**МИНОБРНАУКИ РОССИИ**

**Федеральное государственное бюджетное образовательное учреждение высшего образования**

**«Саратовский государственный технический университет имени Гагарина Ю.А.»**

**(СГТУ имени Гагарина Ю.А.)**

**ПРОФЕССИОНАЛЬНО-ПЕДАГОГИЧЕСКИЙ КОЛЛЕДЖ**

УТВЕРЖДАЮ

Заместитель директора

по учебно-методической работе Профессионально-

педагогического колледжа

СГТУ имени Гагарина Ю.А.

\_\_\_\_\_\_\_\_\_\_ О.В. Зимкова

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_2024 г.

**ЗАДАНИЕ НА ДИПЛОМНЫЙ ПРОЕКТ**

**Студента** Вдовина Кирилла Вячеславовича

**Специальность** 09.02.07 Информационные системы и программирование

**Группа** ИСП-41М

**Тема:** Проектирование и разработка автоматизированной системы для фотостудии

**Руководитель дипломного проекта** Пояркова Т.А., преподаватель ППК СГТУ им. Гагарина Ю. А.

**Рассмотрено на заседании цикловой методической комиссии** информационных технологий, физико-математических и энергетических дисциплин

Протокол № 8 от 01.04.2024 г.

Председатель ЦМК **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**Е.Н. Дмитриева

Дипломный проект способствует систематизации и закреплению знаний выпускника по специальности 09.02.07 Информационные системы и программирование при решении конкретных задач, а также выяснению уровня подготовки выпускника к самостоятельной работе и направлен на проверку качества полученных обучающимся знаний и умений, сформированности общих и профессиональных компетенций, позволяющих решать профессиональные задачи: ОК1- ОК 11, ПК 1.1-1.6, ПК 2.1-2.5, ПК4.1-4.4, ПК11.1-11.6

**Минимальные функциональные требования к системе:**

1. Наличие формы авторизации

2. Отображение данных

3. Добавление, удаление, изменение данных

4. Формирование чеков

5.Отображение должностей

6. Расчет заработной платы сотрудника

7. Экспорт

8. Возможность просмотра услуг

**Пояснительная записка дипломного проекта должна включать:**

Титульный лист

Содержание

Введение

Основная часть

1. Анализ предметной области и разработка требований к системе

1.1 Исследование предметной области. Формирование требований к системе

1.2 Сравнение аналогов программного обеспечения

1.3 Определение объектов базы данных

2. Разработка модулей информационной системы в соответствии с техническим заданием

2.1 Обоснование выбора средств реализации

2.2 Проектирование БД

2.3 Реализация базы данных

2.4 Защита информации в базе данных

2.5 Проектирование системы

2.6 Разработка программных модулей

2.7 Интеграция программных модулей

2.8 Тестирование и отладка

2.9 Рефакторинг программного кода

2.10 Соответствие кода стандартам кодирования

Заключение

Список использованных источников

**Программно-технологическая часть дипломного проекта должна включать:**

**Технологическая документация**

Техническое задание

**Программная часть**

Код программы

**Список рекомендуемых источников:**

**Нормативно – правовые акты**

1. ГОСТ 34.602-2020 Информационная технология. Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированной системы. [Электронный ресурс]. – Режим доступа: <http://www.rugost.com/index.php?option=com_content&view=article&id=96:gost-34602-89&catid=22&Itemid=53>

**Учебники и учебные пособия**

2. Федорова Г.Н. Разработка программных модулей программного обеспечения для компьютерных систем (2-е изд., стер.) учебник.- М.: Академия, 2021

3. Гниденко, И. Г. Технология разработки программного обеспечения: учебное пособие для среднего профессионального образования / И. Г. Гниденко, Ф. Ф. Павлов, Д. Ю. Федоров. — Москва: Издательство Юрайт, 2022. — 235 с. — (Профессиональное образование). — ISBN 978-5-534-05047-9. — Текст: электронный // ЭБС Юрайт [сайт]. — URL: <https://urait.ru>

4. Кумскова, И.А. Базы данных: учебник / Кумскова И.А. — Москва: КноРус, 2022. — 400 с. — (СПО). — ISBN 978-5-406-07467-1. — URL: https://book.ru

5. Перлова О.Н., Проектирование и разработка информационных систем: учебник для студ. учреждений сред.проф. Образования / О.Н.Перлова, О.П.Ляпина, А.В.Гусева. – 2-е изд. стер. – М.: Издательский центр «академия», 2021. – 256 с.

**Дополнительные издания**

6. Рудаков А.В. Технология разработки программных продуктов: учебник для студ. учреждений сред. проф. образования /А.В. Рудаков.- 12-е изд., стер.- Москва: Издательский центр "Академия", 2023.- 208с. ISBN 978-5-4468-7402-6

7. Федорова Г.Н. Осуществление интеграции программных модулей: учебник для студ. учреждений сред. проф. образования /Г.Н. Федорова.- 2-е изд., стер.- Москва: Издательский центр "Академия", 2022.- 288с. ISBN 978-5- 4468-6990-9

**Интернет-ресурсы**

8. Справочник по базам данных [Электронный ресурс]. – Режим доступа: http://msdn.microsoft.com/ru-ru/library/h0y4a0f6.aspx

9. Пошаговые руководства по C# [Электронный ресурс]. – Режим доступа: http://msdn.microsoft.com/ru-ru/library/1dbsh6t3.aspx

10. Учебные руководства [Электронный ресурс]. – Режим доступа: http://msdn.microsoft.com/ru-ru/library/dd492171.aspx

**Требования к предоставлению электронной версии дипломного проекта, презентации.**

Презентация к представлению дипломного проекта, пояснительная записка дипломного проекта и программный продукт записываются на электронный носитель.

**Дата выдачи задания** \_\_\_\_05.04\_\_ 2024\_\_г.

**Дата окончания работы над дипломным проектом**  \_\_\_\_\_10.06\_\_2024\_\_г.

**Руководитель дипломного проекта\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Пояркова Т.А.**

**Студент\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Вдовин К.В.**

Список использованных источников

[1. Анализ предметной области и разработка требований к системе 5](#_Toc165595053)

[1.1 Исследование предметной области. Формирование требований к системе 5](#_Toc165595054)

[1.2 Сравнение аналогов программного обеспечения 5](#_Toc165595055)

[1.3 Определение объектов базы данных 5](#_Toc165595056)

[2. Разработка модулей информационной системы в соответствии с техническим заданием 5](#_Toc165595057)

[2.1 Обоснование выбора средств реализации 5](#_Toc165595058)

[2.2 Проектирование БД 5](#_Toc165595059)

[2.3 Реализация базы данных 5](#_Toc165595060)

[2.4 Защита информации в базе данных 5](#_Toc165595061)

[2.5 Проектирование системы 5](#_Toc165595062)

[2.6 Разработка программных модулей 5](#_Toc165595063)

[2.7 Интеграция программных модулей 5](#_Toc165595064)

[2.8 Тестирование и отладка 5](#_Toc165595065)

[2.9 Рефакторинг программного кода 5](#_Toc165595066)

[2.10 Соответствие кода стандартам кодирования 5](#_Toc165595067)

# 

# Введение

В современном мире сфера фотографии и изобразительного искусства играет значительную роль, становясь неотъемлемой частью жизни общества. С развитием технологий и увеличением спроса на услуги фотостудий возникает необходимость в эффективном управлении процессами съёмки, обработки и предоставления готовых материалов клиентам. В связи с этим актуальной становится задача создания автоматизированных систем, способных оптимизировать работу фотостудии, увеличивать производительность и повышать уровень обслуживания.

Данный дипломный проект посвящён разработке и проектированию автоматизированной системы для фотостудии, которая позволит оптимизировать весь цикл работы: от приёма заказов до передачи готовых фотографий клиентам. Акцент будет сделан на создании удобного и интуитивно понятного интерфейса для пользователей, а также на функциональности системы, включающей в себя возможности управления заказами, планирования съёмок, автоматизации процесса обработки изображений и многого другого.

Целью данного проекта является предложение инновационного решения, способного улучшить эффективность работы фотостудии, повысить качество предоставляемых услуг и удовлетворить потребности клиентов в быстром и качественном выполнении заказов.

