Министерство образования Республики Беларусь

Учреждение образования

Белорусский государственный университет информатики и радиоэлектроники

Институт информационных технологий

Факультет повышения квалификации и переподготовки

Кафедра «Микропроцессорные системы и сети»

КУРСОВОЙ ПРОЕКТ

на тему: Программное средство по бронированию гостиничных номеров

Выполнил:

Слушатель гр. 30333

Калевич С.М.

Проверила:

Желакович И. М.

МИНСК 2024

Министерство образования Республики Беларусь

Учреждение образования

«Белорусский государственный университет

информатики и радиоэлектроники»

Институт информационных технологий

Кафедра «Микропроцессорные системы и сети»

|  |  |  |
| --- | --- | --- |
|  | УТВЕРЖДАЮ | |
|  | Заведующий кафедрой МПСС | |
|  | *И.В.Кашникова* | |
|  |  | |
|  | (подпись) | |
|  |  | 2024 г. |

ЗАДАНИЕ

по курсовому проектированию

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Слушателю | *Калевичу Сергею Михайловичу* | | | |
| Группа | *30333* | | | |
| 1. Тема работы | *Программное средство по бронированию гостиничных номеров* | | | |
| 2. Срок сдачи студентом законченной работы | | | |  |
| 3. Исходные данные к работе | |  | | |
| *Приложение должно предоставлять пользователю возможность выполнять следующие действия: регистрировать заявки на заселение; при отсутствии свободных номеров, соответствующих поданной заявке, предлагать либо другую возможную дату заселения, либо возможный срок пребывания для даты, указанной в заявке; выводить отчет заявках, принятых за указанный период; выводить отчет свободных номерах на указанную дату.*  *Приложение должно выводить в виде диаграмм следующие данные: занятость номеров в указанный период времени; выводить отчет свободных номерах на указанную дату.* | | | | |
| Язык программирования | | | *С#* | |
| Среда разработки | | | *Microsoft Visual Studio2022* | |
| Вид приложения | | | *desktop (WPF)* | |
| Парадигма программирования | | | *ООП* | |
| Способ организации данных | | | *СУБД MySQL* | |
| Технология доступа к данным | | | *Entity Framework* | |
| Архитектурный паттерн | | | *MVVM* | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 4. Содержание расчетно-пояснительной записки (перечень вопросов, которые подлежат разработке) | | | | | |
| *1)* | | *Введение* | | | |
| *2)* | | *Аналитический обзор и постановка задачи* | | | |
| *3)* | | *Разработка методов и моделей* | | | |
| *4)* | | *Разработка проекта программного обеспечения* | | | |
| *5)* | | *Разработка алгоритмов и их описание* | | | |
| *6)* | | *Тестирование полученного программного продукта* | | | |
| *7)* | | *Заключение* | | | |
| *8)* | | *Приложение: Листинг кода с комментариями* | | | |
| 5. Перечень графического материала (с точным обозначением обязательных чертежей и графиков) | | | | | |
| *1)* | | *трассировочная матрица* | | | |
| *2)* | | *USE CASE диаграммы* | | | |
| *3)* | | *схема ресурсов системы* | | | |
| *4)* | | *схема базы данных* | | | |
| *5)* | | *диаграмма классов* | | | |
| *6)* | | *макеты окон* | | | |
| *7)* | | *3 схемы алгоритмов* | | | |
| 6. Консультант по работе | | | | *И.М.Желакович* | |
| 7. Дата выдачи задания | | | | *02.10.2024* | |
| 8. Календарный график работы на весь период проектирования (с обозначением сроков выполнения и трудоемкости отдельных этапов): | | | | | |
| *Разделы 1,2* | | | *01.12.2024 г.* | | *25%* |
| *Раздел 3,4* | | | *03.01.2025 г.* | | *35%* |
| *Раздел 5,6* | | | *12.02.2025 г.* | | *30%* |
| *Раздел 7* | | | *09.03.2025 г.* | | *10%* |
|  | | |  | |  |
| РУКОВОДИТЕЛЬ | | |  | | *И.М.Желакович* |
|  | | | (подпись) | |  |
|  | | |  | |  |
| Задание принял к исполнению | | | *02.10.2024* | |  |
|  | | | (дата) | | (подпись слушателя) |
|  |  | | | | |

**СОДЕРЖАНИЕ**

|  |  |
| --- | --- |
| Введение……………………………………………………………………... | 5 |
| 1  Аналитический обзор и постановка задачи…………………………….. | 6 |
| 1.1  Обзор существующих программных решений для складского учета……………………………………………………….…………………. | 6 |
| 1.2  Анализ предметной области и постановка задач……………..………. | 10 |
| 2  Разработка методов и моделей…………………………….…………….. | 12 |
| 3  Разработка проекта программного обеспечения……………………...... | 14 |
| 3.1  Функциональный анализ предметной области….……………………. | 14 |
| 3.2  Разработка схема базы данных..………………….……………………. | 16 |
| 3.3  Разработка диаграммы классов…….…………….……………………. | 21 |
| 3.4  Разработка графического интерфейса пользователя…………………. | 22 |
| 4  Разработка алгоритмов и их описание….……………………………...... | 29 |
| 5  Тестирование полученного программного продукта…….…………….. | 32 |
| Заключение…………………………………………………………………... | 37 |
| Список использованных источников………………………………………. | 38 |
| Приложение А  (обязательное) Программный код алгоритмов расчета остатков товаров на пунктах хранения ………...………. | 39 |
| Приложение Б (обязательное) Листинг кода разработанного приложения | 40 |

