

Содержание

Введение.....	5
1 Анализ технического задания	6
1.1 Описание предметной области	9
1.2 Поиск и подбор аналогов	11
2. Разработка моделей данных	13
2.1 Концептуальная модель данных.....	13
2.2 Логическая модель данных	15
2.3 Физическая модель данных.....	18
3. Разработка и реализация АИС	23
3.1 SQL запросы	23
3.2 Руководство пользователя.....	30
3.3 Руководство программиста	40
4. Тестирование АИС.....	49
Заключение	56
Список литературы	57
Приложение 1. Модели данных	58
Приложение 2. Текст программы	61

					МИВУ 10.03.01-06ПЗ		
Изм.	Лист	№ докум	Подпись	Дата	АРМ администратора гостиницы		
Разраб		Гусев М.А					
Пров		Колпаков А.А.					
Н. Контр.							
Утв		Орлов А.А.					
					Литера	Лист	Листов
					у	4	66
					МИВлГУ ИБ-122		

Введение

Гостиничный бизнес является одним из быстро развивающихся сфер услуг. Эффективное функционирование современной гостиницы невозможно без применения информационных технологий и систем автоматизации. Одним из ключевых направлений в этой сфере является использование баз данных.

Базы данных позволяют автоматизировать такие процессы как регистрация клиентов, бронирование номеров, учет персонала, отслеживание финансовых показателей и многое другое. Это способствует оптимизации работы гостиницы, повышению качества обслуживания и улучшению финансовых результатов.

В связи с этим актуальной задачей является разработка информационной системы для гостиницы на базе реляционной базы данных. Цель данной курсовой работы – спроектировать и реализовать базу данных для гостиницы (отеля, хостела) с использованием изученных принципов и методов проектирования и разработки баз данных.

В работе будет рассмотрена предметная область гостиницы, спроектированы следующие модели баз данных: концептуальная, физическая и логическая. Также будет разработано графическое приложение на основе языка программирования C# в визуальной среде Visual Studio с использованием СУБД SQLite и по окончании разработки будет произведено тестирование функционала.

Разработанное ПО позволит автоматизировать основные бизнес-процессы гостиницы и обеспечит ее эффективное функционирование.

1 Анализ технического задания

Современная разработка информационных систем требует выбора инструментов, которые наилучшим образом соответствуют требованиям проекта. В контексте создания системы управления базой данных для гостиницы мы рассмотрим и сравним несколько технологий, чтобы понять, какие из них оптимально использовать.

Этот анализ охватывает языки программирования C# и C++, среды разработки Visual Studio и Eclipse, а также системы управления базами данных SQLite и MySQL. Введение этих сравнений поможет нам выявить наиболее подходящие инструменты, учитывая их производительность, простоту использования, гибкость, безопасность и интеграцию с другими системами. Это обеспечит надежность и эффективность системы управления базой данных, поддерживая высокие стандарты разработки и эксплуатации.

1. Характеристики C++:

Производительность: очень высока, поскольку это язык низкого уровня, позволяет манипулировать памятью и системными ресурсами.

Гибкость: предоставляет детальный контроль над аппаратным обеспечением.

Использование: широко используется для разработки системного ПО, драйверов и высокопроизводительных приложений.

2. Характеристики C#:

Простота: более высокого уровня, чем C++, более легкий для изучения и использования.

Платформа: тесно интегрирован с .NET, что облегчает разработку веб-приложений и бизнес-логики.

Безопасность: предоставляет автоматическое управление памятью (сборка мусора), что снижает вероятность ошибок, связанных с памятью.

3. Характеристики Visual Studio:

					МИВУ 10.03.01-06ПЗ	Лист
						6
Изм.	Лист	№ докум	Подпись	Дата		

Функциональность: Обширный набор инструментов для отладки, тестирования и разработки.

Интеграция: отлично интегрируется с экосистемой Microsoft, поддерживает множество языков программирования.

Использование: очень удобная и интуитивно понятная среда.

4. Характеристики Eclipse:

Открытый исходный код: бесплатная и доступная, поддерживаемая сообществом.

Кроссплатформенность: работает на различных операционных системах (Windows, macOS, Linux).

Плагины: огромное количество доступных плагинов для расширения функциональности.

5. Характеристики SQLite

Легковесность: не требует установки и настройки сервера.

Простота: легок в использовании и имеет компактную архитектуру.

Встроенная СУБД: отлично подходит для приложений, где требуется локальное хранилище данных.

6. Характеристики MySQL

Производительность: высокая производительность и масштабируемость для крупных проектов.

Функциональность: поддерживает сложные запросы, транзакции и интеграцию с другими СУБД.

Популярность: широко используется в веб-разработке и поддерживается большим сообществом.

Результат анализа характеристик можно представить в виде таблицы 1 – таблица сравнения.

Таблица 1 – Таблица сравнения

Критерий	C++	C#	Visual Studio	Eclipse	SQLite	MySQL
Производительность	Высокая	Средняя	-	-	Высокая	Высокая
Простота	Сложная	Простая	Удобная	Средняя	Простая	Средняя
Гибкость	Высокая	Средняя	Высокая	Высокая	Низкая	Высокая
Интеграция	Ограниченная	Отличная с .NET	Отличная с экосистемой MS	Кросс-платформенная	Локальное хранилище данных	Веб-разработка
Безопасность	Низкая (ручное управление памятью)	Высокая (автоматическая сборка мусора)	Высокая	Высокая	Высокая	Высокая

Для реализации СУБД гостиницы лучше всего подойдет язык программирования C# со средой разработки Visual Studio и базой данных SQLite, основанной на языке SQL, выбор этих инструментов разработки являются оптимальным выбором.

Они предлагают следующие преимущества:

- C#: Простота использования, высокая производительность, отличная совместимость с .NET и высокая безопасность.
- Visual Studio: Мощная и интуитивно понятная интегрированная среда разработки, поддержка множества языков и богатый набор инструментов.

- SQLite: Легковесная и простая в использовании СУБД, идеально подходящая для встроенных систем и приложений с локальным хранением данных.

Эти инструменты обеспечат стабильную, безопасную и эффективную работу системы управления базой данных для гостиницы.

1.1 Описание предметной области

В данной курсовой работе рассматривается предметная область гостиницы. Гостиница — это организация, которая предоставляет временное жильё (номера) за плату.

Гостиницы классифицируются по следующим признакам:

- По количеству звёзд - от 1 до 5 звёзд. Чем больше звёзд, тем комфортабельнее номера и выше уровень сервиса.

- По типу размещения:

- хостел - самый дешевый вариант, общежитие;

- мотель - номера расположены в отдельно стоящих корпусах вокруг парковки;

- отель - здание или комплекс зданий с общим вестибюлем, этажами, лифтами.

- По целевой направленности:

- деловые;

- семейные;

- туристические;

- апарт-отели - номера-студии с небольшой кухней.

- По сфере обслуживания:

- для взрослых;

- детские;

- молодежные;

- для пансионатного проживания преклонного возраста.

Гостиница предоставляет комплекс услуг по проживанию: сдача в аренду номеров, уборка номеров, смена постельного белья, предоставление приборов и мебели в номерах. Также в ней могут функционировать ресторан, бар, зоны отдыха.

Основные функции гостиницы - предоставление временного жилья, питание, уборка номеров, оздоровление, организация досуга гостей. Качество услуг зависит от класса и специализации конкретного отеля.

Основные цели гостиницы:

- Регистрация клиентов.
- Бронирование номеров.
- Организация проживания клиентов в номерах на определенный период.
- Оказание дополнительных платных услуг.

Основная задача курсовой работы - разработка приложения для автоматизации работы гостиницы. Это позволит упростить ведение учета и облегчить работу персонала.

В гостинице можно выделить следующие сущности:

Клиенты - люди, бронирующие и снимающие номера.

- Атрибуты сущности клиента: ФИО, номер телефон, почтовый адрес.

Номера - помещения, сдаваемые в аренду клиентам.

- Атрибуты сущности номера: тип комнаты, стоимость, статус.

Дополнительные услуги - платные услуги, предоставляемые гостям (химчистка, трансфер и др.).

- Атрибуты сущности дополнительных услуг: сервис, стоимость, доступность.

Персонал - сотрудники гостиницы разных должностей.

- Атрибуты сущности персонала: ФИО, специализация, номер телефона.

Бронирование комнаты — это процесс заблаговременного заказа и блокирования конкретного номера в гостинице на определенный период времени.

- Атрибуты бронирования комнаты: дата заезда и выезда, общая стоимость.

База данных будет содержать информацию о клиентах, номерах, бронировании, оплате проживания, а также о персонале и дополнительных услугах. Это позволит автоматизировать все основные процессы работы гостиницы.

1.2 Поиск и подбор аналогов

При поиске и подборе аналогов баз данных для гостиницы, может быть полезно обратить внимание на решения, которые предлагают схожий функционал.

Ниже представлены популярные системы управления базами данных (СУБД) для управления гостиничными операциями.

1.2.1 Oracle Hospitality OPERA Cloud:

Ссылка: <https://www.oracle.com/hospitality/hotel-property-management/hotel-pms-software/>

Достоинства:

- Интегрированное решение для управления гостиничным бизнесом
- Масштабируемость и гибкость для гостиниц различного размера
- Управление бронированием, учетом, финансами, персоналом

Недостатки:

- Высокая стоимость лицензий и внедрения
- Сложность в настройке и интеграции с другими системами

Особенности:

Облачная архитектура для упрощения обновлений и доступа

1.2.2 Fidelio Suite8 (Amadeus Hospitality):

Достоинства:

- Развитый функционал для гостиничных сетей
- Интеграция с каналами продаж и системами-партнерами
- Инструменты аналитики и отчетности

Недостатки:

- Высокая стоимость владения
- Сложность в настройке и обучении персонала

Особенности:

Ориентация на крупные гостиничные сети и корпорации

1.2.3 Booking.com Connectivity (PMS):

Достоинства:

- Бесплатная облачная PMS-система
- Интеграция с популярными сайтами бронирования
- Простота использования и низкий барьер входа

Недостатки:

- Ограниченный функционал для крупных гостиниц
- Зависимость от политик Booking.com

Особенности:

Ориентация на небольшие и средние отели

1.2.4 Mews (PMS):

Достоинства:

- Современный дизайн и интуитивный интерфейс
- Интеграция с каналами продаж и маркетинговыми инструментами
- Гибкость настроек и автоматизации

Недостатки:

- Сравнительно высокая стоимость лицензий
- Ограниченная функциональность для крупных гостиниц

Особенности:

Ориентация на инновационные подходы в гостиничном бизнесе

2. Разработка моделей данных

Разработка моделей данных включает 3 этапа:

Первый этап — это разработка концептуальной модели, которая включает в себя следующее:

- определение сущностей (объектов, понятий) предметной области и их взаимосвязей;
- выявление атрибутов сущностей и описание их характеристик;
- построение диаграммы сущность-связь (ER-диаграммы) для визуального представления модели.

Второй этап — это разработка логической модели, которая включает в себя следующее:

- трансформация концептуальной модели в структуру, подходящую для конкретной СУБД;
- определение типов данных, ключей, индексов, ограничений целостности;
- нормализация данных для устранения избыточности и аномалий;
- спецификация отношений между сущностями (таблицами).

Третий этап — это разработка физической модели, которая включает в себя следующее:

- оптимизация логической модели для конкретной СУБД и аппаратной платформы;
- определение физического хранения данных: файлов, томов, буферов и т.д.;
- результат - физическая схема базы данных, готовая для реализации.[1]

2.1 Концептуальная модель данных

Концептуальная модель данных является первым и наиболее абстрактным уровнем проектирования базы данных. На этом этапе определяются ключевые сущности предметной области, их атрибуты и взаимосвязи.

Сущность «Клиент» имеет следующие атрибуты: ФИО, номер телефона, E-mail, паспортные данные, гражданство.

Сущность «Бронирование» имеет следующие атрибуты: дата заезда, дата выезда, общая стоимость.

Сущность «Номер» имеет следующие атрибуты: Статус, стоимость, тип комнаты.

Сущность «Персонал» имеет следующие атрибуты: ФИО, специализация, номер телефона, адрес.

Сущность «Дополнительные услуги» имеет следующие атрибуты: Сервис, стоимость, доступность.

Сущность «История бронирования» имеет следующие атрибуты: Дата заезда, дата выезда, ФИО клиента, тип номера.

Сущность «Гости» имеет следующий атрибут: ФИО.

Можно выделить следующие связи между сущностями:

- между сущностями «Клиент» и «Бронирование» существует связь «один-ко-многим», так как один клиент может забронировать несколько номеров;

- между сущностями «Бронирование» и «Номер» - связь «многие-к-одному», множество броней распространяется на один номер;

- между сущностями «Бронирование» и «Услуги» - связь «многие-ко-многим», множество услуг распространяется на множество броней.

- между сущностями «Персонал» и «Номер» - связь «один-ко-многим», так как к одному номеру может быть прикреплено несколько сотрудников.

- между сущностями «Бронирование» и «Гости» - связь "многие-ко-многим", так как одно бронирование может включать нескольких гостей, и один гость может участвовать в нескольких бронированиях.

- между сущностями «Бронирование» и «История бронирования» - связь "один-ко-многим", так как одно бронирование может иметь несколько записей в истории бронирований.

На рисунке 1 изображена концептуальная модель базы данных.

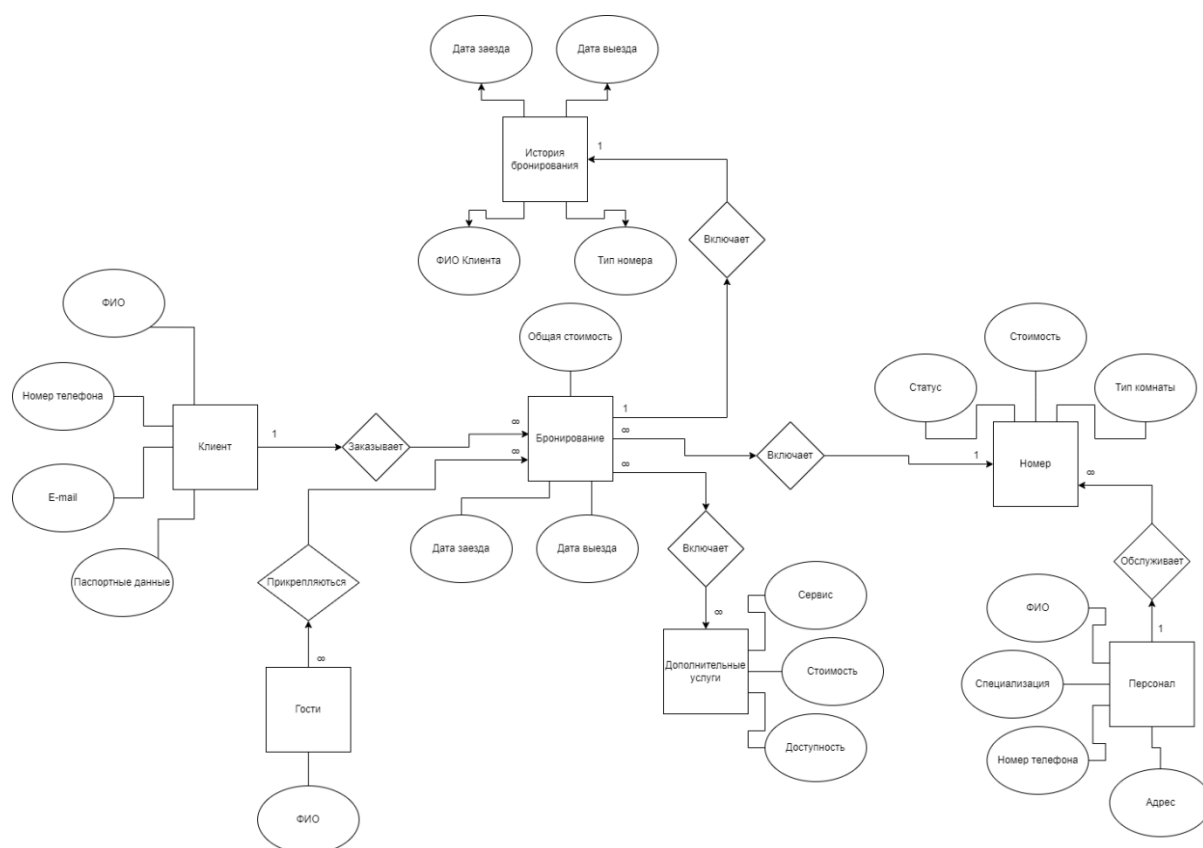


Рисунок 1 – Концептуальная модель базы данных

2.2 Логическая модель данных

Логическая модель данных является следующим этапом после концептуальной модели. На этом этапе концептуальная модель трансформируется в структуру, подходящую для конкретной системы управления базами данных (СУБД)[4].