Введение 2 страницы убери задание оно будет отдельно добавь титульник

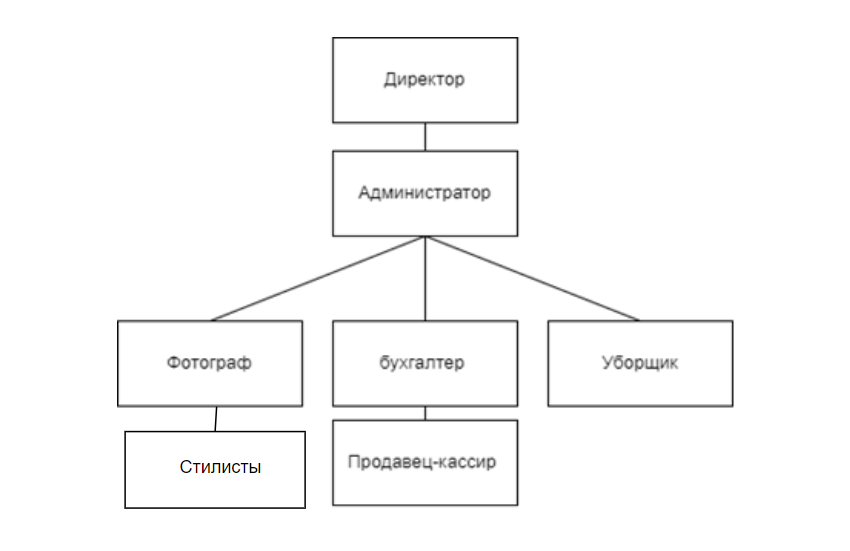
# 1. Анализ предметной области и разработка требований к системе

# 1.1 Исследование предметной области. Формирование требований к системе

Фотостудия - это специализированное предприятие, где осуществляется создание и обработка фотографий, а также организация фотосъемок. Организация работы фотостудии включает в себя процессы приема заказов, планирования съемок, проведения фотосессий и предоставления клиентам готовых фотографий. Основное внимание уделяется не только качеству фотографий, но и высокому уровню обслуживания клиентов.

Структура работы фотостудии обеспечивается качественное обслуживание клиентов в комфортной обстановке. Ниже представлена структура организации работы фотостудии.

Структура фотостудии показана в соответствии с рисунком 1.



Директор - выполняет такие функции как: Разработка стратегии развития бизнеса. Управление ресурсами и персоналом. Принятие ключевых решений по работе студии.

Администратор - выполняет такие функции как: Прием и обработка заказов. Организация расписания съемок. Взаимодействие с клиентами по телефону и по электронной почте. Обеспечение работы студии и поддержание порядка в помещении.

Фотограф - выполняет такие функции как: Подготовка к съемке: выбор оборудования, установка света и фонов. Проведение фотосессий в соответствии с требованиями заказчика. Редактирование и обработка полученных фотографий.

Бухгалтер - выполняет такие функции как: Ведение финансовой документации. Расчет заработной платы сотрудников. Учет расходов и доходов студии. Подготовка отчетности перед налоговыми органами.

Продавец-кассир - выполняет такие функции как: Продажа услуг фотостудии (фотосессий, печати фотографий и т. д.). Обработка платежей и выдача чеков. Консультирование клиентов по ассортименту услуг и ценам.

Уборщик - выполняет такие функции как: Поддержание чистоты и порядка в помещениях студии. Уборка после съемок и подготовка к следующим клиентам. Уход за оборудованием и помещениями.

Стилисты - выполняет такие функции как: Помогают клиентам подобрать образ и стиль для фотосессии.

# 1.2 Сравнение аналогов программного обеспечения

С целью обеспечения высокого уровня качества итоговой работы был проведен анализ аналогов (потенциальных конкурентов). Это позволило определить недостатки и достоинства современных веб-приложений. Прямых аналогов АС фотостудии «Под прицелом» нет, но существуют программные продукты с похожими функционалами.

В процессе анализа конкурентов было выявлено, что сайт для фотостудии «Под прицелом» имеет более удобный и оптимизированный интерфейс, возможность просмотра фотозон и самой студии в целом, имеет более информативные и актуальные данные.

Результат сравнительного анализа приведены в таблице 1.

Таблица 1

Сравнительный анализ веб-сайтов ресторанов

|  |  |  |  |
| --- | --- | --- | --- |
| Название сайта/  Категория | PHOTOLACE | Ellips Studio | ARTLIST |
| Адрес | https://photoplace.pro/saratov | https://ellipsstudio.ru | https://artlist.pro/saratov/studios |
| Удобный интерфейс | + | - | - |
| Прайс с ценами | + | - | + |
| Наличие онлайн бронирования студи | - | - | + |

Таким образом, в результате сравнительного анализа было выявлено, что не все сайты имеют удобный интерфейс

# 1.3 Определение объектов базы данных

На основе анализа информационных потребностей были выявлены следующие сущности (таблицы) базы данных, разрабатываемой АС для организации «Под прицелом»:

Список объектов предметной области представлен в соответствии с таблицей 5.

Таблица 5

Объекты предметной области

|  |  |  |
| --- | --- | --- |
| **№ п/п** | **Наименование объекта** | **Краткое описание** |
| 1 | Сотрудники | Сотрудники, работающие в фотостудии |
| 2 | Оборудование | Имеющиеся оборудование |
| 3 | Аренда оборудования | Возможность арендовать оборудование |
| 4 | Роли | Роль каждого сотрудника |
| 5 | Аренда студии | Возможность аренды студии |
| 6 | Студии | Имевшиеся студии |

В таблице 6 приведены атрибуты сущности «Сотрудники»

Таблица 6

Атрибуты сущности «Сотрудники»

|  |  |  |
| --- | --- | --- |
| **№ п/п** | **Наименование атрибута** | **Краткое описание** |
| 1 | Имя | Необходимо для идентификации сотрудника |
| 2 | Фамилия | Необходимо для идентификации сотрудника |
| 3 | Дата рождения |  |
| 4 | Пол | Необходимо для уточняя сотрудника |
| 5 | Адрес | Необходимо, чтобы знать где живет сотрудник |
| 6 | Телефон | Необходимо, чтобы связаться с сотрудником |
| 7 | электронная почта | Необходимо, для регистрации сотрудника в систему |
| 8 | Пароль | Необходимо для входа сотрудника в систему |

В таблице 7 приведены атрибуты сущности «Оборудование»

Таблица 7

Атрибуты сущности «Оборудование»

|  |  |  |
| --- | --- | --- |
| **№ п/п** | **Наименование атрибута** | **Краткое описание** |
| 1 | Название оборудования | Необходимо для уточнения того, как называется оборудование |
| 2 | Описание | Необходимо, для уточнения информации |
| 3 | Бронирование | Необходимо для бронирование |

В таблице 8 приведены атрибуты сущности «Аренда оборудования»

Таблица 8

Атрибуты сущности «Аренда оборудования»

|  |  |  |
| --- | --- | --- |
| **№ п/п** | **Наименование атрибута** | **Краткое описание** |
| 1 | дата начала | Необходимо для определения начала брони |
| 2 | Дата окончания | Необходимо для определения конца брони |
| 3 | Стоимость | Необходимо для определения стоимости |

В таблице 9 приведены атрибуты сущности «Роли»

Таблица 9

Атрибуты сущности «Роли»

|  |  |  |
| --- | --- | --- |
| **№ п/п** | **Наименование атрибута** | **Краткое описание** |
| 1 | Роли | Необходимо для идентификации роли сотрудника |
| 2 | Наименование роли | Необходимо для определения роли сотрудника |

В таблице 10 приведены атрибуты сущности «Аренда студии»

Таблица 10

Атрибуты сущности «Аренда студии»

|  |  |  |
| --- | --- | --- |
| **№ п/п** | **Наименование атрибута** | **Краткое описание** |
| 1 | Дата начала | Необходимо для определения начала брони |
| 2 | Дата окончания | Необходимо для определения конца брони |

В таблице 11 приведены атрибуты сущности «Студии»

Таблица 11

Атрибуты сущности «Студии»

|  |  |  |
| --- | --- | --- |
| **№ п/п** | **Наименование атрибута** | **Краткое описание** |
| 1 | Название студии | Необходимо для определения студии |
| 2 | Наименование товара | Необходимо для уточнения товара |
| 3 | стоимость бронирования | Необходимо для определения стоимости |

Концептуальная модель базы данных представлена в соответствии с рисунком 17 в виде ER-диаграммы и приведена в первой и второй нормальной форме.

