

Министерство науки и высшего образования Российской Федерации  
ФГБОУ ВО «Кубанский государственный технологический университет»  
(ФГБОУ ВО КубГТУ)

Институт Компьютерных систем и информационной безопасности  
Кафедра Информационных систем и программирования  
Направление подготовки 09.03.03 Прикладная информатика  
Профиль Разработка и внедрение информационных систем

**КУРСОВОЙ ПРОЕКТ**

по дисциплине Разработка приложений баз данных  
(наименование дисциплины)

на тему Информационная система «Фотоцентр»  
(тема курсового проекта)

Выполнил студент Мальцев Михаил Сергеевич курса 3 группы 22-КБ-ПИ1  
(ф.и.о.)

Допущен к защите 11.12.2024

Руководитель (нормоконтролер) работы Вонарх ст. преп. Вонарх Ю.С.

Защищен 11.12.2024 Оценка отм.  
(дата)

Члены комиссии Шкел ст. преп. Кушнир Н.В.  
В.А. доц. Мурлина В.А.  
(подпись, дата, расшифровка подписи)

Краснодар  
2024

ФГБОУ ВО «Кубанский государственный технологический университет»  
(ФГБОУ ВО КубГТУ)

Институт Компьютерных систем и информационной безопасности

Кафедра Информационных систем и программирования

Направление подготовки 09.03.03 Прикладная информатика

Профиль Разработка и внедрение информационных систем

УТВЕРЖДАЮ

Зав. кафедрой М.В. Янаева.

«10» сентября 2024 г.

**ЗАДАНИЕ**

на курсовой проект

Студенту: Мальцеву Михаилу Сергеевичу группы 22-КБ-ПИ1 курса  
(Ф.И.О.) (№ группы и курса)

Тема проекта: Информационная система «Фотоцентр».

(утверждена указанием директора института № 9 от 06.09.2024 )

План проекта:

1. Изучение предметной области, анализ существующих программных (информационных) решений
2. Проектирование ПО, описание алгоритма
3. Программная реализация предлагаемого решения

Объем работы:

- а) пояснительная записка 51 с.
- б) иллюстративная часть 29 лист (-ов)

Рекомендуемая литература

1. Фельдман Г.Г., Новоселов А.П. Автоматизация процессов на предприятии.
2. Умкин Д.Н. Основы систем Баз Данных
3. Григорьева О.В., Фудин Ч.А. Практикум по курсу Базы данных

Срок выполнения: с «10» 09 по «17» 11 2024г.

Срок защиты: «11» 12 2024г.

Дата выдачи задания: «06» 09 2024г.

Дата сдачи проекта на кафедру: «11» 12 2024г.

Руководитель работы Вонарх Ю.С.  
(должность, подпись)

Задание принял студент Мальцев М.С.  
(подпись)

## Реферат

Курсовая работа: 51 страниц, 29 рисунков, 10 таблиц, 6 источников.

ФОТОЦЕНТР, ИНФОРМАЦИОННАЯ СИСТЕМА, БАЗА ДАННЫХ, ВЕБ-ПРИЛОЖЕНИЕ, АДМИНИСТРАТОР, ФОТОГРАФ, МЕНЕДЖЕР, КЛИЕНТ, ЗАКАЗ УСЛУГ, ОТЗЫВЫ, UI/UX ДИЗАЙН, ДОКУМЕНТИРОВАНИЕ, ОТЧЕТЫ, АВТОМАТИЗАЦИЯ ПРОЦЕССОВ, СИСТЕМА УПРАВЛЕНИЯ, API.

Целью данной курсовой работы является разработка информационной системы для фотоцентра, обеспечивающей автоматизацию процессов управления заказами, клиентами и фотографиями, а также предоставляющей удобный интерфейс для различных пользователей системы (администраторов, менеджеров, фотографов и клиентов). Система должна обеспечивать управление заказами, хранение данных о клиентах, заказах и услугах, а также возможность получения отчетности и отзывов о выполненных услугах.

Основные полученные результаты:

- Разработана информационная система для фотоцентра, обеспечивающая эффективное управление заказами, клиентами, фотографиями и услугами.
- Автоматизация процессов управления заказами, что значительно упрощает взаимодействие между клиентами и фотографиями, а также повышает эффективность работы сотрудников.
- Удобный интерфейс для всех типов пользователей (администраторы, менеджеры, фотографы, клиенты), что обеспечивает удобство в работе с системой и доступ к необходимой информации.
- Эффективное хранение данных в базе данных, что позволяет быстро получать информацию о заказах, услугах и клиентах.

- Разработаны отчеты и аналитика для менеджеров и администраторов, что улучшает процесс мониторинга и оценки качества предоставляемых услуг.
- Реализована возможность получения и управления отзывами клиентов, что способствует улучшению качества обслуживания и повышению доверия со стороны клиентов.
- Модуль для загрузки фотографий и работы с заказами дает фотографам возможность оперативно загружать готовые работы и взаимодействовать с клиентами.

## Содержание

Введение .....	5
1 Нормативные ссылки .....	7
2 Описание предметной области .....	8
2.1 Общее описание предметной области .....	8
2.2 Назначение и цели создания ПО .....	9
2.3 Требования к структуре ПО и базе данных .....	11
3 Реализация базы данных средствами выбранной СУБД .....	14
3.1 Выбор СУБД .....	14
3.2 Построение ER- диаграммы .....	15
4 Проектирование структуры приложения .....	17
4.1 Диаграмма прецедентов .....	17
4.2 Выбор среды программирования .....	18
4.3 Окна приложения .....	19
4.4 Формирование отчетной документации .....	30
5 Тестирование программного продукта .....	34
Заключение .....	46
Список использованных источников .....	48
Приложения .....	49
Приложение А .....	49
Приложение Б .....	50

## **Введение**

В современном мире, где спрос на услуги фотоцентров растет, эффективное управление заказами и взаимодействие с клиентами становятся важными аспектами успешной работы. В связи с этим, для повышения качества обслуживания и улучшения внутренней организации работы фотоцентров, требуется создание системы, способной автоматизировать процессы и упростить работу с клиентами и заказами. Разработанная информационная система для фотоцентра призвана решить эти задачи, обеспечив удобный интерфейс для всех пользователей и эффективное взаимодействие между ними.

Целью работы является разработка информационной системы для фотоцентра, которая позволит автоматизировать процессы управления заказами, фотографиями, клиентами и отзывами, а также повысить качество обслуживания за счет упрощения работы сотрудников и предоставления клиентам удобного инструмента для взаимодействия.

Задачи работы включают:

1. Разработку концепции информационной системы для фотоцентра.
2. Проектирование структуры базы данных, необходимой для хранения информации о заказах, клиентах, фотографах и услугах.
3. Реализацию базы данных с использованием выбранной СУБД.
4. Проектирование структуры приложения с учетом всех необходимых функций для пользователей (администраторы, фотографы, менеджеры, клиенты).
5. Реализацию приложения, включая создание интерфейса и функционала для работы с заказами и отзывами.
6. Проведение тестирования программного продукта.

Данная курсовая работа состоит из нескольких разделов. В первом разделе приведены нормативные ссылки, на основе которых разрабатывается программное обеспечение. Во втором разделе описана предметная область, включая назначение и цели создания ПО, требования к структуре, а также

описание входных и выходных документов. В третьем разделе рассматривается проектирование структуры базы данных, включая описание сущностей и построение концептуальной схемы. Четвертый раздел посвящен реализации базы данных средствами выбранной СУБД. В пятом разделе рассмотрено проектирование структуры приложения, включая различные диаграммы и схемы. Шестой раздел описывает реализацию приложения и его функционала, а также работу с базой данных. В седьмом разделе проводится тестирование программного продукта. Заключение подводит итоги работы, а в приложениях приводятся дополнительные материалы и источники.

## 1 Нормативные ссылки

В данной пояснительной записке использованы ссылки на следующие стандарты:

- ГОСТ 19.201-78 Единая система программной документации (ЕСПД). Техническое задание. Требования к содержанию и оформлению;
- ГОСТ 2.105-95 ЕСКД. Общие требования к текстовым документам;
- ГОСТ 19.201-78 Техническое задание. Требования к содержанию и оформлению;
- ГОСТ 19.102-77 Единая система программной документации. Стадии разработки;
- ГОСТ 19.106-78 Единая система программной документации. Требования к программным документам, выполненным печатным способом;
- ГОСТ Р.7.0.5-2008 СИБИД. Библиографическая ссылка. Общие требования и правила составления.
- ГОСТ 12-93 СИБИД. Библиографическая запись. Сокращение слов на русском языке. Общие требования и правила.



## **2 Описание предметной области**

### **2.1 Общее описание предметной области**

Фотоцентр — это организация, предоставляющая широкий спектр услуг в области фотосъемки, обработки фотографий и других сопутствующих услуг, таких как печать, создание фотокниг, монтаж видео и т. д. Основными пользователями таких сервисов являются как частные лица, так и корпоративные клиенты, которым необходимо заказать фотосъемку для различных мероприятий, таких как свадьбы, корпоративы, юбилеи, фото для документов и т. п.

Современный фотоцентр не ограничивается только процессом съемки, но и предоставляет дополнительные услуги, связанные с хранением и обработкой изображений. Для эффективного управления такими услугами необходима информационная система, которая будет автоматизировать работу сотрудников, а также обеспечивать удобство взаимодействия с клиентами.