Основные сущности и их атрибуты:

2.2.1 Клиенты:

- ID_Клиента - уникальный идентификатор клиента
- ФИО - имя, фамилия, отчество клиента
- Дата рождения
- Серия и номер паспорта
- Адрес
- Гражданство

- Email

2.2.2 Бронирование номеров:

- ID_Клиента - связь с клиентом, сделавшим бронирование
- ID_Номера - связь с забронированным номером
- ID_Брони – уникальный идентификатор бронирования
- Общая стоимость

2.2.3 Номера:

- ID_Номера - уникальный идентификатор номера
- ID_Услуги - связь с дополнительными услугами
- Тип номера
- Статус
- Стоимость номера

2.2.4 Дополнительные услуги:

- ID_Услуги - уникальный идентификатор услуги
- Сервис - наименование услуги
- Стоимость
- Доступность

2.2.5 Персонал:

- ID_Персонала - уникальный идентификатор сотрудника
- ФИО
- Специализация
- Номер телефона
- Адрес

2.2.6 Бронирование услуги:

- ID_Брони – связь с бронированием
- ID_Услуги – связь с дополнительными услугами

2.2.7 Гости:

- ID_Гостя – уникальный идентификатор гостя

- ФИО

2.2.8 Бронирование гости

- ID_Гостя - связь с гостем.
- ID_Брони - связь с бронированием.

2.2.9 История бронирования

- ID_Брони - уникальный идентификатор бронирования.
- ID_Клиента - связь с клиентом, сделавшим бронирование.
- ID_Номера - связь с забронированным номером.
- Дата заезда.
- Дата выезда.

Данная логическая модель данных отражает основные сущности, связанные с гостиницей: клиенты, бронирование номеров, бронирование услуг, информация о номерах, дополнительные услуги и персонал, гости, история бронирования. Она позволяет хранить и управлять данными, необходимыми для эффективного функционирования гостиницы.

Такая логическая модель данных представлена на рисунке 2.

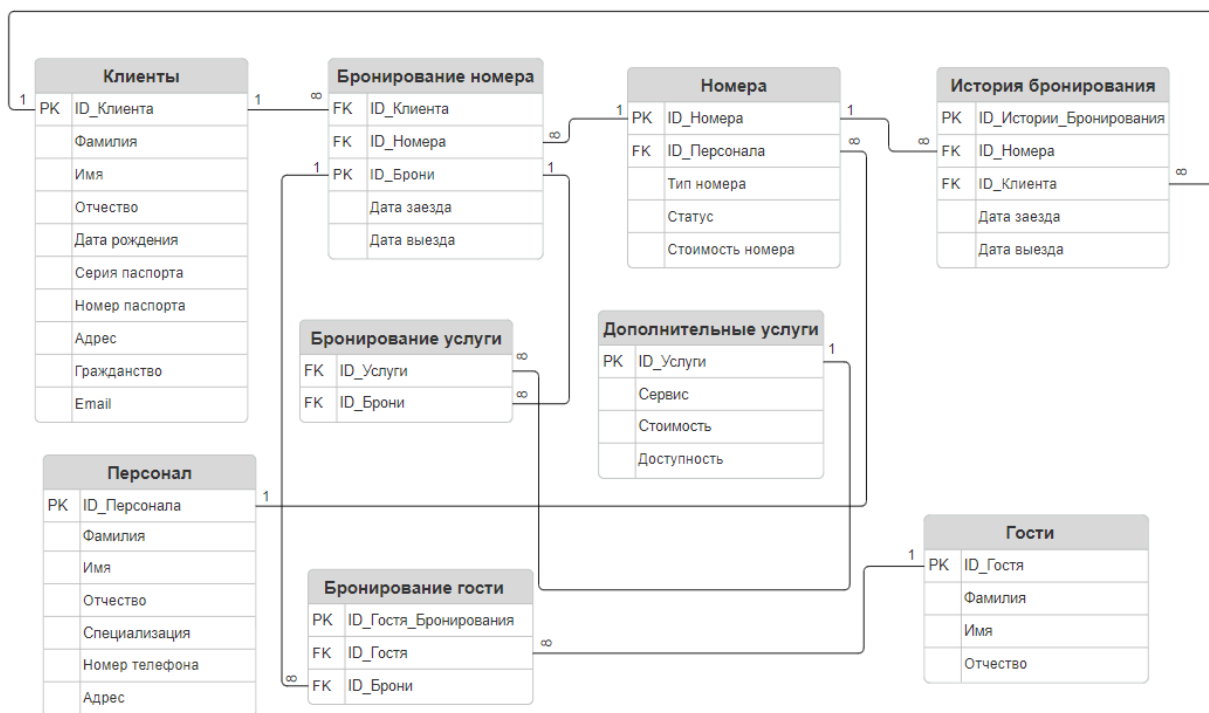


Рисунок 2 – Логическая модель данных

2.3 Физическая модель данных

Физическая модель данных — это этап проектирования базы данных предполагает реализацию логической модели с учетом конкретной системы управления базами данных (СУБД) и требований к производительности системы.[3]

К основным задачам физической модели данных относится:

1) Определение физических характеристик таблиц и полей:

- Типы данных для каждого атрибута (integer, varchar, date и т.д.)
- Размеры полей (длина строк)
- Ограничения целостности данных (первичные/внешние ключи, уникальность, проверка значений)

2) Проектирование физической структуры базы данных:

- Определение физического хранения данных (файлы, тома, блоки)
- Настройка индексов для ускорения доступа к данным
- Решения по партиционированию и кластеризации таблиц
- Механизмы резервного копирования и восстановления данных

3) Оптимизация производительности:

- Анализ запросов и создание оптимальных индексов
- Денормализация данных для повышения скорости выборки
- Использование материализованных представлений
- Настройка буферизации, кэширования, параллелизма

На рисунке 3 изображена физическая модель данных.

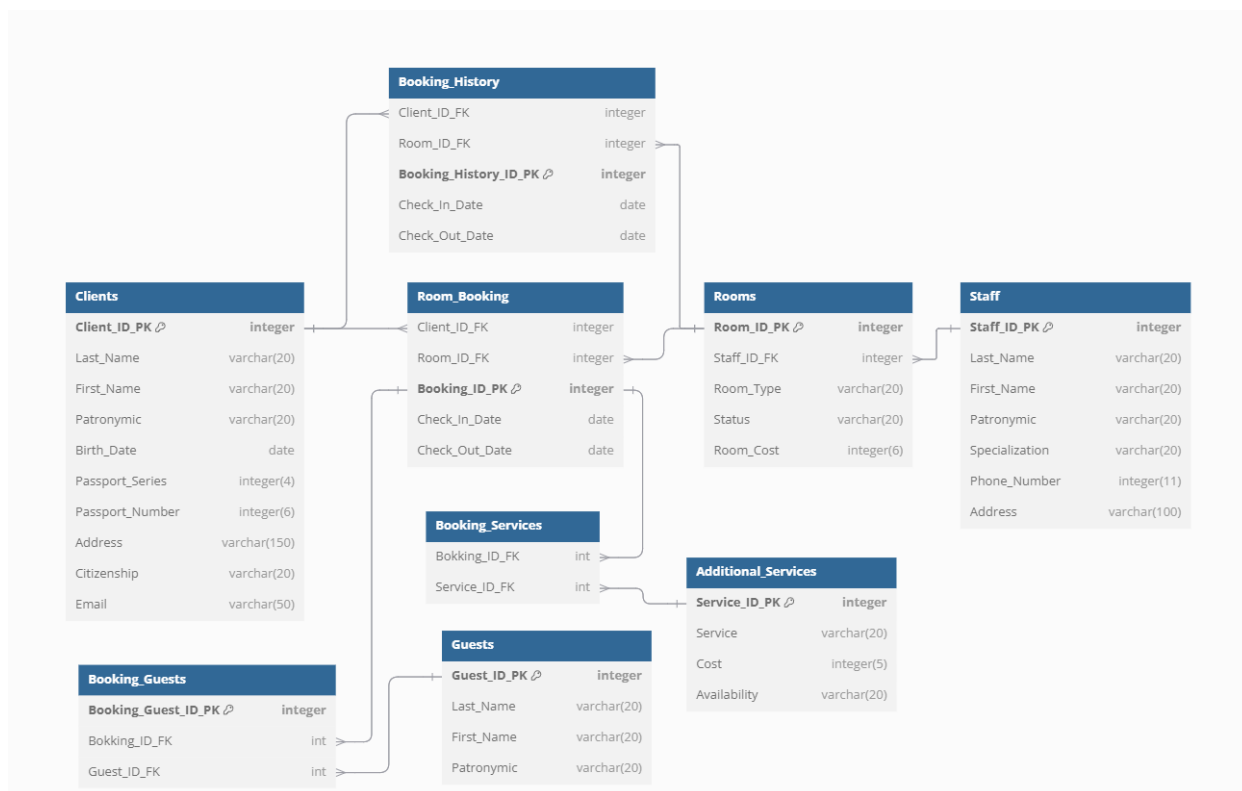


Рисунок 3 – Физическая модель данных

Физическая модель данных состоит из следующих таблиц:

2.3.1 Таблица "Clients":

- Client_ID_PK (integer, primary key)
- Last_Name (LastName(varchar(20)))
- First_Name (FirstName(varchar(20)))
- Patronymic (Patronymic(varchar(20)))
- Birth_Date (dirthday(date))
- Passport_Series (passportSeries(integer(4)))
- Passport_Number (passportNumber(integer(6)))
- Address (Address(varchar(150)))
- Citizenship (Citizenship(varchar(20)))
- Email (email(varchar(50)))

Для данной таблицы могут быть определены:

- первичный ключ Client_ID_PK для уникальной идентификации записей;
- индекс по полю Email для ускорения поиска клиентов по email;

- ограничение уникальности для Passport_Series, Passport_Number.

2.3.2 Таблица "Room_Booking":

- Booking_ID_PK (integer, primary key)
- Client_ID_FK (integer, внешний ключ на таблицу Clients)
- Room_ID_FK (integer, внешний ключ на таблицу Rooms)
- Check_In_Date (CheckinDate(date))
- Check_Out_Date (CheckoutDate(date))

Ключевые особенности:

- связь с таблицами Clients и Rooms через внешние ключи;
- индекс по Client_ID_FK для ускорения поиска бронирований клиента;
- индекс по Check_In_Date и Check_Out_Date для быстрого поиска

свободных номеров;

2.3.3 Таблица "Rooms":

- Room_ID_PK (integer, primary key)
- Staff_ID_FK (integer, внешний ключ на таблицу Staff)
- Room_Type (RoomType(varchar(20)))
- Status (Status(varchar(20)))
- Room_Cost (RoomCost(integer(6)))

Ключевые особенности:

- индекс по Room_Type для группировки и фильтрации;
- индекс по Status для быстрого поиска свободных/занятых номеров;
- ограничение Status проверяет допустимые значения статуса.

2.3.4 Таблица "Staff":

- Staff_ID_PK (integer, primary key)
- Last_Name (LastName(varchar(20)))
- First_Name (FirstName(varchar(20)))
- Patronymic (Patronymic(varchar(20)))
- Specialization (Specializatio(varchar(20)))
- Phone_Number (PhoneNumber(integer(11)))

- Address (Address(varchar(100)))

Ключевые особенности:

- первичный ключ Staff_ID_PK для уникальной идентификации;
- индекс по Specialization для быстрого поиска сотрудников определенной

категории;

- ограничение Phone_Number на уникальность.

2.3.5 Таблица "Additional_Services":

- Service_ID_PK (integer, primary key)
- Service (Service(varchar(20)))
- Cost (Cost(integer(5)))
- Availability (Availability(varchar(20)))

Ключевые особенности:

- первичный ключ Service_ID_PK для уникальной идентификации;
- индекс по Service для быстрого поиска услуг;
- ограничение Availability проверяет активность/неактивность услуги.

2.3.6 Таблица "Booking_Service":

- Booking_ID_FK (integer, внешний ключ на таблицу Room_Booking)
- Service_ID_FK (integer, внешний ключ на таблицу Additional_Services)

Ключевые особенности:

- связь с таблицами Room_Booking и Additional_Services через внешние ключи;
- индекс по Booking_ID_FK и Service_ID_FK для ускорения поиска и фильтрации;

2.3.7 Таблица "Booking_Guests":

- Booking_Guest_ID_PK (integer, primary key, autoincrement)
- Booking_ID_FK (integer, внешний ключ на таблицу Room_Booking)
- Guest_ID_FK (integer, внешний ключ на таблицу Guests)

Ключевые особенности:

- Связь с таблицами Room_Booking и Guests через внешние ключи.

- Индекс по полю Booking_ID_FK для ускорения поиска гостей по бронированию.

- Индекс по полю Guest_ID_FK для быстрого поиска бронирований по гостям.

2.3.8 Таблица "Guests":

- Guest_ID_PK (integer, primary key, autoincrement)
- Last_Name (LastName(varchar(20)))
- First_Name (FirstName(varchar(20)))
- Patronymic (Patronymic(varchar(20)))

Ключевые особенности:

- Первичный ключ Guest_ID_PK для уникальной идентификации гостей.
- Индекс по полям Last_Name, First_Name, и Patronymic для ускорения поиска гостей.

2.3.9 Таблица "Booking_History":

- Booking_History_ID_PK (integer, primary key)
- Client_ID_FK (integer, внешний ключ на таблицу Clients)
- Room_ID_FK (integer, внешний ключ на таблицу Rooms)
- Check_In_Date (CheckinDate(text))
- Check_Out_Date (CheckoutDate(text))

Ключевые особенности:

- Связь с таблицами Clients и Rooms через внешние ключи.
- Индекс по Client_ID_FK для ускорения поиска исторических данных по клиентам.
- Индекс по Room_ID_FK для быстрого поиска исторических данных по комнатам.
- Индекс по Check_In_Date и Check_Out_Date для анализа временных периодов.

3. Разработка и реализация АИС

3.1 SQL запросы

SQL (Structured Query Language) — это стандартный язык для работы с реляционными базами данных, который позволяет выполнять широкий спектр операций: от создания и модификации структуры базы данных до манипуляции данными и их извлечения. Применение SQL-запросов является ключевым аспектом разработки и реализации автоматизированных информационных систем, так как они обеспечивают взаимодействие с базой данных и выполнение различных операций над данными.[2]

3.1.1 Запрос создания таблицы (CREATE TABLE):

Запрос, который создает новую таблицу с наименованием Room_Booking.

```
CREATE TABLE Room_Booking ( Booking_ID_PK INTEGER PRIMARY KEY,  
Client_ID_FK  INTEGER,  Room_ID_FK  INTEGER,  Check_In_Date  DATE,  
Check_Out_Date  DATE,  FOREIGN KEY  (Client_ID_FK)  REFERENCES  
Clients(Client_ID_PK),  FOREIGN KEY  (Room_ID_FK)  REFERENCES  
Rooms(Room_ID_PK) )
```

Этот запрос делает следующие действия:

-CREATE TABLE Room_Booking: Эта команда создаёт новую таблицу с названием Room_Booking.

-Booking_ID_PK INTEGER PRIMARY KEY: создаётся столбец Booking_ID_PK с типом данных INTEGER (целое число), который будет использоваться в качестве первичного ключа. Первичный ключ уникально идентифицирует каждую запись в таблице.

-Client_ID_FK INTEGER: создаётся столбец Client_ID_FK с типом данных INTEGER, который будет использоваться для хранения идентификатора клиента. Этот столбец является внешним ключом, ссылающимся на таблицу Clients.

-Room_ID_FK INTEGER: создаётся столбец Room_ID_FK с типом данных INTEGER, который будет использоваться для хранения идентификатора комнаты. Этот столбец является внешним ключом, ссылающимся на таблицу Rooms.

-Check_In_Date DATE: создаётся столбец Check_In_Date с типом данных DATE для хранения даты заезда.

-Check_Out_Date DATE: создаётся столбец Check_Out_Date с типом данных DATE для хранения даты выезда.

-FOREIGN KEY (Client_ID_FK) REFERENCES Clients(Client_ID_PK): Устанавливается внешний ключ, который связывает столбец Client_ID_FK с первичным ключом Client_ID_PK в таблице Clients. Это означает, что значение в столбце Client_ID_FK должно соответствовать значению в Client_ID_PK таблицы Clients.

-FOREIGN KEY (Room_ID_FK) REFERENCES Rooms(Room_ID_PK): Устанавливается внешний ключ, который связывает столбец Room_ID_FK с первичным ключом Room_ID_PK в таблице Rooms. Это означает, что значение в столбце Room_ID_FK должно соответствовать значению в Room_ID_PK таблицы Rooms.