ВВЕДЕНИЕ

В условиях динамично развивающегося гостиничного бизнеса автоматизация процессов обслуживания клиентов становится критически важной. Одним из ключевых аспектов успешной работы гостиницы является эффективная регистрация заявок на заселение. В гостиничном бизнесе, где важны точность, скорость и удобство обслуживания клиентов, внедрение программного обеспечения для регистрации заявок на заселение играет ключевую роль, поэтому разработка программного обеспечения для регистрации заявок на заселение в номера гостиницы является актуальной задачей, способствующей улучшению качества обслуживания клиентов и оптимизации внутренних процессов гостиницы.

Настоящая курсовая работа посвящена разработке программного средства, которое будет использоваться для автоматизации процесса регистрации заявок на заселение в номера гостиницы. Основной целью данной работы является создание эффективного, удобного и надежного инструмента, который позволит улучшить качество обслуживания клиентов, сократить время обработки заявок и минимизировать вероятность ошибок, связанных с человеческим фактором. Это программное средство позволит автоматизировать процесс обработки заявок, обеспечивая удобство для клиентов и сотрудников гостиницы. Важными задачами разработки являются создание интуитивно понятного интерфейса, обеспечение надежности и безопасности данных, а также интеграция с существующими системами управления гостиницей.

В ходе разработки будут рассмотрены существующие решения в области программного обеспечения для гостиничного бизнеса, проведен анализ их преимуществ и недостатков. На основе этого анализа будет разработан собственный проект программного средства, включающий подробное описание архитектуры системы, функциональных возможностей и этапов реализации. Кроме того, в работе будет представлен план разработки, включающий этапы проектирования, реализации, тестирования и внедрения программы.

Создание программного обеспечения для регистрации заявок на заселение в гостиничные номера является актуальной задачей, так как способствует повышению уровня сервиса и удовлетворенности клиентов, а также оптимизации работы гостиницы в целом.

1 АНАЛИТИЧЕСКИЙ ОБЗОР И ПОСТАНОВКА ЗАДАЧИ

1.1 Обзор существующих программных решений для гостиничного бизнеса

Современные программные решения для гостиничного бизнеса помогают автоматизировать процессы, улучшить обслуживание клиентов и повысить эффективность управления. К наиболее распространенным можно отнести:

**1. OZLocks Hotelier.**

Преимущества:

* Автоматизация всех аспектов управления гостиницей.
* Возможность создания виртуальной модели реальной гостиницы.
* Улучшение управления бронированием и учетом клиентов.

Недостатки:

* Высокая стоимость лицензии.
* Может потребоваться дополнительная настройка для интеграции с другими системами.

**2. Logus Hospitality Management System**

Преимущества:

* Специально разработано для российского рынка.
* Поддержка всех основных функций управления гостиницей.
* Удобный интерфейс и интуитивно понятные функции.

Недостатки:

* Может быть сложным в использовании для новых сотрудников.
* Необходимость регулярных обновлений и поддержки.

**3. FIDELIO**

Преимущества:

* Эффективное бронирование номерного фонда.
* Поддержка гостиниц различной вместимости.
* Повышение качества обслуживания и лояльности гостей.

Недостатки:

* Может потребоваться дополнительная настройка для интеграции с другими системами.
* Высокая стоимость лицензии.

**4. TravelLine**

Преимущества:

* Автоматизация бизнес-процессов для различных типов гостиничных объектов.
* Удобный интерфейс и интуитивно понятные функции.
* Поддержка множества функций управления отелем.

Недостатки:

* Может потребоваться дополнительная настройка для интеграции с другими системами.
* Высокая стоимость лицензии.

**5. 1С. Отель**

Преимущества:

* Автоматизация всех основных процессов управления гостиницей.
* Поддержка различных функций, таких как управление номерным фондом и бронированием.
* Удобный интерфейс и интуитивно понятные функции.

Недостатки:

* Может потребоваться дополнительная настройка для интеграции с другими системами.
* Высокая стоимость лицензии.

Эти программные решения помогают гостиничному бизнесу повысить эффективность и качество обслуживания клиентов, но каждое из них имеет свои особенности и требует тщательного выбора и настройки под конкретные нужды гостиницы.