Основные задачи, решаемые системой для фотоцентра:

1. Управление заказами — отслеживание всех заказов клиентов, их статусов, сроков выполнения и прочей информации, связанной с заказами.
2. Работа с фотографиями — возможность загрузки, обработки и сохранения фотографий, создание отчетности по выполненным работам.
3. Управление пользователями — создание и управление аккаунтами клиентов, фотографов и администраторов.
4. Обратная связь и отзывы — сбор и обработка отзывов клиентов о предоставленных услугах для повышения качества работы и создания прозрачности для пользователей.
5. Аналитика и отчеты — генерация отчетов по заказам, отзывам, финансовым операциям, что позволяет администратору и менеджерам оценивать эффективность работы фотоцентра.

Предметная область включает в себя следующие ключевые объекты:

- Клиенты — пользователи, которые заказывают услуги фотоцентра. Они могут оставлять отзывы, просматривать свои заказы и фотографии.

- Фотографы — сотрудники, которые выполняют съемку, обрабатывают фотографии и загружают их для клиентов.
- Менеджеры — сотрудники, которые занимаются координацией заказов, контролируют их выполнение, а также управляют отношениями с клиентами.
- Администраторы — пользователи с полным доступом к системе, управляющие пользователями, ролями, а также выполняющие администрирование системы и баз данных.

Таким образом, система должна обеспечивать эффективное управление всеми этапами работы фотоцентра — от оформления заказов до получения отзывов и создания отчетности для анализа работы бизнеса.

## **2.2 Назначение и цели создания ПО**

Назначение разрабатываемого программного обеспечения (ПО) заключается в автоматизации процессов работы фотоцентра, обеспечении удобного и эффективного взаимодействия между всеми участниками системы — клиентами, фотографами, менеджерами и администраторами. Система должна минимизировать трудозатраты, повысить оперативность обработки заказов, улучшить качество обслуживания клиентов и упростить администрирование фотоцентра.

Основные цели создания ПО:

- 1 Автоматизация процессов заказа и обработки услуг: Программное обеспечение должно позволять клиентам легко оформлять заказы, а фотографам — быстро и удобно загружать фотографии и отслеживать выполнение заказов. Это обеспечит сокращение времени на обработку заказов и улучшение качества обслуживания.
- 2 Управление клиентами, заказами и отзывами: ПО должно предоставить функциональность для управления базой клиентов, отслеживания

статусов заказов, а также сбора и обработки отзывов, что поможет повысить удовлетворенность клиентов и улучшить качество предоставляемых услуг.

3 Обеспечение удобства работы для всех категорий пользователей: Система должна обеспечивать удобный интерфейс для клиентов, чтобы они могли легко взаимодействовать с фотоцентром, а также предоставлять фотограм, менеджерам и администраторам инструменты для управления заказами, создания отчетов и анализа работы.

4 Обеспечение гибкости и масштабируемости системы: Программное обеспечение должно быть спроектировано таким образом, чтобы в дальнейшем было легко добавлять новые функциональные возможности (например, расширение списка услуг) или изменять существующие в зависимости от нужд бизнеса.

5 Управление доступом и ролями пользователей: Важно обеспечить разграничение доступа для различных пользователей системы. Администраторы, менеджеры, фотографы и клиенты должны иметь различные уровни доступа к данным и функционалу, что обеспечит безопасность данных и контроль за выполнением задач.

6 Интеграция с другими системами: ПО должно иметь возможность интеграции с внешними сервисами для расширения функционала, например, для обработки платежей или автоматической отправки уведомлений.

7 Формирование отчетности: ПО должно предоставлять функционал для формирования отчетов по заказам, финансам, отзывам и другим показателям работы фотоцентра. Это поможет в принятии управленческих решений и оценке эффективности работы.

Таким образом, целью создания программного обеспечения является повышение эффективности работы фотоцентра через внедрение автоматизированной системы, которая улучшит взаимодействие между клиентами и сотрудниками, обеспечит удобство работы для всех пользователей и позволит более эффективно управлять заказами и предоставляемыми услугами.

## 2.3 Требования к структуре ПО и базе данных

Требования к структуре программного обеспечения:

### 1 Многоуровневая архитектура:

Программное обеспечение должно быть построено по многоуровневой архитектуре (например, клиент-серверной), включающей следующие уровни:

а) Уровень пользовательского интерфейса: предоставляет удобный и интуитивно понятный интерфейс для взаимодействия пользователей с системой.

б) Уровень бизнес-логики: обрабатывает данные, обеспечивает выполнение операций, реализует функционал системы.

с) Уровень данных: отвечает за хранение и управление данными в базе данных.

### 2 Платформенная независимость:

ПО должно быть реализовано так, чтобы его можно было запускать на различных платформах при минимальных изменениях.

### 3 Безопасность и управление доступом:

а) Реализовать аутентификацию и авторизацию пользователей на основе ролей (клиент, фотограф, менеджер, администратор).

б) Обеспечить защиту данных от несанкционированного доступа.

### 4 Масштабируемость:

ПО должно быть разработано с учетом возможного увеличения числа пользователей, заказов и объема данных.

### 5 Интуитивно понятный пользовательский интерфейс:

а) Упрощенная навигация и минималистичный дизайн.

б) Поддержка мобильных и настольных устройств.

Требования к структуре базы данных:

### 1 Нормализация:

Структура базы данных должна быть приведена к третьей нормальной форме (3НФ), чтобы исключить избыточность данных и повысить их целостность.

2 Основные таблицы базы данных:

а) Пользователи: информация о клиентах, фотографах, менеджерах и администраторах.

б) Заказы: данные о заказах, включая статус, дату, клиента и фотографа.

с) Фотографии: данные о загруженных фотографиях, связанных с заказами.

д) Отзывы: информация о отзывах клиентов, включая текст и оценку.

3 Целостность данных:

а) Реализация внешних ключей для обеспечения связей между таблицами (например, связь заказов с пользователями и фотографиями).

б) Использование ограничений целостности (например, проверка уникальности, контроль форматов данных).

4 Производительность:

а) Создание индексов для ускорения выполнения часто используемых запросов.

б) Оптимизация структуры таблиц для обработки больших объемов данных.

5 Обеспечение отказоустойчивости:

а) Регулярное резервное копирование данных.

б) Реализация механизма восстановления данных после сбоев.

6 Гибкость в работе с данными:

а) Возможность легко добавлять новые таблицы или изменять структуру существующих.

б) Обеспечение поддержки сложных запросов и отчетности.

7 Совместимость с СУБД:

База данных должна быть совместима с выбранной СУБД (например, Microsoft SQL Server), поддерживающей необходимые функции и инструменты для проектирования, хранения и анализа данных.

Эти требования обеспечат надежность, удобство и безопасность работы системы, а также позволят эффективно обрабатывать данные и выполнять основные функции фотоцентра.

## **3 Реализация базы данных средствами выбранной СУБД**

### **3.1 Выбор СУБД**

#### **Обоснование выбора Microsoft SQL Server**

Для реализации информационной системы фотоцентра была выбрана система управления базами данных (СУБД) Microsoft SQL Server (MS SQL). Это решение принято на основе следующих факторов:

#### **1. Надежность и производительность**

MS SQL Server является одной из самых надежных и высокопроизводительных СУБД, способной обрабатывать большие объемы данных с минимальными задержками. Поддерживает высокую степень параллелизма, что особенно важно для систем с многопользовательским доступом.

#### **2. Инструменты разработки и администрирования**

SQL Server Management Studio (SSMS) предоставляет мощный и удобный интерфейс для управления базой данных, написания запросов и анализа данных. Поддержка средств мониторинга и оптимизации запросов позволяет эффективно управлять ресурсами базы данных.

#### **3. Масштабируемость**

MS SQL Server поддерживает масштабируемость как вертикальную (увеличение ресурсов на одном сервере), так и горизонтальную (распределение данных между несколькими серверами), что делает его подходящим для будущего роста фотоцентра.

#### **4. Безопасность**

MS SQL Server обеспечивает высокий уровень защиты данных благодаря встроенным механизмам шифрования, аутентификации и авторизации. Реализована возможность настройки ролевой модели доступа, что идеально подходит для разграничения прав клиентов, фотографов, менеджеров и администраторов.

#### **5. Поддержка сложных запросов и аналитики**

Возможность написания сложных запросов для формирования отчетов и выполнения аналитических задач. Поддержка хранимых процедур и триггеров для реализации бизнес-логики на стороне базы данных.

#### 6. Совместимость и интеграция

MS SQL Server легко интегрируется с популярными средами разработки, включая .NET, что позволяет реализовать надежное клиентское приложение. Совместимость с различными инструментами анализа данных и визуализации, такими как Power BI.

#### 7. Лицензирование и поддержка

Microsoft предлагает различные варианты лицензирования, включая бесплатную версию SQL Server Express, что делает продукт доступным для малого и среднего бизнеса. Широкая документация и активное сообщество пользователей обеспечивают поддержку в процессе разработки и эксплуатации.

#### Обоснование выбора

Выбор MS SQL Server оправдан необходимостью создания высокопроизводительной, безопасной и масштабируемой системы для фотоцентра. Использование MS SQL Server обеспечивает выполнение всех требований к базе данных и позволяет развивать систему в будущем.

