Министерство образования и науки Республики Башкортостан

Государственное бюджетное профессиональное образовательное учреждение

Уфимский колледж статистики, информатики и вычислительной техники

|  |  |
| --- | --- |
|  | УТВЕРЖДАЮ  Заместитель директора  по учебной работе  \_\_\_\_\_\_\_\_\_\_\_З.З. Курмашева  «\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2024 г. |

АВТОМАТИЗАЦИЯ БРОНИРОВАНИЯ НОМЕРОВ В ОТЕЛЕ

Пояснительная записка к курсовому проекту

МДК.05.02 Разработка кода информационных систем

|  |  |
| --- | --- |
|  | Руководитель проекта  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Е.К. Дмитриева  «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2024 г. |
|  | Студент гр. 21ВЕБ-1  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ П.Г. Валикаева  «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2024г. |

2024

# АННОТАЦИЯ

*Изм.*

*Лист*

*№ докум.*

*Подпись*

*Дата*

*Лист*

*2*

*40.И123-19 09.02.07 КП-ПЗ*

*Разраб.*

*Валикаева П.Г. И.И.*

*Провер.*

*Дмитриева Е.К.*

*Реценз.*

*Н. Контр.*

*Утверд.*

*Автоматизация системы бронирования номеров в отеле*

*Лит.*

*Листов*

*X*

*УКСИВТ 21ВЕБ-1*

Данный курсовой проект представляет собой автоматизированную информационную систему гостиницы и включает в себя хранение и обработку данных, получаемых от пользователя, создание брони и изменение ее статуса, управление номерами, услугами, пользователями .

Целью создания данной АИС является автоматизирование и упрощение бронирования номеров на сайте, также вывод актуальной информации о номерах и услугах. Программа дает возможность покупателю просматривать номера с учетом фильтрации, что гораздо упрощает пользование системой. Также система позволяет пользователю просматривать историю бронирований, что тоже оставляет положительный пользовательский опыт. Визуально пользовательский интерфейс имеет роскошный вид, что привлекает потенциального покупателя.

Для создания данного проекта применялись навыки создания макетов, знание таких языков программирования как: PHP, JavaScript и системы MySQL.

Содержание пояснительной записки

[АВТОМАТИЗАЦИЯ БРОНИРОВАНИЯ НОМЕРОВ В ОТЕЛЕ 1](#_3znysh7)

[2](#_2et92p0)

*Изм.*

*Лист*

*№ докум.*

*Подпись*

*Дата*

*Лист*

*НАЗВАНИЕ ДОКУМЕНТА*

# Введение

Актуальность: данная ИС имеет свою значимость, так как в наше время люди часто путешествуют в разнообразные места. Люди стараются продумывать свою поездку заранее и тщательно выбирают маршруты, а чтобы отдых или место временного пребывания было комфортным необходимо выбрать наиболее подходящее место проживания. Количество гостиниц растет с каждым годом, и выбирать становится все сложнее и сложнее. Наша гостиница предлагает на выбор несколько категорий номеров. Выбрать подходящий номер можно по нескольким вариантам фильтрации. С учетом выбора категории пользователю будет удобнее сортировать перечень доступных номеров по классу номера, его комфортности. Дизайн создается на основании предполагаемой целевой аудитории, чтобы создать наиболее комфортную систему для определенной категории пользователей.

Данная ИС предполагает упрощение выбора номера на основании информации о номере. Номер характеризуется названием, описанием, изображением, количеством номеров в отеле, категорией и коротким названием, которое предположительно должно упростить работу администратора. Категория в свою очередь имеет информацию о названии, количестве комнат в номере, количестве гостей, ценой и площадью. Если при бронировании пользователь не заполнил свой профиль он не сможет совершить его, об этом его уведомит система.

Цель: автоматизация системы бронирования номеров.

Задачи:

* спроектировать информационную систему гостиницы;
* составить наборы данных соответствующих категориям номеров;
* разработать систему бронирования;
* осуществить вывод, редактирование, создание и удаление информации о номерах;
* осуществить вывод, редактирование, создание и удаление информации об услугах;
* реализовать поиск номеров с использованием фильтра;
* осуществить ввод данных с пользовательской стороны.

1. Проектирование информационной системы
2. Описание предметной области

Гостиница

Гостиница предоставляет клиентам возможность бронирования, которое включает в себя выбор даты заезда и выезда, и выбор номера. При выборе номера учитывается его категория, количество гостей на которых рассчитан номер, количество комнат в номере и диапазон цен. Пользователи имеют возможность зарегистрироваться в системе, для этого надо заполнить информацию о пользователе. При бронировании данные будут автоматически заполняться из профиля. Если пользователь впервые пользуется услугами гостиницы, то при бронировании пользователь должен будет зарегистрироваться в системе и заполнить данные профиля. Для ознакомления можно просмотреть список услуг отеля, для бронирования которых необходимо подойти на ресепшн. Незарегистрированные пользователи также могут просматривать каталог номеров.

Пользователями информационной системы отеля являются:

* администратор;
* гость;
* покупатель.

Покупатель должен иметь в системе следующие возможности:

* авторизоваться;
* зарегистрироваться;
* вносить и изменять информацию в своем профиле;
* просматривать информацию о доступных номерах с учетом фильтрации;
* просмотреть подробную информацию о конкретном номере;
* просматривать информацию о дополнительных услугах с использованием фильтрации;
* просматривать историю своих бронирований;
* бронировать номера.

В свою очередь администратор должен иметь возможность:

* авторизоваться;
* управлять данными о номерах;
* управлять категориями номеров;
* просмотр списка бронирующих лиц;
* блокировка пользователей;
* управлять данными о дополнительных услугах;
* управлять категориями услуг;
* просмотр информации о номерах.

В системе можно выделить следующие сущности:

* номер;
* бронь;
* дополнительные услуги;
* покупатель;
* администратор.