1.2 Анализ предметной области и постановка задач

Анализ предметной области программных средств для бронирования гостиничных номеров включает изучение ключевых аспектов, таких как:

1. Функциональность:

* Управление бронированием: автоматизация процесса приема и обработки заявок.
* Интеграция с платежными системами: обеспечение удобной оплаты.
* Аналитика и отчеты: сбор данных для анализа загрузки и доходности.
* Использование данных и анализа поведения гостей для предоставления персонализированных услуг и улучшения клиентского опыта.

1. Технологии:

* Облачные решения: позволяют управлять системой из любой точки мира.
* Мобильные приложения: удобство для клиентов и сотрудников.
* Искусственный интеллект: персонализация предложений и прогнозирование спроса.

1. Тенденции:

* Цифровизация и автоматизация для управления работой гостиницы.
* Увеличение использования AI и машинного обучения.
* Интеграция с платформами OTA (Online Travel Agencies).
* Повышение уровня безопасности данных.

Таким образом созданное приложение решает ряд задач:

* регистрирует заявки клиентов на заселение;
* при отсутствии свободных номеров, соответствующих поданной заявке, предлагает либо другую возможную дату заселения, либо возможный срок пребывания для даты, указанной в заявке;
* выводит отчет о заявках, принятых за указанный период;
* выводит отчет о свободных номерах на указанную дату.

Для выполнения этих задач использованы следующие программные решения:

* Язык программирования – C#.
* Платформа – WPF.NET.
* Среда разработки – Microsoft Visual Studio.
* Парадигма программирования – объектно-ориентированная.
* Способ хранения данных – реляционная БД MySql.

2 РАЗРАБОТКА МЕТОДОВ И МОДЕЛЕЙ

При моделировании программного средства использовались следующие этапы:

1. Моделирование предметной области

* Концептуальная модель (как описано ранее): сущности, связи и атрибуты.
* Процессы: бронирование, отмена, подтверждение брони, интеграция с платежами.
* Создание UML-диаграмм: диаграмм классов, последовательности и состояния.

3. Разработка алгоритмов и методов

4. Разработка пользовательского интерфейса

5. Тестирование

Сам процесс разработки начинается с создания концептуальной модели. Концептуальная модель предметной области "Программное средство по бронированию гостиничных номеров" структурирует основные сущности, их атрибуты и взаимодействия.

**Описание концептуальной модели.**

1. Сущности:

* Гостиницы: включает данные об отелях, таких как название, категория, местоположение, количество номеров.
* Номера: номера гостиниц с характеристиками, такими как тип (стандартный, люкс), стоимость, доступность.
* Клиенты: пользователи, бронирующие номера, с информацией о ФИО, контактах и предпочтениях.
* Бронирования: данные о резервировании номеров, включая даты заезда/выезда, статус (подтверждено, отменено).
* Платежи: информация о способах оплаты, суммах и статусе транзакций.

1. Атрибуты сущностей:

* Гостиницы: название, адрес, рейтинг, количество звезд.
* Номера: ID, тип, цена, статус (свободен/занят).
* Клиенты: ID клиента, ФИО, контактные данные.
* Бронирования: номер брони, даты, связанный клиент и номер.
* Платежи: номер транзакции, дата, сумма, статус.

1. Связи:

* Гостиницы ↔ Номера: "Гостиницы содержат номера."
* Номера ↔ Бронирования: "Номера участвуют в бронированиях."
* Клиенты ↔ Бронирования: "Клиенты осуществляют бронирования."
* Бронирования ↔ Платежи: "Бронирования оплачиваются через платежи."

1. Функциональные компоненты системы:

* Управление списком гостиниц и номеров.
* Бронирование номеров с учетом их доступности.
* Генерация отчетов по бронированиям и клиентам.
* Интеграция с платежными системами для онлайн-оплат.

Эта модель служит основой для проектирования базы данных и функциональных возможностей системы.

3 РАЗРАБОТКА ПРОЕКТА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

3.1 Функциональный анализ предметной области

Анализ предметной области позволяет выделить ее сущности, определить первоначальные требования к функциональности и определить границы проекта. Модель предметной области должна быть документирована, храниться и поддерживаться в актуальном состоянии до этапа реализации.

Для визуализации связей между элементами системы разработана трассировочную матрицу. Трассировочная матрица проекта представлена в таблице 1.

Таблица 1 – Трассировочная матрица

|  |  |  |  |
| --- | --- | --- | --- |
| № | **Требование** | **Функциональность** | **Методы реализации** |
| 1 | Поиск номеров по датам | Возможность поиска свободных номеров по дате | Запрос к базе данных с фильтрацией по условиям |
| 2 | Бронирование номера | Резервирование выбранного номера и фиксация | Изменение статуса номера в базе данных |
| 3 | Управление учетными записями клиентов | Регистрация, вход, хранение данных пользователей | Авторизация с помощью безопасного хранения |
| 4 | Генерация отчетов | Формирование отчетов по бронированиям, клиентам | SQL-запросы для генерации отчетов |
| 5 | Управление доступностью номеров | Автоматическое обновление статуса номеров | Фоновый процесс синхронизации |

Функциональные требования иллюстрирует диаграммы вариантов использования.