### 3.2 Построение ER- диаграммы

На этапе проектирования базы данных была разработана ER-диаграмма (см. рисунок 3.1), отображающая основные сущности системы и их взаимосвязи.

ER-диаграмма позволяет наглядно представить структуру базы данных, включая сущности, атрибуты и связи между ними. Каждая сущность соответствует определенному объекту приложения, а связи отображают их взаимодействие.

При построении диаграммы были определены ключевые атрибуты сущностей и типы связей между ними: «один к одному», «один ко многим» и



«многие ко многим». Это послужило основой для последующего создания таблиц базы данных и реализации логики взаимодействия в приложении.



Рисунок 3.1 – ER-диаграмма

## 4 Проектирование структуры приложения

### 4.1 Диаграмма прецедентов

В рамках проектирования функциональности веб-приложения была разработана диаграмма прецедентов (рисунок 3.2), отражающая основные сценарии взаимодействия пользователей с системой.

Диаграмма прецедентов демонстрирует роли пользователей (актеров) и их действия в системе. Каждому прецеденту соответствует определенная задача или функциональная возможность, которую пользователь может выполнять в приложении.

На диаграмме выделены ключевые взаимодействия, такие как аутентификация, управление данными, выполнение операций и просмотр информации. Построение диаграммы прецедентов позволило определить основные требования к системе и распределить функциональность между различными компонентами приложения.

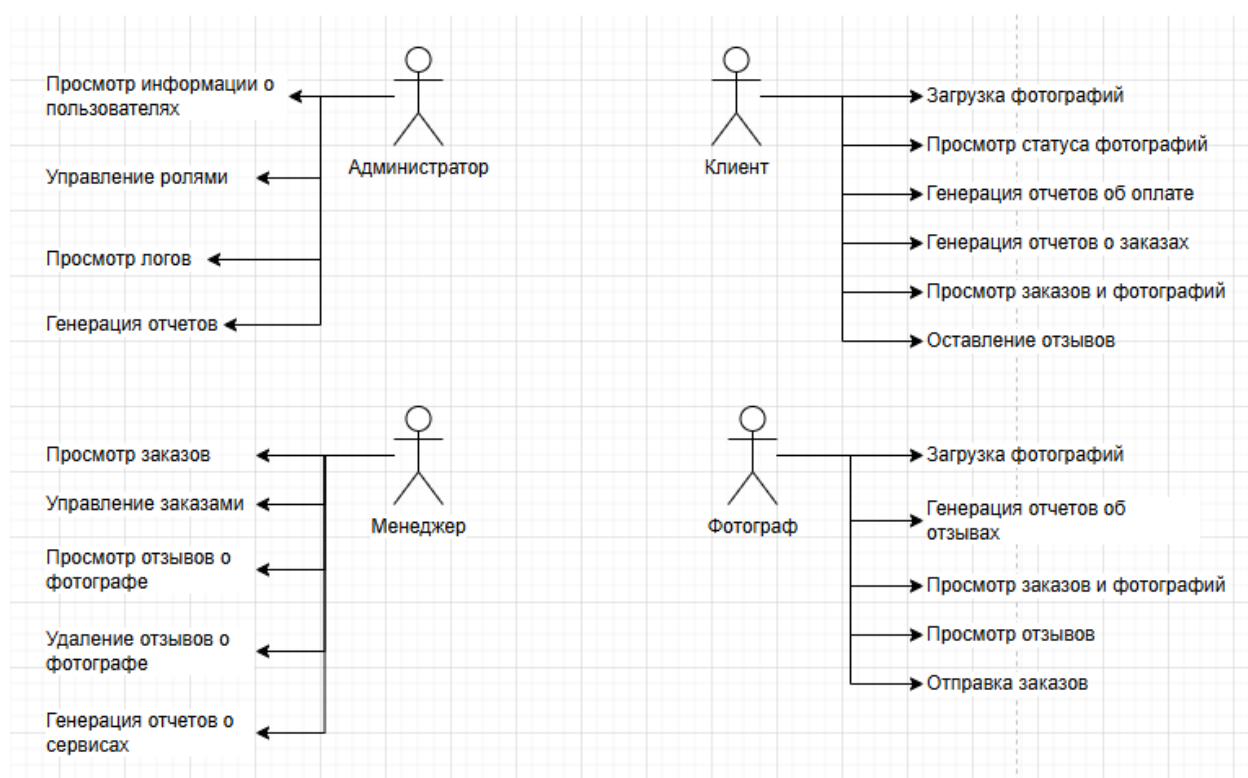


Рисунок 4.2 – Диаграмма прецедентов

## 4.2 Выбор среды программирования

### Выбор среды программирования

Для разработки веб-приложения было решено использовать язык программирования C# и интегрированную среду разработки Visual Studio. Такой выбор обусловлен высокой производительностью, надежностью и широкой функциональностью этих инструментов.

C# — это объектно-ориентированный язык программирования, который предоставляет строгую типизацию, понятный синтаксис и мощные встроенные возможности для работы с данными и взаимодействия с операционной системой. Он идеально подходит для разработки серверной части веб-приложений благодаря интеграции с платформой .NET.

Visual Studio была выбрана как основная интегрированная среда разработки (IDE). Она предоставляет удобный интерфейс, мощный встроенный отладчик и инструменты для автоматизации разработки. Visual Studio поддерживает работу с проектами на C# и Blazor, что позволяет эффективно создавать, тестировать и разворачивать веб-приложения.

Для реализации клиентской части веб-приложения используется Blazor. Это современный фреймворк от Microsoft, который позволяет создавать интерактивные веб-интерфейсы на языке C#. Blazor поддерживает два основных режима работы: Blazor Server и Blazor WebAssembly. Первый вариант выполняет большую часть операций на сервере, что подходит для приложений с высокой степенью безопасности. Второй позволяет запускать приложение на стороне клиента, что обеспечивает более быстрое взаимодействие. Использование Blazor устраняет необходимость писать логику на JavaScript, что упрощает процесс разработки и позволяет работать в едином языковом контексте.

JavaScript будет применяться для реализации отдельных клиентских сценариев, таких как динамическое управление элементами интерфейса или интеграция сторонних библиотек. Это позволит добиться высокой

интерактивности и гибкости приложения, дополняя функционал, предоставляемый Blazor.

Таким образом, выбранный стек технологий, состоящий из C#, Blazor и Visual Studio, в сочетании с возможностями JavaScript для клиентской части, обеспечивает создание производительного, современного и удобного веб-приложения.

### **4.3 Окна приложения**

На этапе разработки интерфейса веб-приложения были спроектированы основные окна, обеспечивающие удобное взаимодействие пользователя с системой.

Каждое окно представляет собой отдельный элемент пользовательского интерфейса, предназначенный для выполнения определенной задачи. Были учтены принципы удобства использования (usability), интуитивности и минимизации количества действий для достижения цели.

Окна приложения включают элементы управления, такие как формы ввода, кнопки, таблицы данных и визуальные индикаторы. Их дизайн был выполнен в соответствии с современными стандартами веб-разработки, что обеспечивает понятный интерфейс и положительный пользовательский опыт.

Ниже представлены окна приложения согласно их связности с определенными ролями и функционалом

# 1. Страницы, связанные с аккаунтом:

В ИС «Фотоцентр» существует система аккаунтов, чтобы войти в аккаунт необходимо ввести корректные имя пользователя и пароль в форму входа (рисунок 4.3).

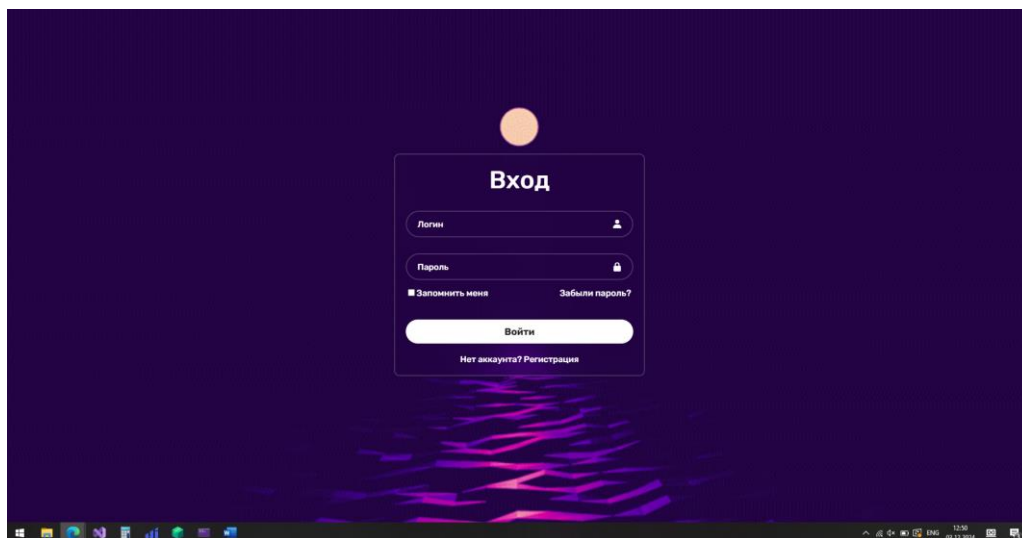


Рисунок 4.3 – Окно входа

Если у вас отсутствует аккаунт, вы можете зарегистрироваться нажав, на кнопку «Регистрация». Здесь, в окне регистрации (рисунок 4.4) необходимо заполнить поля, а после вам будет присвоена роль клиента автоматически.

