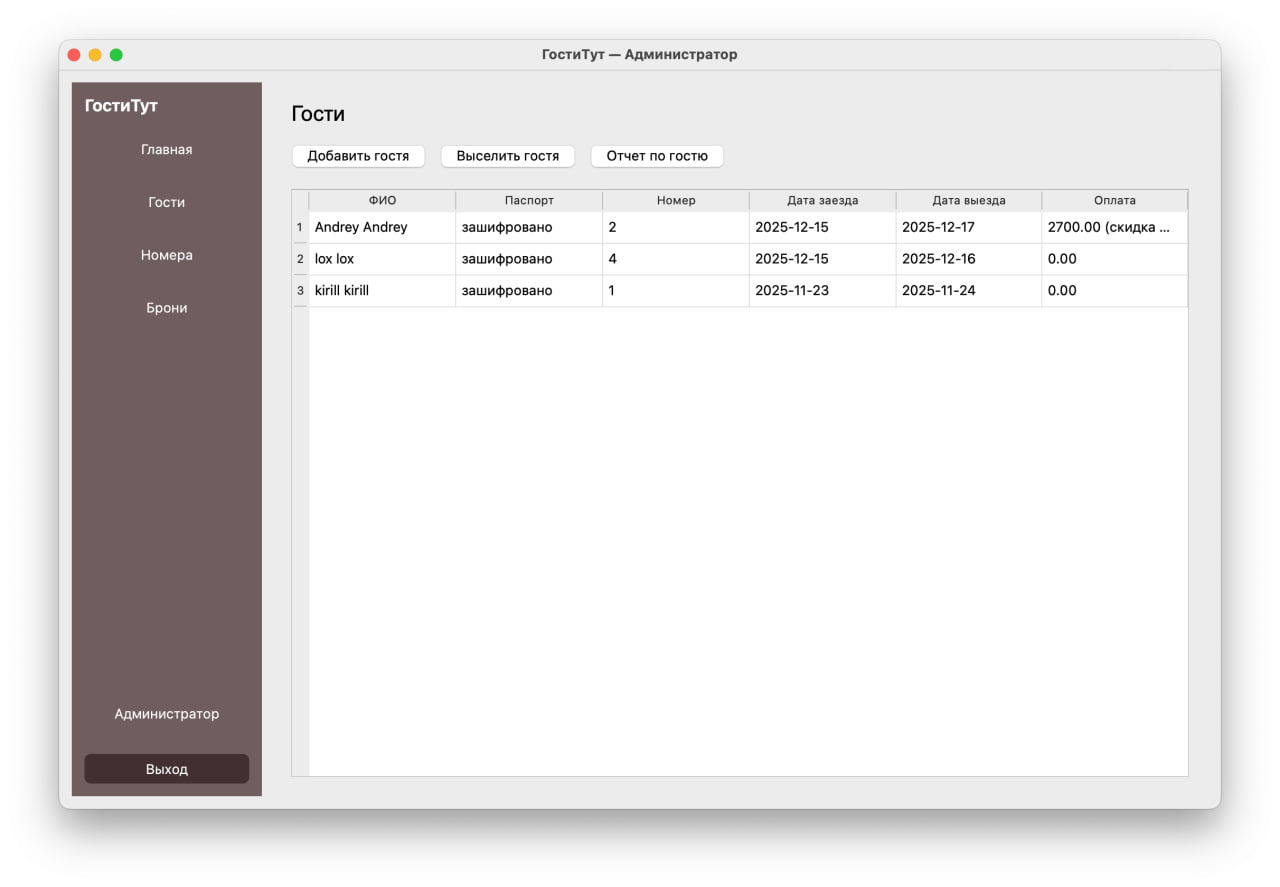
for text, color in (

("Свободен", "#2ecc71"),

("Уборка", "#f1c40f"),

("Занят", "#e74c3c"),

("Бронь", "#95a5a6"),

Рисунок 17 – Окно раздела «Гости»

Изм.

Лист

№ докум.

Подпись

Дата

Лист

36

*090207.249755.ПЗ*

Раздел "Гости" (рисунок 17) - позволяет администратору управлять информацией о гостях, включая добавление, редактирование, выселение и формирование отчетов.