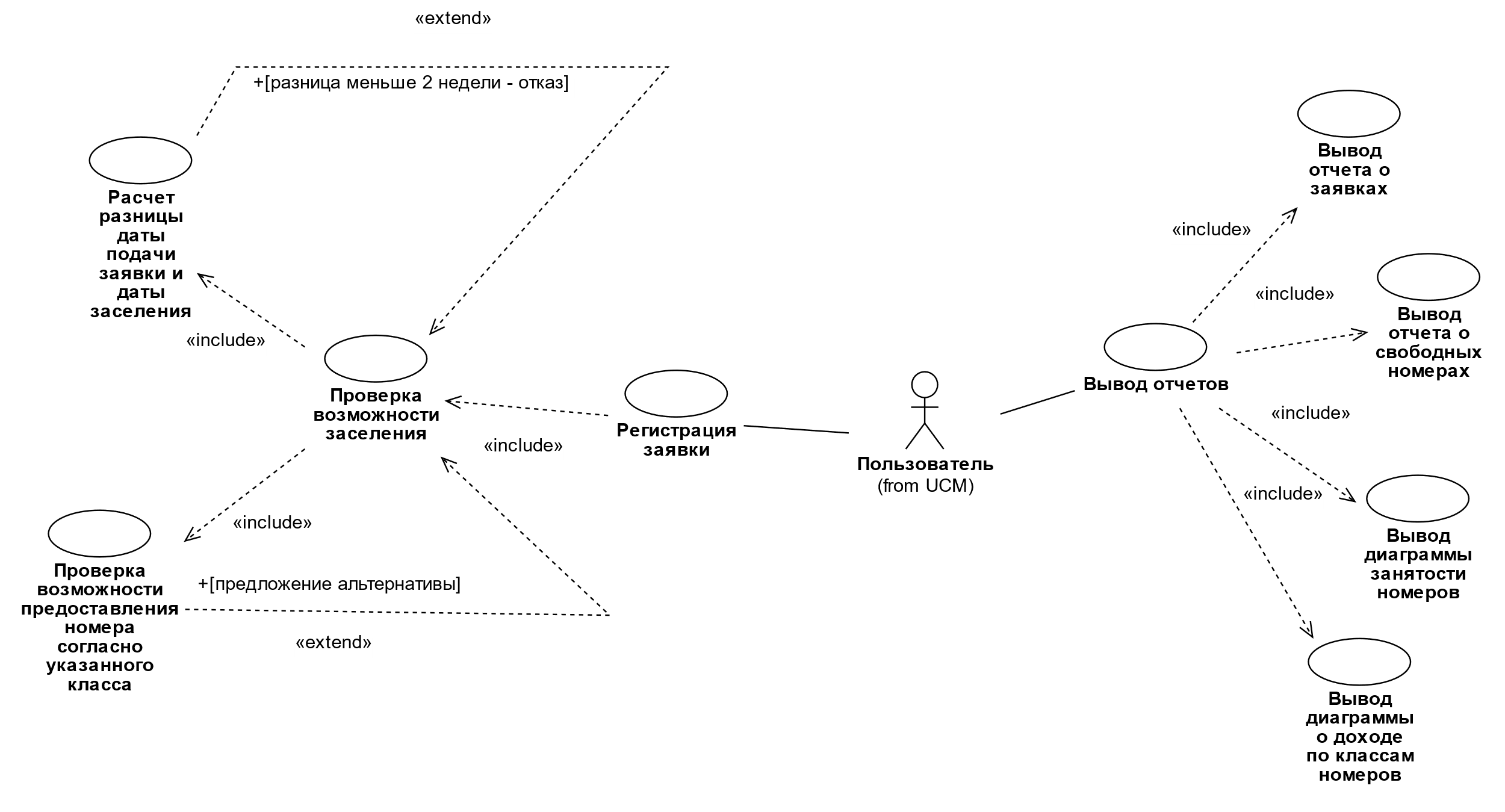


Рисунок 1 – Диаграмма вариантов использования программы пользователем

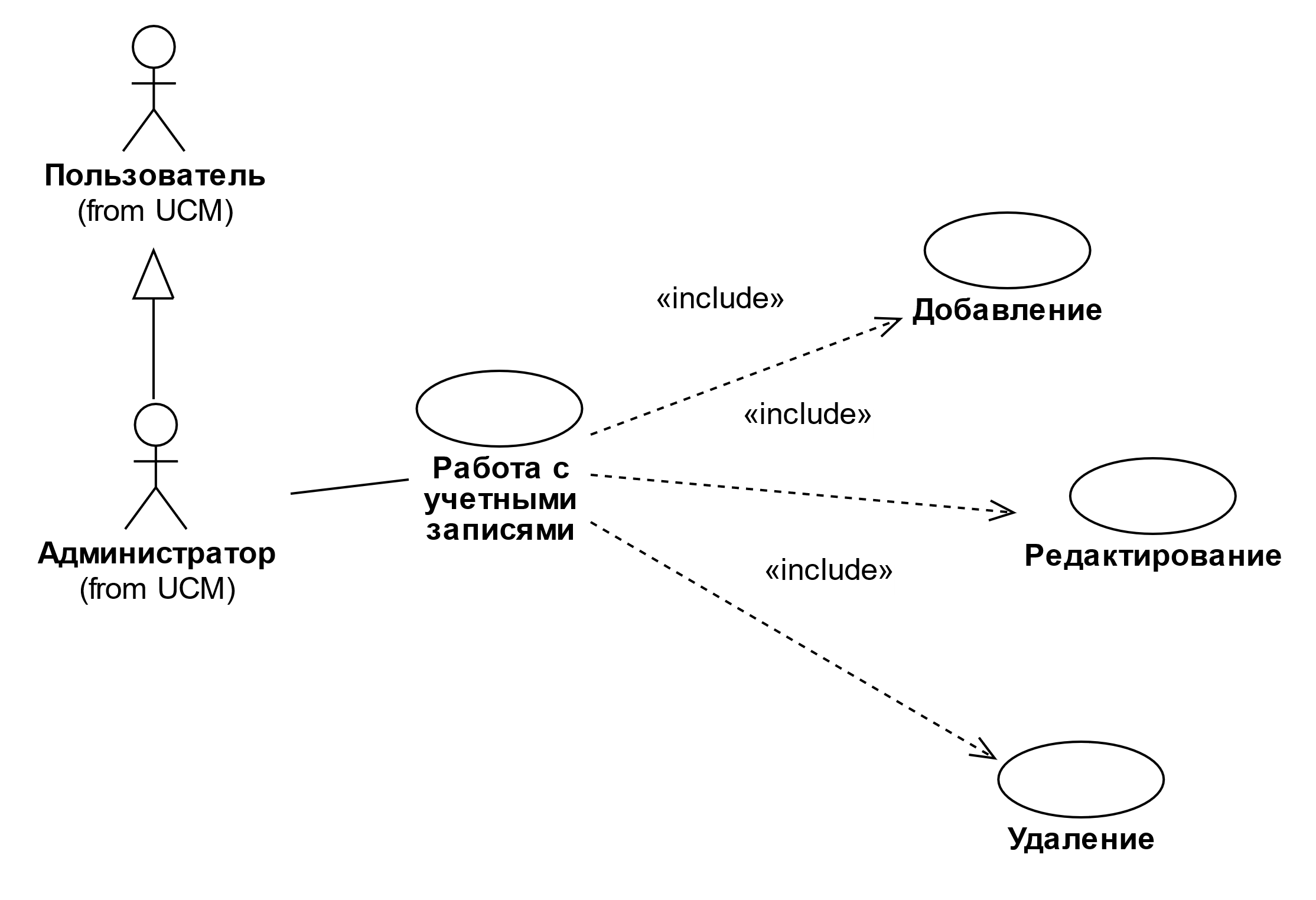


Рисунок 2 – Диаграмма вариантов использования программы администратором

Для отображения конфигурации блоков данных и обрабатывающих блоков используется схема ресурсов системы.

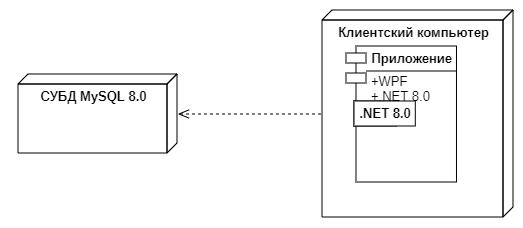


