|  |
| --- |
| МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  федеральное г**о**сударственное АВТОНОМНОЕ образовательное учреждение высшего образования  «Национальный исследовательский ядерный университет «МИФИ» |
| **Димитровградский инженерно-технологический институт –**  филиал федерального государственного автономного образовательного учреждения высшего образования «Национальный исследовательский ядерный университет «МИФИ»  **(ДИТИ НИЯУ МИФИ)** |

**Специальность 09.02.07 Информационные системы и программирование**

**ОТЧЁТ**

**по производственной практике (преддипломной)**

|  |  |  |
| --- | --- | --- |
| **Наименование практики** | ПДП Производственная практика  (преддипломная) | |
| **Обучающийся** | Мясников Денис Сергеевич | |
| **Группа** | 431 | |
| **Место проведения практики:** | ООО «СимбирСофт» | |
| **Сроки прохождения практики:** | с 12.04.2024 по 08.05.2024 | |
| **Руководитель практики от организации/предприятия базы практики:** | Власенко Олег Федосович |  |
| **Руководитель практики от образовательной организации:** | Катина Татьяна Игоревна |  |
| **Итоговая оценка** |  | |
| **Дата** | 08.05.2024 | |

**Димитровград, 2024**

**СОДЕРЖАНИЕ**

[ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ НА ПРОХОЖДЕНИЕ ПРОИЗВОДСТВЕННОЙ ПРАКТИКИ (ПРЕДДИПЛОМНОЙ) 3](#_Toc163586699)

[Введение 3](#_Toc163586700)

[1 Постановка задачи 3](#_Toc163586701)

[1.1 Исследование состояния вопроса 3](#_Toc163586702)

[1.2 Назначение программного продукта 3](#_Toc163586703)

[1.3 Перечень функций программного продукта 3](#_Toc163586704)

[1.4 Описание входных и выходных данных 3](#_Toc163586705)

[1.5 Перечень ограничений, связанный с оборудованием и программным обеспечением 4](#_Toc163586706)

[1.6 Требования к пользовательскому интерфейсу программного продукта 4](#_Toc163586707)

[1.7 Исследование и обоснование выбора инструментальных средств разработки программного продукта 4](#_Toc163586708)

[1.8 Требования к качественным характеристикам программного продукта 4](#_Toc163586709)

[2 Проектирование программного продукта 4](#_Toc163586710)

[2.1 Исследование предметной области 4](#_Toc163586711)

[2.2 Исследование методов проектирования 4](#_Toc163586712)

[2.3 Обоснование выбора методов проектирования 4](#_Toc163586713)

[2.4 Описание функциональной схемы программного продукта 4](#_Toc163586714)

[3 Технология программирования программного продукта 5](#_Toc163586715)

[3.1 Технология программирования программного продукта 5](#_Toc163586716)

[3.1.1 Исследование и обоснование выбора технологий программирования программного продукта 5](#_Toc163586717)

[3.1.2 Описание технологии программирования программного продукта 5](#_Toc163586718)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 6](#_Toc163586719)

[Заключение о прохождении производственной практики (преддипломной) 7](#_Toc163586720)

**ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ НА ПРОХОЖДЕНИЕ ПРОИЗВОДСТВЕННОЙ ПРАКТИКИ (ПРЕДДИПЛОМНОЙ)**

Тема выпускной квалификационной работы (дипломного проекта):

Разработка Web-приложения «BookRoom». Модуль внутреннего обслуживания.

**Введение**

В современном мире, в условиях стремительного развития информационных технологий, создание эффективных и удобных в использовании веб-приложений становится все более актуальной задачей. Особое внимание уделяется разработке приложений, способных обеспечить комфортное взаимодействие пользователя с информационными ресурсами. В рамках данного контекста возникает необходимость в создании веб-приложения «BookRoom», направленного на оптимизацию внутреннего обслуживания сферы гостеприимства.

Актуальность данной темы обусловлена тем, что активное распространение онлайн-сервисов в сфере бронирования и управления гостиничными услугами создает потребность в разработке эффективных и удобных инструментов для внутреннего управления. Введение в действие такого приложения вносит значительные улучшения в организацию процессов и повышает конкурентоспособность предприятий сферы гостеприимства.

Целью настоящей работы является разработка веб-приложения «BookRoom», предназначенного для оптимизации внутреннего обслуживания в сфере гостеприимства.

Объектом исследования настоящей работы является процесс разработки веб-приложения для внутреннего управления гостиничными услугами.

Предметом исследования является анализ, проектирование и разработка модуля внутреннего обслуживания веб-приложения «BookRoom», а также оценка его эффективности.

Для достижения поставленной цели должны быть выполнены следующие задачи:

1. Изучение существующих подходов и технологий в области разработки веб-приложений для гостиничного бизнеса.
2. Исследование предметной области: Анализ существующих веб-приложений для внутреннего обслуживания в сфере гостеприимства, изучение потребностей и требований пользователей.
3. Анализ потребностей и особенностей внутреннего обслуживания в гостиничном бизнесе.
4. Проектирование архитектуры и функционала веб-приложения «BookRoom» для оптимизации внутреннего обслуживания.
5. Разработка модульной структуры: Создание модульной архитектуры приложения для обеспечения гибкости, масштабируемости и легкости поддержки.
6. Разработка сценария интерфейса: Описание пользовательского интерфейса приложения, включая различные сценарии использования, взаимодействие пользователя с системой и пользовательские потоки.
7. Разработка пользовательской документации: Создание инструкций пользователя, руководств по установке и настройке приложения, а также другой документации для облегчения использования и администрирования системы.
8. Разработка и тестирование модуля внутреннего обслуживания веб-приложения.
9. Оценка эффективности разработанного модуля и его применимости на практике.

**1 Постановка задачи**

**1.1 Исследование состояния вопроса**

Актуальность данной работы обусловлена тем, что требования к веб-приложениям для внутреннего обслуживания постоянно меняются в соответствии с развитием технологий и изменениями в бизнес-процессах. Проведение исследования текущего состояния этой области позволит выявить существующие проблемы, потребности и требования пользователей, а также определить новые тенденции и возможности развития. Такой анализ поможет обосновать целесообразность создания нового веб-приложения "BookRoom" для внутреннего обслуживания и определить его функциональные и интерфейсные особенности в соответствии с текущими потребностями пользователей и требованиями рынка.

В рамках данной работы был произведен поиск систем внутреннего управлением отелем в сети Интернет на наличие аналогов. В результате поиска были найдены следующие систем внутреннего управлением отелем, которые представлены на рисунках 1.1–1.3, среди которых выявились положительные и отрицательные стороны.