Пользователь может посмотреть каталог доступных номеров. Каждый номер в системе характеризуется следующими параметрами:

* ID номера;
* длинное название;
* короткое название;
* изображение;
* количество номеров в отеле;
* описание;
* наличие в гостинице (true/false);
* категория.

Категории номеров содержат информацию о :

* ID категории номера;
* название категории;
* площадь номера;
* количество гостей;
* максимальное количество гостей;
* наличие в гостинице (true/false);
* количество комнат;
* цена за категорию.

Также на сайте имеется перечень услуг, характеристиками которого являются:

* ID услуги;
* название;
* описание;
* категория;
* изображение;
* наличие в гостинице (true/false);
* цена;

У каждой услуги есть категория. Категории характеризуются следующими полями:

* ID категории услуги;
* название;
* наличие в гостинице (true/false).

Для того чтобы пользователю забронировать номер необходимо создать профиль. Пользователь характеризуется следующими атрибутами:

* ID пользователя;
* почта;
* пароль;
* роль;
* статус блокировки.

Пользователи делятся на администраторов и покупателей. Информация администратора ограничивается информацией пользователя. В то время как в информацию о покупателе входят данные о:

- ID покупателя;

- ID пользователя;

- имя;

- фамилия;

- отчество;

- дата рождения;

- телефонный номер.

При выборе номера и заполнении данных формируется бронь, которая характеризуется следующими полями:

* ID брони;
* номер;
* пользователь;
* дата заезда;
* дата выезда;
* статус брони.

Ограничения на информацию:

* покупателю не доступны для бронирования номера, которые заняты в те даты, которые были выбраны;
* забронировать номер могут только пользователи, достигшие возраста совершеннолетия;
* бронирование номеров доступно только на ближайшие десять дней;
* администратору не доступна функция бронирования;
* минимальное количество дней бронирования номера: 2 дня;
* максимальное количество гостей в номере: 4 гостя;
* бронирование возможно только при заполнении профиля.

1. Диаграммы UML
   1. Диаграмма прецедентов

Диаграмма прецедентов- также называют use-case diagram или диаграмма вариантов использования. Является исходным концептуальным представлением системы в процессе ее проектирования и разработки. Диаграмма вариантов использования содержит варианты использования системы, действия лиц и связи между ними. Она представляет собой графта, в вершинах которого расположены актеры и прецеденты. Связи между ними это разного вида отношения. Диаграмма прецедентов для системы бронирования номеров представлена на рисунке 1.2.1.

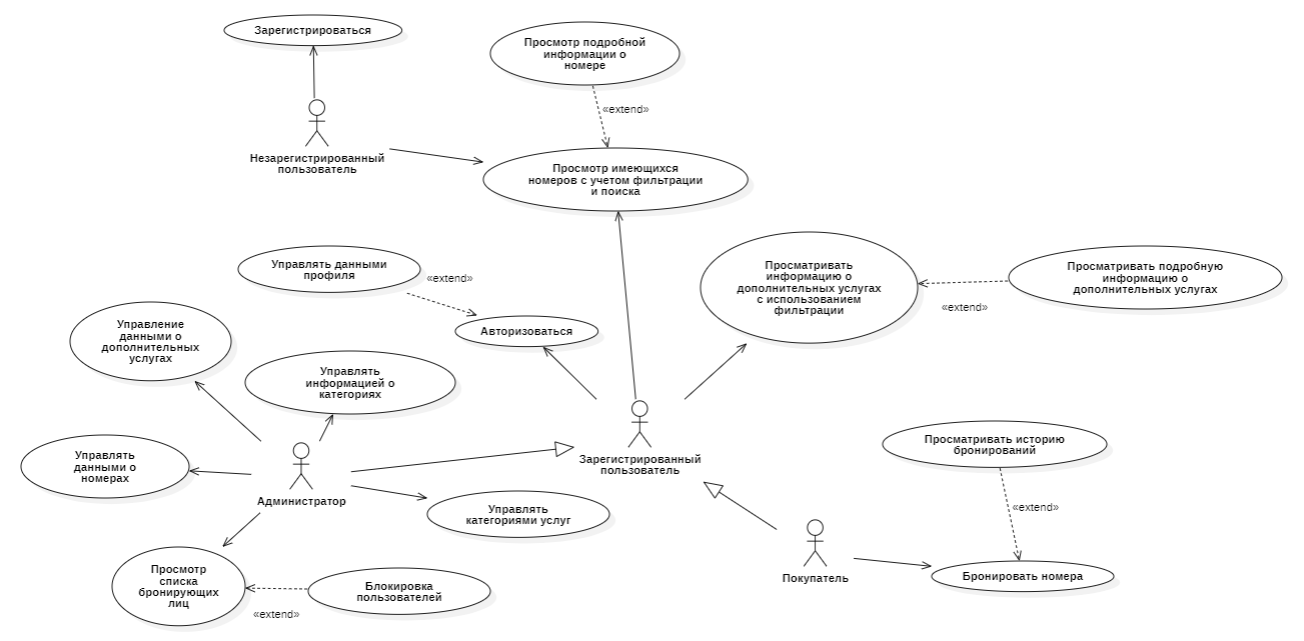


Рисунок 1.2.1 – Диаграмма прецедентов

* 1. Диаграмма активности

Диаграмма активности- также известна как диаграмма деятельности. Она представляет собой блок-схему, которая наглядно показывает, как поток управления переходит от одной деятельности к другой. Действия расположены на беговых дорожках, названиями этих беговых дорожек являются объектами, которые выполняют последовательные действия обращаясь друг к другу. Диаграмма активности изображена на рисунке 1.2.2.