Рисунок 3 – Схема ресурсов системы

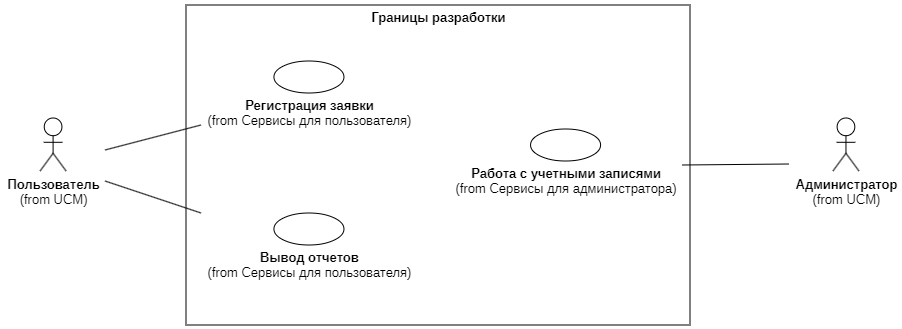


Рисунок 4 – Границы разработки

3.2 Разработка схемы базы данных

Схема БД должна включать набор всех схем ее таблиц, т.е. описание всех колонок этих таблиц (их типов, допустимых значений, связей между таблицами, типа внешних ключей и т.д.), без учета конкретных данных.