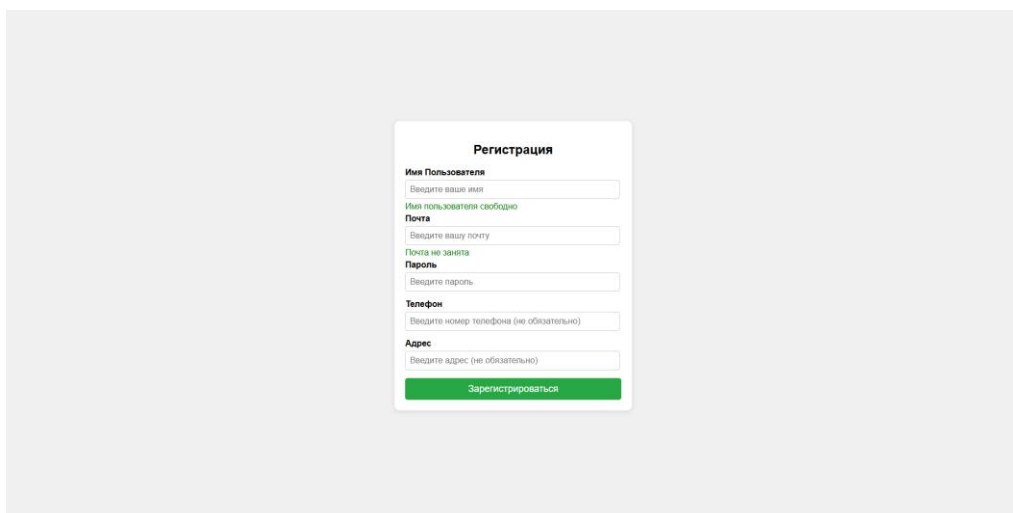


Рисунок 4.4 – Окно регистрации

## 2. Страницы фотографа:

На главной странице (рисунок 4.4) представлена основная информация о функционале сайта и возможностях роли.



Рисунок 4.5 – Главная страница

На странице (рисунок 4.6) отзывов фотограф может посмотреть оставленные клиентами отзывы и получить отчет в виде Word-документа.

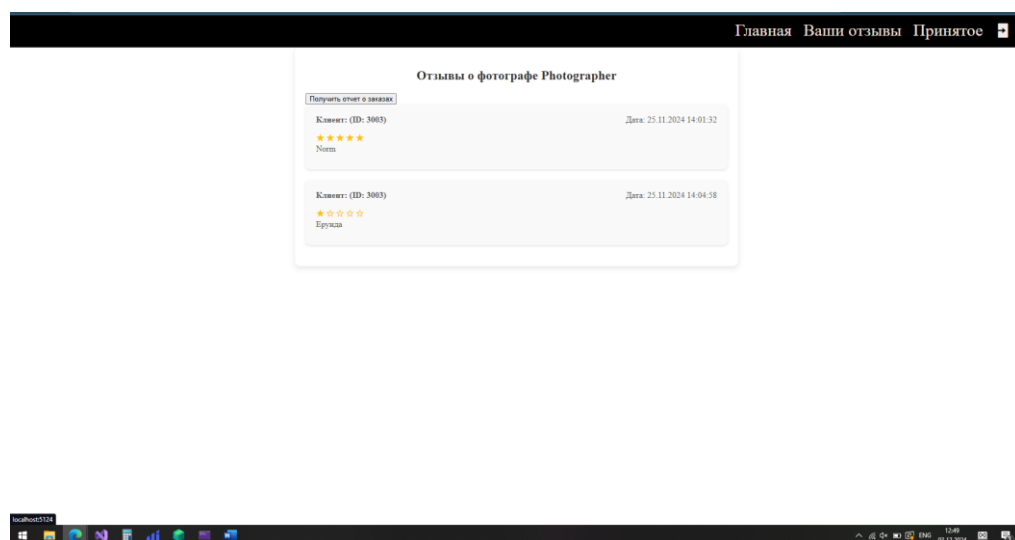


Рисунок 4.6 – Страница отзывов

На странице (рисунок 4.7) принятых заказов фотограф может посмотреть присвоенные заказы в виде таблицы, посмотреть каждый заказ путем

нажатия на кнопку «Посмотреть» и отправить клиенту готовый заказ путем нажатия на кнопку «Отправить». После нажатия на эту кнопку фотографа перекинет на страницу отправки заказа (рисунок 4.8), где ему необходимо загрузить готовое изображения и отправить его клиенту.

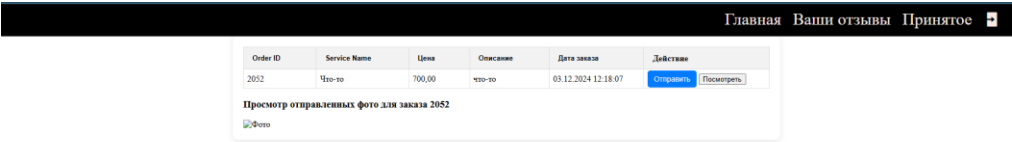


Рисунок 4.7 – Страница принятых заказов

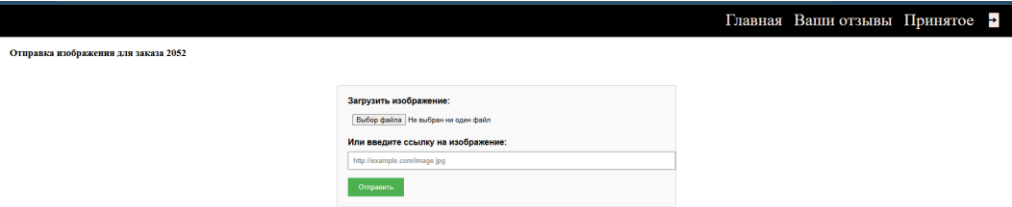


Рисунок 4.8 – Страница отправки заказа

### 3. Страницы клиентов:

На главной странице (рисунок 4.9) представлена основная информация о функционале сайта и возможностях роли.

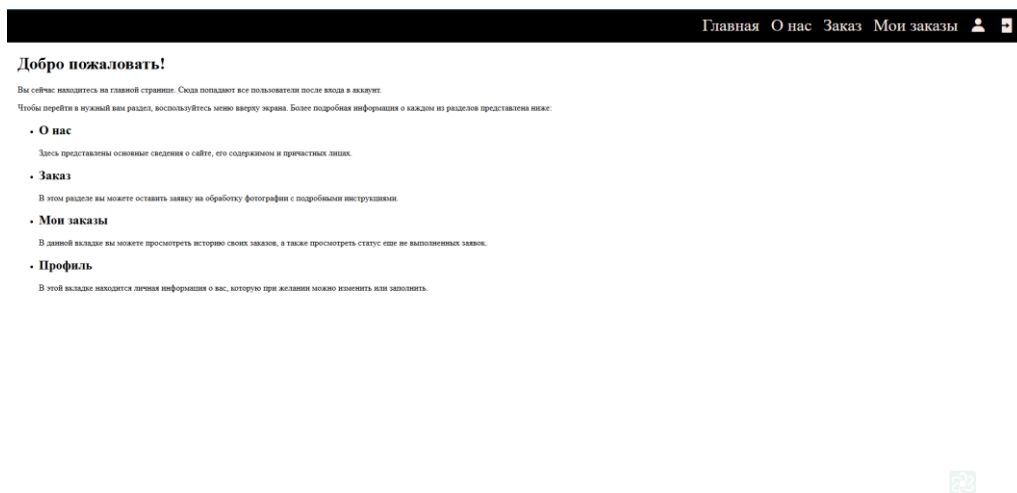


Рисунок 4.9 – Главная страница

На странице «О нас» (рисунок 4.10) представлена основная информация о сервисе и причастных лицах.

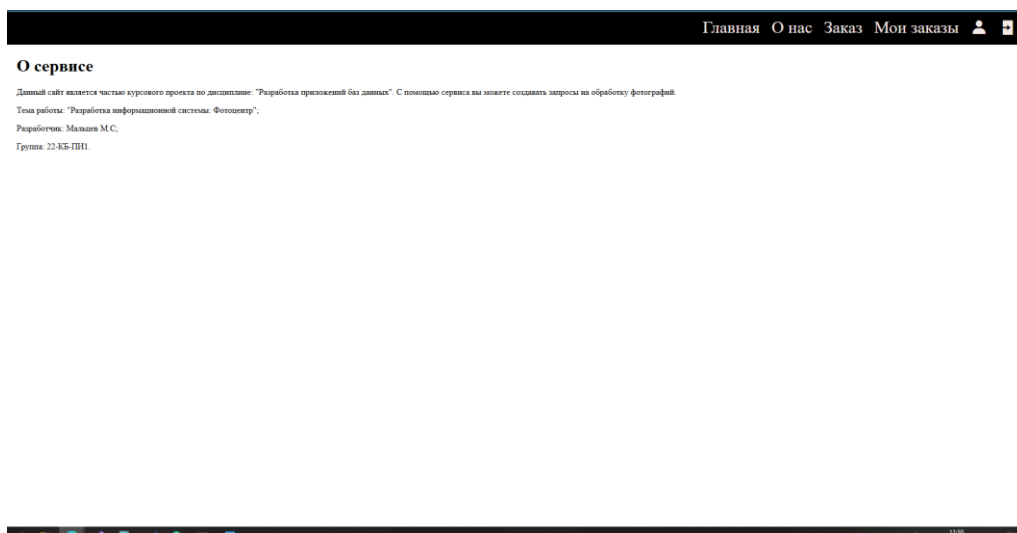


Рисунок 4.10 – Страница «О нас»



На странице нового заказа (рисунок 4.11) клиенту предлагается выбрать нужную услугу, при необходимости загрузить изображение и ввести комментарий для фотографа. После чего пользователь нажимает на кнопку «К оплате» и перейти в окно оплаты заказа (рисунок 4.12).