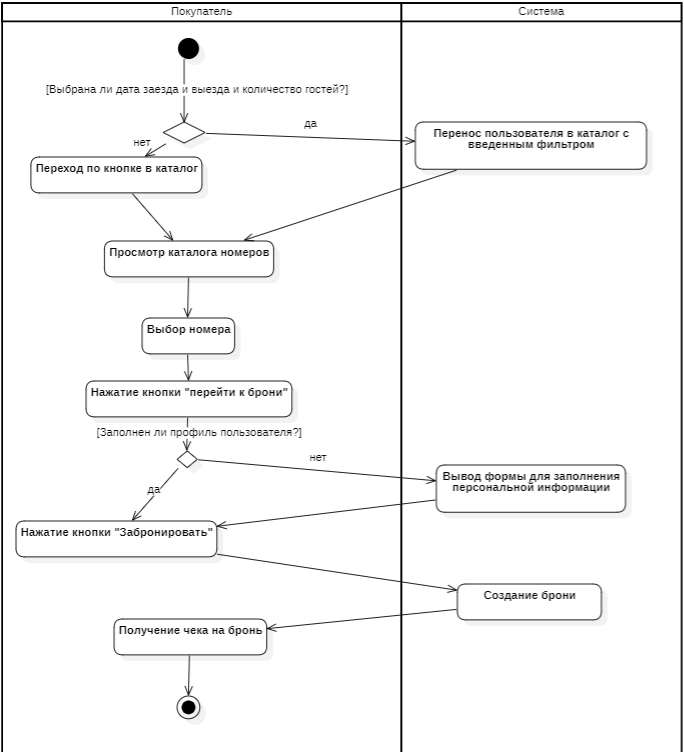


Рисунок 1.2.2 - Диаграмма активности

* 1. Диаграмма классов

Диаграмма классов- структурная диаграмма языка моделирования UML, демонстрирующая общую структуру иерархии классов системы, их коопераций, атрибутов (полей), методов, интерфейсов и взаимодействий между ними. Диаграмма классов изображена на рисунке 1.2.3.

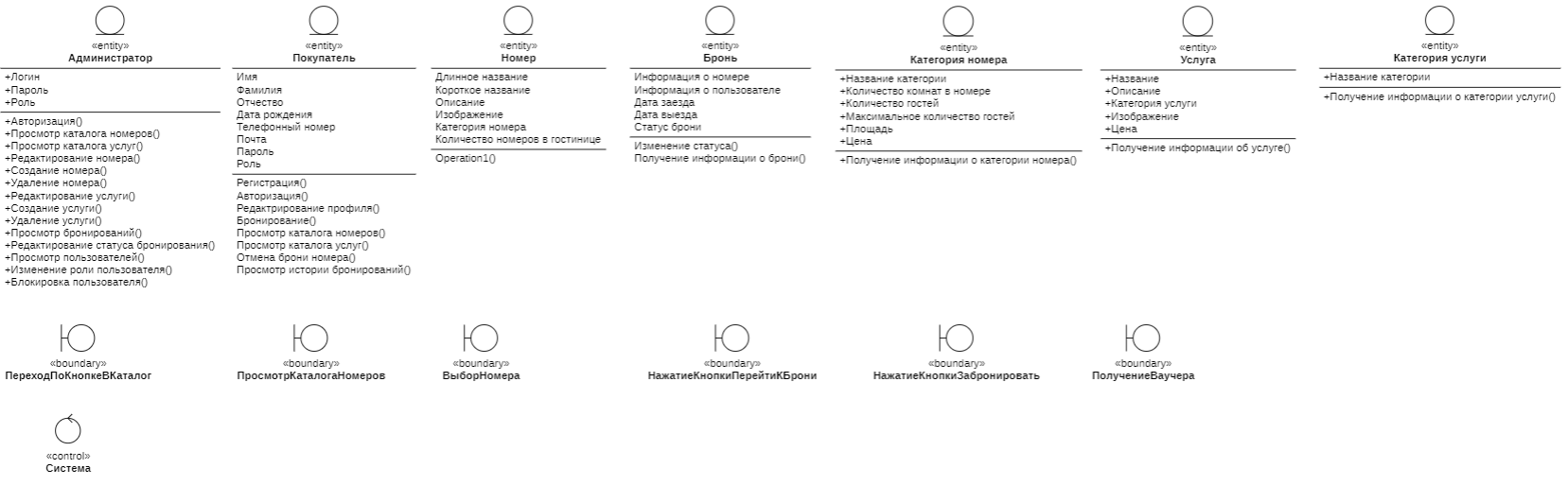


Рисунок 1.2.3 - Диаграмма классов

1. Описание входной информации

Входной информацией являются данные введенные пользователем - администратором, покупателем. Входной информацией покупателя являются персональные данные. Входной информацией администратора является информация о номерах, категориях номеров, услугах, категориях услуг. Пользователь также может менять информацию о себе в личном кабинете.

Одним из вариантов входной информации является информация о пользователе:

* фамилия;
* имя;
* отчество;
* дата рождения;
* номер телефона;
* почта.

Для регистрации:

* почта;
* пароль;
* дата рождения.

Для авторизации:

* почта;
* пароль.

Для изменения логина или пароля входной информацией будут являться:

* почта;
* пароль.

Для админа есть свои входные данные.

Входной информацией для добавления номера является:

* короткое название номера;
* длинное название номера;
* описание номера;
* изображение номера;
* количество номеров в отеле;
* категория номера.

Входной информацией для добавления категории номера:

* название категории номера;
* площадь;
* количество гостей;
* количество комнат;
* цена.

Входной информацией для добавления услуги:

* название услуги;
* описание услуги;
* категория услуги;
* изображение услуги;
* цена.

Входной информацией о категории услуги:

* название категории услуги.

1. Описание выходной информации

Выходной информацией является чек- информация о брони, которая характеризуется информацией о пользователе, информацией о номере, датой заезда, датой выезда, суммой брони.