При составлении схемы БД разрабатываемой программы создано необходимое количество таблиц для хранения нужной информации – Balanses, Movements, ItemMovs, Models, Products, Warehouses, Colors, Sizes, Users, UserRoles.

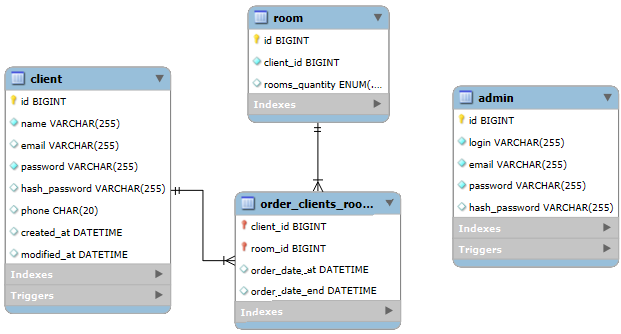


Рисунок 5 – Схема базы данных

Сущность «Клиент» предназначена для хранения информации о пользователях, подавших заявки на заселение. Описание атрибутов представлено в таблице 2.

Таблица 2 − Клиент (client)

|  |  |  |  |
| --- | --- | --- | --- |
| Атрибут | Тип данных | Описание | Обязательный атрибут |
| id (РК) | BIGINT | Код клиента | NOT NULL |
| name | VARCHAR | ФИО клиента | NOT NULL |
| email | VARCHAR | email клиента | NULL |
| password | VARCHAR | Пароль | NOT NULL |
| hash\_password | VARCHAR | Хэшированный пароль | NULL |
| phone | CHAR | Номер телефона | NULL |
| created\_at | DATETIME | Дата создания | NULL |
| modified\_at | DATETIME | Дата изменения | NULL |

Сущность «Номер» предназначена для хранения информации о номерах гостиницы. Описание атрибутов представлено в таблице 3.

Таблица 3 − Клиент (client)

|  |  |  |  |
| --- | --- | --- | --- |
| Атрибут | Тип данных | Описание | Обязательный атрибут |
| id (РК) | BIGINT | Код клиента | NOT NULL |
| client\_id (FK) | BIGINT | ФИО клиента | NOT NULL |
| rooms\_quantity | ENUM | Количество мест в номере | NULL |

Сущность «Заявки» предназначена для хранения информации о поданых заявках. Описание атрибутов представлено в таблице 4.

Таблица 3 − Заявки (client)

|  |  |  |  |
| --- | --- | --- | --- |
| Атрибут | Тип данных | Описание | Обязательный атрибут |
| Client\_id (РК) | BIGINT | Код клиента | NOT NULL |
| Room\_id (РК) | VARCHAR | ФИО клиента | NOT NULL |
| Order\_date\_at | VARCHAR | email клиента |  |
| Order\_date\_end | VARCHAR | Пароль |  |

Сущность «Заявки» предназначена для хранения информации об адмистраторах. Описание атрибутов представлено в таблице 5.

Таблица 3 − Администратор (client)

|  |  |  |  |
| --- | --- | --- | --- |
| Атрибут | Тип данных | Описание | Обязательный атрибут |
| id (РК) | BIGINT | Код клиента | NOT NULL |
| name | VARCHAR | ФИО клиента | NOT NULL |
| password | VARCHAR | Пароль | NOT NULL |
| hash\_password | VARCHAR | Хэшированный пароль | NULL |

3.4 Разработка графического интерфейса пользователя

Исходя из диаграммы вариантов использования и этапа анализа требований был спроектирован простой и интуитивно понятный графический пользовательский интерфейс. Макеты разрабатываемых окон и описание полей приведены на рисунках 6 - 7

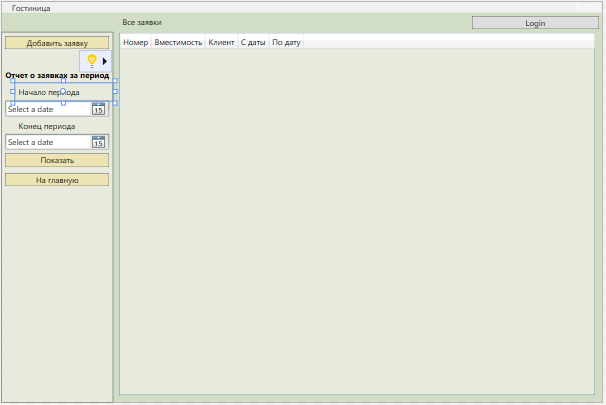


Рисунок 6 – Макет лавного окна

Описание:

* Кнопка «Добавить заявку» – открвыает вспомогательное окно для ввода данных для подачи заявки.
* DatePicker «Начало периода» – выбор начальной даты для вывода отчета.
* DatePicker «Конец периода» – выбор конечной даты для вывода отчета
* Кнопка «Показать» – выводит отчет.
* Кнопка «На главную» – возвращает начальные данные.
* Кнопка «Login» – открывает вспомогательное окно для входа в систему.