Контур.Отель – это сервис по управлению гостиничным бизнесом. С его помощью осуществляется управление фондом мест размещения, работа с бронированием (через booking.com и прочие онлайн-площадки)

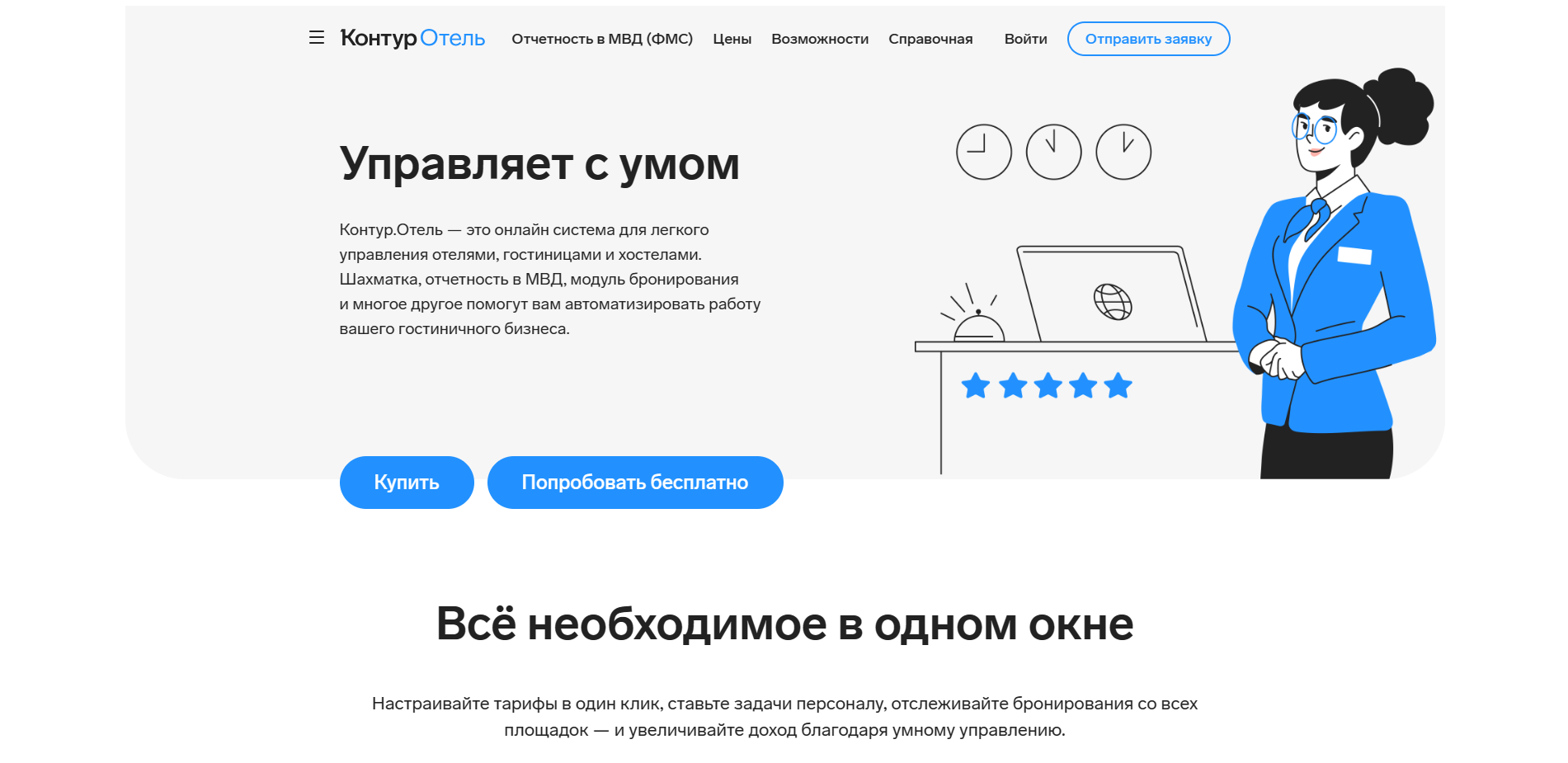


Рисунок 1.1 – Систем внутреннего управлением отелем «Контур.Отель»

Преимущества:

* облачное решение, которое упрощает внедрение и обновление системы;
* интеграция с различными каналами бронирования и онлайн-платформами;
* автоматизация процессов бронирования и управления тарифами.

Недостатки:

* некоторые ограничения в функционале по сравнению с более крупными системами;
* не всегда подходит для крупных отелей с большими объемами бронирований.

TravelLine — Онлайн-платформа для управления гостиницами, хостелами, отелями, загородными домами и квартирами. Пользователь может использовать шаблоны конструктора для создания собственного сайта и его индивидуальной настройки.. Изображение выглядит как текст, снимок экрана, программное обеспечение, веб-страница

Автоматически созданное описание

Рисунок 1.2 – Систем внутреннего управлением отелем «TravelLine»

Преимущества:

* интегрированная система управления отелем и бронированиями с панелью управления;
* гибкие тарифные планы и управление инвентарем;
* интеграция с платежными системами и онлайн-каналами бронирования.

Недостатки:

* некоторые возможности могут быть менее развиты, чем у крупных игроков на рынке;
* могут возникнуть сложности с интеграцией сторонних систем.

Bnovo – Система управления отелем модульного вида, направленная на улучшение эффективности всех процессов на площадках резервирования и рационализированного управления продажами.

Изображение выглядит как текст, снимок экрана, программное обеспечение, мультимедиа

Автоматически созданное описание

Рисунок 1.3 – Систем внутреннего управлением отелем «Bnovo»

Преимущества:

* интегрированная система управления бронированиями и отелем;
* гибкие возможности настройки тарифов и акций;
* интеграция с онлайн-платформами и каналами бронирования.

Недостатки:

* возможны сложности с масштабированием для крупных отелей;
* не всегда оптимальная локализация и поддержка русскоязычных клиентов.

Из проделанного анализа складывается более подробная картина будущего программного продукта, в котором будут учтены сильные и слабые стороны сервисов конкурентов, для выпуска эффективного и приносящего прибыль веб-приложения.

Исследовав состояние вопроса, перейдём к описанию назначения программного продукта.

**1.2 Назначение программного продукта**

Программный продукт «BookRoom» разработан для обеспечения эффективного управления бронированием и внутренним обслуживанием гостиничных номеров. Основными задачами продукта являются автоматизация процессов бронирования, оптимизация работы персонала и повышение удовлетворенности клиентов. Приложение предоставляет пользователям возможность легкого доступа к информации о доступных номерах, гостях, отчетам и статистике, акциях, а также позволяет производить бронирование, управление заказами и номерами с минимальными затратами времени и ресурсов. Назначение программного продукта заключается в повышении эффективности работы гостиничного бизнеса и улучшении пользовательского опыта.

Описав назначение программного продукта, перейдём к составлению перечня функций программного продукта.

**1.3 Перечень функций программного продукта**

Программный продукт должен обеспечивать корректное выполнение следующих функций, для двух групп пользователей.

Функции администратора:

* регистрация, изменение и получение настроек отеля;
* отображение выручки за месяц, выгода по сравнению с предыдущем месяцем, количество гостей, количества комнат;
* получение статистических отчетов и отчетов общего назначения;
* вывод отчетов на печать и на загрузку;
* удаление гостей;
* удаление номеров, редактирование;
* добавление, редактирование, удаление акций;
* добавление, редактирование, удаление расценок.

Общие функции администратора и менеджера:

* авторизация, аутентификация;
* регистрация;
* выход из аккаунта;
* отображение комнат свободные, занятые, чистые, грязные, на проверке, количество посетителей за месяц, активности гостей;
* регистрация, изменение и получение настроек отеля;
* отображение, добавление (оформление), редактирование, удаление гостей;
* просмотр, отмечание выезда гостя;
* просмотр и разблокировка номеров;
* просмотр статистики обслуживания номеров;
* изменение статуса номера (убирается, убран, проверяется);
* создание заявки (нескольких заявок) на ремонт номера;
* блокирование номера на период ремонта;
* отображение информации по ремонтным работам в номерах;
* регулирование статуса ремонта номеров;
* просмотр акций;
* просмотр расценок.