Выходные документы представлены в таблице 1.4.1

Таблица 1.4.1- Описание выходных документов

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование документа (шифр) | Периодичность выдачи документа | Кол-во экз. | Куда передаются |
| Чек | После оформления заказа | 1 | Пользователь |

Форма выходных документов изображена на рисунке 1.4.1

Место для рисунка

Рисунок 1.4.1 - Ваучер (Чек)

1. Концептуальное моделирование

Концептуальное моделирование- процесс создания абстрактных представлений о реальном мире или предметной области. Это набор понятий, связей и ограничений, которые помогают описать и выяснить сущность и структуру Она служит основой для дальнейшего проектирования и реализации информационных систем. Концептуальная модель, ER-диаграмма представлена на рисунке 1.5.1.

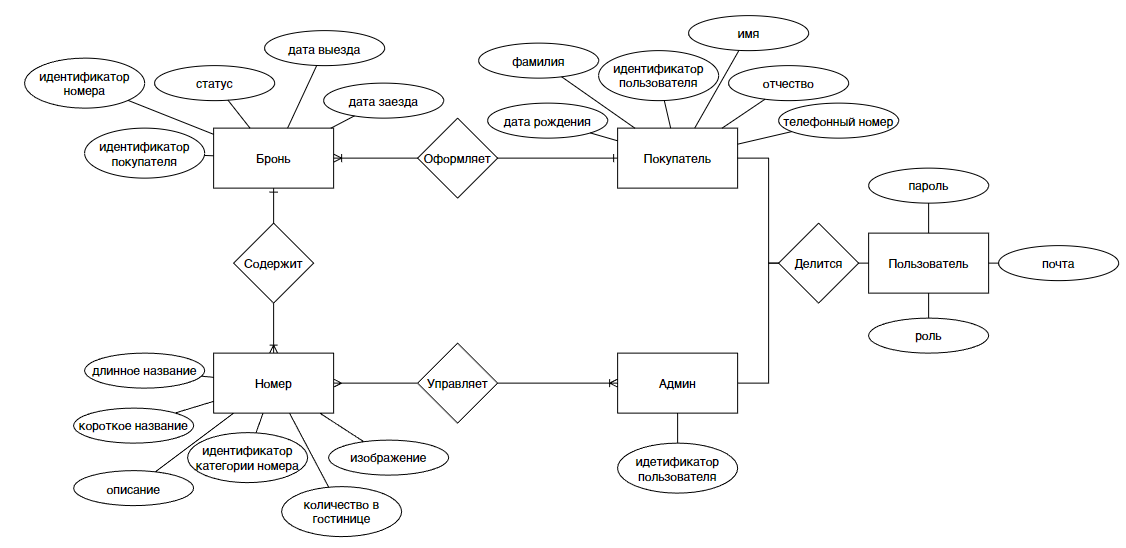


Рисунок 1.5.1 - ER- диаграмма

1. Логическая структура

Логическое моделирование- модель представляющая последовательность действий для передвижения по сайту. Логическая модель изображена на рисунке 1.6.1.

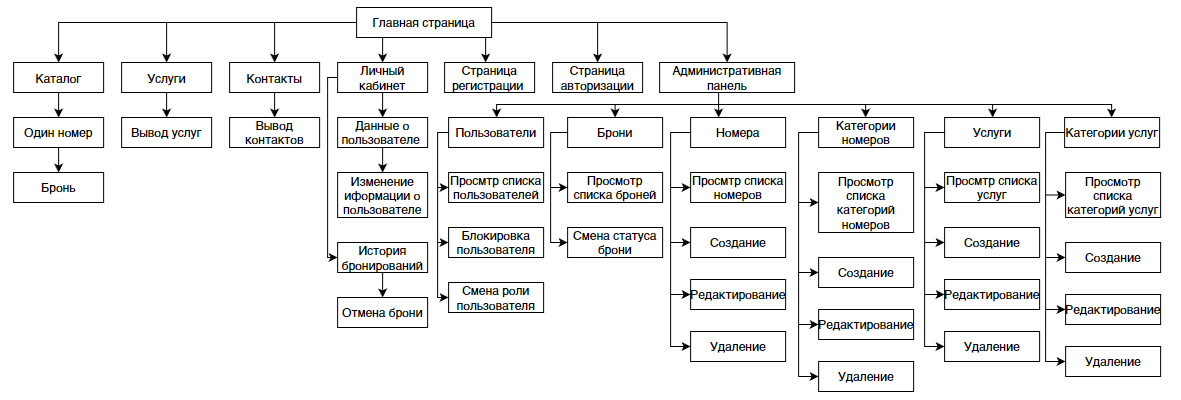


Рисунок 1.6.1 – Логическая модель

1. Описание структуры БД

Описание структуры БД предоставляет информацию о полях в создаваемой базе данных, описание таблиц, полей, находящихся в них, их тип, длину и примечания к ним. Описание структуры базы данных представлена на таблицах 1.7.1- 1.7.7.