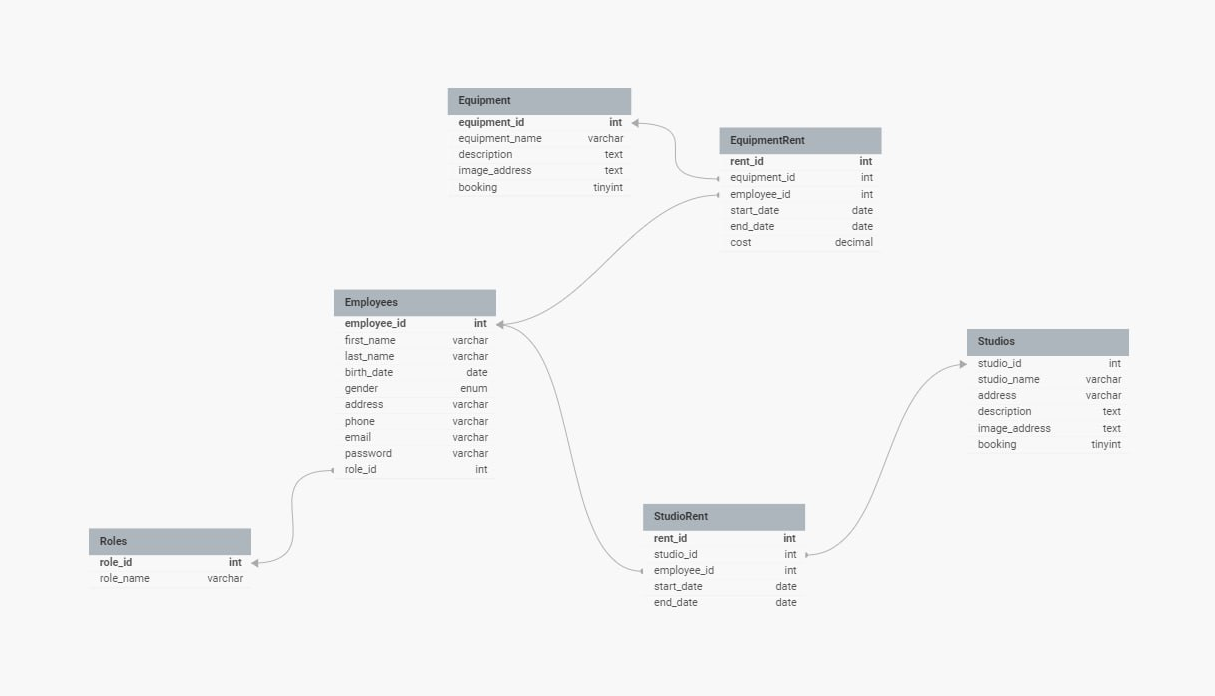


Рисунок 17. ER-диаграмма во второй нормальной форме

# 2. Разработка модулей информационной системы в соответствии с техническим заданием

# 2.1 Обоснование выбора средств реализации

Для реализации серверной части дипломного проекта было решено рассмотреть три варианта используемой СУБД: SQL Server 2022, MySQL и Visual Studio Code

SQL Server 2022 - система управления реляционными базами данных (РСУБД), разработанная корпорацией Microsoft. Основной используемый язык запросов - Transact-SQL.

Достоинства:

1. В SQL Server 2019 имеется большой набор интегрированных служб, расширяющих возможности использования данных: можно составлять запросы, выполнять поиск, проводить синхронизацию, делать отчеты, анализировать данные;
2. Все данные хранятся на основных серверах, входящих в состав центра обработки данных. К ним осуществляется доступ с настольных компьютеров и мобильных устройств. Таким образом, осуществляется контроль над данными вне зависимости от того, где они хранятся;
3. Система SQL Server 2019 позволяет обращаться к данным из любого приложения, разработанного с применением технологий Microsoft .NET и Visual Studio.

Взаимодействовать с такой системой можно как на простых ноутбуках, так и на ПК с мощным процессором, который способен обрабатывать большой объем запросов

Недостатки:

1. Высокая избыточность;
2. Зависимость от операционной среды: СУБД работает только с системой Windows.

MySQL - одна из самых популярных СУБД для веб-приложений. Фактически, является стандартом для веб-серверов, которые работают под управлением операционной системы Linux. MySQL - это бесплатный пакет программ, однако новые версии выходят постоянно, расширяя функционал и улучшая безопасность. Существуют специальные платные версии, предназначенные для коммерческого использования. В бесплатной версии наибольший упор делается на скорость и надежность, а не на полноту функционала.

Достоинства:

1. Пакет MySQL включен в стандартные репозитории наиболее распространённых дистрибутивов операционной системы Linux, что позволяет устанавливать её элементарно просто;
2. Поддерживает набор пользовательских интерфейсов;
3. Может работать с другими базами данных, включая DB2 и Oracle;
4. Скорость и надежность.

Недостатки:

1. MySQL с трудом выполняет несложные задачи, хотя другие системы делают это автоматически, например: создавать инкрементные резервные копии;
2. Сложнее взаимодействовать через WPF, чем с SQL Server 2022, так как наиболее подходит для создания веб-приложений;
3. Отсутствует встроенная поддержка XML или OLAP.

Visual Studio Code— это [редактор исходного кода](https://ru.wikipedia.org/wiki/%D0%A0%D0%B5%D0%B4%D0%B0%D0%BA%D1%82%D0%BE%D1%80_%D0%B8%D1%81%D1%85%D0%BE%D0%B4%D0%BD%D0%BE%D0%B3%D0%BE_%D0%BA%D0%BE%D0%B4%D0%B0). Он имеет многоязычный интерфейс пользователя и поддерживает ряд [языков программирования](https://ru.wikipedia.org/wiki/%D0%AF%D0%B7%D1%8B%D0%BA_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F), [подсветку синтаксиса](https://ru.wikipedia.org/wiki/%D0%9F%D0%BE%D0%B4%D1%81%D0%B2%D0%B5%D1%82%D0%BA%D0%B0_%D1%81%D0%B8%D0%BD%D1%82%D0%B0%D0%BA%D1%81%D0%B8%D1%81%D0%B0), [IntelliSense](https://ru.wikipedia.org/wiki/IntelliSense" \o "IntelliSense), [рефакторинг](https://ru.wikipedia.org/wiki/%D0%A0%D0%B5%D1%84%D0%B0%D0%BA%D1%82%D0%BE%D1%80%D0%B8%D0%BD%D0%B3" \o "Рефакторинг), [отладку](https://ru.wikipedia.org/wiki/%D0%9E%D1%82%D0%BB%D0%B0%D0%B4%D0%BA%D0%B0_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D1%8B), навигацию по коду, поддержку [Git](https://ru.wikipedia.org/wiki/Git" \o "Git) и другие возможности.

Достоинства:

1. Высокая скорость в обслуживании БД;
2. Наличие собственного развитого, простого и удобного языка программирования, мастеров, конструкторов, обработка нескольких баз данных с различными СУБД;

Недостатки:

1. Сложность написания программ
2. Отсутствие средств защиты и восстановления информации
3. Ограничения на объем информации
4. Наличие ошибок при установке связей между таблицами в запросах, формируемых визуальными средствами.

Была составлена сравнительная таблица (таб.16) и на основе нее сделан вывод: наиболее подходящей СУБД для разрабатываемой системы является SQL Server 2022, так как он имеет необходимый и полный функционал, а также с ним наиболее удобно работать с применением технологий Microsoft .NET.

Таблица 16. Сравнительная таблица СУБД

|  |  |  |  |
| --- | --- | --- | --- |
| Характеристики | SQL Server | Visual Studio Code | MySQL |
| Защита и восстановление данных | Высокая степень защиты данных от случайных потерь и от несанкционированных потерь. | Отсутствие средств защиты и восстановления информации | Надежная СУБД |
| Производительность | Высокая производительность | Высокая скорость в обслуживании БД | Высокая скорость работы, есть проблемы с выполнением несложных задач. |
| Простота в использовании | Тесная интеграция с операционной системой Windows | Сложность написания программ | 3/5 Отсутствует встроенная поддержка XML или OLAP, но поддерживает набор пользовательских интерфейсов |

# 2.2 Проектирование БД

В функциональном назначении разрабатываемый программный продукт имеет следующие функции:

* Авторизация
* Просмотр фотостудий
* Бронирование фотостудий
* Просмотр бронирования
* Добавление новой брони

Сущностями базы данных будут являться: «Сотрудники», «Оборудование», «Аренда оборудования», «Роли», «Аренда студии», «Студии»

Атрибуты сущности «Сотрудники» представлены в соответствии с таблицей 12.