Составив перечень функций программного продукта, перейдём к описанию входных и выходных данных.

**1.4 Описание входных и выходных данных**

Входные данные:

1. Авторизация:

* логин;
* пароль.

1. Регистрация нового пользователя:

* логин;
* пароль;
* почта.

1. Регистрация отеля:

* название отеля;
* логотип;
* адрес (Страна, Регион, Город, Улица, Номер дома);
* контактная информация (Почта, Номер телефона);
* информация о владельце (ФИО, Почта, Номер телефона).

1. Изменение настроек отеля:

* идентификатор отеля;
* новая контактная информация;
* новая информация о владельце;
* новое название отеля и логотип.

1. Изменение статуса гостя:

* идентификатор статуса гостя;
* новые значения атрибутов статуса гостя.

1. Изменение статуса обслуживания номера:

* идентификатор статуса обслуживания номера;
* новые значения атрибутов статуса обслуживания номера.

1. Изменение статуса ремонтных работ:

* идентификатор статуса обслуживания номера;
* новые значения атрибутов статуса обслуживания номера.

1. Изменение ремонтных работ:

* идентификатор ремонтных работ;
* новые значения атрибутов ремонтных работ.

1. Создание ремонтных работ:

* название работы;
* описание работы;
* дата начала;
* дата окончания.

1. Удаление ремонтных работ:

* идентификатор акции.

1. Удаление гостевых данных на основе политики хранения данных:

* идентификатор акции.

1. Создание новой расценки:

* код типа комнаты;
* код политики отмены;
* цена.

1. Изменение расценки:

* идентификатор ремонтных работ;
* новые значения атрибутов ремонтных работ.

1. Удаление расценки:

* идентификатор акции.

1. Создание новой акции:

* название акции;
* тип комнаты;
* дата начала;
* дата окончания;
* описание;
* общее количество использований.

1. Изменение акции:

* идентификатор акции;
* новые значения атрибутов акции.

1. Удаление акции:

* идентификатор акции.

1. Создание нового номера:

* номер номера;
* тип комнаты;
* этаж;
* удобства.

1. Изменение данных номера:

* идентификатор номера;
* новые значения атрибутов номера.

1. Изменение статуса номера:

* идентификатор номера;
* новый статус.

1. Удаление номера:

* идентификатор номера.

1. Создание нового бронирования:

* ФИО гостя;
* номер телефона гостя;
* почта гостя;
* код комнаты;
* дата прибытия;
* дата отбытия;
* количество взрослых;
* количество детей;
* код расценки.

1. Удаление бронирования:

* идентификатор бронирования.

Выходные данные:

1. Получение комнат для обслуживания:

* список комнат, требующих обслуживания.

1. Получение статистики для обслуживания:

* статистика об обслуживании номеров (грязный, убирается, проверяется, чистый).

1. Получение ремонтных работ:

* список ремонтных работ, запланированных или выполняемых.

1. Получение статистики ремонтных работ:

* статистика по ремонтным работам (новая, в процессе, завершена, отменена).

1. Получение ремонтных работ конкретной комнаты:

* список ремонтных работ для определенной комнаты.

1. Получение всех номеров:

* список всех номеров в отеле.

1. Получение всех расценок:

* список всех доступных расценок для номеров.

1. Получение статистики "Самое дорогое бронирование":

* информация о самом дорогом бронировании.

1. Получение статистики "Самый выбираемый номер за все время":

* информация о самом часто выбираемом номере за все время.

1. Получение статистики "Самый выбираемый номер за этот месяц":

* информация о самом часто выбираемом номере за текущий месяц.

1. Получение статистики "Средний чек за день в этом месяце":

* средний чек за каждый день текущего месяца.

1. Получение статистики "Самый посещаемый день за все время":

* информация о дне с наибольшим количеством посещений за все время.

1. Получение отчета "Все гости":

* полная информация обо всех гостях, зарегистрированных в отеле.

1. Получение отчета "Все комнаты":

* полная информация обо всех номерах в отеле.

1. Получение отчета "Все расценки":

* полная информация обо всех расценках для номеров.

1. Получение отчета "Все акции":

* полная информация обо всех акциях отеля.

1. Получение отчета "Все пользователи":

* полная информация обо всех пользователях системы.

1. Получение всех акций:

* список всех доступных акций отеля.

1. Получение клиентов для бронирования:

* список всех клиентов, сделавших бронирование.

1. Отметить выезд гостя:

* успешность отметки выезда гостя.

1. Получение статусов комнат:

* статус каждой комнаты (заблокировано, забронировано, доступно, заселено, ожидание.

1. Получение удобств для комнат:

* список удобств, доступных в номерах.

1. Получение типов комнат:

* список всех типов номеров в отеле.

1. Получение статусов акций:

* статус каждой акции (закончилась, новая, активная, максимум, скоро, неактивно.

1. Получение политики отмены:

* правила отмены бронирования для каждой расценки.

1. Получение статусов гостя:

* статус каждого гостя (прибыл, выехал, должен прибыть, должен выехать.

1. Получение политики хранения персональных данных:

* политика отеля по хранению и обработке персональных данных клиентов.

**1.5 Перечень ограничений, связанный с оборудованием и программным обеспечением**

В состав технических средств должен входить компьютер клиента с приведёнными ниже минимальными характеристиками, компонентами:

* процессор Intel Celeron или AMD Athlon x64 или x32;
* объём оперативной памяти 2гб;
* монитор FullHD 1980x1080;
* клавиатура;
* компьютерная мышь.

Для работы клиентской части ограничения на программном уровне:

* браузер Chrome 70;
* браузер FireFox 64;

Для работы приложения требуются следующие технические средства:

* проводная или беспроводная клавиатура и компьютерная мышь;
* монитор с разрешением 1920 на 1860 пикселей.

В состав технических средств, при помощи которых будет работать приложение, должны входить компьютер с приведенными ниже характеристиками, компонентами и возможность любым возможным способом получить файл приложения на устройство.

Минимальные:

* 4 Гб оперативной памяти;
* 4 ядерный процессор на 2.8Ггц;
* пространство на диске 120 Гб;
* SSD/SSD m2 не требуется;
* сетевая карта с пропускной способностью 1Гб/с.

Рекомендуемые

* 16 Гб оперативной памяти;
* 4 ядерный процессор на 3.1Ггц;
* пространство на диске 1024 Гб;
* SSD/SSD m2 обязательно;
* сетевая карта с пропускной способностью 1Гб/с и выше.

Для работы приложения в сервере необходимо программное обеспечение:

* Windows Server 2019/Ubuntu 20.1;
* Docker 22.0.

Определив перечень ограничений, связанных с оборудованием и программным обеспечением, перейдём к составлению требований к пользовательскому интерфейсу программного продукта.

**1.6 Требования к пользовательскому интерфейсу программного продукта**

Интерфейс – это набор правил, определяющих взаимодействие человека с компьютером. Интерфейс – это совокупность средств и методов, позволяющих пользователю взаимодействовать с компьютером, выбирать из множества возможных вариантов действий и получать результат.