Листинг 3 – Фрагмент кода раздела «Гости»

class MainWindow(QMainWindow):

#Главное окно администратора

def build\_guests\_page(self):

#Создание страницы управления гостями

w = QWidget()

v = QVBoxLayout()

v.setContentsMargins(18, 18, 18, 18)

# Заголовок

title = QLabel("Гости")

title.setFont(TITLE\_FONT)

v.addWidget(title)

# Панель управления

btn\_h = QHBoxLayout()

btn\_add = QPushButton("Добавить гостя")

btn\_checkout = QPushButton("Выселить гостя")

btn\_report = QPushButton("Отчет по гостю")

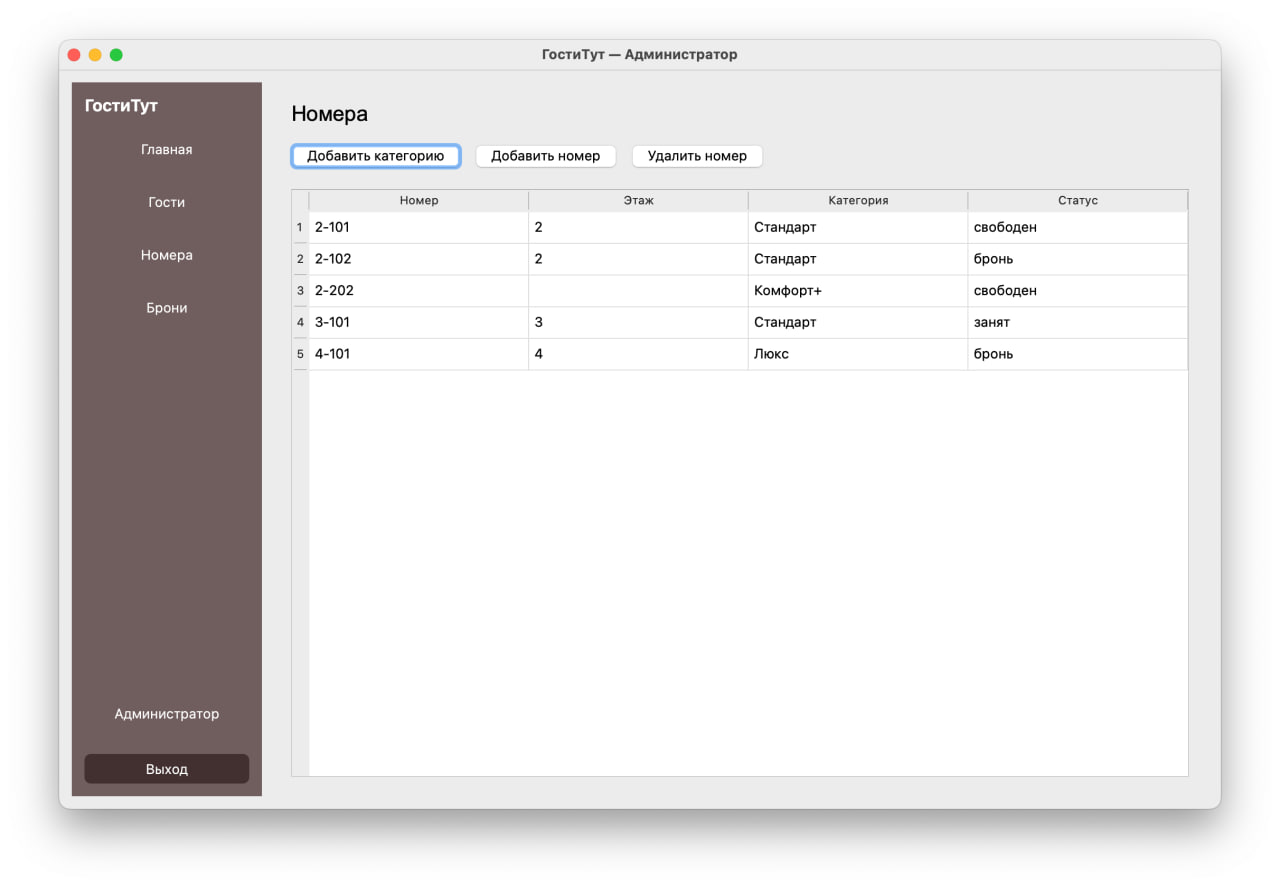
btn\_h.addWidget(btn\_add)

btn\_h.addWidget(btn\_checkout)

btn\_h.addWidget(btn\_report)

btn\_h.addStretch()

v.addLayout(btn\_h)



Изм.

Лист

№ докум.