3.1.2 Запросы выборки из таблиц (SELECT):

1) Запрос выборки и форматирования данных из таблицы Additional_Service

SELECT Service_ID_PK, Service || ' - ' || Cost AS DisplayService FROM Additional_Services

Этот запрос выполняет следующие действия:

-SELECT Service_ID_PK, Service || ' - ' || Cost AS DisplayService: Команда SELECT выбирает два столбца: Service_ID_PK и результат объединения (||) значений столбцов Service и Cost. Оператор || используется для объединения строк. В данном случае, к строке Service добавляется строка ' - ', а затем строка Cost. Полученное значение будет представлено как DisplayService.

-AS DisplayService: Ключевое слово AS используется для присвоения псевдонима результирующему столбцу. В данном случае, объединенное значение столбцов Service и Cost будет отображаться под именем DisplayService.

-FROM Additional_Services: Эта часть запроса указывает, что данные будут извлекаться из таблицы Additional_Services.

2) Запрос для извлечения данных из таблицы Clients

```
SELECT Last_Name, First_Name, Patronymic, Passport_Series, Passport_Number  
FROM Clients WHERE Client_ID_PK = @ClientID
```

Этот запрос выполняет следующие действия:

-SELECT Last_Name, First_Name, Patronymic, Passport_Series, Passport_Number: Команда SELECT используется для выбора перечисленных столбцов: Last_Name (фамилия), First_Name (имя), Patronymic (отчество), Passport_Series (серия паспорта), Passport_Number (номер паспорта).

-FROM Clients: Эта часть запроса указывает, что данные будут извлечены из таблицы Clients.

-WHERE Client_ID_PK = @ClientID: Условие WHERE фильтрует записи, выбирая только те, у которых значение в столбце Client_ID_PK соответствует значению переменной @ClientID.

3) Запрос для извлечения данных из нескольких таблиц (Room_Booking, Clients и Rooms) и объединения этих данных для получения подробной информации о бронировании номеров, клиентах и типах комнат

```
SELECT  
Room_Booking.Booking_ID_PK,  
Room_Booking.Client_ID_FK,  
Clients.Last_Name || ' ' || SUBSTR(Clients.First_Name, 1, 1) || '.' ||  
SUBSTR(Clients.Patronymic, 1, 1) || '.' AS Client_Name,  
Room_Booking.Room_ID_FK,  
Rooms.Room_Type,  
Room_Booking.Check_In_Date,
```

```

Room_Booking.Check_Out_Date
FROM
Room_Booking
JOIN
Clients ON Room_Booking.Client_ID_FK = Clients.Client_ID_PK
JOIN
Rooms ON Room_Booking.Room_ID_FK = Rooms.Room_ID_PK";

```

Этот запрос выполняет следующие действия:

SELECT:

- Room_Booking.Booking_ID_PK: идентификатор бронирования.
- Room_Booking.Client_ID_FK: идентификатор клиента.
- Clients.Last_Name || ' ' || SUBSTR(Clients.First_Name, 1, 1) || '.' || SUBSTR(Clients.Patronymic, 1, 1) || '.' AS Client_Name: формирует строку с фамилией клиента и инициалами имени и отчества. Пример: "Иванов И.И."

- Room_Booking.Room_ID_FK: идентификатор комнаты.
- Rooms.Room_Type: тип комнаты (например, одноместный, двухместный и т.д.).

- Room_Booking.Check_In_Date: дата заезда.

- Room_Booking.Check_Out_Date: дата выезда.

- FROM Room_Booking: указывает основную таблицу Room_Booking, из которой будут извлекаться данные.

- JOIN Clients ON Room_Booking.Client_ID_FK = Clients.Client_ID_PK: объединяет таблицу Room_Booking с таблицей Clients на основе идентификатора клиента, чтобы получить данные о клиентах.

- JOIN Rooms ON Room_Booking.Room_ID_FK = Rooms.Room_ID_PK: объединяет таблицу Room_Booking с таблицей Rooms на основе идентификатора комнаты, чтобы получить данные о типах комнат.

3.1.3 Запросы вставки в таблицу (INSERT):

1) Запрос для вставки новой записи в таблицу Room_Booking

INSERT INTO Room_Booking (Client_ID_FK, Room_ID_FK, Check_In_Date, Check_Out_Date) VALUES (@ClientID, @RoomID, @CheckInDate, @CheckOutDate)

Этот запрос выполняет следующие действия:

-INSERT INTO Room_Booking (Client_ID_FK, Room_ID_FK, Check_In_Date, Check_Out_Date): Эта часть запроса указывает, что мы вставляем новые данные в таблицу Room_Booking и перечисляет столбцы, в которые будут вставлены значения: Client_ID_FK (идентификатор клиента), Room_ID_FK (идентификатор комнаты), Check_In_Date (дата заезда) и Check_Out_Date (дата выезда).

-VALUES (@ClientID, @RoomID, @CheckInDate, @CheckOutDate): Эта часть запроса задает значения, которые будут вставлены в указанные столбцы. Переменные @ClientID, @RoomID, @CheckInDate и @CheckOutDate будут заменены соответствующими данными при выполнении запроса.

2) Запрос для вставки новой записи в таблицу Clients

INSERT INTO Clients (Last_Name, First_Name, Patronymic, Birth_Date, Passport_Series, Passport_Number, Address, Citizenship, Email) VALUES (@LastName, @FirstName, @Patronymic, @BirthDate, @PassportSeries, @PassportNumber, @Address, @Citizenship, @Email)

Этот запрос выполняет следующие действия:

-INSERT INTO Clients (Last_Name, First_Name, Patronymic, Birth_Date, Passport_Series, Passport_Number, Address, Citizenship, Email): Эта часть запроса указывает, что мы вставляем новые данные в таблицу Clients, перечисляя столбцы, в которые будут вставлены значения. Это столбцы для фамилии, имени, отчества, даты рождения, серии паспорта, номера паспорта, адреса, гражданства и электронной почты.

-VALUES (@LastName, @FirstName, @Patronymic, @BirthDate, @PassportSeries, @PassportNumber, @Address, @Citizenship, @Email): Эта часть запроса задает значения, которые будут вставлены в указанные столбцы. Переменные @LastName, @FirstName, @Patronymic, @BirthDate, @PassportSeries,

@PassportNumber, @Address, @Citizenship и @Email будут заменены соответствующими данными при выполнении запроса.

3.1.4 Запросы обновления записей в таблицах (UPDATE):

1) Запрос для обновления существующей записи в таблице Room_Booking

```
UPDATE Room_Booking SET Client_ID_FK=@ClientID,  
Room_ID_FK=@RoomID,Check_In_Date=@CheckInDate,Check_Out_Date=@Check  
OutDate WHERE Booking_ID_PK=@BookingID
```

Этот запрос выполняет следующие действия:

-UPDATE Room_Booking SET: Эта часть запроса указывает, что мы обновляем данные в таблице Room_Booking.

-Client_ID_FK = @ClientID: устанавливает новое значение для столбца Client_ID_FK (идентификатор клиента), используя значение переменной @ClientID.

-Room_ID_FK = @RoomID: устанавливает новое значение для столбца Room_ID_FK (идентификатор комнаты), используя значение переменной @RoomID.

-Check_In_Date = @CheckInDate: устанавливает новое значение для столбца Check_In_Date (дата заезда), используя значение переменной @CheckInDate.

-Check_Out_Date = @CheckOutDate: устанавливает новое значение для столбца Check_Out_Date (дата выезда), используя значение переменной @CheckOutDate.

-WHERE Booking_ID_PK = @BookingID: указывает условие, которое определяет, какую запись обновить. В данном случае, обновляется запись, у которой значение в столбце Booking_ID_PK (идентификатор бронирования) соответствует значению переменной @BookingID.

2) Запрос для обновления существующей записи в таблице Clients

```
UPDATE Clients SET Last_Name=@LastName, First_Name=@FirstName,  
Patronymic=@Patronymic,Birth_Date=@BirthDate,Passport_Series=@PassportSeries,P  
assport_Number=@PassportNumber,Address=@Address,Citizenship=@Citizenship,Em  
ail=@Email WHERE Client_ID_PK=@ClientId
```

Этот запрос выполняет следующие действия:

-UPDATE Clients SET: Эта часть запроса указывает, что мы обновляем данные в таблице Clients.

-Last_Name = @LastName: устанавливает новое значение для столбца Last_Name (фамилия), используя значение переменной @LastName.

-First_Name = @FirstName: устанавливает новое значение для столбца First_Name (имя), используя значение переменной @FirstName.

-Patronymic = @Patronymic: устанавливает новое значение для столбца Patronymic (отчество), используя значение переменной @Patronymic.

-Birth_Date = @BirthDate: устанавливает новое значение для столбца Birth_Date (дата рождения), используя значение переменной @BirthDate.

-Passport_Series = @PassportSeries: устанавливает новое значение для столбца Passport_Series (серия паспорта), используя значение переменной @PassportSeries.

-Passport_Number = @PassportNumber: устанавливает новое значение для столбца Passport_Number (номер паспорта), используя значение переменной @PassportNumber.

-Address = @Address: устанавливает новое значение для столбца Address (адрес), используя значение переменной @Address.

-Citizenship = @Citizenship: устанавливает новое значение для столбца Citizenship (гражданство), используя значение переменной @Citizenship.

-Email = @Email: устанавливает новое значение для столбца Email (электронная почта), используя значение переменной @Email.

-WHERE Client_ID_PK = @ClientId: указывает условие, которое определяет, какую запись обновить. В данном случае, обновляется запись, у которой значение в столбце Client_ID_PK (идентификатор клиента) соответствует значению переменной @ClientId.

3.1.5 Запросы удаления записи из таблицы (DELETE)

1) Запрос для удаления записей из таблицы Booking_Service

DELETE FROM Booking_Service WHERE Booking_ID_FK=@BookingID

Этот запрос выполняет следующие действия:

-DELETE FROM Booking_Service: Эта часть запроса указывает, что мы хотим удалить записи из таблицы Booking_Service.

-WHERE Booking_ID_FK=@BookingID: Условие WHERE ограничивает удаление тех записей, у которых значение в столбце Booking_ID_FK (идентификатор бронирования) совпадает со значением переменной @BookingID.

2) Запрос для удаления записи из таблицы Clients

DELETE FROM Clients WHERE Client_ID_PK=@ClientId

Этот запрос выполняет следующие действия:

-DELETE FROM Clients: Эта часть запроса указывает, что мы хотим удалить записи из таблицы Clients.

-WHERE Client_ID_PK = @ClientId: Условие WHERE ограничивает удаление только тех записей, у которых значение в столбце Client_ID_PK (идентификатор клиента) совпадает со значением переменной @ClientId.

3.2 Руководство пользователя

При открытии приложения первым шагом для пользователя является окно авторизации, показанное на рисунке 1, которое служит для идентификации и аутентификации пользователей.

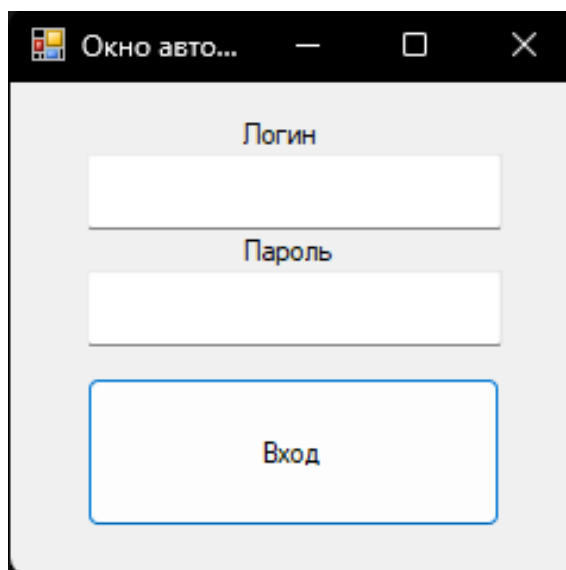


Рисунок 1 – окно авторизации

В окне авторизации пользователь вводит логин и пароль от своей учетной записи, что показано на рисунках 2,3,4 и соответственно если логин и пароль правильные, то он попадает в рабочее окно, что показано рисунках 5,6,7.

The screenshot shows a window titled "Окно авто...". It contains two input fields: "Логин" (Login) with the text "admin" and "Пароль" (Password) with masked characters "*****". Below the fields is a button labeled "Вход" (Login).

Рисунок 2 – ввод данных от учетной записи администратора

The screenshot shows a window titled "Окно авто...". It contains two input fields: "Логин" (Login) with the text "sysadmin" and "Пароль" (Password) with masked characters "*****". Below the fields is a button labeled "Вход" (Login).

Рисунок 3 – ввод данных от учетной записи системного администратора

The screenshot shows a window titled "Окно авто...". It contains two input fields: "Логин" (Login) with the text "staff" and "Пароль" (Password) with masked characters "*****". Below the fields is a button labeled "Вход" (Login).

Рисунок 4 – ввод данных от учетной записи сотрудника

Бронирование номеров

Клиенты

История бронирования

Введите ФИО для поиска

Поиск

Декабрь 2024

Пн	Вт	Ср	Чт	Пт	Сб	Вс
25	26	27	28	29	30	1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31	1	2	3	4	5

Сегодня: 04.12.2024

ФИО Клиента	Номер Комнаты	Тип Комнаты	Дата Звезда	Дата Выезда
Иванов И.И.	10	Люкс	03.12.2024 17:20	08.12.2024 17:20
Иванов И.И.	1	Эконом	01.12.2024 15:12	01.12.2024 15:12
Николаев Н.Н.	6	Стандарт	01.12.2024 15:12	25.12.2024 15:12
Иванов И.И.	2	Эконом	01.12.2024 15:12	04.12.2024 15:12
Иванов И.И.	3	Эконом	01.12.2024 15:12	07.12.2024 15:12
*				

Добавление гостя

Добавление

Редактирование

Удаление

Дата заезда

4 декабря 2024 г.

Дата выезда

4 декабря 2024 г.

Выбор клиента

1 Эконом 2000

Выбор комнаты

Иванов Иван

Выбор услуги

☐ WiFi - 50
 ☐ Завтрак - 150
 ☐ Парковка - 200
 ☐ СПА - 300
 ☐ Трансфер - 400

Рисунок 5 – рабочее окно администратора

Меню системного администратора

Номера

Сотрудники

Дополнительные услуги

Рисунок 6 – меню системного администратора

Рабочее окно сотрудника

Введите ФИО для поиска

Поиск

ФИО сотрудника	Номер комнаты	Тип комнаты
Соловьев Егор ...	1	Эконом
Крылова Мария...	2	Эконом
Соловьев Егор ...	3	Эконом
Крылова Мария...	4	Эконом
Горбачева Ирин...	5	Стандарт
Горбачева Ирин...	6	Стандарт
Шестакова Еле...	7	Стандарт
Горбачева Ирин...	8	Люкс
Соловьев Егор ...	9	Люкс
Шестакова Еле...	10	Люкс
*		

Рисунок 7 - рабочее окно сотрудника

Рабочее окно администратора или же окно бронирования номеров, которое показано на рисунке 2, имеет следующие функции:

1) Кнопка перехода на форму клиентов, форма клиента показана на рисунке 8.

2) Кнопка перехода на форму история бронирования, в этой форме администратор может увидеть прошлые бронирования, а также при помощи строки поиска по ФИО, может найти необходимого клиента, форма история бронирования показана на рисунке 9.

3) Поле поиска клиента в таблице бронирования по ФИО.

4) Календарь с выделенными датами, до кого числа занята конкретная комната.

5) Кнопка перехода на форму гостей, в этой форме администратор может заполнить ФИО гостя, для того, чтоб добавить его в базу данных и прикрепить гостя к необходимому бронированию, форма гостей показана на рисунке 10.

6) Кнопки добавления, удаления и редактирования записей в таблице бронирования.

7) Таблица бронирования в которой выделены цветом записи с разным сроком окончания бронирования, красный цвет обозначает что до конца бронирования остался 1 день, желтый цвет означает что до конца бронирования осталось 3 дня, зеленый цвет означает что до конца бронирования осталось 5 дней, все что больше 5 дней не выделяется цветом, что показано на рисунке 11.

8) При выделении записи в таблице бронирования можно получить всю информацию по бронированию, которая показывается в нижних таблицах, это показано на рисунке 12.

9) Для добавления бронирования в правой стороне нам необходимо выбрать дату заезда, дату выезда, клиента, номер и дополнительные услуги, а после добавления этой записи, мы можем добавить к бронированию гостей, это показано на рисунках 13,14,15.