Дизайн пользовательского интерфейса должен адаптироваться под основные разрешения экранов настольных, портативных и планшетных устройств. Максимальное разрешение по горизонтали – 1920px, минимальное – 1080px. В диапазоне указанных разрешений горизонтальная прокрутка не должна появляться. В случае необходимости в макет можно добавить вертикальную прокрутку с использованием CSS3. В этом случае при прокрутке вверх в колонках контент должен автоматически сдвинуться вниз на определенное расстояние. Этот метод можно применить только к версиям макета для всех типов устройств. Для версий макетов для планшетов и ноутбуков необходимо использовать CSS3 для прокрутки по вертикали.

Интерфейс системы, включая графики и диаграммы, должен адаптироваться под разрешение экрана.

Элементы интерфейса (пункты меню, кнопки, поля ввода в формах, раскрывающиеся списки, и т.д.) должны адаптироваться по размеру под устройство, на котором просматривается система, и под основной сценарий использования данного устройства. Дизайн пользовательского интерфейса должен быть разработан с учетом принципа сохранения работоспособности при потере части функциональности системы. Некоторые элементы дизайна могут иметь упрощенный вид, но основной функционал должен оставаться доступным. В случае использования в системе элементов интерфейса, которые не имеют визуального контекста (кнопки, например), пользователю должно быть описано, каким образом можно использовать данный элемент.

Дизайн пользовательского интерфейса должен корректно отображаться и работать на настольных компьютерах, ноутбуках, планшетных устройствах. Также должен корректно работать и на мобильных устройствах: смартфонах, планшетах. Кроме того, дизайн должен быть удобным для пользователей разных возрастов, социальных групп и уровня подготовки, поэтому при разработке дизайна пользовательского интерфейса нужно учитывать особенности аудитории и их потребности.

Из вышеперечисленных критериев и условий были сделаны следующие выводы, о том каким должен быть графический интерфейс.

1. Минимизация усилий пользователя при выполнении работы:

* сокращение длительности операций чтения, редактирования и поиска информации;
* уменьшение времени навигации и выбора команды;

1. Возможность использовать различные интерфейсы с одним и тем же приложением.
2. Возможность развивать внешний вид без каких-либо проблем.
3. Приложение должно допускать возможность ввода данных и команд множеством разных способов (клавиатура, мышь, другие устройства) и многовариантность доступа к прикладным функциям (иконы, «горячие клавиши», меню), кроме того, программа должна учитывать возможность перехода и возврат от окна к окну, от режима к режиму, и правильно обрабатывать такие ситуации.
4. Для создания пользовательского интерфейса должны предоставляться отдельные ресурсы, направленные на хранение и обработку данных, необходимых для поддержки пользователя.
5. При переходе на другую аппаратную (программную) платформу, должен осуществляется автоматически перенос и пользовательского интерфейса, и конечного приложения.
6. Расположение элементов обосновано анатомически: так, чтобы ими было удобно управлять с помощью одной руки, а пальцы не промахивались мимо кнопки.
7. Крупные элементы интерфейса – блоки меню, кнопки или иконки, которые заполняют экран полностью: так, чтобы их было хорошо видно даже при маленьком разрешении.
8. Использовать элементы, соответствующие своему предназначению, к примеру: списки, заголовки, текст, картинка и т. д.

Составив требования к пользовательскому интерфейсу программного продукта, перейдём к исследованию и обоснованию выбора инструментальных средств разработки программного продукта.

**1.7 Исследование и обоснование выбора инструментальных средств разработки программного продукта**

Для разработки базы данных интернет-магазина можно использовать следующие СУБД:

1. PostgreSQL.

* преимущества:

1. мощные и надёжные механизмы транзакций и репликации,
2. наследование,
3. легкая расширяемость,
4. позволяет создавать отказоустойчивые среды.

* недостатки:

1. отсутствует встроенная поддержка управлениями ресурсами,
2. для сложных задач автоматизации потребуется дополнительное ПО,
3. отсутствует встроенной поддержки балансировки нагрузки,
4. относительно маленькая коммерческая поддержка.
5. MySQL.

* преимущества:

1. высокая скорость работы,
2. бесплатная лицензия,
3. поддержка движков: MyISAM, InnoDB,
4. поддерживается почти всеми CMS.

* недостатки:

1. отсутствует механизм блокировки во время транзакций,
2. неэффективна в больших базах,
3. несоответствие стандарту MySQL,
4. медленное развитие сообществом.
5. M ongoDB.

* преимущества:

1. отсутствие схемы или конкретной структуры,
2. кроссплатформенная,
3. данные хранятся в BSON формате,
4. балансировка нагрузки.

* недостатки:

1. подходит под очень крупные проекты,
2. отсутствует поддержка транзакций,
3. не соответствует принципу ACID,
4. не получится реализовать бизнес-логику.
5. MSSQL.

* преимущества:

1. поддержка решений от Microsoft (Excel, Access),
2. удобный поиск,
3. автоматизация рутинных административных задач.

* недостатки:

1. работает только на ОС Windows,
2. высокий порог вхождения,
3. высокая стоимость.

В результате исследования выбран PostgreSQL, потому что он эффективен для работы с большими объёмами данных и имеет встроенные средства защиты от утечек данных.

Для разработки и администрирования баз данных PostgreSQL можно использовать следующие пользовательские интерфейсы, которые визуализируют работу с базами данных:

1. pgAdmin – программное обеспечение с открытым исходным кодом для разработки и администрирования баз данных PostgreSQL и производных от них баз данных, таких как EnterpriseDB Postgres Plus Advanced Server или Greenplum.