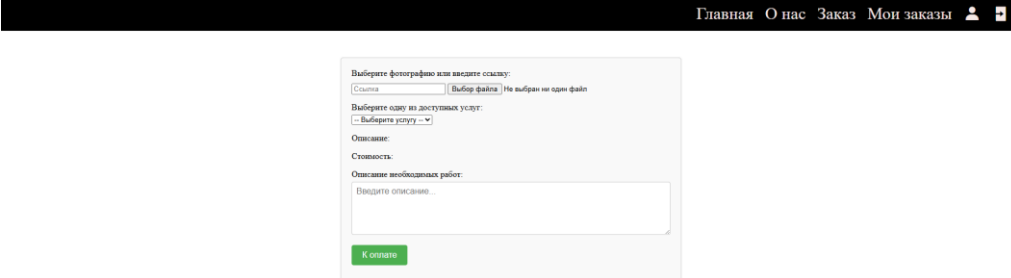


Рисунок 4.11 – Страница нового заказа

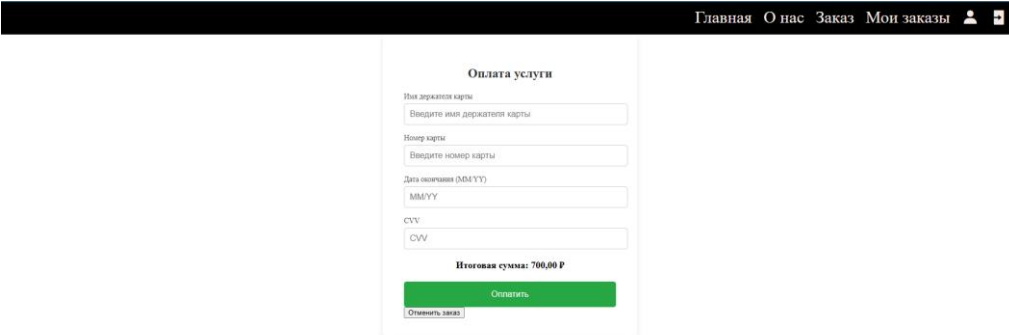


Рисунок 4.12 – Окно оплаты заказа

На странице с историей заказов (рисунок 4.13) клиент может посмотреть в виде таблицы созданные им заказы. Также если заказ готов, то клиент может нажать на кнопку «Готово к просмотру». После нажатия клиента перекинет на

страницу готового заказа (рисунок 4.14), где клиент сможет просмотреть отправленное фотографом изображение и оставить отзыв.

Главная О нас Заказ Мои заказы

Информация о заказах

9 items


Номер заказа	Фотограф	Сервис	Дата	Стоимость	Статус
49	Photographer	Фотосоп	2024-11-19 09:11:36	350,00	<a href="#">Готово к просмотру</a>
50	Photographer	Фотосоп	2024-11-19 09:21:12	350,00	<a href="#">Готово к просмотру</a>
51	Photographer	Фотосоп	2024-11-19 09:22:56	350,00	<a href="#">Готово к просмотру</a>
52	Photographer	Фотосоп	2024-11-19 09:31:25	350,00	<a href="#">Готово к просмотру</a>
1052	Photographer	Фотосоп	2024-11-25 02:00:55	350,00	<a href="#">Готово к просмотру</a>
2052	Не назначен	Что-то	2024-12-03 12:18:07	700,00	<a href="#">Ждем присланного фотографа</a>
2053	Не назначен	Фотосоп	2024-12-03 12:18:24	350,00	<a href="#">Ждем присланного фотографа</a>
2054	Не назначен	Что-то еще	2024-12-03 12:18:35	770,00	<a href="#">Ждем присланного фотографа</a>
2055	Не назначен	Что-то	2024-12-03 12:50:54	700,00	<a href="#">Ожидает оплаты</a>

Получить отчет о заказах

Рисунок 4.13 – Страница с историей заказов

Главная О нас Заказ Мои заказы

Готовый заказ



Фотограф: Photographer

Ваш отзыв:

★★★★★

Рисунок 4.14 – Страница готового заказа

Страница пользователя (рисунок 4.15) предоставляет пользователю основную информацию о нем, т.е. ту, которую он ввел при регистрации. При желании пользователь может отредактировать данные о себе.

Главная О нас Заказ Мои заказы

Профиль пользователя

Номер клиента:

3003

Имя:

Client

Почта:

client@mail.ru

Телефон:

89183606226

Адрес:

address

Дата регистрации:

19.11.2024 20:56:30

Обновить

Создать новый профиль

Получить код

Рисунок 4.15 – Страница профиля

#### 4. Страницы администратора:

На главной странице (рисунок 4.16) представлена основная информация о функционале сайта и возможностях роли.

Главная Журнал Пользователи Сотрудники

Добро пожаловать!

Добро пожаловать, администратор! Здесь вы можете управлять пользователями и настройками сайта.

• Управление пользователями

Добавление, удаление и редактирование пользователей.

• Настройки

Настройка системы и управление правами доступа.

Рисунок 4.16 – Главная страница

На странице с логами (рисунок 4.17) администратору представлены зарегистрированные изменения в базе данных, внесенные пользователями сервиса.

Главная Журнал Пользователи Сотрудники

ID лог	ID пользователя	Действие	Время	Детали	Таблица	Серьезность
273 itemsPage 1 of 14						
Log_M	User_M	Action_Type	Action_Timestamp	Details	Table_Name	Severity_Level
13126	3002	Регистрация	2024-11-19 08:55:45	Имя Admin, Почта: admin@mail.ru, Телефон: , Адрес:	Clients	info
13127	3003	Регистрация	2024-11-19 08:56:30	Имя Client, Почта: client@mail.ru, Телефон: , Адрес:	Clients	info
13128	3004	Регистрация	2024-11-19 08:56:51	Имя Manager, Почта: manager@mail.ru, Телефон: , Адрес:	Clients	info
13129	3005	Регистрация	2024-11-19 08:57:13	Имя Photographer, Почта: photographer@mail.ru, Телефон: , Адрес:	Clients	info
13130	3002	Вход	2024-11-19 08:57:40	Имя Admin	Users	info
13131	3004	Изменение записи	2024-11-19 08:57:50	Клиент: 3004, Изменения: False	Clients	info
13132	3002	Изменение роли	2024-11-19 08:57:50	Пользователь: 3004, Новая роль: manager, Старая роль: client	Users	info
13133	3005	Изменение записи	2024-11-19 08:57:52	Клиент: 3005, Изменения: False	Clients	info
13134	3003	Вход	2024-11-19 08:58:24	Имя Client	Users	info
13135	3004	Вход	2024-11-19 08:58:36	Имя Manager	Users	info
13137	3004	Добавление сервиса	2024-11-19 09:02:49	Новый сервис: , Цена: ,	Services	info

Скачать отчет по логам

Рисунок 4.17 – Окно с логами

На странице с пользователями (рисунок 4.18) представлена основная информация о пользователях, также у администратора имеется возможность изменения роли пользователя путем выбора из выпадающего списка. Можно просмотреть информацию о клиенте нажав на Id клиента. Она выведется в окне информации о пользователе (рисунок 4.19).

Главная Журнал Пользователи Сотрудники

ID	Имя	Роль	Дата регистрации		
5 itemsPage 1 of 1					
User_M	User ID	Username	Role	Date_Registered	User ID
3002	3002	Admin	admin	2024-11-19 08:55:44	admin
3003	3003	Client	client	2024-11-19 08:56:30	client
3004	3004	Manager	manager	2024-11-19 08:56:51	manager
3005	3005	Photographer	photographer	2024-11-19 08:57:13	photographer
3006	3006	Manams	client	2024-11-20 08:09:27	client

Рисунок 4.18 – Страница с пользователями

Главная Журнал Пользователи Сотрудники

Профиль пользователя

Номер клиента:  
3003

Имя:  
Client

Почта:  
client@mail.ru

Телефон:  
89183606226

Адрес:  
address

Дата регистрации:  
19.11.2024 20:56:38

Рисунок 4.19 – Окно с информацией о пользователях

На странице с информацией о сотрудниках (рисунок 4.20) представлена основная информация о фотографах. Кнопка «Просмотреть отзывы» переносит администратора на страницу с отзывами (рисунок 4.6).

Главная Журнал Пользователи Сотрудники

ID

Plus

User_Id	Username	Rating	Orders_In_Progress	Orders_Complete	Просмотр
3003	Photographer	5,2	-1	7	<a href="#">Посмотреть отзывы</a>

Рисунок 4.20 – Окно с информацией о сотрудниках

## 5. Страницы менеджера:

На главной странице (рисунок 4.21) представлена основная информация о функционале сайта и возможностях роли.