Таблица 1.7.1 - «Order» (Заказ)

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечания |
| Идентификатор брони | ID\_BOOK | INT(11) | Первичный ключ |
| Идентификатор покупателя | ID\_USER | INT(11) | Внешний ключ к Buyer |
| Идентификатор номера | ID\_ROOM | INT(11) | Внешний ключ к Room |
| Дата заселения | DATE\_ARRIVAL | DATETIME | Обязательное поле |
| Дата выезда | DATE\_DEPARTURE | DATETIME | Обязательное поле |
| Статус брони | STATUS\_BOOK | SET(‘Принят’,’Выполнен’,’Отменен’) | Обязательное поле |

Таблица 1.7.2 - «Room» (Номер)

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечания |
| Идентификатор номера | ID\_ROOM | INT(11) | Первичный ключ |
| Короткое название номера | SHORT\_NAME\_ROOM | VARCHAR(150) | Обязательное поле |
| Название номера | LONG\_NAME\_ROOM | VARCHAR(270) | Обязательное поле |
| Идентификатор категории номера | ID\_CAT\_ROOM | INT(11) | Внешний ключ к categories room |
| Описание номера | DESC\_ROOM | TEXT | Обязательное поле |
| Количество номеров в отеле | AMOUNT\_IN\_HOTEL | INT(11) | Обязательное поле |
| Изображение номера | IMG\_ROOM | VARCHAR(150) | Обязательное поле |

Таблица 1.7.3 - «Categories room» (Категория номера)

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечания |
| Идентификатор категории номера | ID\_CAT\_ROOM | INT(11) | Первичный ключ |
| Название категории номера | NAME\_CAT\_ROOM | VARCHAR(200) | Обязательное поле |
| Площадь номера | SQUARE\_CAT\_ROOM | INT(11) | Обязательное поле |
| Количество комнат | AMOUNT\_ROOM\_IN\_ROOM | INT(11) | Обязательное поле |
| Количество гостей | NUMBER\_PERS | INT(11) | Обязательное поле |
| Цена категории номера | PRICE\_CAT\_ROOM | DECIMAL(8, 2) | Обязательное поле |

Таблица 1.7.4 - «Service» (Услуга)

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечания |
| Идентификатор услуги | ID\_SERVICE | INT(11) | Первичный ключ |
| Название услуги | NAME\_SERVICE | VARCHAR(150) | Обязательное поле |
| Описание услуги | DESC\_SERVICE | TEXT | Обязательное поле |
| Изображение услуги | IMG\_SERVISE | VARCHAR(150) | Обязательное поле |
| Идентификатор категории услуги | CAT\_SERVICE | INT(11) | Внешний ключ к categories services |
| Цена услуги | PRICE\_SERVICE | DECIMAL(7, 2) | Обязательное поле |

Таблица 1.7.5 - «Categories service» (Категория услуги)

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечания |
| Идентификатор категории услуги | ID\_CAT\_SERVICE | INT(11) | Первичный ключ |
| Название категории услуги | NAME\_CAT\_SERVICE | VARCHAR(150) | Обязательное поле |

Таблица 1.7.6 - «User» (Пользователь)

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечания |
| 1 | 2 | 3 | 4 |
| Идентификатор пользователя | ID\_USER | INT(11) | Первичный ключ |
| Почта | EMAIL | VARCHAR(100) | Обязательное уникальное поле |
| Пароль | PASSWORD | VARCHAR(100) | Обязательное поле |
| Роль | USER\_ROLE | VARCHAR(5) | Обязательное поле |

Таблица 1.7.7 - «Buyer» (Покупатель)

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечания |
| Идентификатор покупателя | ID\_BUYER | INT(11) | Первичный ключ |
| Идентификатор покупателя | ID\_USER | INT(11) | Внешний ключ к users |
| Имя покупателя | NAME\_BUYER | VARCHAR(50) | Необязательное поле |
| Фамилия покупателя | SNAME\_ BUYER | VARCHAR(50) | Необязательное поле |
| Отчество покупателя | PATHRONYMIC | VARCHAR(50) | Необязательное поле |
| Дата рождения | BIRTH\_ BUYER | DATE | Необязательное поле |
| Телефонный номер | PHONE\_ BUYER | VARCHAR(11) | Необязательное поле |
| Статус покупателя | BLOCKED | ENUM(‘0’,’1’) | Обязательное поле |

1.8 Контрольный пример

Контрольный пример предоставляет входные данные для проверки работы сайта. Контрольные примеры предоставлены в таблицах 1.8.1 1.8.

Таблица 1.8.1 - Контрольный пример (Регистрация пользователя)

|  |  |  |
| --- | --- | --- |
| Почта | Пароль | Дата рождения |
| kalina@mail.ru | 12345678 | 31-10-2005 |
| dimaLov@gmail.com | dimkalovv | 18-06-2007 |
| goroch@mail.ru | serezhkaGor | 05-08-1991 |
| minskiye@gmail.com | MinskTime | 18-10-2018 |
| aksiniya@mail.com | 8952812 | 11-04-56 |

Таблица 1.8.2- Контрольный пример (Авторизация)

|  |  |
| --- | --- |
| Почта | Пароль |
| admin@mail.ru | cropp |
| user@mail.ru | user |
| hope@mail.ru | hope |
| trapbed@mail.ru | trapbed |
| akakiy@mail.ru | akakiy |

Таблица 1.8.3- Контрольный пример (Изменение данных профиля)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Имя | Фамилия | Отчество | Дата рождения | Телефонный номер | Почта | Пароль |
| Рита | Усманова | Львовна | 11-06-2002 | 89123456789 | user@mail.ru | user |
| Константин | Петров | Андреевич | 18-09-1995 | 89987654321 | hope@mail.ru | hope |
| - | - | - | - | - | admin@mail.ru | admin |
| Трапбед | Трапбедов | Трапбедович | 31-10-2005 | 891234567810 | trapbed@mail.ru | yayavygoda |
| Николай | Гоголь | Васильевич |  |  | akakiy@mail.ru | 01-03-1809 |

Таблица 1.8.4- Контрольный пример (Смена логина или пароля)

|  |  |
| --- | --- |
| Логин | Пароль |
| user@mail.ru | firstUser |
| yourhope@mail.ru | imyourhope |
| admin@mail.ru | admin |
| trapbed@mail.ru | yamaguchi |
| milanas@mail.ru | milana |