Подпись

Дата

Лист

37

*090207.249755.ПЗ*

Рисунок 18 – Окно раздела «Номера»

Раздел "Номера" (рисунок 18) - предоставляет инструменты для управления номерным фондом: добавление/удаление номеров, редактирование категорий.

Листинг 4 – Фрагмент кода раздела «Номера»

class MainWindow(QMainWindow):

#Главное окно администратора

def build\_rooms\_page(self):

#Создание страницы управления номерами

w = QWidget()

v = QVBoxLayout()

v.setContentsMargins(18, 18, 18, 18)

# Заголовок

title = QLabel("Номера")

title.setFont(TITLE\_FONT)

v.addWidget(title)

# Панель управления

btn\_h = QHBoxLayout()

btn\_add\_cat = QPushButton("Добавить категорию")

btn\_add\_room = QPushButton("Добавить номер")

btn\_del\_room = QPushButton("Удалить номер")

btn\_h.addWidget(btn\_add\_cat)

Изм.

Лист

№ докум.

Подпись

Дата

Лист

38

*090207.249755.ПЗ*

btn\_h.addWidget(btn\_add\_room)

btn\_h.addWidget(btn\_del\_room)

btn\_h.addStretch()

v.addLayout(btn\_h)

# Таблица номеров

self.rooms\_table = QTableWidget(0, 4)

self.rooms\_table.setHorizontalHeaderLabels(

["Номер", "Этаж", "Категория", "Статус"]