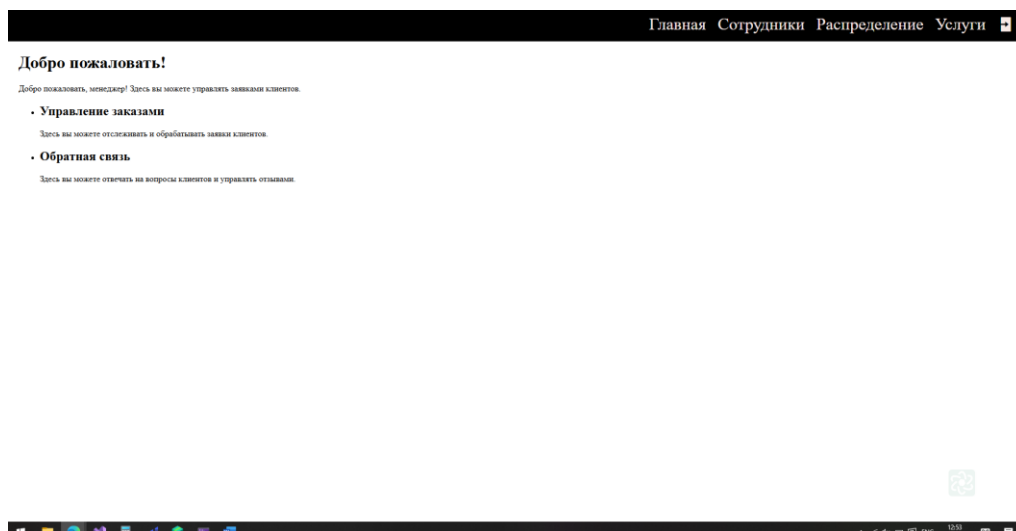


Рисунок 4.21 – Главная страница

На странице распределения заказов (рисунок 4.22) менеджерам предлагается распределить заказы клиентов между фотографами путем выбора соответствующих заказов и фотографов в выпадающих меню и нажатия на кнопку «Назначить фотографа».

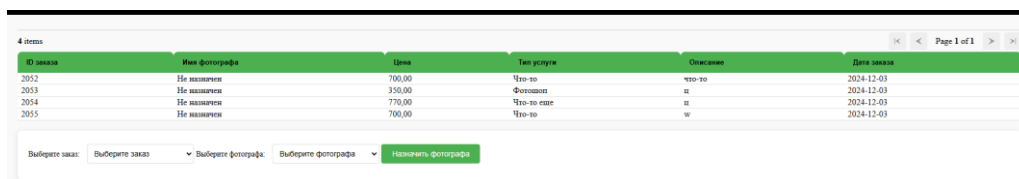


Рисунок 4.22 – Окно распределения заказов

На странице сервисов (рисунок 4.23) у менеджера имеются возможности редактирования, удаления и добавления сервисов.

Название сервиса:  Цена:

3 items Page 1 of 1

Название	Цена	Описание	Действия
Фотошоп	350,00	Изменим ваши фотографии по вашему желанию	Удалить Изменить
Что-то	700,00	Делаем что-то	Удалить Изменить
Что-то еще	770,00	Делаем что-то еще	Удалить Изменить

**Добавить новый сервис**

Название сервиса:

Цена:

Описание:

Рисунок 4.23 – Окно сервисов

#### 4.4 Формирование отчетной документации

Для обеспечения необходимой отчетности в процессе работы с приложением был разработан механизм формирования отчетной документации.

Отчетная документация включает в себя различные виды отчетов, которые предоставляют пользователю информацию о выполненных действиях, статистику, результаты операций и другие важные данные. Эти отчеты могут быть представлены в различных форматах, таких как PDF, Excel или HTML, в зависимости от требований пользователя.

Процесс формирования отчетов был автоматизирован, что позволяет легко генерировать и экспортировать документацию по запросу. Такая система отчетности помогает пользователям анализировать данные, принимать обоснованные решения и поддерживать прозрачность работы приложения.

Результат формирования отчетов представлен на рисунках 4.24 – 4.28.

### Отчет по заказам клиента

Имя клиента: 3003

Почта клиента: client@mail.ru

Телефон: 89183606226

Адрес: adress

ID Заказа	Дата заказа	Сумма	Описание
49	19.11.2024	350,00	Добавьте черный круг в центре
50	19.11.2024	350,00	я напишу плохой отзыв
51	19.11.2024	350,00	<u>test</u>
52	19.11.2024	350,00	w
1052	25.11.2024	350,00	<u>Da</u>

Рисунок 4.24 – Отчет по заказам клиента

### Отчет по платежам клиента

Имя клиента: 3003

Почта клиента: client@mail.ru

Телефон: 89183606226

Адрес: adress

ID Платежа	Дата Платежа	Сумма Платежа
4	19.11.2024	350,00
5	19.11.2024	350,00
6	19.11.2024	350,00
7	19.11.2024	350,00
1007	25.11.2024	350,00

Рисунок 4.25 – Отчет по платежам



## Отчет по логам действий

ID Лога	ID Пользователя	Тип действия	Время действия	Детали	Таблица	Уровень важности
13126	3002	Регистрация	19.11.2024 20:55:45	Имя: Admin, Почта: admin@mail.ru, Телефон: Адрес:	Clients	info
13127	3003	Регистрация	19.11.2024 20:56:30	Имя: Client, Почта: client@mail.ru, Телефон: Адрес:	Clients	info
13128	3004	Регистрация	19.11.2024 20:56:51	Имя: Manager, Почта: manager@mail.ru, Телефон: Адрес:	Clients	info
13129	3005	Регистрация	19.11.2024 20:57:13	Имя: Photographer, Почта: photographer@mail.ru, Телефон: Адрес:	Clients	info
13130	3002	Вход	19.11.2024 20:57:40	Имя: Admin	Users	info
13131	3004	Изменение записи	19.11.2024 20:57:50	Клиент: 3004, Изменение: False	Clients	info
13132	3002	Изменение роли	19.11.2024 20:57:50	Пользователь: 3004, Новая роль: manager, Старая роль: client	Users	info
13133	3005	Изменение записи	19.11.2024 20:57:52	Клиент: 3005, Изменение: False	Clients	info
13134	3003	Вход	19.11.2024 20:58:24	Имя: Client	Users	info
13135	3004	Вход	19.11.2024 20:58:36	Имя: Manager	Users	info

Рисунок 4.26 – Отчет по действиям пользователей

## Отчет по отзывам фотографа

ID Отзывы	ID Клиента	ID Фотографа	Текст отзыва	Оценка	Дата отзыва	ID Заказа
1004	3003	3005	Norm	5	25.11.2024 14:01:32	49
1005	3003	3005	Ерунда	1	25.11.2024 14:04:58	1052

Рисунок 4.27 – Отчет отзывам клиента

## Отчет по услугам

ID Услуги	Название	Цена	Описание
9	Фотошоп	350,00 Р	Изменим ваши фотографии по вашему желанию
11	Что-то	700,00 Р	Делаем что-то
12	Что-то еще	770,00 Р	Делаем что-то еще

Рисунок 4.28 – Отчет по услугам

## 5 Тестирование программного продукта

В процессе разработки приложения было проведено тестирование программного продукта, направленное на проверку функциональности, производительности и безопасности системы.

Тестирование включало несколько этапов, таких как юнит-тестирование, интеграционное тестирование и тестирование пользовательского интерфейса. На каждом этапе проводились проверки отдельных компонентов приложения и их взаимодействия между собой, что позволяло выявить и исправить возможные ошибки до финальной версии продукта.

Кроме того, были использованы различные тестовые данные для проверки корректности работы системы в различных сценариях. Результаты тестирования были зафиксированы в виде отчетов и таблиц, содержащих информацию о выполненных тестах, выявленных ошибках и принятых мерах по их устранению.

Таблицы 5.1 – 5.10, содержат подробную информацию о каждом проведенном тесте, включая описание, ожидаемые результаты и фактические результаты. Также в таблицах отображены статистические данные по времени выполнения тестов и процент успешных тестов.

Тестирование помогло обеспечить высокое качество программного продукта, минимизировать количество ошибок и повысить стабильность системы.

Таблица 5.1 – Тест «ConvertToBase64\_ShouldReturnCorrectString»

Наименование	Описание
Наименование проекта	Фотоцентр
Номер версии	1.0
Имя тестера	Михаил
Даты тестирования	25.11.2024
Test Case #	TC_Converter_1

Продолжение таблицы №5.1

Приоритет тестирования	Высокий
Название тестирования/Имя	ConvertToBase64_ShouldReturnCorrectString
Резюме испытания	Проверить, что метод ConvertToBase64 возвращает строку в правильном формате base64 для фото.
Шаги тестирования	<ol style="list-style-type: none"> <li>1. Создать объект Photo с тестовыми данными (байты и тип контента).</li> <li>2. Вызвать метод ConvertToBase64 с созданным объектом.</li> <li>3. Убедиться, что метод возвращает строку в формате base64.</li> </ol>
Данные тестирования	<ul style="list-style-type: none"> <li>- Байты изображения: {1, 2, 3}</li> <li>- Тип контента: "image/png"</li> </ul>
Ожидаемый результат	Возвращённая строка должна начинаться с data:image/png;base64, и быть корректной.
Фактический результат	Строка верна и начинается с указанного префикса.
Предпосылки	Методы в классе PhotoConverter должны быть доступны для вызова.
Постусловия	Метод корректно обработал данные и вернул строку в формате base64.
Статус (Pass/Fail)	Pass
Комментарии	Нет дополнительных комментариев.