Таблица 1

Атрибуты сущности «Сотрудники»

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **№ п/п** | **Наименование** | Идентификатор | Тип | Длина | Ограничения |
| **1** | идентификатор сотрудника | employee id | Числовой | 50 | Уникальный |
| 2 | имя | first name | Текстовый | 50 | NOT NULL |
| 3 | фамилия | last name | Текстовый | 50 | NOT NULL |
| 4 | дата рождения | birth date | Числовой | 50 | NOT NULL |
| 5 | пол | gender address | Текстовый |  | NOT NULL |
| 6 | телефон | phone | Числовой | 20 | NOT NULL |
| 7 | электронная почта | email | Текстовый | 50 | NOT NULL |
| 8 | пароль | password | Числовой | 50 | NOT NULL |
| 9 | идентификатор должности | role id | Числовой | 50 | Уникальный |

В таблице 13 приведены атрибуты сущности «Оборудование»

Таблица 13

Атрибуты сущности «Оборудование»

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **№ п/п** | **Наименование атрибута** | Идентификатор | Тип | Длина | Ограничения |
| 1 | идентификатор оборудования | equipment id | Числовой | 20 | Уникальный |
| 2 | название | name | Текстовый | 50 | NOT NULL |
| 3 | оборудования описание | description | Текстовый | 50 | NOT NULL |
| 4 | изображение адрес | image address | Текстовый | 50 | NOT NULL |
| 5 | бронирование | booking | Текстовый |  | NOT NULL |

В таблице 14 приведены атрибуты сущности «Аренда оборудования»

Таблица 14

Атрибуты сущности «Аренда оборудования»

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **№ п/п** | **Наименование атрибута** | Идентификатор | Тип | Длина | Ограничения |
| 1 | идентификатор арендатора идентификатор | rent id | Числовой | 20 | Уникальный |
| 2 | оборудования идентификатор | equipment id | Числовой | 20 | NOT NULL |
| 3 | сотрудника | employee id | Числовой | 20 | NOT NULL |
| 4 | дата начала | start date | Тип даты | 10 | NOT NULL |
| 5 | дата окончания | end date | Тип даты | 10 | NOT NULL |
| 6 | Цена | cost | Денежный |  | NOT NULL |

В таблице 14 приведены атрибуты сущности «Роли»

Таблица 15

Атрибуты сущности «Роли»

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **№ п/п** | **Наименование атрибута** | Идентификатор | Тип | Длина | Ограничения |
| 1 | идентификатор роли | role id | Текстовый | 20 | Уникальный |
| 2 | имя роли | role name | Текстовый | 20 | NOT NULL |

В таблице 15 приведены атрибуты сущности «Аренда студии»

Таблица 15

Атрибуты сущности «Аренда студии»

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **№ п/п** | **Наименование атрибута** | Идентификатор | Тип | Длина | Ограничения |
| 1 | идентификатор арендатора | rent id | Числовой | 20 | Уникальный |
| 2 | Идентификатор судии | studio id | Числовой | 20 | Уникальный |
| 3 | идентификатор сотрудника | employee id | Числовой | 20 | Уникальный |
| 4 | дата начала | start date | Числовой | 10 | NOT NULL |
| 5 | Дата окончания | end date | Тип даты | 10 | NOT NULL |

В таблице 16 приведены атрибуты сущности «Студии»

Таблица 16

Атрибуты сущности «Студии»

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **№ п/п** | **Наименование атрибута** | Идентификатор | Тип | Длина | Ограничения |
| 1 | идентификатор | studio id | Числовой | 20 | Уникальный |
| 2 | Название студии | studio name | Текстовый | 20 | NOT NULL |
| 3 | Адрес студии | address | Текстовый | 20 | NOT NULL |
| 4 | описание | description | Текстовый | 10 | NOT NULL |
| 5 | Адрес изображения | image address | Текстовый | 10 | NOT NULL |
| 6 | бронирование | booking | Числовой |  | NOT NULL |
|  | Стоимость | cost | Денежный |  | NOT NULL |

На основе приведенных таблиц была составлена ER-диаграмма 3-го уровня, представленная в соответствии с рисунком.18

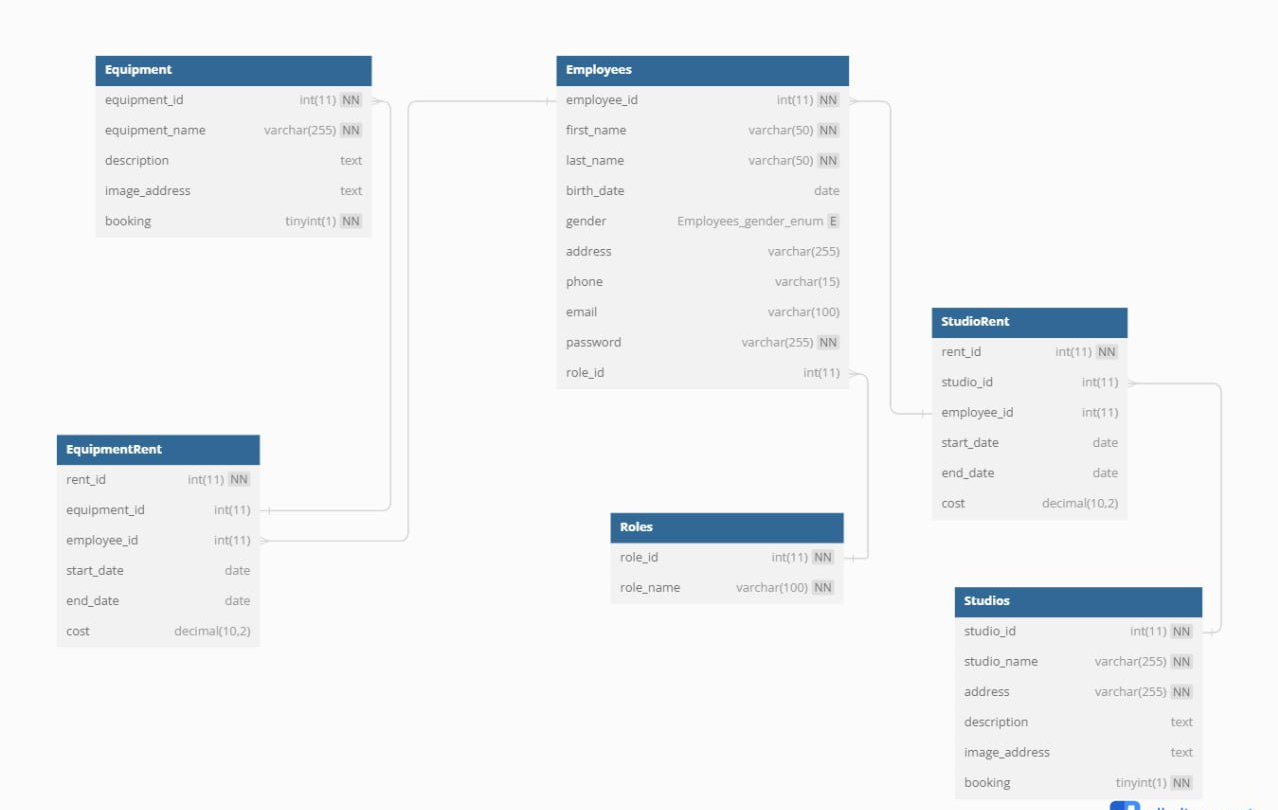


Рисунок 18. ER-диаграмма базы данных, разработанная в DB Designer

# 2.3 Реализация базы данных

Для того, чтобы разрабатываемое программное обеспечение работало с данными, необходимо спроектировать базу данных.