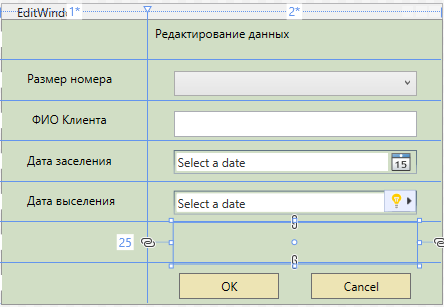


Рисунок 6 – Макет вспомогательного окна

Описание:

* Комбобокс «Размер номера» – выбирает на сколько мест следует выбрать номер.
* Поле «ФИО Клиента» – для ввода ФИО клиента.
* DatePicker «Дата заселения» – выбор планируемой даты заселения.
* DatePicker «Дата выселения» – выбор планируемой даты заселения..
* Кнопка «OК» – для подтверждения выбора.
* Кнопка «Cancel» – для отмены выбора.

4 РАЗРАБОТКА АЛГОРИТМОВ И ИХ ОПИСАНИЕ

Исходя из поставленных задач разрабатываются следующие алгоритмы:

**1. Регистрация заявок на заселение**

Цель: Создать новую запись заявки на заселение с указанием деталей (даты, номера, клиента).

Шаги алгоритма:

1. Пользователь вводит данные заявки (дата заезда/выезда, тип номера, имя клиента).
2. Проверить наличие свободного номера на указанные даты.
3. Если номер доступен:

* Создать новую запись в базе заявок.
* Изменить статус номера на "Забронирован".
* Уведомить клиента о подтверждении заявки.

1. Если номер недоступен, передать управление следующему алгоритму (поиск альтернатив).

**2. Предложение альтернатив при отсутствии свободных номеров**

Цель: Найти возможную альтернативу для указанной даты или предложить доступные даты.

**Шаги алгоритма:**

1. 1Проверить, имеются ли аналогичные номера на другие даты, ближайшие к указанным.
2. Если таких дат нет, определить возможный срок пребывания на указанные даты.
3. Сформировать список предложений и отправить клиенту.

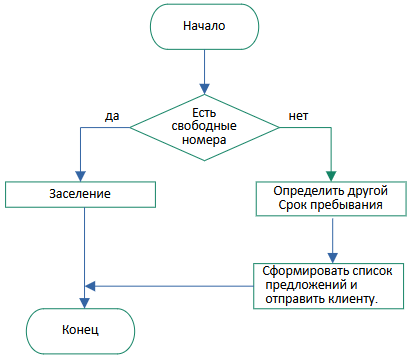


Рисунок 7 – Алгоритм предложения альтернатив

**3. Вывод отчета о заявках за период**

Цель: Сформировать список всех заявок за определённый временной промежуток.

**Шаги алгоритма:**

1. Получить от пользователя начало и конец периода.
2. Извлечь из базы данные всех заявок, соответствующих указанным датам.
3. Сгруппировать заявки по статусу, клиентам или номерам
4. Сформировать отчет и отобразить его.

**4. Вывод отчета о свободных номерах на указанную дату**

Цель: Показать список доступных номеров на заданную дату.

**Шаги алгоритма:**

Получить дату, указанную пользователем.

1. Извлечь из базы список всех номеров.
2. Исключить номера, занятые на указанную дату.
3. Отобразить список свободных номеров.

5 ТЕСТИРОВАНИЕ ПОЛУЧЕННОГО ПРОГРАММНОГО ПРОДУКТА

Тестирование программного обеспечения является очень важным и в то же время трудоемким видом деятельности. Тестирование – это прежде всего оценка промежуточных продуктов, созданных в процессе разработки программного обеспечения. Это гораздо более широкое поле деятельности, нежели просто проверка некоторых частей или программной системы в целом с целью определения степени ее соответствия техническим требованиям [7].

В данной работе была выбрана методология функционального тестирования.

Функциональное тестирование – это тестирование программного обеспечения в целях проверки реализуемости функциональных требований, то есть способности программного обеспечения в определенных условиях решать задачи, нужные пользователям. Функциональные требования определяют, что именно делает программное обеспечение, какие задачи оно решает [8].

Функциональное тестирование предполагает составление плана тестирования, который строится на основе вариантов использования и включает в себя описание тестов и их результаты.

После завершения проекта былa призведена проверка корректности выполнения всех функций системы:

**Поиск номеров:**

* Тестирование поиска номеров по различным критериям (даты, тип номера).
* Проверка отображения корректного списка доступных номеров.

**Регистрация бронирования:**

* Создание новой заявки и сохранение данных.
* Обработка конфликтных ситуаций (например, одновременные заявки на один номер).