ClientForm

Введите ФИО для поиска в таблице клиентов

Поиск

	Фамилия	Имя	Отчество	Дата Рождения	Серия Паспорта	Номер Паспорта	Адрес	Гражданство	Email
▶	Иванов	Иван	Иванович	01.01.1990	1234	567890	Москва, Красна...	Россия	ivanov@example...
	Петров	Петр	Петрович	14.02.1985	2345	678901	Санкт-Петербур...	Россия	petrov@example...
	Сидоров	Сидр	Сидорович	25.03.1980	3456	789012	Казань, улица Б...	Россия	sidorov@exampl...
	Кузнецов	Алексей	Алексеевич	18.04.1975	4567	890123	Новосибирск, К...	Россия	kuznetsov@exa...
	Попов	Сергей	Сергеевич	10.05.1982	5678	901234	Екатеринбург, у...	Россия	popov@example...
	Васильев	Василий	Васильевич	22.06.1991	6789	412345	Челябинск, ули...	Россия	vasilev@exampl...
	Смирнов	Андрей	Андреевич	15.07.1983	7890	123456	Омск, улица Пе...	Россия	smimov@exampl...
	Михайлов	Михаил	Михайлович	30.08.1979	8901	234567	Самара, улица ...	Россия	mikhailov@exam...
	Федоров	Федор	Федорович	19.09.1988	9012	345678	Уфа, проспект ...	Россия	fedorov@exampl...
	Николаев	Николай	Николаевич	11.10.1984	1237	456789	Волгоград, улиц...	Россия	nikolaev@exam...
	Павлов	Павел	Павлович	01.11.1986	1236	567891	Калининград, ул...	Россия	pavlov@example...
	Алексеев	Алексей	Алексеевич	24.12.1992	2348	678902	Томск, улица Га...	Россия	alekseev@exam...
	Андреев	Андрей	Андреевич	12.01.1993	3456	789013	Пермь, улица К...	Россия	andreev@exampl...
	Макаров	Макар	Макарович	05.02.1987	4567	890124	Саранск, просп...	Россия	makarov@exam...
	Григорьев	Григорий	Григорьевич	18.03.1981	5678	901235	Киров, улица Ок...	Россия	grigorev@exampl...
	Лебедев	Лев	Львович	11.04.1989	6789	182346	Владимир, улиц...	Россия	lebedev@exampl...
	Киселев	Кирилл	Кириллович	20.05.1978	7890	123457	Воронеж, просп...	Россия	kiselev@example...
	Семенов	Семен	Семенович	17.06.1976	8901	234568	Тула, улица Сов...	Россия	semenov@exam...
	Степанов	Степан	Степанович	09.07.1990	9012	345679	Рязань, улица Е...	Россия	stepanov@exam...
	Егоров	Егор	Егорович	14.08.1985	1239	456790	Курск, улица Пе...	Россия	egorov@example...

Фамилия

Имя

Отчество

Дата рождения

4 декабря 2024 г.

Серия паспорта

Номер паспорта

Полный адрес

Гражданство

Электронная почта

Добавление

Редатирование клиента

Удаление записи

Рисунок 8 – форма клиента

HistoryForm

	ID Бронирования	ФИО Клиента	Номер Комнаты	Тип Комнаты	Дата Зеезда	Дата Выезда
▶	2	Иванов И.И.	6	Стандарт	2024-11-26 16:4...	2024-11-30 16:4...
	3	Иванов И.И.	1	Эконом	2024-11-26 16:5...	2024-11-26 16:5...
	4	Иванов И.И.	10	Люкс	2024-11-26 16:5...	2024-12-03 16:5...
	5	Кузнецов А.А.	9	Люкс	2024-11-26 16:5...	2024-12-02 16:5...
*						

Удаление записи из истории бронирования

Поиск

Введите ФИО для поиска

Рисунок 9 – форма истории бронирования

GuestForm

Карпенко Илья Иванович

Карпенко Илья Иванович

Фамилия

Карпенко

Имя

Илья

Отчество

Иванович

Добавить нового гостя

Удалить гостя

Прикрепить гостя к бронированию

Открепить гостя от брони

Редактирование гостя

Рисунок 10 – форма гостя

	ФИО Клиента	Номер Комнаты	Тип Комнаты	Дата Заезда	Дата Выезда
▶	Иванов И.И.	10	Люкс	03.12.2024 17:20	08.12.2024 17:20
	Петров П.П.	2	Эконом	04.12.2024 13:15	05.12.2024 13:15
	Кузнецов А.А.	3	Эконом	04.12.2024 13:15	07.12.2024 13:15
	Павлов П.П.	7	Стандарт	04.12.2024 13:15	18.12.2024 13:15
*					

Рисунок 11 – таблица бронирования

	ФИО Клиента	Номер Комнаты	Тип Комнаты	Дата Заезда	Дата Выезда
	Иванов И.И.	10	Люкс	03.12.2024 17:20	08.12.2024 17:20
	Петров П.П.	2	Эконом	04.12.2024 13:15	05.12.2024 13:15
	Кузнецов А.А.	3	Эконом	04.12.2024 13:15	07.12.2024 13:15
▶	Павлов П.П.	7	Стандарт	04.12.2024 13:15	18.12.2024 13:15
*					

WiFi Завтрак Парковка		Номер комнаты	Класс номера	Стоимость номера в сутки
	▶	7	Стандарт	3000
	*			

	Фамилия	Имя	Отчество	Серия паспорта	Номер паспорта
▶	Павлов	Павел	Павлович	1236	567891
*					

Рисунок 12 – получение подробной информации при выделении записи в таблице бронирования

Добавление

Редактирование

Удаление

Дата заезда
4 декабря 2024 г.

Дата выезда
10 декабря 2024 г.

Выбор клиента
9 Люкс 5000

Выбор комнаты
Семенов Семен

Выбор услуги

☒ WiFi - 50
☒ Завтрак - 150
☐ Парковка - 200
☐ СПА - 300
☐ Трансфер - 400

Рисунок 13 - добавление клиента в таблицу бронирования

ФИО Клиента	Номер Комнаты	Тип Комнаты	Дата Заезда	Дата Выезда
Иванов И.И.	10	Люкс	03.12.2024 17:20	08.12.2024 17:20
Петров П.П.	2	Эконом	04.12.2024 13:15	05.12.2024 13:15
Кузнецов А.А.	3	Эконом	04.12.2024 13:15	07.12.2024 13:15
Павлов П.П.	7	Стандарт	04.12.2024 13:15	18.12.2024 13:15
Семенов С.С.	9	Люкс	04.12.2024 13:15	10.12.2024 13:15

WiFi

Завтрак

Номер комнаты

Класс номера

Стоимость номера в сутки

Фамилия

Имя

Отчество

Серия паспорта

Номер паспорта

Семенов

Семен

Семенович

8901

234568

Добавление гостя

Добавление

Карпенко Илья Иванович

Семенова Лидия Андреевна

Фамилия

Имя

Патроним

Отчество

Адрес

Семенова Лидия Андреевна

Добавить нового гостя

Удалить гостя

Прикрепить гостя к бронированию

Открепить гостя от брони

Редактирование гостя

Рисунок 14 – прикрепление гостя к бронированию

ФИО Клиента	Номер Комнаты	Тип Комнаты	Дата Заезда	Дата Выезда
Иванов И.И.	10	Люкс	03.12.2024 17:20	08.12.2024 17:20
Петров П.П.	2	Эконом	04.12.2024 13:15	05.12.2024 13:15
Кузнецов А.А.	3	Эконом	04.12.2024 13:15	07.12.2024 13:15
Павлов П.П.	7	Стандарт	04.12.2024 13:15	18.12.2024 13:15
Семенов С.С.	9	Люкс	04.12.2024 13:15	10.12.2024 13:15

Семенова Лидия Андреевна

Рисунок 15 – результат бронирования

Форма клиента, которая показана на рисунке 8 имеет следующие функции:

- 1) Кнопки добавления, удаления и редактирования записей в таблице бронирования.
- 2) Строку поиска клиента по ФИО.

3) Для добавления клиента в таблицу, необходимо заполнить все поля, как показано на рисунках 16, 17.

Фамилия	Максимов
Имя	Андрей
Отчество	Андреевич
Дата рождения	2 февраля 2000 г.
Серия паспорта	3334
Номер паспорта	555667
Полный адрес	Владимир, ул. Мира, 5
Гражданство	Россия
Электронная почта	Max@example.com
<input type="button" value="Добавление"/>	

Рисунок 16 – добавление клиента в таблицу

	Фамилия	Имя	Отчество	Дата Рождения	Серия Паспорта	Номер Паспорта	Адрес	Гражданство	Email
	Кузнецов	Алексей	Алексеевич	18.04.1975	4567	890123	Новосибирск, К...	Россия	kuznetsov@exa...
	Попов	Сергей	Сергеевич	10.05.1982	5678	901234	Екатеринбург, у...	Россия	popov@example....
	Васильев	Василий	Васильевич	22.06.1991	6789	412345	Челябинск, ули...	Россия	vasiliev@exampl...
	Смирнов	Андрей	Андреевич	15.07.1983	7890	123456	Омск, улица Ле...	Россия	smimov@exampl...
	Михайлов	Михаил	Михайлович	30.08.1979	8901	234567	Самара, улица ...	Россия	mikhailov@exam...
	Федоров	Федор	Федорович	19.09.1988	9012	345678	Уфа, проспект ...	Россия	fedorov@exampl...
	Николаев	Николай	Николаевич	11.10.1984	1237	456789	Волгоград, улиц...	Россия	nikolaev@examp...
	Павлов	Павел	Павлович	01.11.1986	1236	567891	Калининград, ул...	Россия	pavlov@example...
	Алексеев	Алексей	Алексеевич	24.12.1992	2348	678902	Томск, улица Га...	Россия	alekseev@exam...
	Андреев	Андрей	Андреевич	12.01.1993	3456	789013	Пермь, улица К...	Россия	andreev@exampl...
	Макаров	Макар	Макарович	05.02.1987	4567	890124	Саранск, просп...	Россия	makarov@examp...
	Григорьев	Григорий	Григорьевич	18.03.1981	5678	901235	Киров, улица Ок...	Россия	grigorev@exampl...
	Лебедев	Лев	Львович	11.04.1989	6789	182346	Владимир, улиц...	Россия	lebedev@exampl...
	Киселев	Кирилл	Кириллович	20.05.1978	7890	123457	Воронеж, просп...	Россия	kiselev@example...
	Семенов	Семен	Семенович	17.06.1976	8901	234568	Тула, улица Сов...	Россия	semenov@exam...
	Степанов	Степан	Степанович	09.07.1990	9012	345679	Рязань, улица Е...	Россия	stepanov@exam...
	Егоров	Егор	Егорович	14.08.1985	1239	456790	Курск, улица Ле...	Россия	egorov@example...
▶	Максимов	Андрей	Андреевич	02.02.2000	3334	555667	Владимир, ул. М...	Россия	Max@example.c...
*									

Рисунок 17 – результат добавления клиента в таблицу

Меню системного администратора, показанное на рисунке 6, имеет следующие функции:

1) Кнопка перехода на форму номера, в этой форме системный администратор может, добавить новые номера введя необходимые данные в строки, а также отредактировать и удалить существующие записи о номерах, а также прикрепить конкретного человека, который отвечает за этот номер, данная форма на рисунке 18.

2) Кнопка перехода на форму сотрудника, в этой форме системный администратор может добавить нового сотрудника введя необходимые данные в строки, а так же отредактировать или удалить существующие записи о сотрудниках и еще системный администратор так же может найти сотрудника в строке поиска введя его ФИО, данная форма показана на рисунке 19.

3) Кнопка перехода на форму дополнительных услуг, в этой форме системный администратор может добавить новую услугу введя необходимые данные в строки, а также отредактировать и удалить существующие записи о дополнительных услугах, данная форма показана на рисунке 20.

	ID комнаты	Тип комнаты	Статус	Стоимость	ID сотрудника	Сотрудник
▶	1	Эконом	Доступен для б...	2000	10	Соловьев Егор
	2	Эконом	Доступен для б...	2000	11	Крылова Мария
	3	Эконом	Доступен для б...	2000	10	Соловьев Егор
	4	Эконом	Доступен для б...	2000	11	Крылова Мария
	5	Стандарт	Доступен для б...	3000	13	Горбачева Ирина
	6	Стандарт	Доступен для б...	3000	13	Горбачева Ирина
	7	Стандарт	Доступен для б...	3000	12	Шестакова Елена
	8	Люкс	Доступен для б...	5000	13	Горбачева Ирина
	9	Люкс	Доступен для б...	5000	10	Соловьев Егор
	10	Люкс	Доступен для б...	5000	12	Шестакова Елена
*						

Рисунок 18 – форма номера

	ID сотрудника	Фамилия	Имя	Отчество	Специализация	Номер телефона	Адрес
▶	1	Кузнецов	Николай	Петрович	Manager	79001112233	Москва, Тверск...
	2	Смирнов	Алексей	Иванович	Cleaner	79001234567	Санкт-Петербур...
	3	Попов	Сергей	Сергеевич	Technician	79007654321	Новосибирск, К...
	4	Соколов	Александр	Викторович	Receptionist	79009876543	Екатеринбург, ...
	5	Новиков	Дмитрий	Алексеевич	Security	79005432109	Казань, Пушкин...
	6	Федоров	Анатолий	Иванович	Chef	79007651234	Уфа, Октября, 20
	7	Морозов	Максим	Сергеевич	Gardener	79003214567	Самара, Набере...
	8	Волков	Владимир	Петрович	Electrician	79008765432	Волгоград, Тра...
	9	Зайцев	Григорий	Андреевич	Plumber	79004567891	Нижний Новгор...
	10	Соловьев	Егор	Дмитриевич	Housekeeper	79003219876	Челябинск, Ком...
	11	Крылова	Мария	Александровна	Housekeeper	79004561234	Москва, Садова...
	12	Шестакова	Елена	Викторовна	Housekeeper	79005432187	Санкт-Петербур...
	13	Горбачева	Ирина	Сергеевна	Housekeeper	79007654320	Казань, Песная...
	14	Иванова	Ольга	Ивановна	Receptionist	79003219865	Новосибирск, С...
	15	Петрова	Татьяна	Петровна	Receptionist	79007865432	Екатеринбург, ...

Рисунок 19 – форма сотрудника

	ID услуги	Услуга	Стоимость	Доступность
▶	1	WiFi	50	Активно
	2	Завтрак	150	Активно
	3	Парковка	200	Активно
	4	СПА	300	Активно
	5	Трансфер	400	Активно
	6	Фитнес	250	Активно
	7	Бассейн	500	Активно
	8	Прачечная	100	Активно
	9	Уборка	70	Активно
	10	Конференц-зал	600	Активно

Дополнительная услуга

Стоимость

Доступность

Добавить
Редактировать
Удалить

Рисунок 20 – форма дополнительных услуг

Рабочее окно сотрудника, показанное на рисунке 7, предназначено для того, чтобы сотрудник мог отслеживать за какими номерами он закреплен, для этого есть строка поиска, куда сотрудник вписывает свое ФИО, и в результате поиска он видит за какими комнатами он закреплен, это показано на рисунке 21.

Введите ФИО для поиска
Горбачева Ирина Сергеевна
Поиск

	ФИО сотрудника	Номер комнаты	Тип комнаты
▶	Горбачева Ирин...	5	Стандарт
	Горбачева Ирин...	6	Стандарт
	Горбачева Ирин...	8	Люкс
*			

Рисунок 21 – рабочее окно сотрудника

3.3 Руководство программиста

3.3.1 Введение

Проект написан на языке программирования C# в среде Visual Studio с использованием базы данных SQLite. Проект включает в себя формы для различных типов пользователей, таких как клиенты, гости, администраторы и сотрудники.

3.3.2 Установка и настройка системы

Требования к системе:

- Visual Studio 2019 или новее
- .NET Framework 4.7.2 или новее[5]
- Библиотека SQLite

Установка необходимых инструментов и библиотек:

- Откройте Visual Studio.
- Создайте или откройте проект.
- Установите пакет SQLite через NuGet Package Manager: Install-Package

System.Data.SQLite[6]

3.3.3 Структура проекта

Проект состоит из следующих файлов и директорий:

- ClientForm.cs: Форма для клиента.
- GuestForm.cs: Форма для гостя.
- HistoryForm.cs: Форма для просмотра истории бронирований.
- MainForm.cs: Форма бронирования.
- RoomForm.cs: Форма для управления комнатами.
- ServiceForm.cs: Форма для управления услугами.
- StaffForm.cs: Форма для управления сотрудниками.
- StaffFormWork.cs: Форма для работы сотрудников.
- SysAdminForm.cs: Форма для системного администратора.
- LoginForm.cs: Форма входа в приложение.