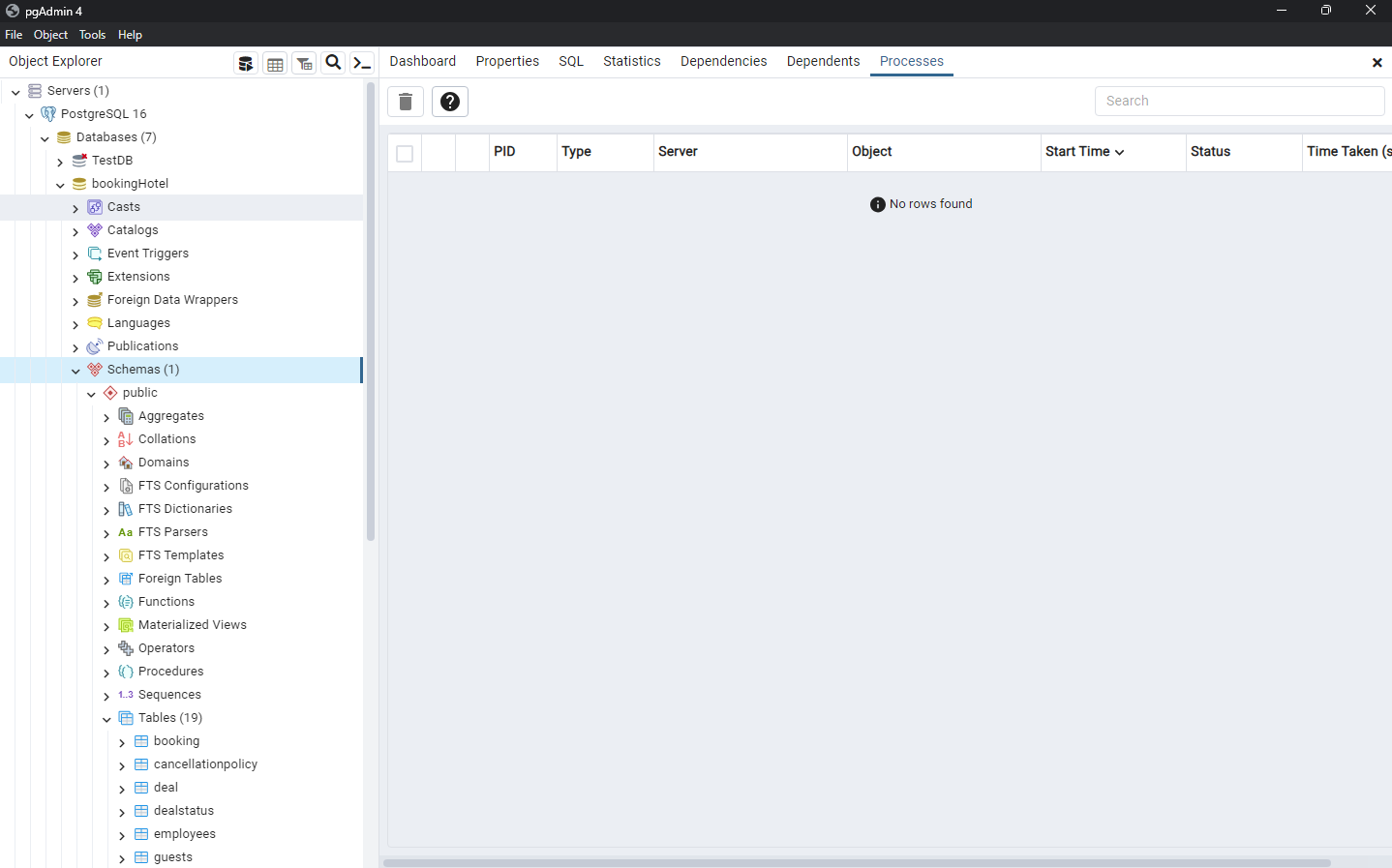


Рисунок 1.4 – Программное обеспечение pgAdmin

Преимущества:

* совместим с Linux, Windows, macOS;
* позволяет работать с несколькими серверами одновременно;
* экспорт в CSV;
* планирование запросов;
* возможность отслеживать ваши сеансы, блокировки БД с помощью панели мониторинга;
* ярлыки в редакторе SQL для более удобной работы;
* встроенный отладчик процедурного языка;
* тщательная документация и активное сообщество.

Недостатки:

* медленный и не всегда интуитивно понятный пользовательский интерфейс по сравнению с платными конкурентами;
* тяжелый;
* высокий порог вхождения;
* для работы с несколькими базами данных одновременно потребуются продвинутые навыки.

DBeaver – это клиентское программное приложение SQL и инструмент администрирования баз данных. Для реляционных баз данных он использует интерфейс прикладного программирования JDBC (API) для взаимодействия с базами данных через драйвер JDBC.

Изображение выглядит как текст, снимок экрана, число, программное обеспечение

Автоматически созданное описание

Рисунок 1.5 – Клиентское программное приложение DBeaver

Преимущества:

* кроссплатформенность;
* поддержка более 80 баз данных;
* визуальный конструктор, позволяющий добавлять запросы без навыков работы с SQL;
* несколько представлений данных;
* импорт/экспорт данных в CSV, HTML, XML, JSON, XLS, XLSX;
* повышенная безопасность данных;
* полнотекстовый поиск данных и возможность отображения результатов в виде таблиц/представлений;
* доступен бесплатный тарифный план.

Недостатки:

* низкая производительность по сравнению с конкурентами;
* слишком частые обновления, что раздражает;
* после некоторого времени бездействия DBeaver отключается от базы данных.

Navicat – инструмент для разработки и администрирования баз данных, который работает на любом сервере MySQL, начиная с версии 3.21.

Изображение выглядит как текст, снимок экрана, программное обеспечение, Значок на компьютере

Автоматически созданное описание

Рисунок 1.6 – Инструмент Navicat

Преимущества:

* простая и быстрая установка;
* поддержка Windows, Linux, iOS;
* удобный визуальный конструктор SQL;
* автодополнение кода;
* инструмент моделирования данных: управление объектами базы данных, схемами проектирования;
* планировщик заданий: запускайте задания, получайте уведомления о завершении задания;
* синхронизация источников данных;
* импорт/экспорт данных в Excel, Access, CSV и другие форматы;
* защита данных с помощью SSH и SSL;
* использование облачных сервисов Amazon, Google и др.

Недостатки:

* низкая производительность GUI;
* высокая цена по сравнению с конкурентами;
* одна лицензия ограничена одной платформой (вам понадобятся 2 отдельные лицензии для PostgreSQL и MySQL);
* множество дополнительных возможностей, требующих времени для изучения.

DataGrip – это интегрированная среда разработки (IDE) для разработчиков баз данных, которая предоставляет консоль запросов, навигацию по схемам, планы объяснений, интеллектуальное завершение кода, анализ в реальном времени и быстрые исправления, рефакторинг, интеграцию управления версиями и другие функции.

Изображение выглядит как текст, снимок экрана, программное обеспечение, число

Автоматически созданное описание

Рисунок 1.7 – Интегрированная среда разработки DataGrip

Преимущества:

* кроссплатформенность (поддержка Windows, macOS, Linux);
* простая навигация по схеме;
* настраиваемый UI с консолью для обеспечения безопасности выполняемой работы;
* быстрое обнаружение ошибок;
* встроенная система контроля версий;
* поддержка MySQL, SQLite, MariaDB, Cassandra и других;
* отчеты с возможностью их интеграции с диаграммами и графиками;
* автодополнение кода.

Недостатки:

* высокая цена;
* высокое потребление оперативной памяти;
* сложный процесс отладки ошибок;
* длинная кривая обучения;
* не предназначен для использования в качестве облачного веб-приложения;

В результате исследования пользовательских интерфейсов был выбран DataGrip, потому что у него удобный интерфейс, быстрый отклик, а также интеграция с почти любой базой данных.

Для разработки системы внутреннего управления отелем можно использовать следующие языки программирования высокого уровня:

JavaScript – мультипарадигменный язык программирования. Поддерживает объектно-ориентированный, императивный и функциональный стили.

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

Рисунок 1.8 – Синтаксис языка JavaScript

Преимущества:

* подходит для клиентской и серверной стороны;
* огромное количество библиотек;
* низкий порог входа;
* популярность.

Недостатки:

* слабая типизация, которая позволяет складывать строку с числом. Эта особенность породила огромное количество мемов;
* запутанный синтаксис.

Python – высокоуровневый язык программирования общего назначения с динамической строгой типизацией и автоматическим управлением памятью, ориентированный на повышение производительности разработчика, читаемости кода и его качества, а также на обеспечение переносимости написанных на нём программ.

Изображение выглядит как текст, снимок экрана, программное обеспечение, Операционная система

Автоматически созданное описание

Рисунок 1.9 – Синтаксис языка Python

Преимущества:

* лаконичность: на Python нужно писать гораздо меньше кода для выполнения команд;
* Python применяют в веб-разработке, в машинном обучении, аналитике данных, играх, научных исследованиях;
* востребованность на рынке: на Python пишут и в небольших студиях, и в огромных корпорациях.

Недостатки:

* машинное обучение и наука о данных требуют знания соответствующих разделов математики.

C# – современный объектно-ориентированный и типобезопасный язык программирования. C# позволяет разработчикам создавать разные типы безопасных и надежных приложений, выполняющихся в .NET.