Таблица 1.8.5- Контрольный пример (Создание/редактирование карточки номера)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Короткое название номера | Длинное название номера | Описание номера | Изображение номера | Количество номеров в отеле | Категория номера |
| 2 СтМ | Двухместный стандарт с видом на море | Красивый вид открывается с чудесного, уютного номера. Номер оснащен кондиционером, поэтому вам будет комфортно как в жару, так и в холод | standardSea.png | 3 | 1 |
| 4 Се | Семейный номер с тремя комнатами | Номер подходит для семей, с маленькими детьми, которые могут кричать круглые сутки. Также в номере имеется два санузла | famThreeRoom.png | 2 | 6 |
| Люкс 3 П | Люкс на три персоны | Люксовый номер на три персоны со всеми возможными привилегиями | Lux3P.png | 2 | 2 |
| Стандарт 1 К 1 | Одиночный номер категории стандарт | Номер категории стандарт для активных персон, которые не могут сидеть на месте и в номере им нужен минимальный комплект услуг. | standMin.png | 5 | 1 |
| 1 Э | Номер класса эконом на одного | Если вы отправились в путешествие в одиночку этот номер специально для вас. В номере есть все необходимое для жизни одного человека. | onePerson.png | 4 | 5 |

Таблица 1.8.6- Контрольный пример (Создание/редактирования категории номера)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название | Площадь(кв.м.) | Количество гостей | Количество комнат | Цена (руб) |
| Апартамент | 40 | 5 | 4 | 20000.00 |
| Джуниор- сьюит | 35 | 3 | 2 | 14000.00 |
| Студия | 25 | 2 | 1 | 7000.00 |
| Первая категория | 20 | 2 | 1 | 8000.00 |
| Делюкс | 25 | 2 | 1 | 10000.00 |

Таблица 1.8.7- Контрольный пример (Добавление услуги)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название | Описание | Категория | Изображение | Цена (руб) |
| Завтрак в номер | Наш официант донесет завтрак в нужное время прямо к вам в номер | Мероприятия & развлечения | brakfastInRoom.png | 800.00 |
| Детская вечерняя программа | Хотите вечером побыть вдвоем а детей не с кем оставить? У нас имеется детская программа. Детям не скучно, а вы получите драгоценные часы отдыха и спокойствия. | Детям | childAftrnooon.png | 500.00 |
| Вечерние дискотеки | Скучаете по своей молодости или фанат песен прошлых времен? Окунитесь в атмосферу того времени. Почувствуйте себя снова молодыми. | Музыка & танцы | olDisco.png | 500.00 |
| Мини библиотека |  |  |  | 0.00 |
| Доступ в бассейн | В гостинице имеется бассейн | Мероприятия & развлечения | swimpool.png | 700.00 |

1.9 Общие требования к программному продукту

1. Описание требований к программному продукту
2. Краткое описание проекта

Данный программный продукт создан для сервиса по онлайн-бронированию номеров -“LION”.

1. Целевые рабочие задачи:

Автоматизация бронирования номеров в гостинице.

1. Ссылки на нормативные документы:

- ;

1. Технические, программные средства:

Для ввода программного продукта необходимые следующие технические и программные средства:

1. Объем основной памяти:

- минимум 500 мб.

1. Типы периферийных устройств:

- устройства ввода;

- устройства вывода;

- устройства ввода- вывода.

1. Оборудование ввода и вывода:

Устройства вывода:

- монитор.

Устройства ввода:

- мышь;

- клавиатура;

- тачпад.

1. Сетевое оборудование:
2. Системные и прочие программные средства:

- СУБД MySQL;

- редактор кода VS Code;

- локальный веб- сервер Open Server Panel с поддержкой модулей: Apache, PHP, MySQL.

1. Интерфейсы или продукты необходимые для поддержки работы программного продукта:

-

1. Физические компоненты поставляемого продукта:

-

1. Вид поставляемых программ:

- ;

1. Необходимое программное обеспечение:

- Visual Studio Code;

- Open Server Panel.

1. Описание функциональных возможностей
2. Функции, выполняемые продуктом

Программный продукт должен выполнять следующие целевые рабочие задачи:

* поиск номера с использованием фильтра по некоторым критериям: количество гостей, количество комнат, категория и выбор дат;
* просматривать возможные услуги, предоставляемые гостиницей с учетом сортировки;
* регистрация и авторизация пользователей;
* изменение пользователем данных пользователя в личном кабинете;
* изменение пользователем пароля в личном кабинете;
* управление администратором информацией о номерах;
* управление администратором информацией о категориях номеров;
* управление администратором информацией об услугах;
* управление информацией о категориях услуг.

1. Граничные значения:

Граничные значения для ввода даты рождения при регистрации в системе представлены на таблице 1.9.1

|  |  |  |
| --- | --- | --- |
| № сценария тестирования | Описание | Ожидаемый результат |
| 1 | Ввод в поле “дата рождения” текущей даты или даты позже текущего даты | Вывод сообщения с текстом “Вы слишком молоды чтобы пользоваться нашей системой!” и невозможность зарегистрироваться в системе |
| 2 | Ввод в поле “дата рождения” даты рождения раньше текущей даты, но возраст пользователя меньше 18 | Вывод сообщения с текстом “Вам еще нет 18! Вы не сможете забронировать номер!  ”. Регистрация в системе. |
| 3 | Ввод в поле “дата рождения” даты рождения раньше текущей даты и возраст пользователя больше 18 | Вывод сообщения с текстом “Вы успешно зарегистрировались!  ”. Регистрация в системе. |

1. Надежность программного продукта заключается в:

- проверке достоверных данных при регистрации;

- выводе актуальных данных на страницах продукта;

- уведомлении пользователя о неправильно введенных данных;

- уведомлении пользователя об отсутствии результата запроса.

1. Описание эффективности:

- запуск системы - 2 сек;

- отправка сообщения на почту - 10 сек.

1. Средства предотвращения несанкционированного доступа к программам и данным:

- переадресация при попытке перехода в административную панель через строку запроса URL на главную страницу.