3.3.4 Описание форм

1) ClientForm.cs:

- Конструктор ClientForm: инициализирует форму, устанавливает соединение с базой данных и загружает данные клиентов.
- Метод LoadData: открывает соединение с базой данных, выполняет SQL-запрос для получения данных клиентов, заполняет DataTable данными и отображает их в DataGridView. Также настраивает отображение столбцов.
- Метод dataGridView1_SelectionChanged: обрабатывает изменение выбранной строки в DataGridView и заполняет соответствующие текстовые поля данными выбранного клиента.
- Метод button1_Click: обрабатывает нажатие кнопки добавления нового клиента, выполняет SQL-запрос для вставки новых данных клиента в базу данных и обновляет данные в DataGridView.
- Метод button3_Click: обрабатывает нажатие кнопки удаления выбранного клиента, выполняет SQL-запрос для удаления клиента из базы данных и обновляет данные в DataGridView.
- Метод button2_Click: обрабатывает нажатие кнопки обновления данных выбранного клиента, выполняет SQL-запрос для обновления данных клиента в базе данных и обновляет данные в DataGridView.
- Метод button4_Click: обрабатывает нажатие кнопки поиска клиента по фамилии, имени и отчеству, выполняет SQL-запрос для поиска и отображает результаты в DataGridView.

2) GuestForm.cs:

- Конструктор GuestForm: инициализирует форму, устанавливает соединение с базой данных и загружает списки жильцов и прикрепленных жильцов для данного бронирования. Также устанавливает обработчик событий для изменения выбранного элемента в listBox1.
- Метод LoadGuests: выполняет SQL-запрос для получения списка всех жильцов из базы данных, заполняет DataTable данными и отображает их в listBox1.

- Метод LoadAttachedGuests: выполняет SQL-запрос для получения списка жильцов, прикрепленных к текущему бронированию, заполняет DataTable данными и отображает их в listBox2.

- Метод button1_Click: обрабатывает нажатие кнопки для добавления нового жилья, выполняет SQL-запрос для вставки данных нового жилья в базу данных и обновляет список всех жильцов.

- Метод button2_Click: обрабатывает нажатие кнопки для удаления выбранного жилья из базы данных, выполняет SQL-запрос для удаления выбранного жилья и обновляет списки всех жильцов и прикрепленных жильцов.

- Метод button3_Click: обрабатывает нажатие кнопки для прикрепления выбранного жилья к текущему бронированию, выполняет SQL-запрос для добавления записи в таблицу Booking_Guests и обновляет список прикрепленных жильцов.

- Метод button4_Click: обрабатывает нажатие кнопки для отвязывания выбранного жилья от текущего бронирования, выполняет SQL-запрос для удаления записи из таблицы Booking_Guests и обновляет список прикрепленных жильцов.

3) HistoryForm.cs:

- Конструктор HistoryForm: инициализирует форму, устанавливает соединение с базой данных и загружает данные истории бронирований.

- Метод LoadData: открывает соединение с базой данных, выполняет SQL-запрос для получения данных истории бронирований, заполняет DataTable данными и отображает их в dataGridViewHistory. Также настраивает отображение столбцов.

- Метод HistoryForm_Load: обрабатывает загрузку формы. (В данном коде метод пустой, можно использовать для инициализации дополнительных данных или настроек при загрузке формы).

- Метод button1_Click: обрабатывает нажатие кнопки для удаления выбранной записи истории бронирований. Проверяет, выбрана ли запись,

запрашивает подтверждение удаления, выполняет SQL-запрос для удаления записи и обновляет данные в dataGridViewHistory.

- Метод button2_Click: обрабатывает нажатие кнопки для поиска записей в истории бронирований по фамилии, имени и отчеству клиента. Выполняет SQL-запрос с параметрами поиска и отображает результаты в dataGridViewHistory.

4) MainForm.cs:

- Конструктор MainForm: инициализирует форму, устанавливает соединение с базой данных, загружает данные клиентов, комнат, услуг, бронирований и истории. Также устанавливает обработчик для события изменения выбранной строки в dataGridView1.

- Метод LoadServices: выполняет SQL-запрос для получения списка дополнительных услуг, заполняет DataTable данными и отображает их в checkedListBox1.

- Метод LoadClientDetails: выполняет SQL-запрос для получения данных клиента по его ID и отображает их в dataGridView2.

- Метод LoadRoomDetails: выполняет SQL-запрос для получения данных комнаты по ее ID, отображает данные в dataGridView3 и выделяет занятые даты в календаре.

- Метод HighlightOccupiedDates: выполняет SQL-запрос для получения дат занятости комнаты и выделяет эти даты в календаре monthCalendar1.

- Метод dataGridView1_SelectionChanged: обрабатывает изменение выбранной строки в dataGridView1, загружает данные выбранного бронирования, данные клиента, данные о комнате, данные о жильцах и выбранные услуги.

- Метод LoadClients: выполняет SQL-запрос для получения списка клиентов, заполняет DataTable данными и отображает их в comboBox2.

- Метод LoadRooms: выполняет SQL-запрос для получения списка доступных для бронирования комнат, заполняет DataTable данными и отображает их в comboBox1. Также устанавливает обработчик для события изменения выбранного элемента в comboBox1.

- Метод ComboBox1_SelectedIndexChanged: обрабатывает изменение выбранного элемента в comboBox1, выделяет занятые даты для выбранной комнаты в календаре.

- Метод LoadBookings: выполняет SQL-запрос для получения данных о всех бронированиях, заполняет DataTable данными и отображает их в dataGridView1.

- Метод HighlightExpiringBookings: выделяет цветом строки в dataGridView1, в зависимости от оставшихся дней до даты выезда.

- Метод LoadSelectedServices: выполняет SQL-запрос для получения списка выбранных услуг для данного бронирования и отображает их в listBox1.

- Метод button5_Click: обрабатывает нажатие кнопки для добавления нового бронирования. Проверяет наличие пересечений с существующими бронированиями, вставляет новое бронирование в таблицу Room_Booking и добавляет выбранные услуги для этого бронирования в таблицу Booking_Service.

- Метод button1_Click: обрабатывает нажатие кнопки для обновления выбранного бронирования. Обновляет данные бронирования и перезаписывает связанные услуги в таблице Booking_Service.

- Метод button2_Click: обрабатывает нажатие кнопки для удаления выбранного бронирования. Запрашивает подтверждение удаления, добавляет запись в таблицу истории бронирований, удаляет запись из основной таблицы бронирований и обновляет данные.

- Метод historyButton_Click: обрабатывает нажатие кнопки для открытия формы истории бронирований HistoryForm.

- Метод button3_Click: обрабатывает нажатие кнопки для управления жильцами выбранного бронирования. Открывает форму GuestForm и обновляет список жильцов для выбранного бронирования.

- Метод ProcessCmdKey: обрабатывает нажатие клавиши F5 для обновления приложения. Вызывает метод RefreshApplication.

- Метод RefreshApplication: обновляет выделенные даты в календаре monthCalendar, подсвечивая занятые даты для выбранного номера.

- Метод `button4_Click`: обрабатывает нажатие кнопки для поиска бронирований по имени клиента. Выполняет SQL-запрос с параметрами поиска и отображает результаты в `dataGridView1`.

5) RoomForm.cs:

- Конструктор `RoomForm`: инициализирует форму, устанавливает соединение с базой данных SQLite, загружает данные комнат и сотрудников, устанавливает обработчик для события изменения выбранной строки в `dataGridView1`.

- Метод `dataGridView1_SelectionChanged`: обрабатывает изменение выбранной строки в `dataGridView1`. Заполняет текстовые поля и выпадающий список `comboBox1` данными выбранной комнаты.

- Метод `LoadData`: открывает соединение с базой данных, выполняет SQL-запрос для получения данных о комнатах, заполняет `DataTable` данными и отображает их в `dataGridView1`. Также настраивает заголовки столбцов.

- Метод `LoadStaffData`: выполняет SQL-запрос для получения списка сотрудников, заполняет `DataTable` данными и отображает их в `comboBox1`.

- Метод `button1_Click`: обрабатывает нажатие кнопки для добавления новой комнаты. Выполняет SQL-запрос для вставки данных новой комнаты в базу данных и обновляет данные в `dataGridView1`.

- Метод `button2_Click`: обрабатывает нажатие кнопки для обновления данных выбранной комнаты. Выполняет SQL-запрос для обновления данных комнаты в базе данных и обновляет данные в `dataGridView1`.

- Метод `button3_Click`: обрабатывает нажатие кнопки для удаления выбранной комнаты. Выполняет SQL-запрос для удаления комнаты из базы данных и обновляет данные в `dataGridView1`.

6) ServiceForm.cs:

- Конструктор `ServiceForm`: инициализирует форму, устанавливает соединение с базой данных SQLite, загружает данные дополнительных услуг и

устанавливает обработчик для события изменения выбранной строки в dataGridView1.

- Метод LoadData: открывает соединение с базой данных, выполняет SQL-запрос для получения данных о дополнительных услугах, заполняет DataTable данными и отображает их в dataGridView1. Также настраивает заголовки столбцов.

- Метод dataGridView1_SelectionChanged: обрабатывает изменение выбранной строки в dataGridView1. Заполняет текстовые поля данными выбранной услуги.

- Метод dataGridView1_CellContentClick: (В данном коде метод пустой. Можно использовать для обработки нажатий на ячейки таблицы dataGridView1).

- Метод button1_Click: обрабатывает нажатие кнопки для добавления новой услуги. Выполняет SQL-запрос для вставки данных новой услуги в базу данных и обновляет данные в dataGridView1.

- Метод button2_Click: обрабатывает нажатие кнопки для обновления данных выбранной услуги. Выполняет SQL-запрос для обновления данных услуги в базе данных и обновляет данные в dataGridView1.

- Метод button3_Click: обрабатывает нажатие кнопки для удаления выбранной услуги. Выполняет SQL-запрос для удаления услуги из базы данных и обновляет данные в dataGridView1.

7) StaffForm.cs

- Конструктор StaffForm: инициализирует форму, устанавливает соединение с базой данных SQLite, загружает данные сотрудников и устанавливает обработчик для события изменения выбранной строки в dataGridView1.

- Метод LoadData: открывает соединение с базой данных, выполняет SQL-запрос для получения данных о сотрудниках, заполняет DataTable данными и отображает их в dataGridView1. Также настраивает заголовки столбцов.

- Метод dataGridView1_SelectionChanged: обрабатывает изменение выбранной строки в dataGridView1. Заполняет текстовые поля данными выбранного сотрудника.

- Метод `button1_Click`: обрабатывает нажатие кнопки для добавления нового сотрудника. Выполняет SQL-запрос для вставки данных нового сотрудника в базу данных и обновляет данные в `dataGridView1`.

- Метод `button2_Click`: обрабатывает нажатие кнопки для обновления данных выбранного сотрудника. Выполняет SQL-запрос для обновления данных сотрудника в базе данных и обновляет данные в `dataGridView1`.

- Метод `button3_Click`: обрабатывает нажатие кнопки для удаления выбранного сотрудника. Выполняет SQL-запрос для удаления сотрудника из базы данных и обновляет данные в `dataGridView1`.

- Метод `button4_Click`: обрабатывает нажатие кнопки для поиска сотрудников по имени, фамилии и отчеству. Выполняет SQL-запрос с параметрами поиска и отображает результаты в `dataGridView1`.

8) `StuffFormWork.cs`

- Конструктор `StuffFormWork`: инициализирует форму, устанавливает соединение с базой данных SQLite и загружает данные сотрудников и комнат.

- Метод `LoadData`: открывает соединение с базой данных, выполняет SQL-запрос для получения данных о сотрудниках и комнатах, заполняет `DataTable` данными и отображает их в `dataGridView1`. Также настраивает заголовки столбцов.

- Метод `button1_Click`: обрабатывает нажатие кнопки для поиска сотрудников по ФИО. Выполняет фильтрацию данных в `DataTable` на основе введенного значения и обновляет данные в `dataGridView1`.

9) `SysAdminForm.cs`

- Конструктор `SysAdminForm`: инициализирует форму и настраивает компоненты пользовательского интерфейса.

- Метод `button1_Click`: обрабатывает нажатие кнопки для открытия формы управления сотрудниками `StuffForm`. Создает экземпляр формы `StuffForm` и отображает его.

- Метод `button2_Click`: обрабатывает нажатие кнопки для открытия формы управления комнатами `RoomForm`. Создает экземпляр формы `RoomForm` и отображает его.

- Метод `button3_Click`: обрабатывает нажатие кнопки для открытия формы управления дополнительными услугами `ServiceForm`. Создает экземпляр формы `ServiceForm` и отображает его.

10) `LoginForm.cs`:

- Конструктор `LoginForm`: инициализирует форму, создает словарь пользователей с ролями и паролями, и настраивает компоненты пользовательского интерфейса.

- Метод `btnLogin_Click_Click`: обрабатывает нажатие кнопки входа. Получает введенные пользователем логин и пароль, проверяет их соответствие данным из словаря `userRoles`. Если логин и пароль совпадают, скрывает форму входа и открывает форму, соответствующую роли пользователя (администратор, системный администратор или сотрудник). Если данные не совпадают, выводит сообщение об ошибке.

4. Тестирование АИС

Тестирование автоматизированной информационной системы (АИС) является неотъемлемой частью процесса разработки, направленной на обеспечение ее корректной работы, надежности и безопасности. При открытии приложения первым шагом для пользователя является окно авторизации, показанное на рисунке 22, которое служит для идентификации и аутентификации пользователей. Это окно играет важную роль в защите данных и управлении доступом к функционалу системы, что делает его критически важным элементом для тестирования.

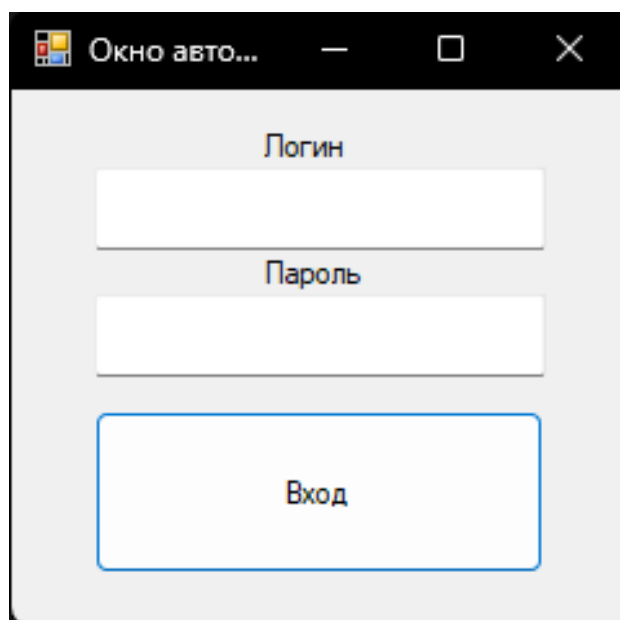


Рисунок 22 – окно авторизации

В окне авторизации пользователь вводит логин и пароль от своей учетной записи, если логин и пароль правильные, то пользователь попадает в рабочее окно, а если логин или пароль, который был введен не корректен, то приложение выдаст ошибку, что логин и пароль не правильный, это показано на рисунке 23.

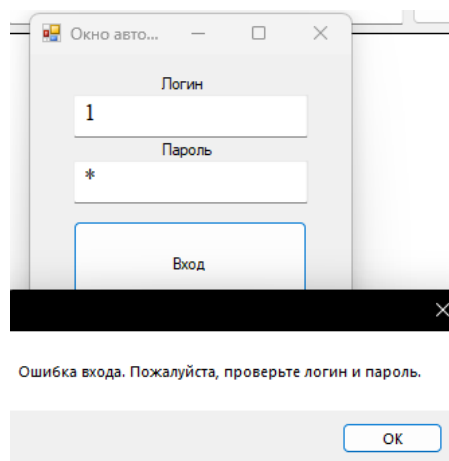


Рисунок 23 – ошибка входа, если логин и/или пароль указан не правильный

Если в таблице бронирования, таблице клиенты, таблице сотрудники, таблице номера и таблице дополнительные услуги необходимо отредактировать запись или ее удалить, и пользователь не выделит запись, которую необходимо отредактировать или удалить, но нажмет на кнопку «редактировать» или «удалить», то получит следующие сообщения, показанные на рисунках 24, 25, 26, 27, 28, 29.