Создаём базу данных .photostudio

Код создания БД:

CREATE DATABASE IF NOT EXISTS photostudio;

Результат выполненного запроса по созданию БД представлен в соответствии с рисунком 19

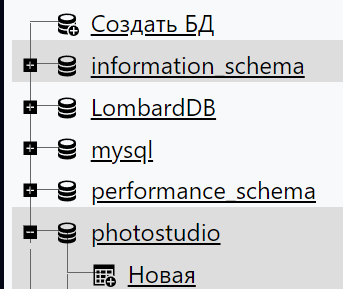


Рисунок.19 Результат выполненного запроса по созданию БД

Чтобы создать таблицу «Сотрудники», которая в базе данных будет называться [Employees](http://127.0.0.1:120/openserver/phpmyadmin/index.php?route=/sql&db=photostudio&table=Employees&pos=0), нужно создать следующий запрос:

Код запроса на создание таблицы «Сотрудники»:

CREATE TABLE `Employees` (

`employee\_id` int(11) NOT NULL,

`first\_name` varchar(50) COLLATE utf8mb4\_unicode\_ci NOT NULL,

`last\_name` varchar(50) COLLATE utf8mb4\_unicode\_ci NOT NULL,

`birth\_date` date DEFAULT NULL,

`gender` enum('Male','Female') COLLATE utf8mb4\_unicode\_ci DEFAULT NULL,

`address` varchar(255) COLLATE utf8mb4\_unicode\_ci DEFAULT NULL,

`phone` varchar(15) COLLATE utf8mb4\_unicode\_ci DEFAULT NULL,

`email` varchar(100) COLLATE utf8mb4\_unicode\_ci DEFAULT NULL,

`password` varchar(255) COLLATE utf8mb4\_unicode\_ci NOT NULL,

`role\_id` int(11) DEFAULT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci;

Код запроса на заполнение таблицы «Сотрудники»

INSERT INTO `Employees` (`employee\_id`, `first\_name`, `last\_name`, `birth\_date`, `gender`, `address`, `phone`, `email`, `password`, `role\_id`) VALUES

(1, 'Иван', 'Иванов', '1990-05-15', 'Male', 'Москва', '123456789', 'ivan@example.com', 'ivanpassword', 1),

(2, 'Елена', 'Петрова', '1985-08-20', 'Female', 'Санкт-Петербург', '987654321', 'elena@example.com', 'elenapassword', 2),

(3, 'Алексей', 'Сидоров', '1992-12-10', 'Male', 'Киев', '456789123', 'alex@example.com', 'alexpassword', 3),

(4, 'Мария', 'Смирнова', '1988-03-25', 'Female', 'Минск', '789123456', 'maria@example.com', 'mariapassword', 4),

(5, 'Павел', 'Козлов', '1995-07-03', 'Male', 'Одесса', '321654987', 'pavel@example.com', 'pavelpassword', 5),

Результат выполненных запросов на создание таблицы «Сотрудники» представлен в соответствии с рисунком 20

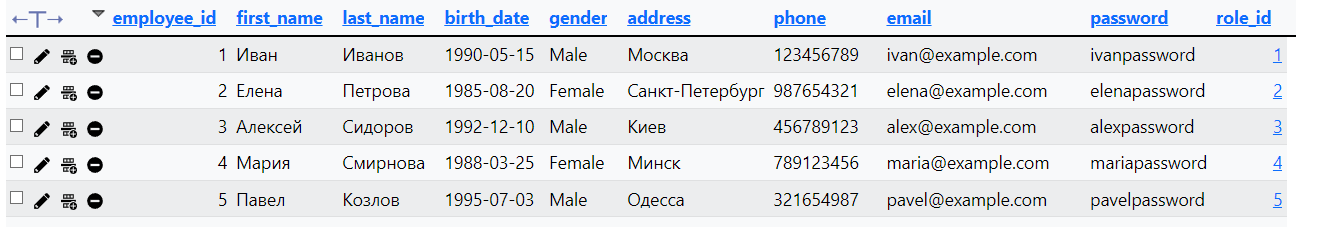


Рисунок .20 Результат выполненных запросов на создание таблицы «Сотрудники»

Чтобы создать таблицу «Оборудование», которая в базе данных будет называться , [Equipment](http://127.0.0.1:120/openserver/phpmyadmin/index.php?route=/sql&db=photostudio&table=Equipment&pos=0) нужно создать следующий запрос:

CREATE TABLE `Equipment` (

`equipment\_id` int(11) NOT NULL,

`equipment\_name` varchar(255) COLLATE utf8mb4\_unicode\_ci NOT NULL,

`description` text COLLATE utf8mb4\_unicode\_ci DEFAULT NULL,

`image\_address` text COLLATE utf8mb4\_unicode\_ci DEFAULT NULL,

`booking` tinyint(1) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci;

Код запроса на заполнение таблицы «Оборудование»

INSERT INTO `Equipment` (`equipment\_id`, `equipment\_name`, `description`, `image\_address`, `booking`) VALUES

(1, 'Canon EOS 5D Mark IV', 'Профессиональная фотокамера', 'http://example.com/camera1.jpg', 0),

(2, 'Nikon D850', 'Высококлассная зеркальная камера', 'http://example.com/camera2.jpg', 0),

(3, 'Profoto B10', 'Мощное студийное освещение', 'http://example.com/lighting1.jpg', 0),

(4, 'Manfrotto 055', 'Стабильный и надежный штатив', 'http://example.com/tripod1.jpg', 0),

(5, 'DJI Ronin-S', 'Гироскопический стабилизатор для видеосъемки', 'http://example.com/gimbal1.jpg', 0),

Результат выполненных запросов на создание таблицы «Оборудование» представлен в соответствии с рисунком 21

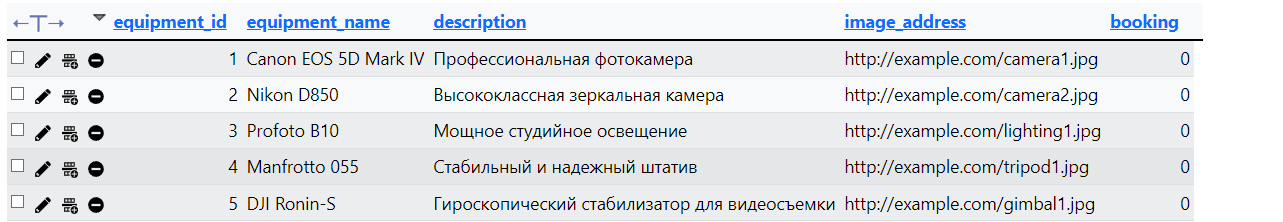


Рисунок 21. Результат выполненных запросов на создание таблицы «Оборудование»

Чтобы создать таблицу «Аренда оборудования», которая в базе данных будет называться , [EquipmentRent](http://127.0.0.1:120/openserver/phpmyadmin/index.php?route=/sql&db=photostudio&table=EquipmentRent&pos=0) нужно создать следующий запрос:

CREATE TABLE `EquipmentRent` (

`rent\_id` int(11) NOT NULL,

`equipment\_id` int(11) DEFAULT NULL,

`employee\_id` int(11) DEFAULT NULL,

`start\_date` date DEFAULT NULL,

`end\_date` date DEFAULT NULL,

`cost` decimal(10,2) DEFAULT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci;

Код запроса на заполнение таблицы «Аренда Оборудования»

INSERT INTO `EquipmentRent` (`rent\_id`, `equipment\_id`, `employee\_id`, `start\_date`, `end\_date`, `cost`) VALUES

(1, 1, 1, '2024-05-01', '2024-05-05', '100.00'),

(2, 2, 2, '2024-05-01', '2024-05-05', '150.00'),

(3, 3, 3, '2024-05-01', '2024-05-05', '200.00'),

(4, 4, 4, '2024-05-01', '2024-05-05', '50.00'),

(5, 5, 5, '2024-05-01', '2024-05-05', '120.00');

Результат выполненных запросов на создание таблицы «Аренда Оборудования» представлен в соответствии с рисунком 22

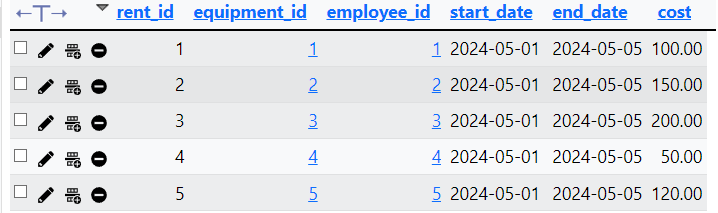


Рисунок 22. Результат выполненных запросов на создание таблицы «Аренда Оборудования»