**Изображение выглядит как текст, снимок экрана, дисплей, программное обеспечение

Автоматически созданное описание**

Рисунок 1.10 – Синтаксис языка C#

Преимущества:

* наличие ООП;
* кроссплатформенность;
* автоматическая сборка мусора и управление памятью;
* много библиотек и готовых решений.

Недостатки:

* C# менее гибкий, так как в основном зависит от платформы .NET.

Java – строго типизированный объектно-ориентированный язык программирования общего назначения, разработанный компанией Sun Microsystems.

Изображение выглядит как текст, программное обеспечение, Значок на компьютере, веб-страница

Автоматически созданное описание

Рисунок 1.11 – Синтаксис языка Java

Преимущества:

* кроссплатформенность – Java работает на большом количестве ОС;
* наличие ООП;
* большое сообщество.

Недостатки:

* дополнительный слой в виде виртуальной машины немного снижает производительность;
* многословный код — иногда нужно написать больше кода, чем на других языках.

Из языков был выбран JavaScript. JavaScript является одним из наиболее распространенных языков программирования в веб-разработке. Использование одного языка как на клиентской, так и на серверной стороне позволяет создавать приложения с более простой архитектурой и повышает переиспользуемость кода. Так же JavaScript имеет огромное сообщество разработчиков и обширную экосистему библиотек и фреймворков как на стороне сервера (например, Express.js, Koa.js), так и на стороне клиента (например, React, Angular, Vue.js). Это делает разработку, масштабирование и поддержку приложений более удобными и эффективными.

Исследовав и обосновав выбор инструментальных средств разработки программного продукта, перейдём к определению требований к качественным характеристикам программного продукта.

**1.8 Требования к качественным характеристикам программного продукта**

Качество программного обеспечения – способность программного продукта при заданных условиях удовлетворять установленным или предполагаемым потребностям. Является одним из основных критериев качества программного обеспечения.

В широком смысле качество программного обеспечения – это степень соответствия характеристик программного продукта и условий эксплуатации.

Качество программного обеспечения оценивается по совокупности его свойств и включает в себя следующие аспекты:

* надежность – способность ПО выполнять требуемые задачи в обозначенных условиях на протяжении заданного промежутка времени или указанное количество операций. Атрибуты данной характеристики – это завершенность и целостность всей системы, способность самостоятельно и корректно восстанавливаться после сбоев в работе, отказоустойчивость.
* эффективность – определяется способностью ПО решать задачи, которые соответствуют зафиксированным и предполагаемым потребностям пользователя, при заданных условиях использования ПО. Т.е. эта характеристика отвечает за то, что ПО работает исправно и точно, функционально совместимо, соответствует стандартам отрасли и защищено от несанкционированного доступа.
* совместимость – характеризует ПО с точки зрения легкости его переноса из одного окружения (software/hardware) в другое
* удобство использования – возможность легкого понимания, изучения, использования и привлекательности ПО для пользователя
* уровень безопасности – в приложении должны быть минимизированы уровни риска потери или утечки данных из-за небезопасных протоколов
* возможность сопровождения – легкость, с которой ПО может анализироваться, тестироваться, изменяться для исправления дефектов, для реализации новых требований, для облегчения дальнейшего обслуживания и адаптироваться к имеющемуся окружению.

Проанализировав требования, следует вывод, что интернет-магазин должен быть удобным в использовании, должен иметь красивый дизайн и легкое обслуживание системы.

Определив требования к качественным характеристикам программного продукта, перейдём к проектированию программного продукта.

**2 Проектирование программного продукта**

**2.1 Исследование предметной области**

Предметная область – множество всех предметов, свойства которых и отношения, между которыми рассматриваются в научной теории. В логике – подразумеваемая область возможных значений предметных переменных логического языка.

Предметная область программного продукта "BookRoom" охватывает все аспекты управления гостиничным бизнесом. Она включает в себя не только операции бронирования номеров, но и учет гостей, а также управление акциями, расценками и ремонтными работами.

1. Бронирование номеров:

* Этот процесс включает в себя резервирование жилых помещений гостиницы на определенные даты для гостей.
* Важно обеспечить удобство и простоту для гостей при выборе и бронировании номеров.

1. Учет гостей:

* Важная часть работы гостиницы - поддержание базы данных о гостях, включая их контактные данные.
* Учет гостей помогает в обеспечении персонализированного обслуживания и удовлетворении потребностей клиентов.

1. Управление акциями и расценками:

* Гостиницы могут предлагать различные акции и специальные предложения для привлечения клиентов.
* Управление расценками включает в себя установку цен на проживание и дополнительные услуги в зависимости от сезона, спроса и других факторов.

1. Обслуживание и ремонт номеров:

* Этот аспект включает в себя обеспечение чистоты, порядка и технического состояния номеров гостиницы.
* Необходимо вести учет ремонтных работ и обеспечить их своевременное выполнение для удовлетворения потребностей гостей.

1. Настройка отеля:

* Администраторы гостиницы должны иметь возможность настраивать различные параметры и функциональность программного продукта в соответствии с потребностями и требованиями отеля.

Эти аспекты совместно образуют предметную область "BookRoom", цель которой - обеспечить высокое качество обслуживания гостей, оптимизацию работы персонала и увеличение прибыли гостиницы.

Предметная область "BookRoom" состоит из двух основных зон ответственности:

1. Управление гостиничным комплексом, включая бронирование, администрирование номеров и обслуживание гостей.
2. Управление административными аспектами, такими как настройка отеля, управление персоналом и анализ данных.

Выделим базовые сущности данной предметной области, которые образуют структуру проектируемой ИС:

1. Бронирования. Атрибуты продукции – код бронирования, код гостя, код комнаты, код расценки, дата прибытия, дата отбытия, количество взрослых, количество детей, всего к оплате.
2. Гости. Атрибуты пользователя – код гостя, код комнаты, код статуса гостя, номер гостя, фамилия, имя, отчество, номер телефона, почта.
3. Статус гостя. Атрибуты роли – код статус гостя, статус.
4. Статус номера гостя. Атрибуты роли – код статуса номера гостя, статус номера.
5. Пользователи. Атрибуты – код пользователя, логин пароль, почта, роль.
6. Акции. Атрибуты – код акции, код типа комнаты, код статуса акции, название акции, номер акции, сколько осталось использований, начало акции, конец акции, сколько всего использований, описание акции.
7. Статус акции. Атрибуты – код статуса акции, статус.
8. Политика отмены. Атрибуты продукции – код политики отмены, политика отмены, описание.
9. Номера. Атрибуты пользователя – код номера, код статуса номера, код типа номера, код статуса номера гостя, код ремонтных работ, код удобств, номер комнаты, этаж номера.
10. Тип номера. Атрибуты роли – код типа номера, тип номера.
11. Удобства номера. Атрибуты – код удобства, удобство.
12. Статус номера. Атрибуты – код статуса номера, статус.
13. Ремонтные работы номера. Атрибуты – код ремонтных работ, код комнаты, код статуса ремонтных работ, название работы, описание, дата начала, дата окончания, блокировать ли номер.
14. Статус ремонтных работ. Атрибуты – код статуса ремонтных работ, статус.
15. Расценка. Атрибуты – код расценки, код типа комнаты, код политики отмены, код акции, цена.
16. Параметры отеля. Атрибуты продукции – код параметра отеля, код политики хранения персональных данных, название отеля, логотип, страна, регион, город, улица, номер дома, количество этажей, количество номеров, контактная почта, контактный номер телефона, ФИО владельца, почта владельца, номер телефона владельца.
17. Политика хранения персональных данных. Атрибуты пользователя – код политика хранения персональных данных, политика хранения персональных данных.