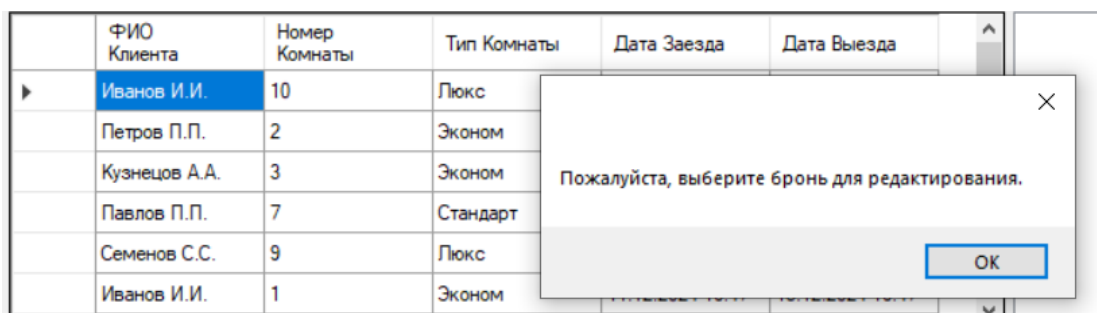


Рисунок 24 – уведомление о том, что не выбрана запись для редактирования в таблице бронирования

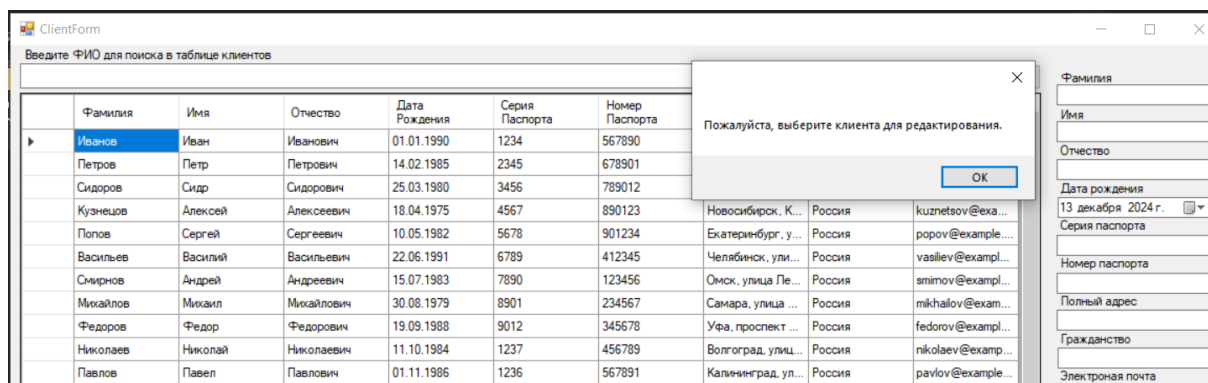


Рисунок 25 – уведомление о том, что не выбрана запись для редактирования в таблице клиенты

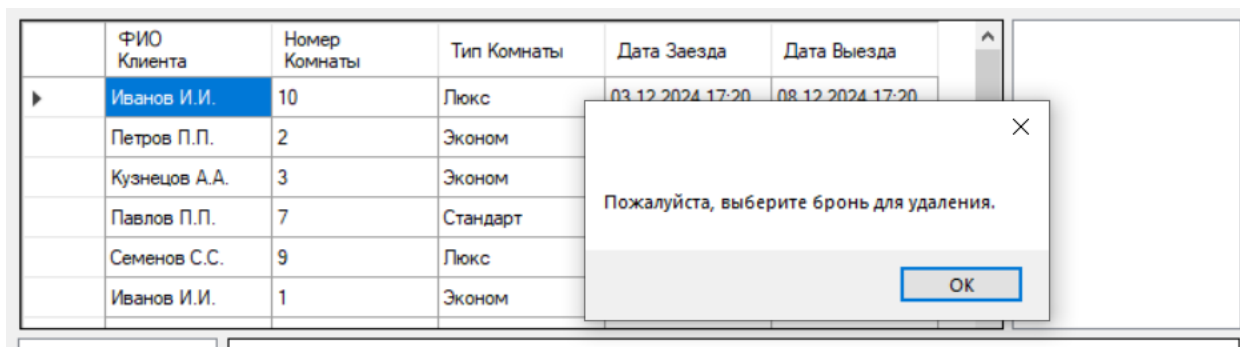


Рисунок 26 – уведомление о том, что не выбрана запись для удаления в таблице бронирования

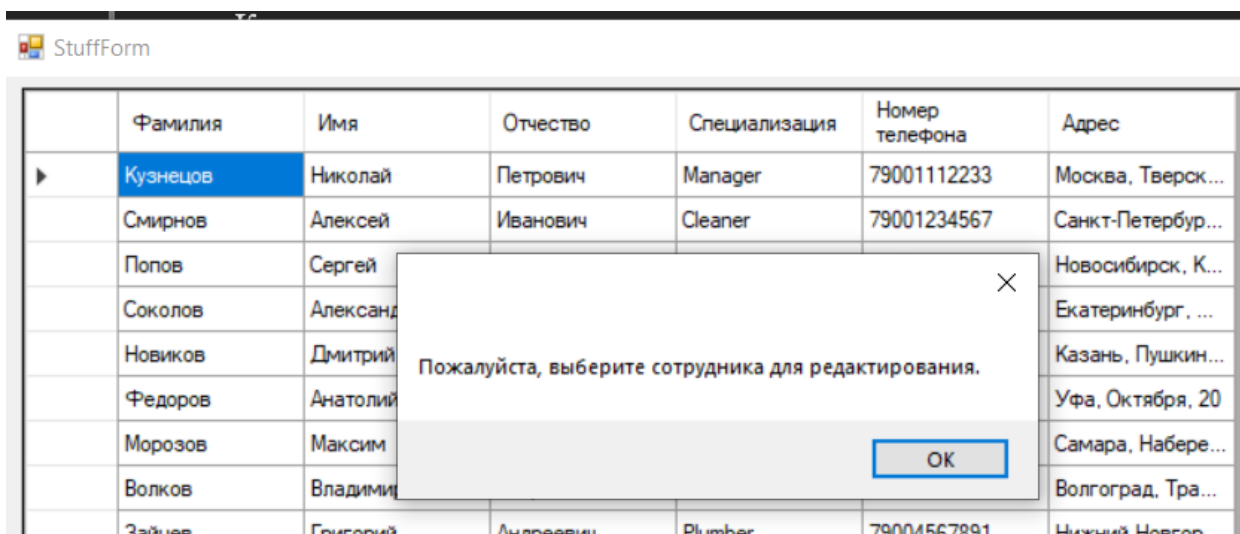


Рисунок 27 – уведомление о том, что не выбрана запись для редактирования в таблице сотрудники

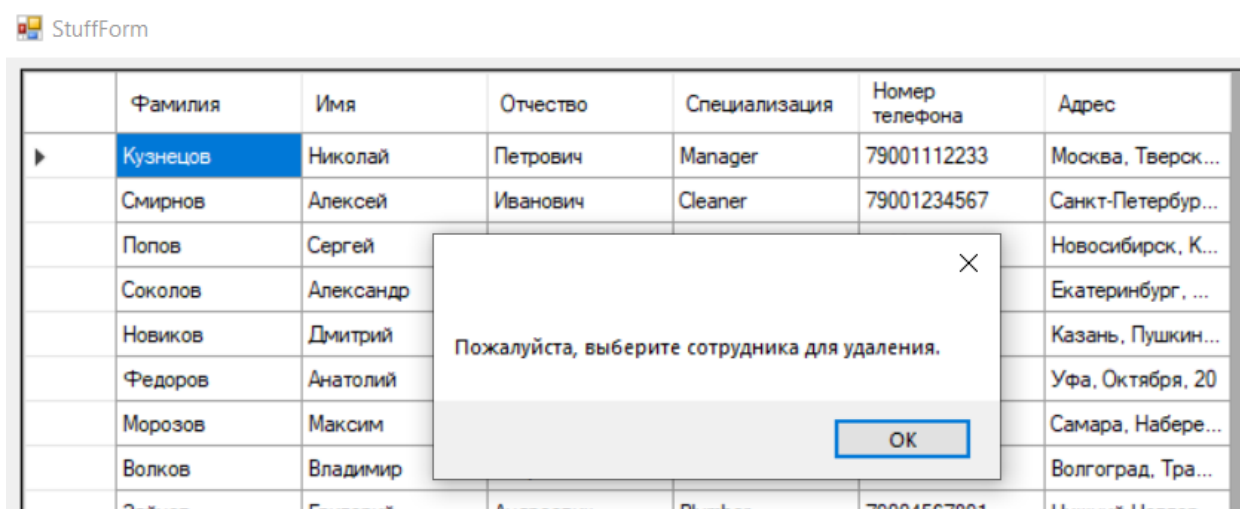


Рисунок 28 – уведомление о том, что не выбрана запись для удаления в таблице сотрудники

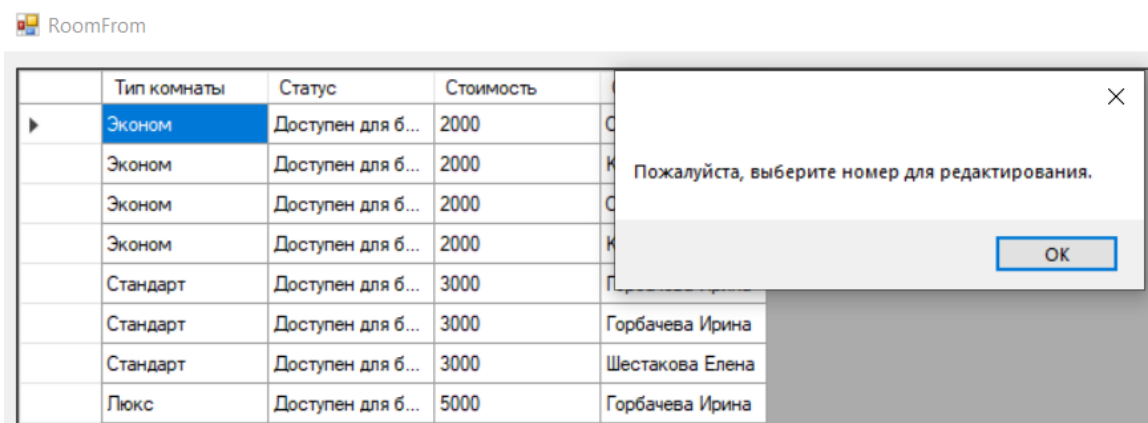


Рисунок 29 – уведомление о том, что не выбрана запись для редактирования в таблице номера

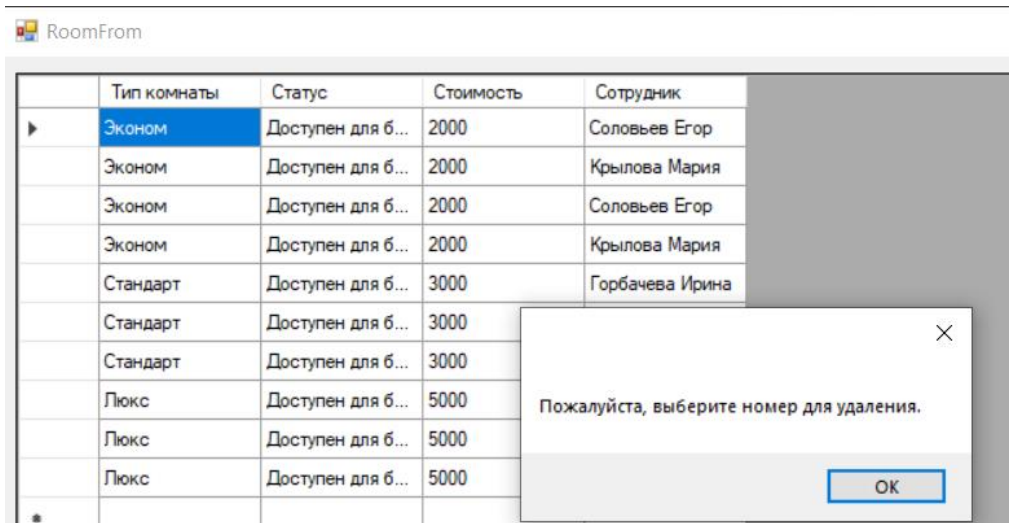


Рисунок 30 – уведомление о том, что не выбрана запись для удаления в таблице номера

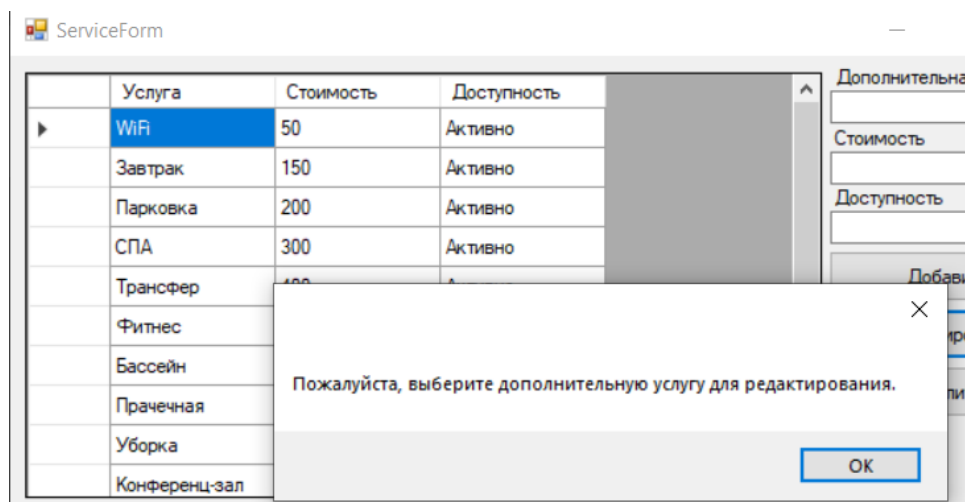


Рисунок 31 – уведомление о том, что не выбрана запись для редактирования в таблице дополнительные услуги

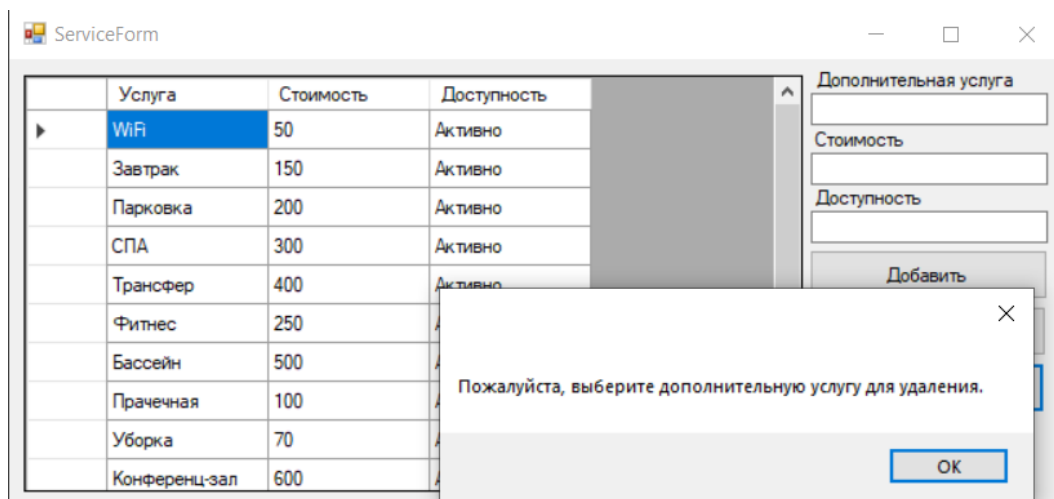


Рисунок 32 – уведомление о том, что не выбрана запись для удаления в таблице дополнительные услуги

Когда пользователь удаляет запись, появляется окно подтверждения удаления записи, показанное на рисунках 33, 34.