Чтобы создать таблицу «Роли», которая в базе данных будет называться , Roles нужно создать следующий запрос:

CREATE TABLE `Roles` (

`role\_id` int(11) NOT NULL,

`role\_name` varchar(100) COLLATE utf8mb4\_unicode\_ci NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci;

Код запроса на заполнение таблицы «Роли»

INSERT INTO `Roles` (`role\_id`, `role\_name`) VALUES

(1, 'Администратор'),

(2, 'Фотограф'),

(3, 'Администратор студии'),

(4, 'Продавец-кассир'),

(5, 'Уборщик');

Результат выполненных запросов на создание таблицы «Роли» представлен в соответствии с рисунком 23



Рисунок.23 Результат выполненных запросов на создание таблицы «Роли»

Чтобы создать таблицу «Аренда студии», которая в базе данных будет называться , StudioRent нужно создать следующий запрос:

CREATE TABLE `StudioRent` (

`rent\_id` int(11) NOT NULL,

`studio\_id` int(11) DEFAULT NULL,

`employee\_id` int(11) DEFAULT NULL,

`start\_date` date DEFAULT NULL,

`end\_date` date DEFAULT NULL,

`cost` decimal(10,2) DEFAULT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci;

Код запроса на заполнение таблицы «Аренда студии»

INSERT INTO `StudioRent` (`rent\_id`, `studio\_id`, `employee\_id`, `start\_date`, `end\_date`, `cost`) VALUES

(1, 1, 1, '2024-05-01', '2024-05-05'),

(2, 2, 2, '2024-05-01', '2024-05-05'),

(3, 3, 3, '2024-05-01', '2024-05-05'),

(4, 4, 4, '2024-05-01', '2024-05-05'),

(5, 5, 5, '2024-05-01', '2024-05-05');

Результат выполненных запросов на создание таблицы «Аренда студии» представлен в соответствии с рисунком 24

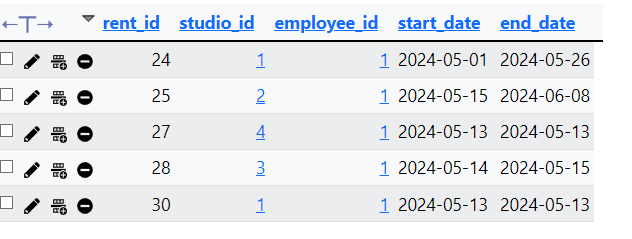


Рисунок 24. Результат выполненных запросов на создание таблицы «Аренда студии»

Чтобы создать таблицу « студии», которая в базе данных будет называться , Studios нужно создать следующий запрос:

CREATE TABLE `Studios` (

`studio\_id` int(11) NOT NULL,

`studio\_name` varchar(255) COLLATE utf8mb4\_unicode\_ci NOT NULL,

`address` varchar(255) COLLATE utf8mb4\_unicode\_ci NOT NULL,

`description` text COLLATE utf8mb4\_unicode\_ci DEFAULT NULL,

`image\_address` text COLLATE utf8mb4\_unicode\_ci DEFAULT NULL,

`booking` tinyint(1) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci;

Код запроса на заполнение таблицы «студии»

INSERT INTO `Studios` (`studio\_id`, `studio\_name`, `address`, `description`, `image\_address`, `booking`) VALUES

(1, 'Studio One', 'Улица Пушкина, дом Колотушкина', 'Лучшая фотостудия в городе', 'http://example.com/studio1.jpg', 0),

(2, 'Art Studio', 'Проспект Ленина, дом 123', 'Уютная и творческая атмосфера', 'http://example.com/studio2.jpg', 0),

(3, 'PhotoLab', 'ул. Гагарина, дом 15', 'Профессиональное оборудование и высококлассные услуги', 'http://example.com/studio3.jpg', 0),

(4, 'Fashion Studio', 'пр-т Победы, дом 20', 'Модная фотостудия для съемок рекламы и моделей', 'http://example.com/studio4.jpg', 0),

(5, 'Golden Hour', 'ул. Советская, дом 30', 'Современная студия для креативных съемок', 'http://example.com/studio5.jpg', 0);

Результат выполненных запросов на создание таблицы «студии» представлен в соответствии с рисунком 25

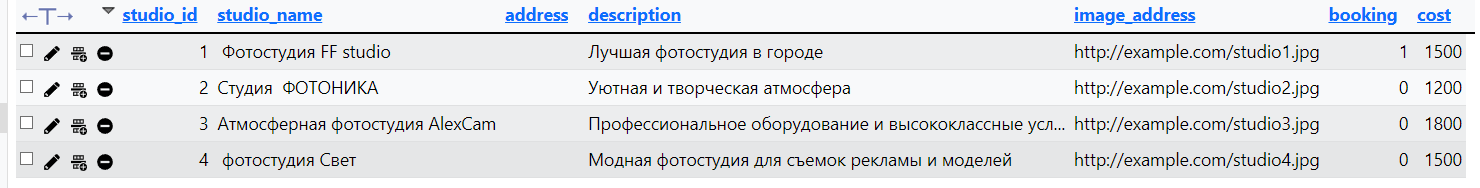


Рисунок .25 Результат выполненных запросов на создание таблицы «студии»

Реляционная диаграмма базы данных «photostudio» представлена в соответствии с рисунком 26.

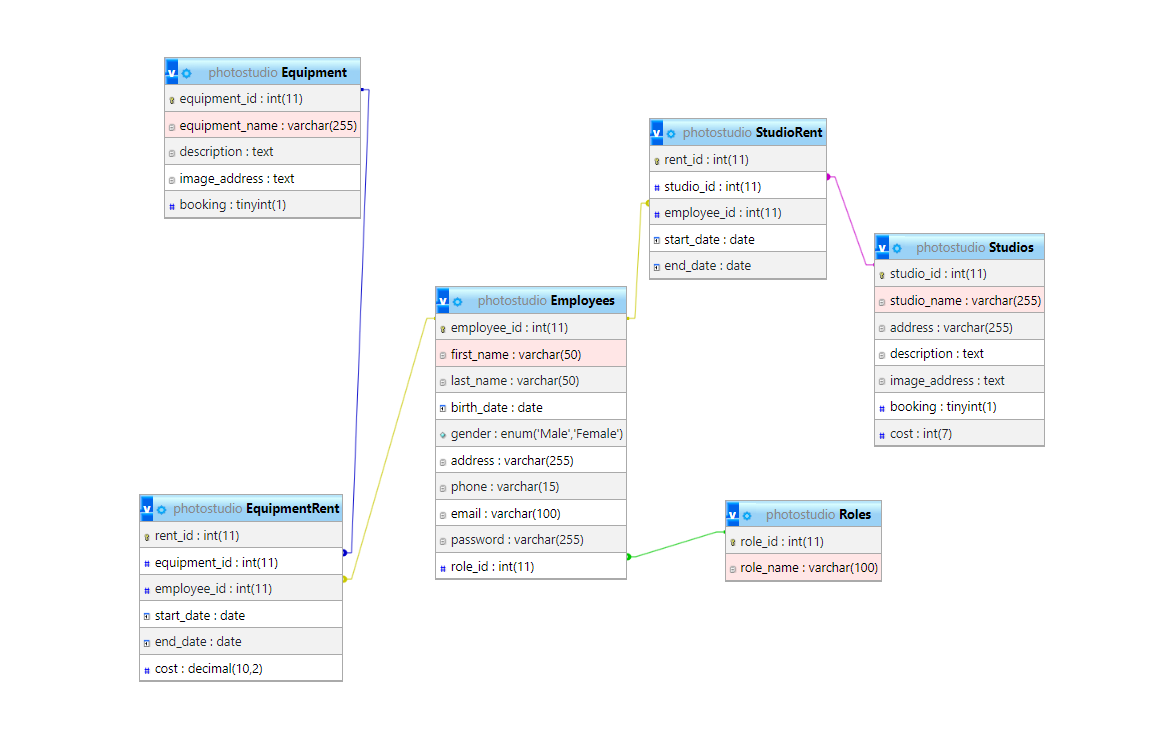


Рисунок 26. Реляционная диаграмма базы данных «photostudio»

# 2.4 Защита информации в базе данных

# 2.5 Проектирование системы

Предметной областью, изучаемой в ходе написания проекта для преддипломной работы, является Фотостудия.

Сайт создаётся для автоматизации работы Фотостудии «Под прицелом».