Таблица 5.2 – Тест «ConvertToBase64\_ShouldReturnEmptyString\_WhenImageIsNull»

Наименование	Описание
Наименование проекта	Фотоцентр
Номер версии	1.0
Имя тестера	Михаил
Даты тестирования	25.11.2024

Продолжение таблицы №5.2

Test Case #	TC_Converter_2
Название тестирования/Имя	ConvertToBase64_ShouldReturnEmptyString_WhenImageIsNull
Резюме испытания	Проверить, что метод возвращает пустую строку, если Image равно null.
Шаги тестирования	1. Создать объект Photo с Image = null и валидным Content_Type. 2. Вызвать метод ConvertToBase64. 3. Убедиться, что метод возвращает пустую строку.
Данные тестирования	- Байты изображения: null - Тип контента: «image/png»
Ожидаемый результат	Метод должен вернуть string.Empty.
Фактический результат	Возвращает пустую строку, как ожидается.
Предпосылки	Класс PhotoConverter должен быть доступен для тестирования.
Постусловия	Метод корректно обработал null как входные данные.
Статус (Pass/Fail)	Pass
Комментарии	Тест необходим для проверки обработки потенциальных ошибок.

Таблица 5.3 – Тест « ConvertToBase64\_ShouldReturnEmptyString\_WhenContentTypesIsNull »

Наименование	Описание
Наименование проекта	Фотоцентр
Номер версии	1.0
Имя тестера	Михаил
Даты тестирования	25.11.2024
Test Case #	TC_Converter_3

Продолжение таблицы №5.3

Приоритет тестирования	Средний
Название тестирования/Имя	ConvertToBase64_ShouldReturnEmptyString_WhenContentTypeIsNull
Резюме испытания	Проверить, что метод возвращает пустую строку, если Content_Type равно null.
Шаги тестирования	<ol style="list-style-type: none"> <li>1. Создать объект Photo с валидным Image и Content_Type = null.</li> <li>2. Вызвать метод ConvertToBase64.</li> <li>3. Убедиться, что метод возвращает пустую строку.</li> </ol>
Данные тестирования	<ul style="list-style-type: none"> <li>- Байты изображения: { 1, 2, 3 }</li> <li>- Тип контента: null</li> </ul>
Ожидаемый результат	Метод должен вернуть string.Empty.
Фактический результат	Пустая строка
Предпосылки	Класс PhotoConverter должен быть доступен для тестирования.
Постусловия	Метод корректно обработал null как входные данные.
Статус (Pass/Fail)	Pass
Комментарии	Тест проверяет сценарий с отсутствием типа контента.

Таблица 5.4 – Тест «ConvertToBase64\_ShouldReturnEmptyString\_WhenImageAndContentTypeAreNull»

Наименование	Описание
Наименование проекта	Фотоцентр
Номер версии	1.0
Имя тестера	Михаил
Даты тестирования	25.11.2024
Test Case #	TC_Converter_4

Продолжение таблицы №5.4

Приоритет тестирования	Средний
Название тестирования/Имя	ConvertToBase64_ShouldReturnEmptyString_WhenImageAndContentTypeAreNull
Резюме испытания	Проверить, что метод возвращает пустую строку, если Image и Content_Type равны null.
Шаги тестирования	1. Создать объект Photo с Image = null и Content_Type = null. 2. Вызвать метод ConvertToBase64. 3. Убедиться, что метод возвращает пустую строку.
Данные тестирования	- Байты изображения: null - Тип контента: null
Ожидаемый результат	Метод должен вернуть string.Empty.
Фактический результат	Возвращает пустую строку, как ожидается.
Предпосылки	Класс PhotoConverter должен быть доступен для тестирования.
Постусловия	Метод корректно обработал null как входные данные.
Статус (Pass/Fail)	Pass
Комментарии	Тест проверяет работу метода с полностью некорректными данными.

Таблица 5.5 – Тест «HashPassword\_ShouldReturnCorrectHash»

Наименование	Описание
Наименование проекта	Фотоцентр
Номер версии	1.0
Имя тестера	Михаил
Даты тестирования	25.11.2024

Продолжение таблицы №5.5

Test Case #	TC_PasswordHelper_1
Приоритет тестирования	Высокий
Название тестирования/Имя	HashPassword_ShouldReturnCorrectHash
Резюме испытания	Проверить, что метод HashPassword возвращает корректный хэш фиксированной длины для пароля.
Шаги тестирования	1. Вызвать метод HashPassword с тестовым паролем «TestPassword123». 2. Проверить, что возвращаемое значение не null. 3. Проверить, что длина возвращаемой строки равна 64 символам.
Данные тестирования	Пароль: «TestPassword123»
Ожидаемый результат	Метод возвращает строку длиной 64 символа (SHA-256).
Фактический результат	Соответствует ожидаемому результату.
Предпосылки	Убедитесь, что метод PasswordHelper.HashPassword реализован и доступен для вызова.
Постусловия	Корректный хэш был сгенерирован и прошёл проверки.
Статус (Pass/Fail)	Pass
Комментарии	Тест проверяет корректность работы метода с одним значением.

Таблица 5.6 – Тест «HashPassword\_ShouldReturnSameHashForSamePassword»

Наименование	Описание
Наименование проекта	Фотоцентр
Номер версии	1.0
Имя тестера	Михаил
Даты тестирования	25.11.2024
Test Case #	TC_PasswordHelper_2



Продолжение таблицы №5.6

Приоритет тестирования	Высокий
Название тестирования/Имя	HashPassword_ShouldReturnSameHashForSamePassword
Резюме испытания	Проверить, что метод HashPassword возвращает одинаковые хэши для одного и того же пароля.
Шаги тестирования	1. Вызвать метод HashPassword с тестовым паролем «TestPassword123» дважды. 2. Сравнить два возвращаемых значения.
Данные тестирования	Пароль: «TestPassword123»
Ожидаемый результат	Оба вызова возвращают одинаковую строку.
Фактический результат	Соответствует ожидаемому результату.
Предпосылки	Убедитесь, что метод PasswordHelper.HashPassword реализован и доступен для вызова.
Постусловия	Корректный хэш был сгенерирован и одинаков для идентичных входных данных.
Статус (Pass/Fail)	Pass
Комментарии	Тест проверяет консистентность работы метода для одинакового ввода.

Таблица 5.7 – Тест

«HashPassword\_ShouldReturnDifferentHashForDifferentPasswords»

Наименование	Описание
Наименование проекта	Фотоцентр
Номер версии	1.0
Имя тестера	Михаил
Даты тестирования	25.11.2024

Продолжение таблицы №5.7

Test Case #	TC_PasswordHelper_3
Приоритет тестирования	Высокий
Название тестирования/Имя	HashPassword_ShouldReturnDifferentHashForDifferentPasswords
Резюме испытания	Проверить, что метод HashPassword возвращает разные хэши для разных паролей.
Шаги тестирования	1. Вызвать метод HashPassword с паролями «TestPassword123» и «AnotherPassword456». 2. Сравнить два возвращаемых значения.
Данные тестирования	Пароли: «TestPassword123», «AnotherPassword456»
Ожидаемый результат	Метод возвращает разные строки для двух различных паролей.
Фактический результат	Соответствует ожидаемому результату.
Предпосылки	Убедитесь, что метод PasswordHelper.HashPassword реализован и доступен для вызова.
Постусловия	Корректный хэш был сгенерирован и различается для разных данных.
Статус (Pass/Fail)	Pass
Комментарии	Тест проверяет уникальность хэша для разных входных данных.

Таблица 5.8 – Тест «LoginPage\_ShouldRenderFormElements»

Наименование	Описание
Наименование проекта	Фотоцентр
Номер версии	1.0
Имя тестера	Михаил
Даты тестирования	25.11.2024
Test Case #	TC_LoginPage_1

Продолжение таблицы №5.8

Приоритет тестирования	Высокий
Название тестирования/Имя	LoginPage_ShouldRenderFormElements
Резюме испытания	Убедиться, что все элементы формы (поля ввода и кнопки) корректно отображаются на странице.
Шаги тестирования	1. Рендерить компонент страницы входа (Login). 2. Найти все элементы input и проверить их атрибуты. 3. Убедиться, что элементы имеют placeholder, required, name, и class.
Данные тестирования	Нет данных, тест проверяет элементы интерфейса.
Ожидаемый результат	Элементы формы отображаются корректно с указанными атрибутами.
Фактический результат	Соответствует ожидаемому результату.
Предпосылки	Компонент Login реализован и доступен для тестирования.
Постусловия	Проверка завершена, элементы интерфейса отображаются корректно.
Статус (Pass/Fail)	Pass
Комментарии	Тест проверяет только статические элементы страницы без взаимодействия.