Система создаётся для следующих групп пользователей:

* администратор;
* менеджер.

Функциональные возможности:

* управление бронированиями и регистрацией гостей;
* управление статусами номеров (свободные, занятые, на ремонте);
* управление информацией о гостях (регистрация, редактирование, удаление);
* управление акциями и специальными предложениями;
* управление расценками на различные услуги и номера;
* управление ремонтными работами и обслуживанием номеров;
* генерация отчетов о выручке, статистике и активности гостей;
* настройка параметров отеля (контактные данные, политика отмены и пр.).

Готовые запросы:

* регистрация;
* авторизация;
* получение всех пользователей;
* изменение пользователя;
* удаление пользователя;
* получение комнат для обслуживания;
* получение статистика для обслуживания;
* изменение статуса обслуживания комнаты;
* получение ремонтных работ;
* получение статистки ремонтных работ;
* получение ремонтных работ конкретной комнаты;
* изменение статуса ремонтных работ;
* изменение ремонтных работ;
* создание ремонтных работ;
* удаление ремонтных работ;
* получение всех номеров;
* создание номера;
* изменение номера;
* изменение статуса номера;
* удаление номера;
* получение всех расценок;
* создание расценки для типа комнат;
* создание расценки для типа комнат с учетом скидки;
* изменение расценки;
* удаление расценки;
* получение статистики "Самое дорогое бронирование";
* получение статистики "Самое выбираемый номер за все время";
* получение статистики "Самое выбираемый номер за этот месяц";
* получение статистики "Средний чек за день в этом месяце";
* получение статистики "Самый посещаемый день за все время";
* получение отчета "Все гости";
* получение отчета "Все комнаты";
* получение отчета "Все расценки";
* получение отчета "Все акции";
* получение отчета "Все пользователи";
* регистрация отеля;
* изменение настроек отеля;
* получение настроек отеля;
* получение всех акций;
* создание акции;
* изменение акции;
* удаление акции;
* создание бронирования;
* получение клиентов для бронирования;
* удаление бронирования;
* изменение статуса гостя (прибыл, должен прибыть, должен выехать);
* отметить выезд гостя;
* удаление гостевых данных на основе политики хранения данных;
* получение статусов комнат;
* получение удобств для комнат;
* получение типов комнат;
* получение статусов акций;
* получение политики отмены;
* получение статусов гостя;
* получение статусов гостевых комнат;
* получение статуса ремонта комнат;
* получение политики хранения персональных данных.

Составим схему инфологической модели в виде ER – диаграммы. ER – диаграмма представлена на рисунке 2.1.

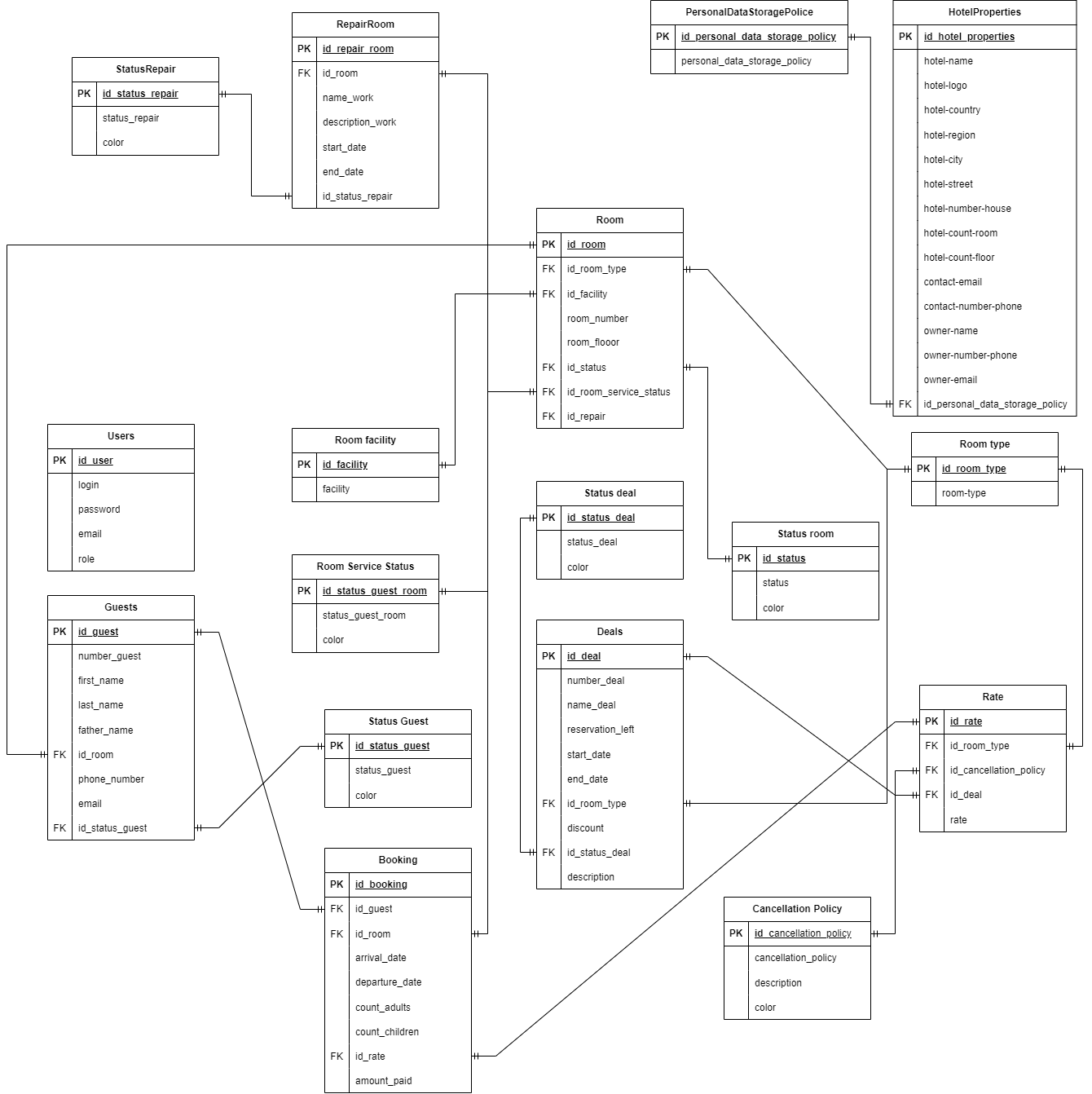


Рисунок 2.1 – ER-диаграмма

**2.2 Исследование методов проектирования**

Исследование методов проектирования программного обеспечения является важным этапом в разработке любого проекта. Этот процесс позволяет разработчикам выбрать оптимальные методы и подходы для создания эффективной, удобной в использовании и надежной системы.

Использование исследования методов проектирования позволяет:

* Оценить существующие методы и подходы к разработке ПО.
* Выбрать наиболее подходящий метод, учитывая особенности проекта, требования заказчика и опыт команды разработки.
* Обосновать выбор метода перед заказчиком и заинтересованными сторонами.
* Улучшить процесс разработки и повысить качество конечного продукта.