Данное приложение должно выполнять следующие задачи:

1. Добавлять строки в таблицы БД;
2. Реализовано окно авторизации.
3. Добавление записи на аренду
4. Исправно запускаться и функционировать на ПК с ОС «Windows» удовлетворяющем системным требованиям.

Контекстная диаграмма автоматизированной системы (далее АС) представлена в соответствии с рисунком 8.

Входными данными являются:

* Пользователи
* Выбор услуги
* Выбор оборудования
* Бронирование фотостудии

Выходными данными являются:

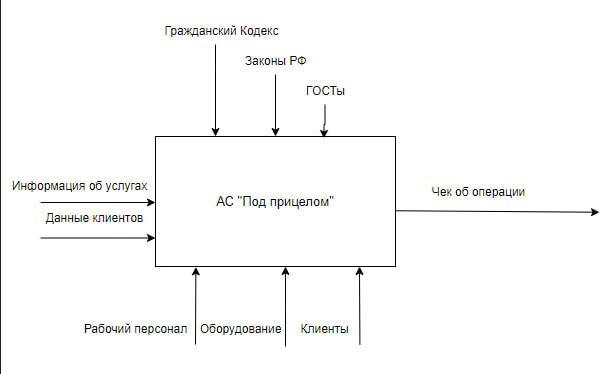
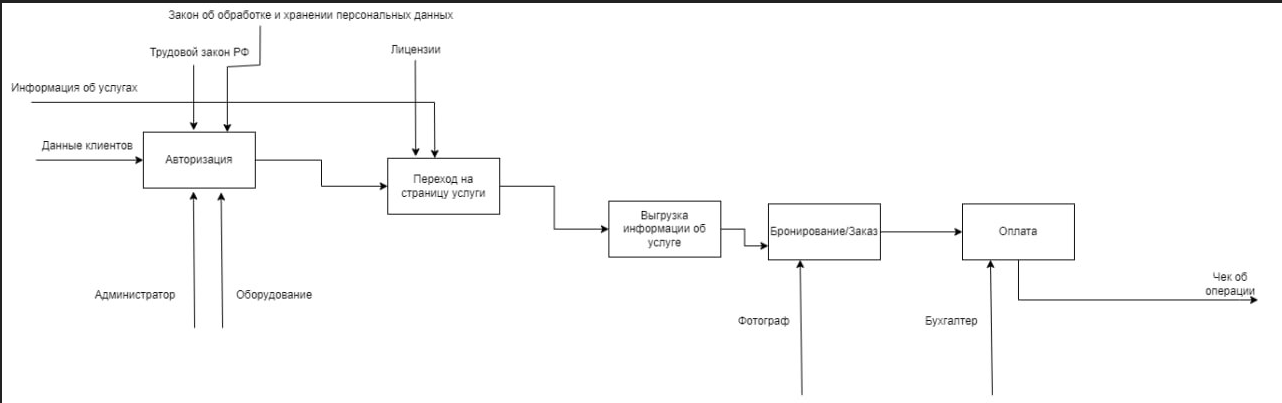
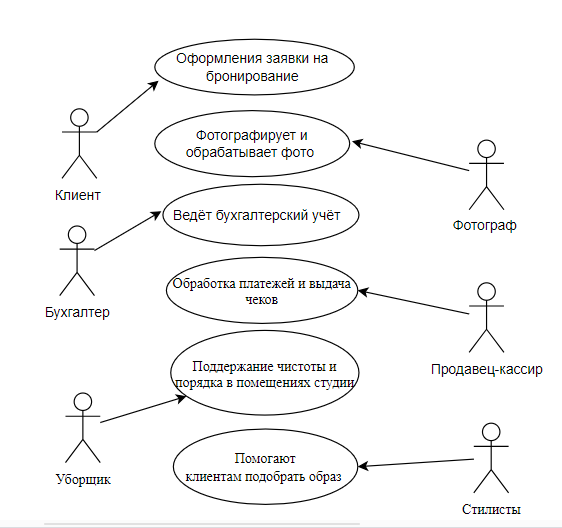
* Уведомление о бронирование
* Рисунок 8. Контекстная диаграмма АС
* Диаграмма декомпозиций контекстной диаграммы АС представлена в соответствии с рисунком 9.
* 

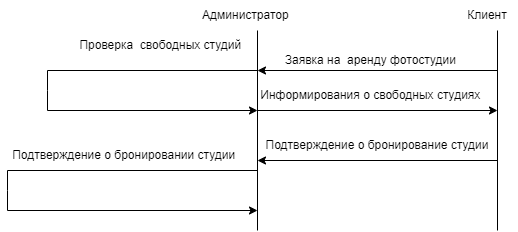
Рисунок 8 и 9 не стыкуются Рисунок 9. Диаграмма декомпозиций АС

В соответствии с рисунком 10 представлена диаграмма вариантов ис.



Почему эти люди не взаимодействуют никак Рисунок 10. Диаграмма вариантов использования АС

Диаграмма последовательностей, представленная в соответствии с рисунком 11. Данная диаграмма используется для более детального описания логики сценариев использования.



Где линейка времени и загнутая стрелка дугой Рисунок 11. Диаграмма последовательностей АС

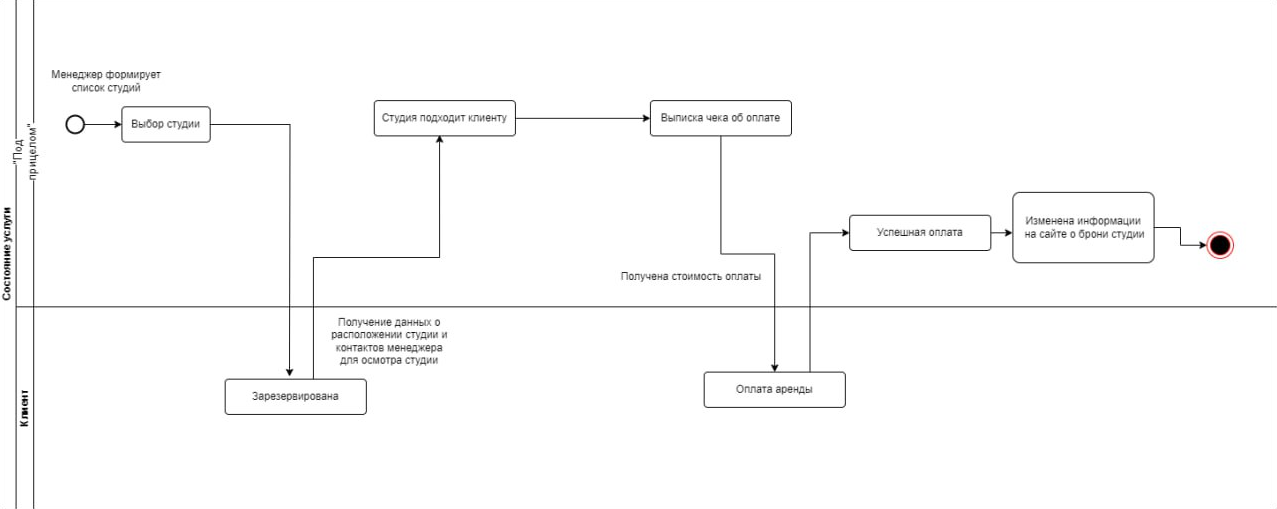
Диаграмма состояний, представленная в соответствии с рисунком 12 используется для описания поведения АС. 

Рисунок 12 Диаграмма состояний АС

Диаграмма классов демонстрирует общую структуру иерархии [классов](https://ru.wikipedia.org/wiki/%D0%9A%D0%BB%D0%B0%D1%81%D1%81_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5)) системы, их коопераций, атрибутов (полей), [методов](https://ru.wikipedia.org/wiki/%D0%9C%D0%B5%D1%82%D0%BE%D0%B4_(%D1%8F%D0%B7%D1%8B%D0%BA%D0%B8_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F)), интерфейсов и взаимосвязей (отношений) между ними. Диаграмма представлена в соответствии с рисунком 13.