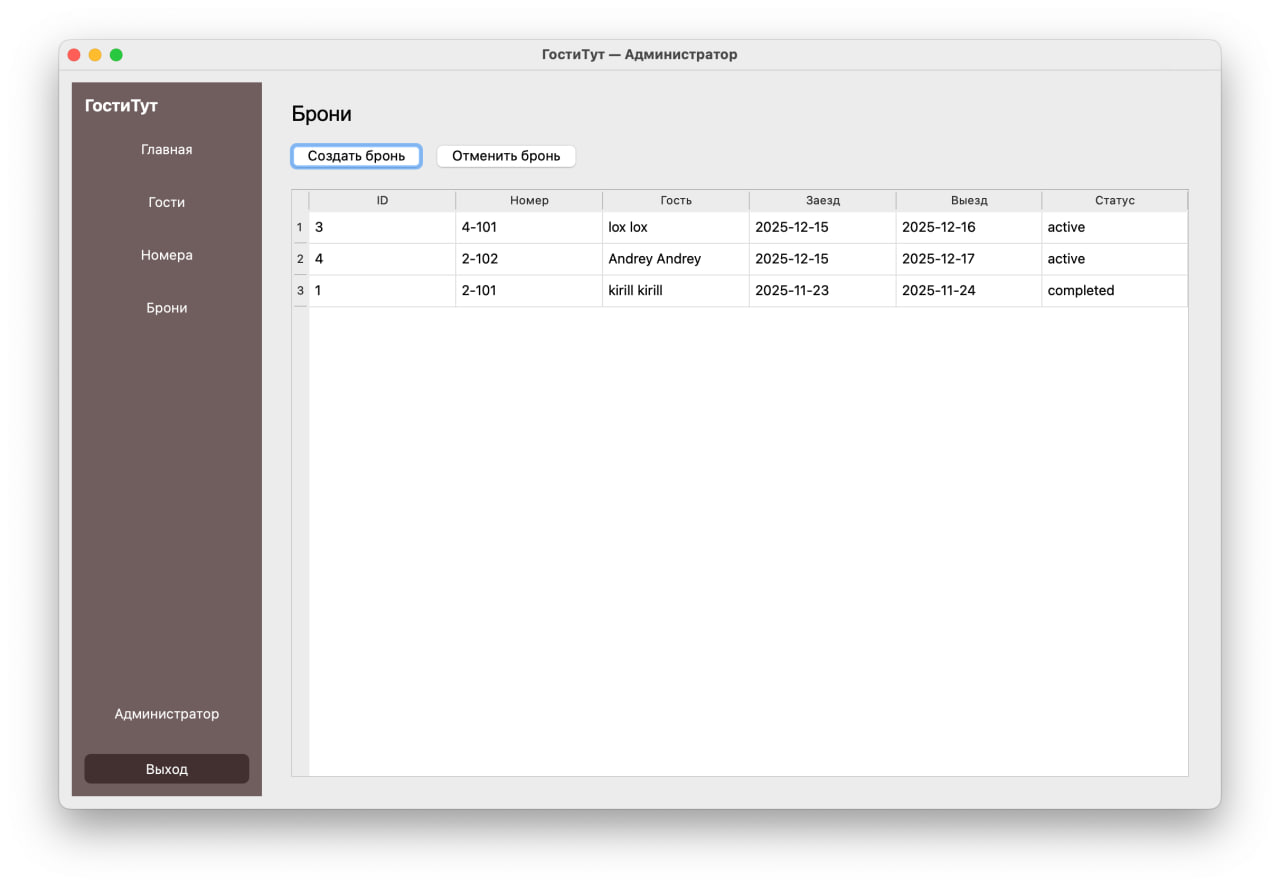


Рисунок 19 – Окно раздела «Брони»

Раздел "Брони" (рисунок 19) - реализует функционал бронирования номеров, включая создание, редактирование и отмену бронирований.

Листинг 5 – Фрагмент кода раздела «Брони»

class MainWindow(QMainWindow):

#Главное окно администратора

def build\_bookings\_page(self):

#Создание страницы управления бронированиями

Изм.

Лист

№ докум.

Подпись

Дата

Лист

39

*090207.249755.ПЗ*

w = QWidget()

v = QVBoxLayout()

v.setContentsMargins(18, 18, 18, 18)

# Заголовок

title = QLabel("Брони")

title.setFont(TITLE\_FONT)

v.addWidget(title)

# Панель управления

btn\_h = QHBoxLayout()

btn\_create = QPushButton("Создать бронь")

btn\_cancel = QPushButton("Отменить бронь")

btn\_h.addWidget(btn\_create)

btn\_h.addWidget(btn\_cancel)

btn\_h.addStretch()

v.addLayout(btn\_h)

# Таблица бронирований

self.bookings\_table = QTableWidget(0, 6)

self.bookings\_table.setHorizontalHeaderLabels(

["ID", "Номер", "Гость", "Заезд", "Выезд", "Статус"]

## **4.2 Реализация программного кода**

Информационная система для управления гостиницей «ГостиТут» имеет 6 моделей данных: admins, room\_types, rooms, guests, bookings, room\_status\_history, с которыми реализованы основные функции взаимодействия с базой данных.

Изм.

Лист

№ докум.

Подпись

Дата

Лист

40

*090207.249755.ПЗ*

Для отображения номеров и их статусов используется функция def build\_main\_page(self).

Исходный код функции def check\_login(self) представлен на рисунке 20.



Рисунок 20 – Функция входа в систему

Функция проверяет логин и пароль в базе данных. Пароль хешируется алгоритмом SHA-256, затем выполняется запрос к таблице admins. При успешной проверке открывается главное окно приложения, иначе выводится сообщение об ошибке.

Изм.

Лист

№ докум.

Подпись

Дата

Лист

41

*090207.249755.ПЗ*

Для сохранения данных о госте в базе данных используется функция def save() внутри dialog\_edit\_guest. Для остальных страниц используются функции, работающие подобным же образом.

Исходный код функции def save() внутри dialog\_edit\_guest представлен на рисунке 21.



Рисунок 21 – Функция сохранения данных

Код обрабатывает заполненные поля формы редактирования гостя. При нажатии кнопки «Сохранить» данные валидируются, паспортные данные шифруются алгоритмом AES, затем выполняется SQL-запрос UPDATE для обновления записи в таблице guests.

Изм.

Лист

№ докум.

Подпись

Дата

Лист

42

*090207.249755.ПЗ*

Для удаления записи из базы данных используется функция def action\_delete\_room(self). Для остальных страниц используются функции, работающие подобным же образом.

Исходный код функции def action\_delete\_room(self) представлен на рисунке 22.



Изм.

Лист

№ докум.

Подпись

Дата

Лист

43

*090207.249755.ПЗ*

Рисунок 22 – Функция удаления данных

Функция проверяет выделение строки в таблице, показывает диалоговое окно с подтверждением удаления и опцией удаления категории. Выполняется SQL-запрос DELETE по уникальному полю id, при необходимости удаляется категория, если у неё не осталось номеров. После удаления таблица обновляется.

Для создания главной страницы с отображением номеров используется функция def build\_main\_page(self). Аналогичные функции используются для создания других страниц интерфейса.

Исходный код функции def build\_main\_page(self) представлен на рисунке 23.

Изм.

Лист

№ докум.