**Отчеты:**

* Формирование отчетов по заявкам и свободным номерам на заданный период.

**Примеры произведенных тестов**

Отображение всех заявок при старте приложения представлено на рисунке 8:

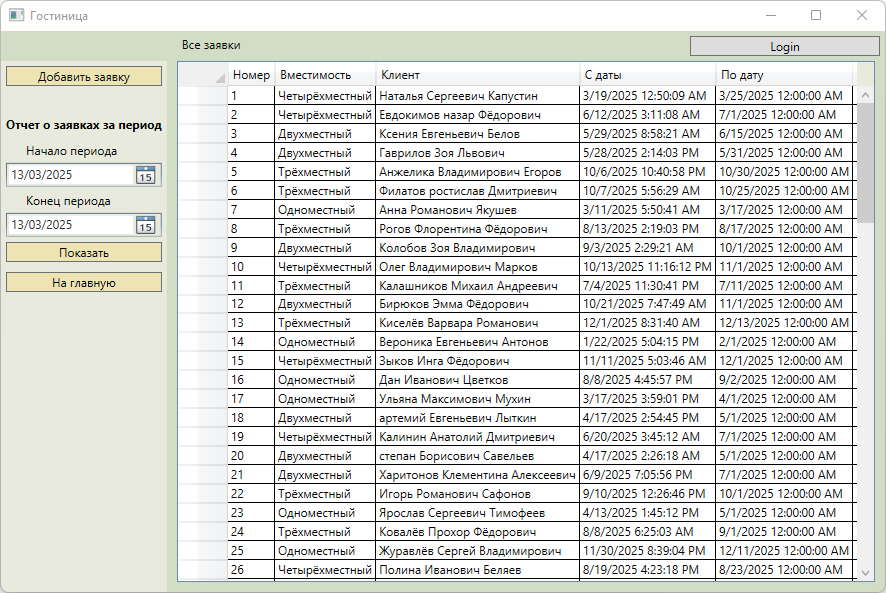


Рисунок 8 – Отображение всех заявок

Отображение заявок за указанный период представлено на рисунке 9:



Рисунок 9 – Отображение заявок за указанный период

Возврат предыдущего вывода представлен на рисунке 10:

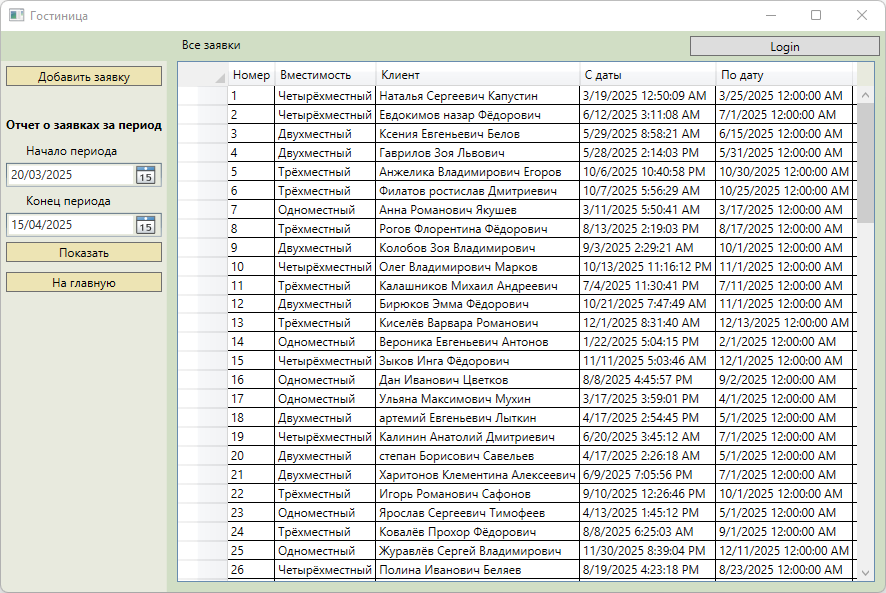


Рисунок 10– Возврат предыдущего вывода

Добавление заявки представлено на рисунке 11:

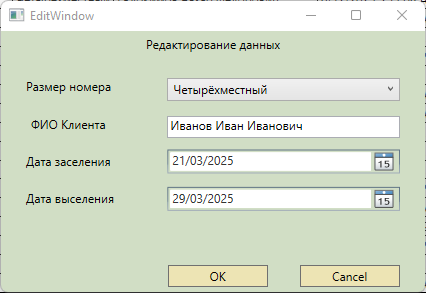


Рисунок 11– Добавление заявки

Результат добавления заявки представлен на рисунке 12

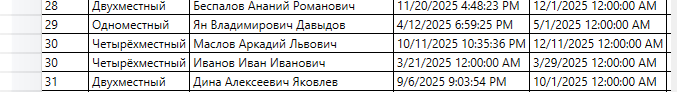


Рисунок 12 – Результат добавления заявки