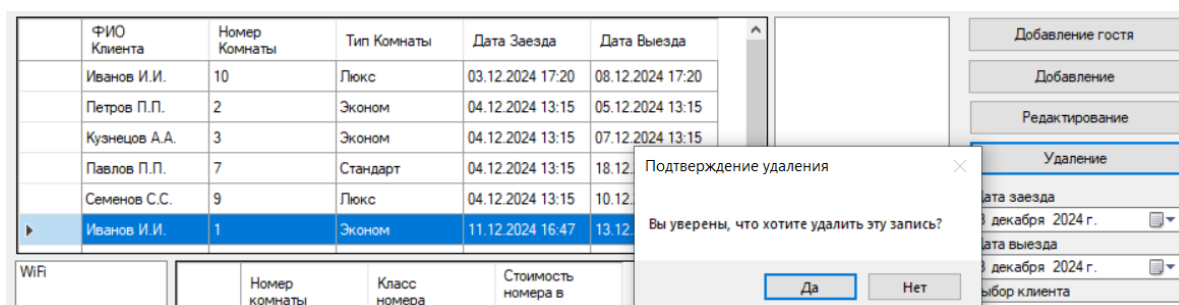


Рисунок 33 – окно подтверждения удаления записи в таблице бронирования

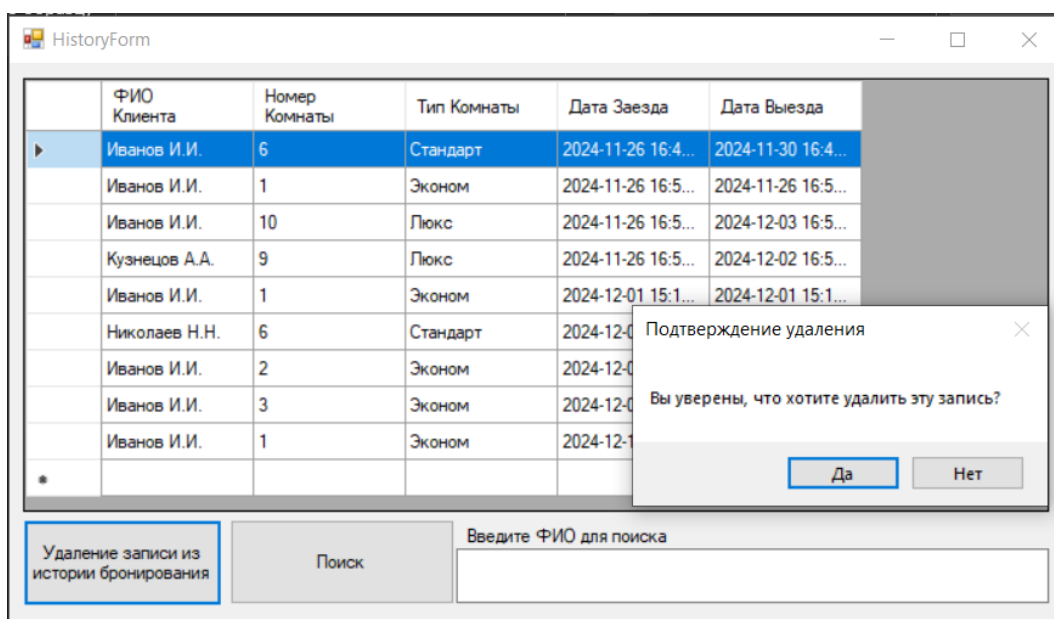


Рисунок 34 – окно подтверждения удаления записи в таблице история бронирования

Когда пользователь добавляет гостя, ему необходимо выделить нужную запись к которой он хочет прикрепить гостей, если запись не выбрать и нажать на кнопку «Добавить гостя», то пользователь получит следующее сообщение показано на рисунке 35.

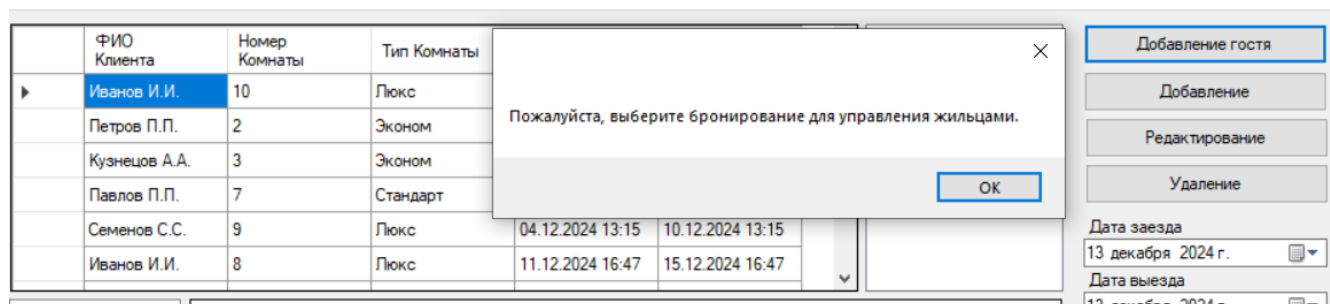


Рисунок 35 – уведомление о том, что для добавления гостя необходимо выбрать запись, к которой его нужно прикрепить

После нажатия на кнопку «Выход из учетной записи» пользователь получит уведомление, точно он хочет выйти из учетной записи или нет, данное окно показано на рисунке 36.

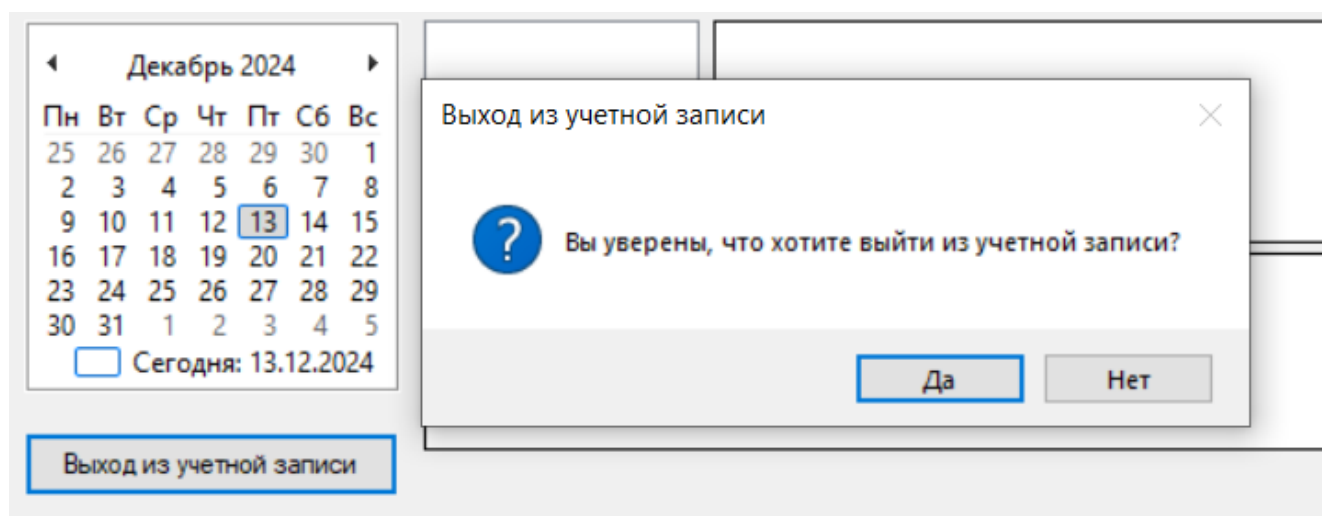


Рисунок 36 – окно подтверждения выхода из учтённой записи

Когда пользователь добавляет записи о бронировании, то может получить следующее уведомление показанное на рисунке 31, которое означает, что выбранная комната для бронирования уже занята и необходимо выбрать другую.

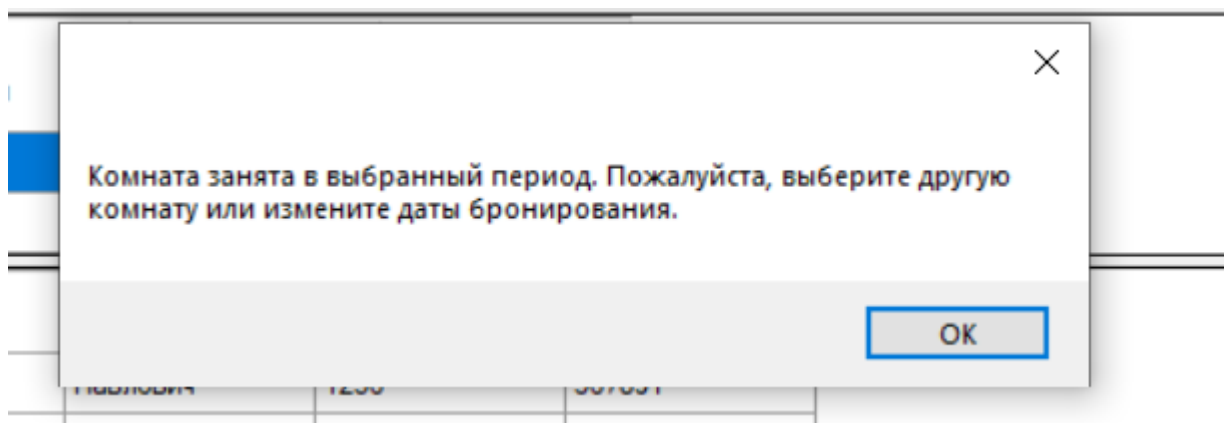


Рисунок 37 – уведомление о том, что выбранная комната уже занята

Заключение

В рамках данной курсовой работы была разработана автоматизированная информационная система (АИС) для гостиницы с использованием SQLite и языка программирования C# в среде Windows Forms. Основной целью проекта было создание удобного и эффективного инструмента для управления процессами бронирования номеров и учета клиентов, что позволяет улучшить качество обслуживания и оптимизировать работу гостиницы.

В процессе разработки был выполнен анализ требований, спроектирована архитектура базы данных, и реализован пользовательский интерфейс, обеспечивающий интуитивно понятное взаимодействие с системой. Особое внимание было уделено ключевым аспектам, таким как безопасность данных, целостность информации и удобство использования.

Применение SQLite в качестве СУБД обеспечило легкость интеграции и достаточную производительность для решения поставленных задач. Использование языка C# и платформы Windows Forms позволило создать функциональное и визуально приятное приложение, которое удовлетворяет потребностям как сотрудников гостиницы, так и ее клиентов.

Проведенное тестирование подтвердило работоспособность и надежность разработанной системы. Основные функции, такие как управление бронированиями, учет клиентов, генерация отчетов и авторизация пользователей, показали стабильные результаты и высокую степень соответствия заданным требованиям.

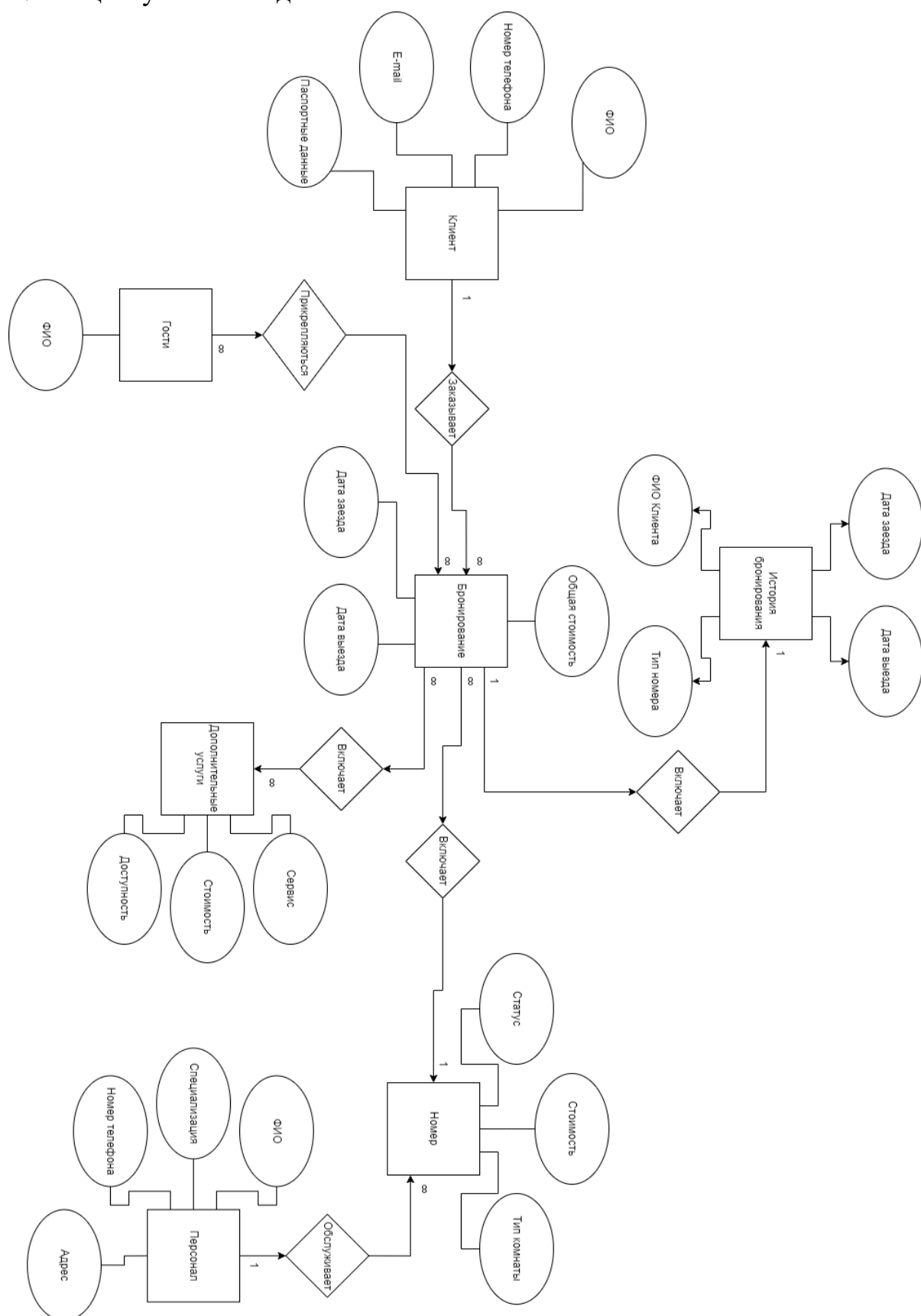
Таким образом, выполненная работа демонстрирует возможность создания эффективных информационных систем для гостиничного бизнеса с использованием современных технологий. Разработанная АИС может быть легко адаптирована и масштабирована для других предприятий гостиничного сектора, что открывает перспективы для дальнейшего развития и совершенствования системы.

Список литературы

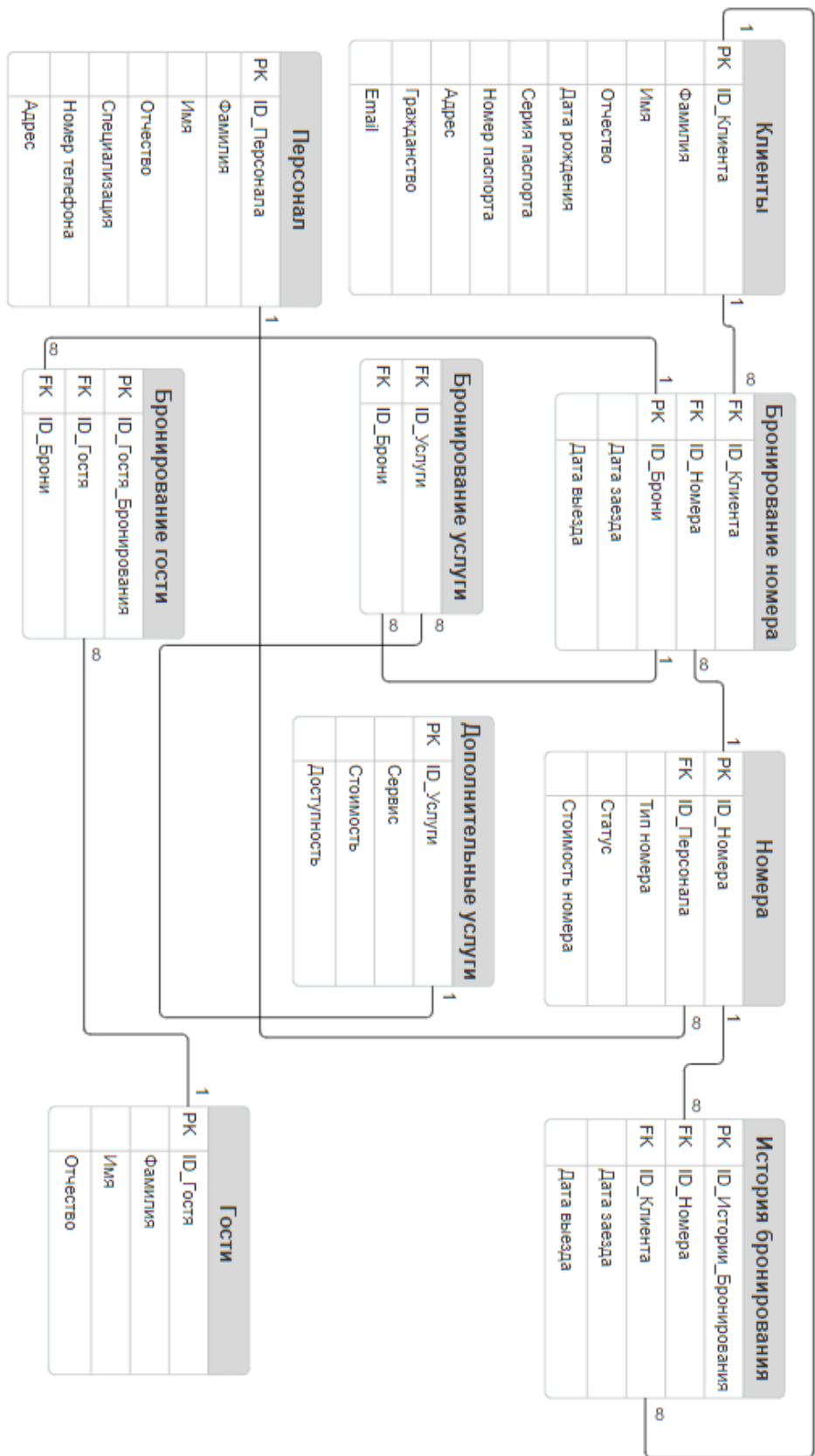
1. Гарсия-Молина, Х., Ульман, Дж. Д., Видом, Дж. "Основы систем баз данных." – Москва: Вильямс, 2020. – 336с.
2. Грин, М. "Пособие по SQLite: основы и практическое применение." – СПб.: БХВ-Петербург, 2018. – 369с.
3. Кортес, Р., Барчулевич, Е. В., Артемьев, А. А. "Базы данных. Теория и практика." – Москва: Научный мир, 2019. – 459с.
4. Кауфман, О. "SQLite для профессионалов." – СПб.: Питер, 2020. – 576с.
5. Троелсен, Э., Джепперсен, Ф. "Язык программирования C# 8.0 и платформы .NET Core 3.0." – СПб.: Питер, 2020. – 819с.
6. Хортон, М. "C# и .NET: руководство для начинающих." – СПб.: Символ-Плюс, 2019. – 413с.