Подпись

Дата

Лист

44

*090207.249755.ПЗ*



Рисунок 23 – Функция создания страницы «Главная»

Функция создаёт главную страницу приложения с отображением номеров по категориям. Номера отображаются в виде плиток с цветовой индикацией статуса. Для каждой категории создаётся отдельная колонка, в которой размещаются номера соответствующего типа.

Для формирования отчета используется функция def action\_guest\_report(self).

Исходный код функции def action\_guest\_report(self) представлен на рисунке 24.

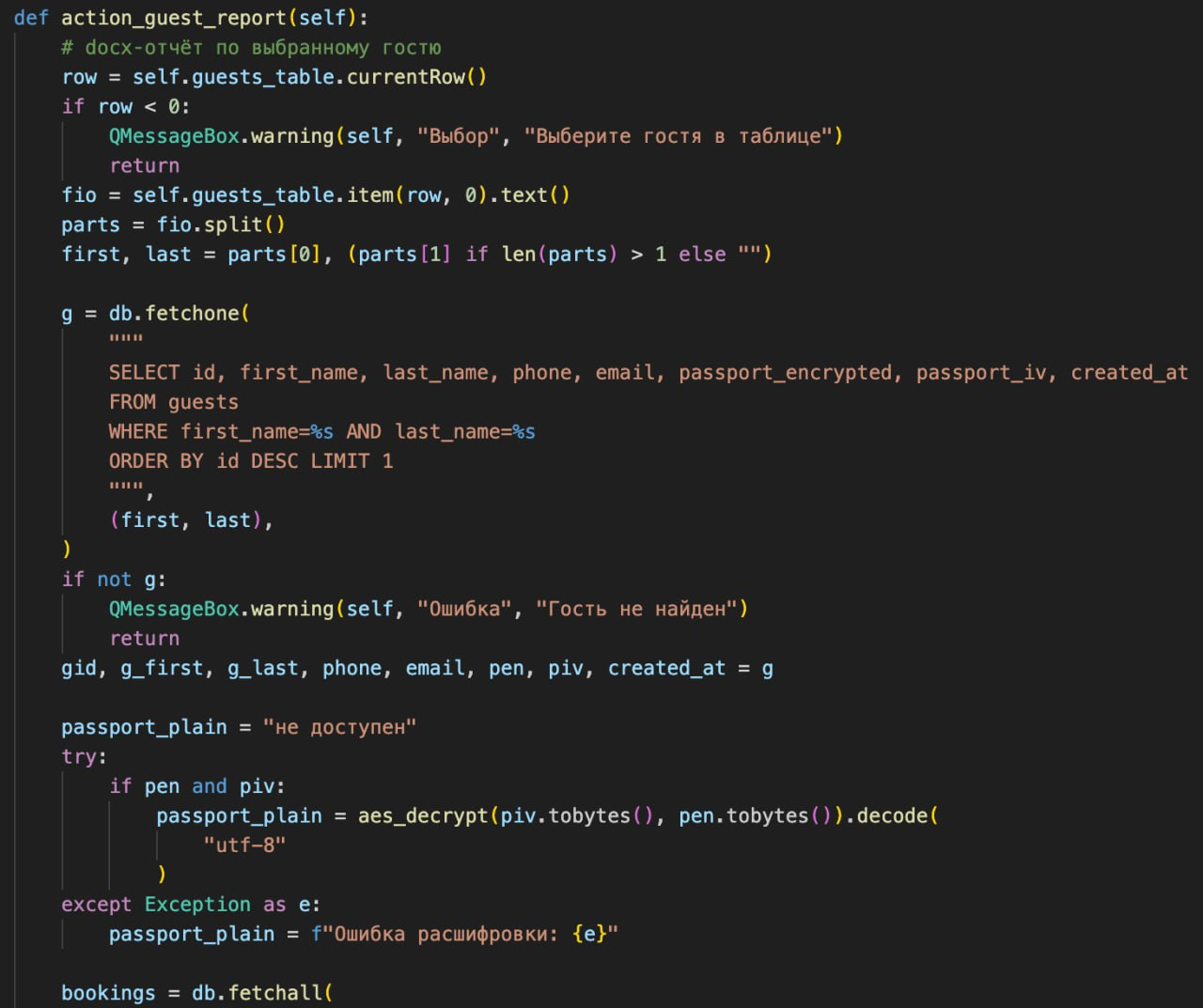


Рисунок 24 – Функция создания отчета

Функция формирует отчет в формате DOCX по выбранному гостю. Из базы данных извлекаются данные гостя и его бронирования, паспортные данные расшифровываются. Отчет включает информацию о госте и таблицу бронирований, затем сохраняется в папку reports с уникальным именем файла.