Рассмотрим следующие методы проектирования:

1. Waterfall (Каскадная модель)

Методология Waterfall предполагает линейную последовательность этапов разработки: сначала проводится анализ, затем проектирование, разработка, тестирование и внедрение. Этот метод подходит для проектов с четко определенными требованиями и низкой вероятностью их изменения.

Преимущества:

* Четкая структура и последовательность этапов.
* Простота планирования и управления проектом.

Недостатки:

* Не гибок в случае изменения требований.
* Риски возникновения проблем на поздних этапах разработки.
* Трудно оценить все требования заранее.

2. Agile

Методология Agile ориентирована на гибкость и адаптивность к изменениям. Она предполагает разработку продукта в небольших итерациях, что позволяет быстро реагировать на обратную связь заказчика и изменения требований.

Преимущества:

* Гибкость и возможность быстрого реагирования на изменения.
* Постоянное взаимодействие с заказчиком.
* Повышение мотивации команды разработки.

Недостатки:

* Требует высокой степени самоорганизации команды.
* Не всегда подходит для крупных и сложных проектов.

3. Spiral

Методология Spiral представляет собой комбинацию итеративного и инкрементального подходов, с постоянной оценкой рисков. Она подходит для проектов с высоким уровнем риска, где требуется поэтапная разработка и постоянное управление рисками.

Преимущества:

* Основан на постоянном учете рисков.
* Гибкий и адаптивный подход к разработке.
* Позволяет быстро выявлять и исправлять проблемы.

Недостатки:

* Требует дополнительного времени и ресурсов из-за дополнительных итераций.
* Сложность планирования и управления проектом.

4. RAD (Rapid Application Development)

Метод RAD направлен на создание быстрого прототипа продукта с минимальным набором функций, который затем постепенно дорабатывается и расширяется. Он подходит для проектов с ограниченными ресурсами и сроками.

Преимущества:

* Быстрое создание прототипа для быстрого понимания требований заказчика.
* Ускорение процесса разработки благодаря повторному использованию компонентов.

Недостатки:

* Риск возникновения проблем в процессе масштабирования и доработки.
* Требует высокой квалификации команды для быстрого развертывания.

5. DevOps

DevOps — это подход к разработке и внедрению программного обеспечения, который объединяет разработку и операции с целью автоматизации процессов и улучшения сотрудничества между командами.

Преимущества:

* Автоматизация процессов разработки и внедрения.
* Улучшение сотрудничества и коммуникации между разработчиками и операциями.
* Ускорение цикла разработки и доставки продукта.

Недостатки:

* Требует значительных изменений в культуре и процессах организации.
* Не всегда применим для небольших проектов или проектов с ограниченным бюджетом.

**2.3 Обоснование выбора методов проектирования**

Выбор подходящего метода проектирования играет ключевую роль в успехе проекта, так как от этого зависит эффективность, гибкость и качество разработки. Неправильный выбор метода может привести к проблемам в ходе разработки, увеличению времени и затрат, а также к неудовлетворенности заказчика.

Для проекта "BookRoom", системы управления гостиничным бизнесом, выбор метода проектирования является критическим шагом, определяющим успешность разработки и удовлетворение требований заказчика. В данном контексте предпочтительным методом проектирования является Agile. Рассмотрим подробнее причины такого выбора:

Гибкость и возможность быстрого реагирования на изменения в отрасли или запросы заказчика.

Гостиничный бизнес динамичен и подвержен постоянным изменениям. Agile методология предоставляет гибкий фреймворк, позволяющий быстро адаптироваться к новым требованиям и изменениям в отрасли. Регулярные итерации разработки позволяют оперативно внедрять новые функции или модифицировать существующие в ответ на запросы заказчика или изменения в бизнес-процессах.

Постоянное взаимодействие с заказчиком, что поможет лучше понять его потребности и ожидания.

В проекте "BookRoom" важно поддерживать постоянное взаимодействие с заказчиком для того, чтобы гарантировать, что разрабатываемая система соответствует его потребностям и ожиданиям. Agile методология предусматривает регулярные обзоры промежуточных результатов работы с заказчиком, что позволяет оперативно обсуждать требования, вносить изменения и уточнять детали.

Возможность быстрой разработки итераций продукта для оперативного предоставления новых функциональностей или исправления проблем.

В гостиничном бизнесе важно оперативно реагировать на изменения рыночной ситуации и запросы клиентов. Agile методология позволяет быстро создавать итерации продукта и оперативно внедрять новые функциональности или исправлять обнаруженные проблемы. Это обеспечивает высокую реактивность системы и удовлетворенность пользователей.

Создание адаптивной системы, которая легко изменяется в ответ на изменения в бизнес-процессах или требованиях рынка.

Гостиничный бизнес подвержен постоянным изменениям, и важно, чтобы система управления была способна легко адаптироваться к этим изменениям. Agile методология способствует созданию адаптивной системы, которая легко изменяется в ответ на изменения в бизнес-процессах или требованиях рынка. Это позволяет системе "BookRoom" оставаться актуальной и конкурентоспособной на протяжении всего срока эксплуатации.

Таким образом, учитывая динамичность гостиничного бизнеса, необходимость постоянного взаимодействия с заказчиком и оперативную реакцию на изменения в требованиях рынка, Agile методология является наиболее подходящей для проекта "BookRoom". Ее принципы гибкости, реактивности и постоянного взаимодействия помогут обеспечить успешное развитие и удовлетворение потребностей заказчика.

**2.4 Описание функциональной схемы программного продукта**

**2.5 Описание модульной структуры программного продукта**

**2.6 Описание структурной схемы программного продукта**

**2.7 Описание сценария пользовательского интерфейса**

**3 Технология программирования программного продукта**

**3.1 Технология программирования программного продукта**

**3.1.1 Исследование и обоснование выбора технологий программирования программного продукта**

**3.1.2 Описание технологии программирования программного продукта**

**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

**1 Нормативно-правовые акты**

1.

2.

3.

**2 Учебная и специальная литература**

4.

5.

6.

**3 Электронные ресурсы**

**7.**

8.

9.

**Заключение  
о прохождении производственной практики (преддипломной)**

Обучающийся/аяся ФИО полностью 4 курса группы 431 специальности 09.02.07 Информационные системы и программирование проходил/а производственную практику (преддипломную) с 12.04.2024 по 08.05.2024 в наименование организации/предприятия базы практики.

По итогам производственной практики обучающийся/аяся представил/а в ДИТИ НИЯУ МИФИ следующие документы:

1. Дневник по производственной практике (преддипломной) (включая аттестационный лист и характеристику).

2. Отчёт по производственной практике (преддипломной).

Отчёт по производственной практике (преддипломной) соответствует следующим характеристикам (нужное подчеркнуть):

- представлен своевременно/ несвоевременно;

- правильно и в полном объёме выполнены задания/ задания выполнены с ошибками/ задания не выполнены;

- приложения представлены в полном объеме/ приложения представлены в неполном объеме/ приложения не представлены.

Оформление дневника, отчета и приложений (нужное подчеркнуть):

- соответствует оформление дневника / не соответствует оформление дневника;

- соответствует оформление отчёта/ не соответствует оформление отчёта.

Результаты защиты отчёта производственнойпрактике (преддипломной), замечания и предложения:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Заключение о прохождении производственной практики (преддипломной):**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

(оценка прописью)

**Руководитель практики от**

**образовательной организации \_\_\_\_\_\_\_\_\_** И.О. Фамилия

подпись

08.05.2024