2. Экспериментальный раздел

2.1. Описание программы

Описание программы включает в себя описание модулей. Описание модулей- таблица, отражающая все используемые классы и их содержимое- методы. Прописывается файл и все хранящиеся в нем методы с кратким описанием функций, происходящих внутри класса (модуля). Описание модулей продемонстрировано в таблице 2.1.

Таблица 2.1. - Описание модулей (классов)

|  |  |
| --- | --- |
| Методы | Назначение |
| Класс Connect.php | |
| \_\_construct() | Метод для обращения к базе данных. |
| Класс User.php | |
| user\_exist($email, $pass) | Проверка на наличие пользователя с полученными почтой и паролем в базе данных. Возвращает булевое значение. |
| check\_bday($bday) | Проверка дня рождения, метод возвращает возраст и создает сессию с результатом проверки на совершеннолетие- булевый тип данных. |
| signin($pass, $email) | Функция авторизации. Внутри функции производится валидация полученных данных. Выводит сообщение пользователю и переадресует на главную страницу. Создает сессии с id пользователя, почтой и ролью пользователя. |
| signup($email, $bday, $pass) | Функция регистрации. Производит валидацию получаемых данных. Обращается к методам: user\_exist($email, $pass) и check\_bday($bday). Выводит сообщения пользователю о результате регистрации. Создает сессии, хранящие в себе id, email, role пользователя. |
| get\_info\_user($id) | Метод для получения информации о пользователе по id. |
| get\_email\_pass($id) | Метод для получение почты и пароля пользователя по id. |
| add\_info($id, $name, $sname, $path, $bday, $phone, $email) | Метод внесения и обновления данных о пользователе в профиле и базе данных. |
| changeLoginPass($id, $email, $pass) | Метод для изменения исключительно почты и пароля в профиле пользователя и в базе данных. |
| user\_exist\_email($email) | Проверка существования пользователя по почте |
| recoverAcc($email, $pass) | Восстановление аккаунта пользователя по почте со сменой пароля |
| get\_info\_user\_books($id\_user) | Получение информации о пользователе по идентификатору для бронирования номера |
| Класс Rooms.php | |
| exist\_rows($rooms) | Метод для проверки наличия существования номера . |
| output\_info($query) | Метод вывода информации о всех номерах. |
| get\_room\_in\_room() | Метод получения количества комнат в номере для фильтра. |
| get\_cat\_name() | Функция для получения категорий в фильтр. |
| get\_end\_pers($num) | Метод для создания фильтра по количеству гостей. |
| get\_rooms() | Метод получения номеров для вывода. |
| get\_one\_room($id\_room) | Метод для получения информации об одном номере. |
| search\_catalog($numPers, $numRooms, $cat, $priceForm, $priceTo) | Метод для создания запроса с фильтром для вывода номеров в каталоге. |
| Класс Services.php | |
| get\_services() | Метод для получения услуг. |
| get\_cat\_serv() | Метод для получения категорий услуг |
| services\_all($cat) | Метод для получения всех услуг по категории. |
| get\_array\_from\_id\_cat\_service() | Получение id и названий категорий услуг. |
| get\_id\_cat\_serv($id) | Получение категории услуги по id услуги. |
| get\_services\_by\_cat($id\_cat) | Получение всей информации об услугах по категории. Возвращение информации в виде массива из информации о количестве строк и массива из полученных данных. |
| get\_more\_info($what, $id) | Функция для получения описания услуги. |
| get\_title($id, $what) | Возвращает заголовок исходя из запрашиваемых данных. |
| Класс Admin\_info.php | |
| get\_info\_room() | Метод для получения информации о номерах. |
| get\_info\_room\_by\_id($id) | Получение информации об одном номере. |
| get\_info\_user() | Метод для получения информации о всех пользователях. |
| get\_info\_buyer($id) | Функция для получения информации о покупателе по id. |
| get\_cats\_room\_exist() | Метод проверяющий статус существования категории номера |
| get\_cats\_room() | Метод для получения информации о категориях номеров. |
| get\_all\_book() | Метод для получения информации о бронях |
| check\_name\_room($long, $short) | Проверка на существование введенных администратором названий номера. |
| get\_info\_serv() | Метод для получения информации об услугах. |
| get\_info\_cat\_serv() | Метод для получения информации о категориях услуг. |
| get\_more\_info($what, $id) | Метод для получения подробной информации по переданным данным. |
| get\_title($id, $what) | Метод, принимающий значение поля “смотреть” и создающий строку для заголовка модального окна. |
| get\_info\_one\_serv($id) | Возвращает массив из выборочных данных. |
| get\_info\_one\_service($id) | Возвращает массив из всех данных. |
| get\_serv\_by\_cat($id\_cat\_serv) | Метод возвращает массив из количества строк по запросу и массива данных. |
| get\_info\_cat\_serv\_by\_name($name) | Метод возвращает массив из количества строк по запросу и массива данных по запросу на поиск категорий по названию. |
| get\_info\_one\_cat\_serv($id) | Получение информации об одной категории услуг по идентификатору |
| check\_exist\_room\_with\_cat($id\_cat\_room) | Проверка на существование номера по категории. |
| get\_info\_cat\_room\_by\_id($id) | Получение информации о категории номера по идентификатору. |
| get\_info\_cat\_room\_by\_name($name) | Получение информации о категории номера по названию. |
| Класс Admin\_change.php | |
| block\_user($id) | Метод для блокировки пользователя по id. |
| unblock\_user($id) | Метод для разблокировки пользователя по id. |
| change\_book\_status($action, $id\_book) | Метод для смены статуса брони. Функция принимает действие и id брони. |
| upadate\_status\_service($id, $cat) | Функция, меняющая статус услуги на удаленное и наоборот. |
| update\_status\_cat\_service($id, $act) | Метод, меняющий статус категории услуги на удаленное или наоборот. |
| update\_name\_cat\_serv($id, $name) | Функция, меняющая название категории по id категории. |
| update\_status\_room($id, $act) | Изменение статуса номера на удаленное или наоборот, определяется по действию ($act) |
| update\_room($id, $long, $short, $desc, $img, $cat, $amount) | Изменение информации о номере. |
| change\_status\_cat\_room($id\_cat\_room, $act) | Изменение статуса категории номера на удаленное или наоборот. |
| update\_name\_cat\_room($id, $name, $square, $max, $amount\_room\_in\_room, $price) | Изменение информации о категории номера по $id категории. |
| change\_role($to, $id) | Изменение роли пользователя по $id, где роль на которую происходит смена $to. |
| Класс Admin\_create.php | |
| create\_service($name,$desc, $cat, $price, $img) | Создание услуги по полученным данным из формы. |
| create\_cat\_service($name) | Создание категории услуги по дынным из формы. |
| create\_room($long\_name, $short\_name, $desc, $cat, $amount\_rooms, $img, $tmp) | Создание номера по полученным данным из формы и передаваемым из промежуточного файла. |
| create\_cat\_room($name, $square, $max, $amount\_room\_in\_room, $price, $count) | Создание категории номера с данными из формы. |
| Класс Booking.php | |
| get\_email\_user($idUser) | Получение почты пользователя для ввода на страницу бронирования. |
| get\_name\_room($idRoom) | Получение названия номера для ввода на страницу бронирования. |
| check\_date($dateArrival, $dateDeparture) | Проверка на отправку дат заезда и выезда |
| get\_array\_book($idRoom) | Получение массива информации о номере |
| get\_exist\_dates($dateArrival, $dateDepart) | Проверка дат на бронирование из таблицы occupied\_rooms. |
| all\_dates\_between($dateArrival, $dateDepart) | Получение массива из дат между выбранными датами. |
| crate\_arr\_non\_exist\_dates($dateArrival, $dateDepart, $exist, $array\_between\_dates) | Создание массива из дат, которых нет в таблице occupied\_rooms. |
| check\_key($date, $id\_room) | Проверка ключа с id номера в таблице occupied\_rooms. |
| insert\_book($array\_between\_dates, $exist\_dates, $non\_exist\_dates, $array\_rooms, $id\_room | Метод, который заносит данные в таблицу occupied\_rooms. |
| insert\_into\_book($idUser, $idRoom, $dateArrival, $dateDepart, $old\_info) | Метод, заносящий данные в таблицу book |