Изм.

Лист

№ докум.

Подпись

Дата

Лист

45

*090207.249755.ПЗ*

# **ЗАКЛЮЧЕНИЕ**

В ходе выполнения курсового проекта была разработана и подробно описана информационная система, предназначенная для автоматизации деятельности гостиницы. Проект охватывает этапы от анализа предметной области и проектирования до программной реализации, предоставляя комплексный подход к управлению ключевыми бизнес-процессами в гостиничной сфере.

Разработанная информационная система целенаправленно соответствует потребностям целевого сегмента, обеспечивая эффективное управление номерным фондом, обработку бронирований, безопасный учет данных гостей с применением криптографического шифрования и формирование необходимой отчетности.

В рамках проектирования системы были получены и применены практические навыки моделирования и построения DFD-диаграмм для анализа потоков информации, Use Case-диаграмм для выявления функциональных требований и взаимодействия с пользователем, а также создание логических и физических моделей базы данных в нотации IDEF1X.

Результатом работы является функционирующее приложение, реализующее заявленный в техническом задании функционал, использующее СУБД PostgreSQL.

Изм.

Лист

№ докум.

Подпись

Дата

Лист

46

*090207.249755.ПЗ*

# **СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. ГОСТ 7.32–2017. Система стандартов по информации, библиотечному и издательскому делу. Отчет о научно-исследовательской работе. Структура и правила оформления: утвержден и введен в действие Приказом Росстандарта от 31.05.2017 № 493-ст: дата введения 01.07.2018. – Москва: Стандартинформ, 2018. – 20 с.
2. ГОСТ 19.201–78. Единая система программной документации. Техническое задание. Требования к содержанию и оформлению: утвержден и введен в действие Постановлением Госстандарта СССР от 11.12.1978 № 3447: дата введения 01.07.1979. – Москва: Издательство стандартов, 1979. – 8 с.
3. ГОСТ 34.602–2020. Информационная технология. Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированной системы: утвержден и введен в действие Приказом Росстандарта от 10.12.2020 № 1806-ст: дата введения 01.07.2021. – Москва: Стандартинформ, 2021. – 24 с.
4. Завьялов, А. В. Диаграммы UML для анализа и проектирования информационных систем: учебно-методическое пособие / А. В. Завьялов. – Москва: РТУ МИРЭА, 2021. – 65 с. – URL: <https://e.lanbook.com/book/218630> (дата обращения: 12.11.2025).
5. Грекул, В. И. Проектирование информационных систем: учебник и практикум для вузов / В. И. Грекул, Н. Л. Коровкина, Г. А. Левочкина. – 2-е изд., перераб. и доп. – Москва: Издательство Юрайт, 2024. – 423 с. – (Высшее образование). – ISBN 978-5-534-17841-8. – URL: <https://urait.ru/bcode/536901> (дата обращения: 03.04.2024).
6. Верников, А. А. Методологии IDEF0 и IDEF1X: учебный курс / А. А. Верников. – URL: <https://www.cfin.ru/vernikov/idef/idef1x.shtml?ysclid=mj5oooqbwh891568969> (дата обращения: 24.11.2025).

Изм.

Лист

№ докум.

Подпись

Дата

Лист

47

*090207.249755.ПЗ*

1. Кинзябулатов, Р. А. IDEF0. Знакомство с нотацией и пример использования / Р. А. Кинзябулатов. – URL: <https://www.trinion.org/blog/idef0-znakomstvo-s-notaciey-i-primer-ispolzovaniya> (дата обращения: 25.11.2025).
2. PyQt6 – полное руководство [Электронный ресурс]. – URL: <https://habr.com/ru/companies/skillfactory/articles/599599/> (дата обращения: 10.10.2025).

Изм.

Лист

№ докум.

Подпись

Дата

Лист

48

*090207.249755.ПЗ*

# **ПРИЛОЖЕНИЕ А**

Изм.

Лист

№ докум.

Подпись

Дата

Лист

49

*090207.249755.ПЗ*

**Техническое задание**

# **ПРИЛОЖЕНИЕ Б**

Изм.

Лист

№ докум.

Подпись

Дата

Лист

50

*090207.249755.ПЗ*

**Исходный код программы**

Исходный код программы представлен по ссылке: <https://github.com/Kirysha337/Project_GostiTut>