Таблица 5.9 – Тест

«LoginPage\_ShouldShowErrorMessageForInvalidCredentials»

Наименование	Описание
Наименование проекта	Фотоцентр
Номер версии	1.0
Имя тестера	Михаил
Даты тестирования	25.11.2024

Продолжение таблицы №5.9

Test Case #	TC_LoginPage_2
Приоритет тестирования	Высокий
Название тестирования/Имя	LoginPage_ShouldShowErrorMessageForInvalidCredentials
Резюме испытания	Проверить, что при вводе неверных данных на странице входа отображается сообщение об ошибке.
Шаги тестирования	<ol style="list-style-type: none"> <li>1. Мокировать репозиторий пользователя (IUserRepository), чтобы возвращать null для любого ввода.</li> <li>2. Рендерить компонент Login.</li> <li>3. Ввести неверные данные пользователя ("invalidUser", "wrongPassword") в поля формы.</li> <li>4. Нажать кнопку отправки формы.</li> <li>5. Проверить, что на странице отображается сообщение об ошибке с текстом "Неверное имя пользователя или пароль".</li> </ol>
Данные тестирования	Имя пользователя: "invalidUser", Пароль: "wrongPassword".
Ожидаемый результат	Сообщение об ошибке "Неверное имя пользователя или пароль" отображается на экране.
Фактический результат	Соответствует ожидаемому результату.
Предпосылки	Мокирование IUserRepository реализовано, компонент Login доступен для тестирования.
Постусловия	Ввод неверных данных приводит к отображению сообщения об ошибке.
Статус (Pass/Fail)	Pass
Комментарии	Этот тест проверяет обработку ошибок и отображение сообщений на уровне пользовательского интерфейса.

Таблица 5.10 – Тест «LogActionAsync\_ShouldAddLogToDatabase»

Наименование	Описание
Наименование проекта	Фотоцентр
Номер версии	1.0
Имя тестера	Михаил
Даты тестирования	25.11.2024
Test Case #	TC_ActionLogger_1
Приоритет тестирования	Высокий
Название тестирования/Имя	LogActionAsync_ShouldAddLogToDatabase
Резюме испытания	Проверить, что метод LogActionAsync корректно добавляет запись в базу данных.
Шаги тестирования	<ol style="list-style-type: none"> <li>1. Настроить базу данных в памяти с помощью UseInMemoryDatabase.</li> <li>2. Создать экземпляр ActionLogger с использованием тестовой базы данных.</li> <li>3. Определить тестовые данные: userId, actionType, details, tableName, severityLevel.</li> <li>4. Вызвать метод LogActionAsync с тестовыми данными.</li> <li>5. Проверить, что запись с указанным userId существует в базе данных.</li> <li>6. Убедиться, что значения полей в базе данных совпадают с переданными данными.</li> </ol>
Данные тестирования	userId: 1, actionType: "TestAction", details: "Test details", tableName: "TestTable", severityLevel: "Info".
Ожидаемый результат	Запись успешно добавляется в базу данных, значения всех полей совпадают с переданными данными.
Фактический результат	Соответствует ожидаемому результату.

## Продолжение таблицы №5.10

Предпосылки	Настройка базы данных в памяти, корректная реализация класса ActionLogger.
Постусловия	Запись корректно добавлена в базу данных, таблица ActionLogs содержит новые данные.
Статус (Pass/Fail)	Pass
Комментарии	Этот тест покрывает только успешный сценарий добавления записи в базу данных.

Результаты выполнения тестов представлены на рисунке 4.29.

Тестирование ▼	Длительн.
TestProject (10)	947
TestProject (10)	947
PhotoConverterTests (4)	2
ConvertToBase64_ShouldReturnEmptyString_WhenImagesNull	< 1
ConvertToBase64_ShouldReturnEmptyString_WhenImageAndContentTypeAreNull	< 1
ConvertToBase64_ShouldReturnEmptyString_WhenContentTypesNull	2
ConvertToBase64_ShouldReturnCorrectString	< 1
PasswordHelperTests (3)	9
HashPassword_ShouldReturnSameHashForSamePassword	< 1
HashPassword_ShouldReturnDifferentHashForDifferentPasswords	8
HashPassword_ShouldReturnCorrectHash	1
LoginPageTest (2)	474
LoginPage_ShouldShowErrorMessageForInvalidCredentials	124
LoginPage_ShouldRenderFormElements	350
ActionLoggerTests (1)	462
LogActionAsync_ShouldAddLogToDatabase	462

Рисунок 4.29 – Результаты тестирования

## Заключение

Подводя итоги, хочется отметить, что проектирование и разработка информационной системы для фотоцентра стало важным шагом в цифровизации его бизнес-процессов. В условиях современного мира, где многие бизнесы переходят в онлайн, веб-приложение для фотоцентра предоставляет удобный интерфейс для взаимодействия с клиентами и упрощает внутренние процессы компании. Система автоматизирует ряд рутинных задач, таких как: оформление заказов на фотопродукцию, просмотр и редактирование информации о заказах, управление данными клиентов и генерация отчетов.

Исследование, проведенное в ходе работы, позволило проанализировать лучшие практики разработки информационных систем, начиная с анализа потребностей фотоцентра и заканчивая выбором оптимальных инструментов для веб-разработки и проектирования базы данных. Это дало возможность создать эффективное решение, отвечающее специфическим требованиям бизнеса.

Практическая часть работы была сосредоточена на разработке веб-приложения, которое обеспечивает легкость и удобство взаимодействия пользователей с системой. Были реализованы ключевые функции, такие как оформление и отслеживание заказов, а также администрирование данных. Важным этапом стало создание базы данных, которая хранит информацию о клиентах, заказах и операциях с фотографиями. Также был реализован механизм логирования для отслеживания действий пользователей и обеспечения безопасности данных.

Процесс тестирования системы показал её стабильность, безопасность и соответствие функциональным требованиям, что подтверждает ее готовность к эксплуатации. Тестирование обеспечило надежность приложения и выявило возможные уязвимости, которые были успешно устранены.

Таким образом, конечным продуктом является веб-приложение для клиентов и сотрудников фотоцентра, полностью протестированное и готовое к реальному использованию. Разработанная база данных поддерживает все

необходимые функции для эффективного управления заказами, логированием и работы с клиентами. Полученные результаты показывают, что применение современных технологий в разработке веб-приложений и баз данных позволяет создать качественную информационную систему, которая будет удобна в использовании и способна значительно улучшить бизнес-процессы фотоцентра.



### **Список использованных источников**

1. Белоногов Г.Г., Новоселов А.П. Автоматизация процессов накопления, поиска и обобщения информации / - М.: Наука, 2017. - 256 с.
2. Ульман Дж. Основы систем баз данных / - М.: Финансы и статистика, 2017. - 292 с.
3. Пинягина О.В., Фукин И.А. Практикум по курсу «Базы данных»: учеб. пособие / Казан. (Приволж.) федер. ун-т. - Казань: Казанский университет, 2012. - 91 с.
4. Проектирование и реализация баз данных в СУБД MySQL с использованием MySQL Workbench: учеб. пособие / С.А. Мартишин [и др.] - М.: ИД ФОРУМ: НИЦ Инфра-М, 2012. - 160 с.
5. Пирогов В.Ю. Информационные системы и базы данных: организация и проектирование: учеб. пособие / - СПб.: БХВ-Петербург, 2009. - 528 с.
6. Дунаев В.В. Базы данных. Язык SQL для студента. - 2-е изд., доп. и перераб.- СПб.: БХВ-Петербург, 2007.

## Тест антиплагиат

49

## Приложение Б

### Листинг

```
using Microsoft.AspNetCore.Authentication.Cookies;
using Microsoft.EntityFrameworkCore;
using Фотоцентр.Components;
using Фотоцентр.Services;
using Фотоцентр.Data;
using Microsoft.AspNetCore.Identity;

var builder = WebApplication.CreateBuilder(args);

// Add services to the container.
builder.Services.AddRazorComponents()
    .AddInteractiveServerComponents();

builder.Services.AddDbContextFactory<AppDbContext>(options =>
    options.UseSqlServer(builder.Configuration.GetConnectionString("DefaultCon-
nection")),
    ServiceLifetime.Scoped);

builder.Services.AddSession(options =>
{
    options.IdleTimeout = TimeSpan.FromMinutes(30);
    options.Cookie.HttpOnly = true;
    options.Cookie.IsEssential = true;
});

builder.Services.AddControllersWithViews();
```

```
builder.Services.AddAuthentication(CookieAuthenticationDefaults.AuthenticationScheme)
```

```
.AddCookie(options =>
{
    options.Cookie.Name = "auth_token";
    options.LoginPath = "/login";
    options.Cookie.MaxAge = TimeSpan.FromMinutes(30);
    options.AccessDeniedPath = "/error";
});
```

```
builder.Services.AddHttpContextAccessor();
builder.Services.AddAuthorization();
builder.Services.AddCascadingAuthenticationState();
builder.Services.AddBlazorBootstrap();
builder.Services.AddScoped<IActionLogger, ActionLogger>();
builder.Services.AddQuickGridEntityFrameworkAdapter();
builder.Services.AddControllersWithViews();
builder.Services.AddRazorPages();
```

```
var app = builder.Build();
```

```
// Configure the HTTP request pipeline.
```

```
if (!app.Environment.IsDevelopment())
{
    app.UseExceptionHandler("/Error", createScopeForErrors: true);
}
```

```
app.UseStaticFiles();
app.UseAuthentication();
```

```
app.UseAuthorization();  
app.UseAntiforgery();  
app.UseSession();  
app.MapRazorComponents<App>()  
    .AddInteractiveServerRenderMode();  
app.Run();
```