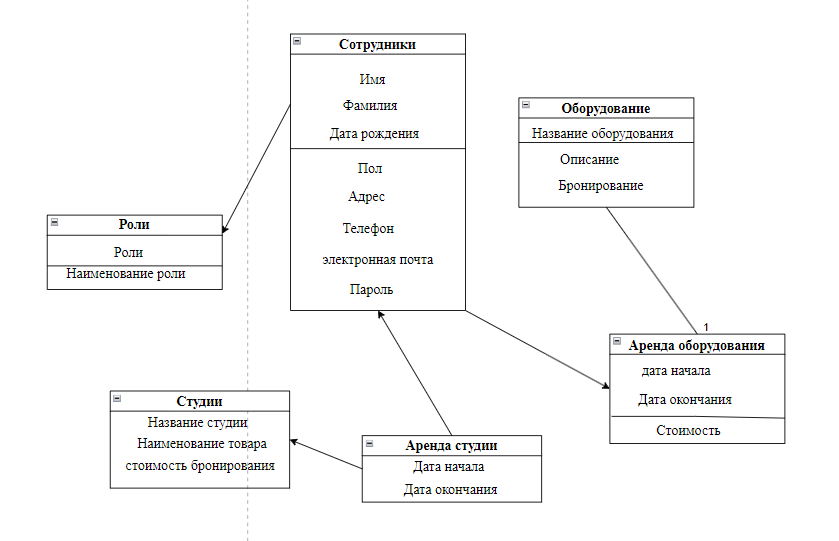


Рисунок 13. Диаграмма классов АС

# 2.6 Разработка программных модулей

# 2.7 Интеграция программных модулей

# 2.8 Тестирование и отладка

Для обеспечения высокого уровня функциональности программного продукта необходимо провести тестирование с выбором оптимальной методики. Учитывая отсутствие сложной функциональности обработки данных и ориентацию приложения на бизнес, основное внимание уделено приемочному тестированию. Этот вид тестирования включает запуск всего приложения и воспроизведение действий пользователя для проверки качества функционирования системы в целом. В таблицах 1-4 приведены некоторые примеры тест-кейсов.

Таблица 1.

Тестовый пример 1

|  |  |
| --- | --- |
| Тестовый пример | Тест-1 |
| Приоритет тестирования | Высокий |
| Заголовок/название теста | Запуск веб-приложения |
| Краткое изложение теста | Запуск и проверка инициализации веб приложения |
| Этапы теста | 1. Открыть браузер.  2. Ввести в адресную строку file:///D:/ospane/domains/index.html |
| Тестовые данные | Адрес  file:///D:/ospane/domains/index.html |
| Ожидаемый результат | Открытие и инициализация веб приложения, отображение она авторизации. |
| Статус | Успех |
| Предварительное условие | Отсутствует |
| Постусловие | Отсутствует |
| Примечания/комментарии | Отсутствуют |

Таблица 2.

Тестовый пример 2

|  |  |
| --- | --- |
| Тестовый пример | Тест-2 |
| Приоритет тестирования | Высокий |
| Заголовок/название теста | Проверка элементов авторизации |
| Краткое изложение теста | Проверка функциональности элементов авторизации |
| Этапы теста | Нажать кнопку зарегистрироваться |
| Тестовые данные | Активация элементов регистрации |
| Ожидаемый результат | Добавление записи в бд |
| Статус | Успех |
| Предварительное условие | Нахождение на странице авторизации. |
| Постусловие | Отсутствует |
| Примечания/комментарии | Отсутствуют |

Таблица 3.

Тестовый пример 3

|  |  |
| --- | --- |
| Тестовый пример | Тест-2 |
| Приоритет тестирования | Высокий |
| Заголовок/название теста | Проверка элементов межстраничной навигации |
| Краткое изложение теста | Проверка функциональности элементов навигации между страницами сайта. |
| Этапы теста | Нажать кнопку на главной странице |
| Тестовые данные | Активация элементов межстраничной навигации |
| Ожидаемый результат | Перемещение на выбранную страницу сайта. |
| Статус | Успех |
| Предварительное условие | Нахождение на главной странице веб приложения. |
| Постусловие | Отсутствует |
| Примечания/комментарии | Отсутствуют |

Таблица 4.

Тестовый пример 4

|  |  |
| --- | --- |
| Тестовый пример | Тест-4 |
| Приоритет тестирования | Высокий |
| Заголовок/название теста | Бронирование студии |
| Краткое изложение теста | Заполнение формы тестовыми данными. |
| Этапы теста | 1. Перейти на страницу «Бронь».  2. Выбрать подходящую студию и заполнить поля бронирования  «Дата начала:»- «Дата окончания:»  3. Нажать кнопку “Забронировать”. |
| Тестовые данные |  |
| Ожидаемый результат | Открытие всплывающего окна с сообщением об успешной записи на услугу. |
| Статус | Провал сайт выдаёт ошибку 403 |
| Предварительное условие | Нахождение на четвертой странице веб-приложения |
| Постусловие | Отсутствует |
| 0,Примечания/комментарии | Отсутствует |

Тестов мало оформи больше тестов ввиде сводной таблицы Таблица 8 – Результаты тестирования системы с указанием ожидаемых и фактических результатов

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| № | Название | Шаги | Тестовые данные | Ожидаемый результат | Фактический результат | Итог |
| 1 | Проверка авторизации пользователя | 1. Запустить приложение 2. Ввести данные 3. Нажать кнопку авторизации | GalinaML  1234 | Успех | Успех | Пройден |

# 2.9 Рефакторинг программного кода

В данной главе будут описаны исправления ошибки из главы 1.4 тест-кейса

Текст ошибки представлен в соответствии с рисунком 7.

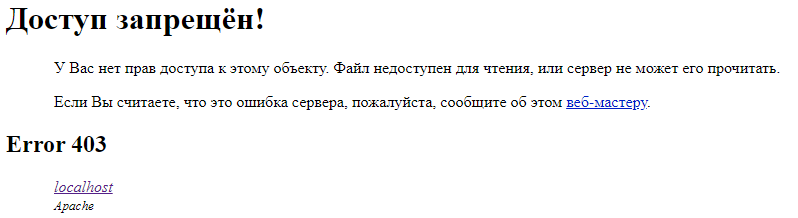


Рисунок 7. Ошибка, при которой не функционирует кнопка боронование

Исходный код в с ошибкой:

// Проверяем, забронирована ли студия

                if ($row["booking"] == 0) {

                    // Выводим кнопку "Забронировать" только если студия доступна для бронирования

                    echo "<a href='.php?studio\_id=" . $row["studio\_id"] . "' class='btn-book'>Забронировать</a>";

                }

Исправленный код:

 // Проверяем, забронирована ли студия

                if ($row["booking"] == 0) {

                    // Выводим кнопку "Забронировать" только если студия доступна для бронирования

                    echo "<a href='bron1.php?studio\_id=" . $row["studio\_id"] . "' class='btn-book'>Забронировать</a>";

                }

                echo "</li>";

# 2.10 Соответствие кода стандартам кодирования

При написании сайта был применен стиль написания кода Buble на JS. Вот основные принципы этого стандарта:

* Выравнивание. Для выравнивания блоков кода используются пробелы (2 пробела на 1 уровень выравнивания), знаки табуляции не используются.
* Длина строки. 80 символов (если это возможно).
* Однострочные комментарии. Такие комментарии используются в коде.
* Многострочные комментарии. Эти комментарии используются для документирования кода.
* Объявление переменных. Переменные всегда объявляются в явном виде для предотвращения загрязнения глобального объекта. Ключевое слово var не используется. Если значение переменной в ходе выполнения программы менять не планируется, её объявляют в виде константы (подобные константы нередко тоже называют «переменными») с помощью ключевого слова const, используя его по умолчанию — кроме тех случаев, когда менять значение переменной планируется. В таких случаях используется ключевое слово let.
* Константы. Если некие значения в программе являются константами, их имена составляют из прописных букв. Например — CAPS. Для разделения частей имён, состоящих из нескольких слов, используется знак подчёркивания (\_).

Вот пример кода:

var swiper = new Swiper(".swiper-container-wrapper .swiper-container", {

preloadImages: false,

autoplay: {

delay: 4000,

disableOnInteraction: false

},

init: false,

loop: false,

speed: 1200,

grabCursor: true,

mousewheel: {

enable: true

},

keyboard: true,

simulateTouch: true,

parallax: true,

effect: "slide",

pagination: {

el: ".swiper-slide-pagination",

clickable: true

},

navigation: {

nextEl: ".slide-next",

prevEl: ".slide-prev"

},

scrollbar: {

el: ".swiper-scrollbar",

draggable: true

}

});