Приложение 1. Модели данных

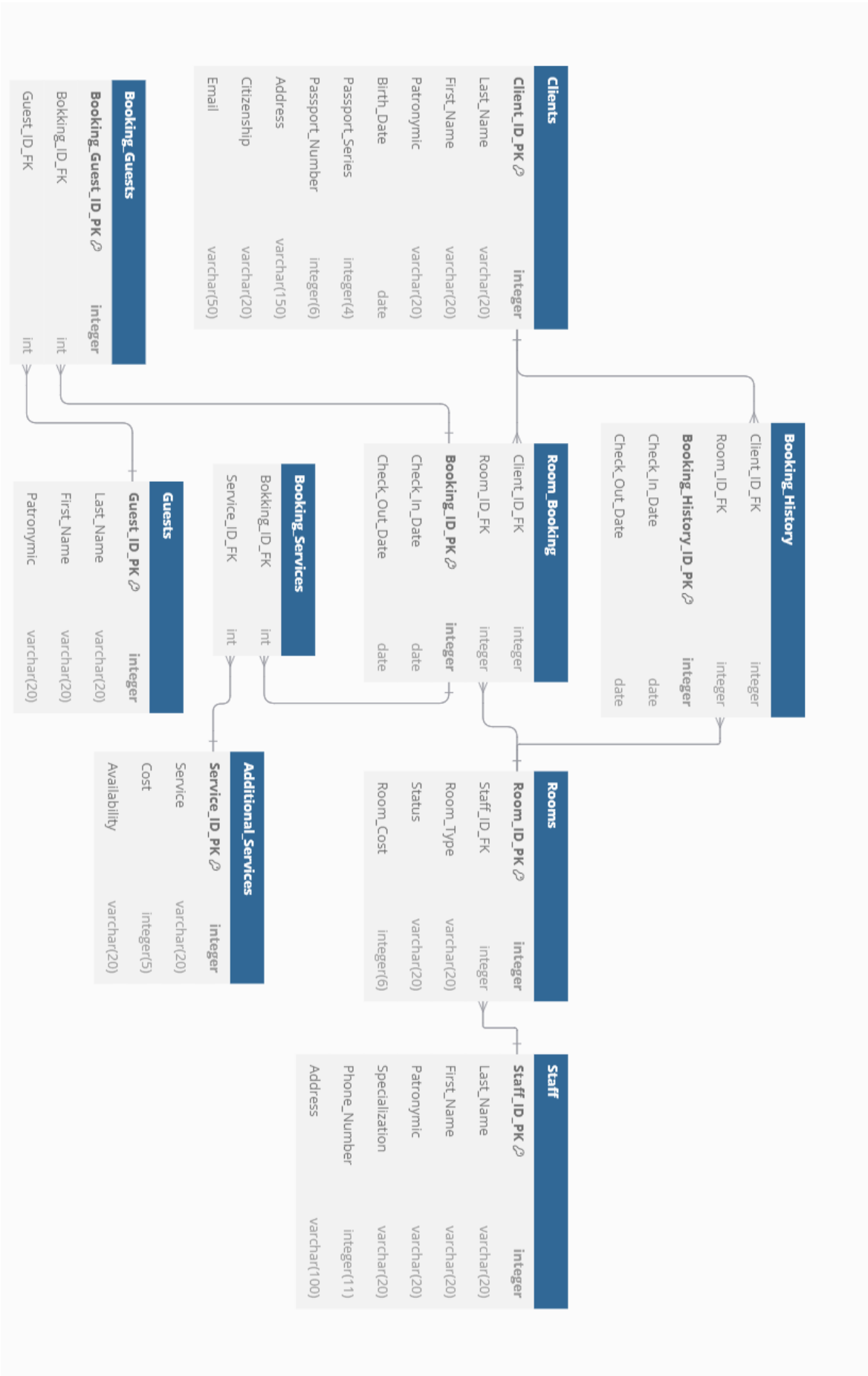
1. Концептуальная модель



2. Логическая модель



3. Физическая модель



Приложение 2. Текст программы

Вся программа расположена на Git: <https://github.com/depxd/hotel3/tree/main>

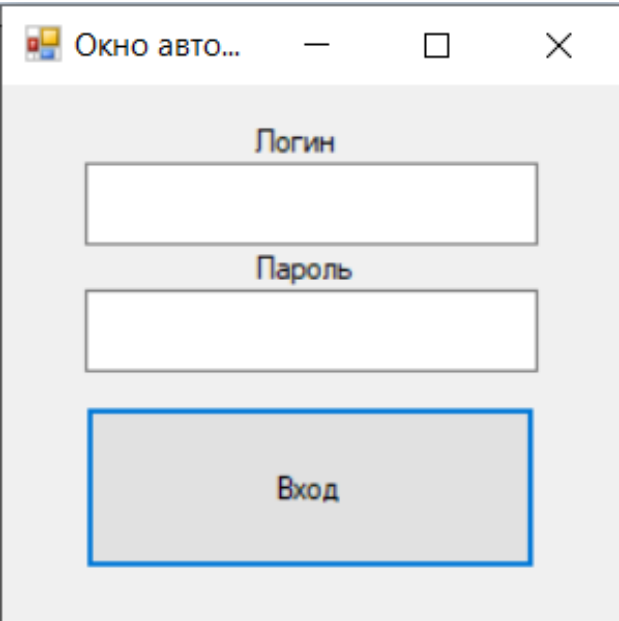


Рисунок 38 – окно авторизации

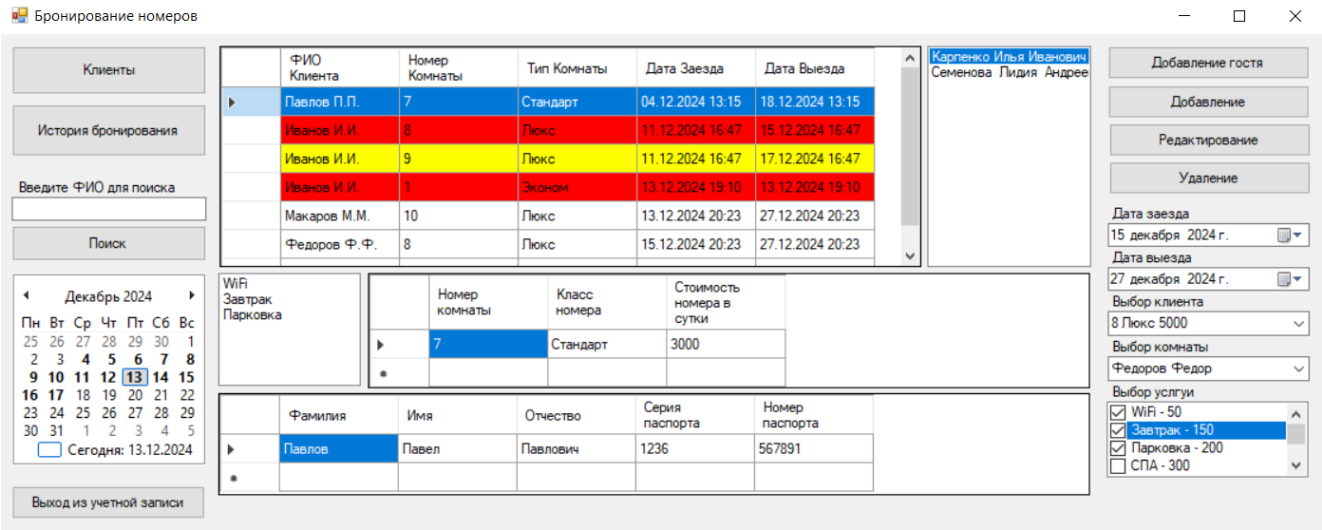


Рисунок 39 – окно бронирования

ClientForm

Введите ФИО для поиска в таблице клиентов

Поиск

	Фамилия	Имя	Отчество	Дата Рождения	Серия Паспорта	Номер Паспорта	Адрес	Гражданство	Email
▶	Иванов	Иван	Иванович	01.01.1990	1234	567890	Москва, Красна...	Россия	ivanov@example...
	Петров	Петр	Петрович	14.02.1985	2345	678901	Санкт-Петербур...	Россия	petrov@example...
	Сидоров	Сидр	Сидорович	25.03.1980	3456	789012	Казань, улица Б...	Россия	sidorov@examl...
	Кузнецов	Алексей	Алексеевич	18.04.1975	4567	890123	Новосибирск, К...	Россия	kuznetsov@exa...
	Попов	Сергей	Сергеевич	10.05.1982	5678	901234	Екатеринбург, у...	Россия	popov@example...
	Васильев	Василий	Васильевич	22.06.1991	6789	412345	Челябинск, ули...	Россия	vasiliev@examl...
	Смирнов	Андрей	Андреевич	15.07.1983	7890	123456	Омск, улица Ле...	Россия	smimov@examl...
	Михайлов	Михаил	Михайлович	30.08.1979	8901	234567	Самара, улица ...	Россия	mikhailov@exam...
	Федоров	Федор	Федорович	19.09.1988	9012	345678	Уфа, проспект ...	Россия	fedorov@examl...
	Николаев	Николай	Николаевич	11.10.1984	1237	456789	Волгоград, улиц...	Россия	nikolaev@examp...
	Павлов	Павел	Павлович	01.11.1986	1236	567891	Калининград, ул...	Россия	pavlov@example...
	Алексеев	Алексей	Алексеевич	24.12.1992	2348	678902	Томск, улица Га...	Россия	alekseev@exam...
	Андреев	Андрей	Андреевич	12.01.1993	3456	789013	Пермь, улица К...	Россия	andreev@examl...
	Макаров	Макар	Макарович	05.02.1987	4567	890124	Саранск, просп...	Россия	makarov@examp...
	Григорьев	Григорий	Григорьевич	18.03.1981	5678	901235	Киров, улица Ок...	Россия	grigorev@examl...
	Лебедев	Лев	Львович	11.04.1989	6789	182346	Владимир, улиц...	Россия	lebedev@examl...
	Киселев	Кирилл	Кириллович	20.05.1978	7890	123457	Воронеж, просп...	Россия	kiselev@example...
	Семенов	Семен	Семенович	17.06.1976	8901	234568	Тула, улица Сов...	Россия	semenov@exam...
	Степанов	Степан	Степанович	09.07.1990	9012	345679	Рязань, улица Е...	Россия	stepanov@exam...
	Егоров	Егор	Егорович	14.08.1985	1239	456790	Курск, улица Пе...	Россия	egorov@example...

Фамилия

Имя

Отчество

Дата рождения

Серия паспорта

Номер паспорта

Полный адрес

Гражданство

Электронная почта

Добавление

Редатирование клиента

Удаление записи

Рисунок 40 – окно клиенты

HistoryForm

Удаление записи из истории бронирования

Поиск

Введите ФИО для поиска

	ФИО Клиента	Номер Комнаты	Тип Комнаты	Дата Заезда	Дата Выезда
▶	Петров П.П.	2	Эконом	2024-12-04 13:1...	2024-12-05 13:1...
	Кузнецов А.А.	3	Эконом	2024-12-04 13:1...	2024-12-07 13:1...
	Семенов С.С.	9	Люкс	2024-12-04 13:1...	2024-12-10 13:1...
*					

Рисунок 41 – окно история бронирования

GuestForm

Карпенко Илья Иванович
Семенова Лидия Андреевна
Иванов Максим Иванович
Павлова Екатирина Андреевна
Макров Олег Макарьевич
Федорова Ольга Максимовна

Карпенко Илья Иванович
Семенова Лидия Андреевна
Павлова Екатирина Андреевна

Фамилия
Имя
Отчество

Добавить нового гостя

Удалить гостя

Прикрепить гостя к бронированию

Открепить гостя от брони

Редактирование гостя

Рисунок 42 – окно гостя

Меню системного администратора

Номера

Сотрудники

Дополнительные услуги

Выход

Рисунок 43 – меню системного администратора

RoomFrom

	Тип комнаты	Статус	Стоимость	Сотрудник
▶	Эконом	Доступен для б...	2000	Соловьев Егор
	Эконом	Доступен для б...	2000	Крылова Мария
	Эконом	Доступен для б...	2000	Соловьев Егор
	Эконом	Доступен для б...	2000	Крылова Мария
	Стандарт	Доступен для б...	3000	Горбачева Ирина
	Стандарт	Доступен для б...	3000	Горбачева Ирина
	Стандарт	Доступен для б...	3000	Шестакова Елена
	Люкс	Доступен для б...	5000	Горбачева Ирина
	Люкс	Доступен для б...	5000	Соловьев Егор
	Люкс	Доступен для б...	5000	Шестакова Елена
*				

Тип комнаты

Статус

Стоимость

Закрепить сотрудника за номером

Кузнецов Николай ▾

Добавить

Редактировать

Удалить

Рисунок 44 – окно номера

StuffForm

	Фамилия	Имя	Отчество	Специализация	Номер телефона	Адрес
▶	Кузнецов	Николай	Петрович	Manager	79001112233	Москва, Тверск...
	Смирнов	Алексей	Иванович	Cleaner	79001234567	Санкт-Петербур...
	Попов	Сергей	Сергеевич	Technician	79007654321	Новосибирск, К...
	Соколов	Александр	Викторович	Receptionist	79009876543	Екатеринбург, ...
	Новиков	Дмитрий	Алексеевич	Security	79005432109	Казань, Пушкин...
	Федоров	Анатолий	Иванович	Chef	79007651234	Уфа, Октября, 20
	Морозов	Максим	Сергеевич	Gardener	79003214567	Самара, Набере...
	Волков	Владимир	Петрович	Electrician	79008765432	Волгоград, Тра...
	Зайцев	Григорий	Андреевич	Plumber	79004567891	Нижний Новгор...
	Соловьев	Егор	Дмитриевич	Housekeeper	79003219876	Челябинск, Ком...
	Крылова	Мария	Александровна	Housekeeper	79004561234	Москва, Садова...
	Шестакова	Елена	Викторовна	Housekeeper	79005432187	Санкт-Петербур...
	Горбачева	Ирина	Сергеевна	Housekeeper	79007654320	Казань, Лесная...
	Иванова	Ольга	Ивановна	Receptionist	79003219865	Новосибирск, С...
	Петрова	Татьяна	Петровна	Receptionist	79007865432	Екатеринбург, ...

Введите ФИО для поиска

Поиск

Фамилия

Имя

Отчество

Специализация

Номер телефона

Адрес

Добавить

Редактировать

Удалить

Рисунок 45 – окно сотрудники

ServiceForm

	Услуга	Стоимость	Доступность
▶	WiFi	50	Активно
	Завтрак	150	Активно
	Парковка	200	Активно
	СПА	300	Активно
	Трансфер	400	Активно
	Фитнес	250	Активно
	Бассейн	500	Активно
	Прачечная	100	Активно

Дополнительная услуга

Стоимость

Доступность

Добавить
Редактировать
Удалить

Рисунок 46 – окно дополнительные услуги

Рабочее окно сотрудника

Введите ФИО для поиска

	ФИО сотрудника	Номер комнаты	Тип комнаты
▶	Соловьев Егор ...	1	Эконом
	Крылова Мария...	2	Эконом
	Соловьев Егор ...	3	Эконом
	Крылова Мария...	4	Эконом
	Горбачева Ирин...	5	Стандарт
	Горбачева Ирин...	6	Стандарт
	Шестакова Еле...	7	Стандарт
	Горбачева Ирин...	8	Люкс
	Соловьев Егор ...	9	Люкс
	Шестакова Еле...	10	Люкс
*			

Рисунок 47 – рабочее окно сотрудника