Логическая структура представлена на рисунке 2.1.

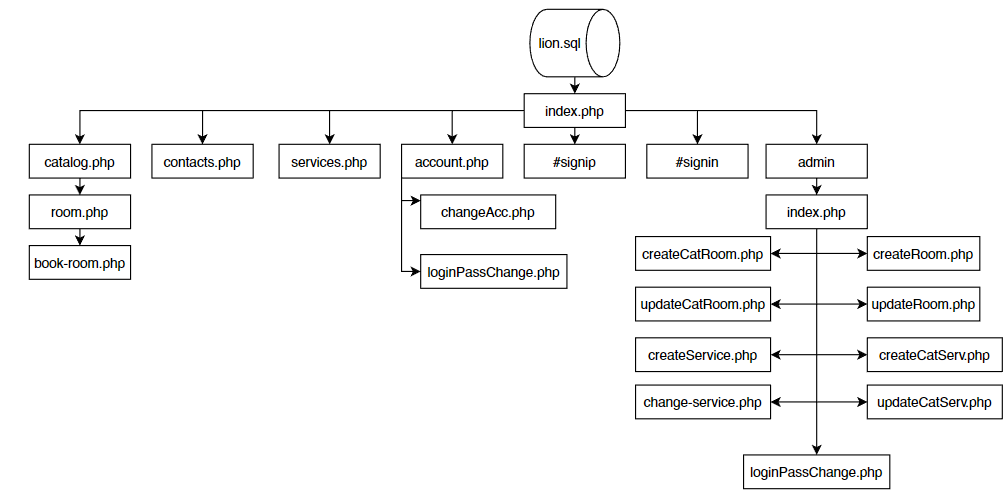


Рисунок 2.1- Логическая структура

Физическая структура представлена на рисунке 2.2

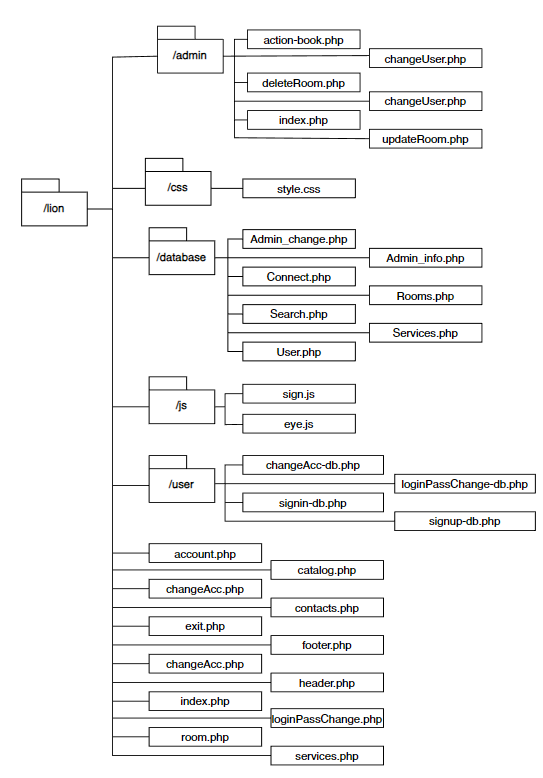


Рисунок 2.2- Физическая структура