Частное учреждение образования

«Колледж бизнеса и права»

Создание веб-приложения для автоматизации изготовления и продажи гитар и сопутствующих товаров

Пояснительная записка

к курсовому проекту по предмету

«Конструирование программ и языки программирования»

КП Т.197005.401

Руководитель проекта (А.В.Кривошеина)

Обучающейся (К.С.Жданович)

2024

СОДЕРЖАНИЕ

Изм.

Лист

№ докум.

Подпись

Датаа

Лист

2

КП Т.197005.401

Разраб.

Жданович К.С.

Провер.

Кривошеина А.В.

Реценз.

Н. Контр.

Утверд.

Лит.

Листов

КБП

56

у

[Введение 4](#_Toc170306410)

[1 Описание задачи 6](#_Toc170306411)

[1.1 Анализ предметной области 6](#_Toc170306412)

[1.2 Постановка задачи 6](#_Toc170306413)

[2 Проектирование веб-приложения 8](#_Toc170306414)

[2.1 Проектирование модели 8](#_Toc170306415)

[2.2 Требования к веб-приложению 10](#_Toc170306416)

[2.3 Структура веб-приложения 13](#_Toc170306417)

[2.4 Проектирование макета веб-приложения 14](#_Toc170306418)

[2.5 Программно-технические средства, необходимые для разработки приложения 18](#_Toc170306419)

[2.6 Защита и сохранность данных 19](#_Toc170306420)

[2.7 Организация и ведение информационной базы (модели) 20](#_Toc170306421)

[3 Реализация веб-приложения 25](#_Toc170306422)

[3.1 Описание разделов веб-приложения 25](#_Toc170306423)

[3.2 Разработка административной части приложения 26](#_Toc170306424)

[3.3 Разработка клиентской части приложения 27](#_Toc170306425)

[3.4 Описание используемых функций и процедур 28](#_Toc170306426)

[3.5 Функциональное тестирование 29](#_Toc170306427)

[4 Применение 33](#_Toc170306428)

[4.1 Назначение веб-приложения 33](#_Toc170306429)

[4.2 Программно-аппаратное обеспечение сервера и клиента 34](#_Toc170306430)

[Заключение 35](#_Toc170306431)

[Список использованных источников 36](#_Toc170306432)

[Приложение А 37](#_Toc170306433)

[Приложение Б 45](#_Toc170306434)

*Создание веб-приложения для автоматизации изготовления и продажи гитар и сопутствующих товаров*

# Введение

В наше время магазины и производители музыкальных инструментов, таких как гитары, сталкиваются с рядом сложностей при организации процесса заказа, изготовления и продажи своей продукции. Если эту задачу выполнять вручную, без подходящей системы, возникает риск ошибок, неэффективности и трудностей в управлении заказами.

В таком случае создание веб-приложения для автоматизации изготовления и продажи гитар и сопутствующих товаров является эффективным решением. Благодаря веб-приложению, клиенты могут безопасно и удобно оформлять заказы на гитары и аксессуары, выбирая нужные характеристики и комплектацию. Это позволяет минимизировать время ожидания и устранить возможные ошибки, связанные с ручным вводом информации.

Веб-приложение предоставляет пользователям простой и интуитивно понятный интерфейс, где они могут выбирать доступные модели гитар, комплектующие, аксессуары, а также отслеживать статус своих заказов. Автоматизированная система также облегчает работу персонала магазина или производства. Ресурсы, которые ранее были затрачены на ручную обработку и управление заказами, теперь могут быть перераспределены на более важные задачи. Сотрудники могут мониторить и управлять заказами через веб-интерфейс, а также быстро получать доступ к необходимой информации о клиентах и сформированных заказах.

Целью разработки веб-приложения "Создание веб-приложения для автоматизации изготовления и продажи гитар и сопутствующих товаров" является обеспечение эффективного и удобного процесса заказа, производства и реализации музыкальных инструментов и аксессуаров.

Актуальность создания данного приложения заключается в возможности для клиентов просматривать информацию о доступных моделях гитар, комплектующих и аксессуаров, а также оформлять заказы на изготовление индивидуальных инструментов. Приложение также позволяет минимизировать время ожидания и предотвращает возможные ошибки, связанные с ручным вводом информации, повышая эффективность деятельности магазина или производства.

Пояснительная записка состоит из четырех разделов, в которых расписаны шаги проектирования, разработки, а также описание алгоритмов и входных\выходных данных.

В первом разделе «Описание задачи» описываются цели создания приложения и проектирование содержимого приложения. Проводится исследование предметной области, задачи, которые планируется решать посредством приложения, определяется целевая аудитория, описываются алгоритмы получения промежуточных и итоговых показателей, обоснование необходимости компьютерной обработки информации или процессов.

Во втором разделе «Проектирование веб-приложения» описываются средства защиты будущего веб-приложения, накладываемые ограничения на интерфейс, общие требования к интерфейсу (расположение, размер, форма, шрифт и прочее для всех элементов управления), требования к аппаратным и операционным ресурсам (операционная система, объем памяти устройства, доступ к интернету и его скорость, дополнительные устройства (мышь, клавиатура, принтер и т.д.)).

В третьем разделе «Реализации веб-приложения» перечисляются инструменты разработки, необходимые для разработки веб-приложения, порядок авторизации пользователей, организацию данных, а также применяемые технологии.

В четвертом разделе «Применение» описывается назначение приложения. Анализируется программное обеспечение, которое необходимо для функционирования веб-приложения.

В заключении описывается выполнение поставленной задачи, степень соответствия проектных решений задания, причины несоответствия, если таковые имеются.

В списке информационных источников перечисляются информационные источники, которые были использованы в процессе разработки веб-приложения.

В приложении А представлен текст программных модулей веб-приложения.

В приложении Б представлены формы входных и выходных документов.

В приложении В представлен результаты работы веб-приложения.

# Описание задачи

## Анализ предметной области

Создание веб-приложения для автоматизации процесса изготовления и продажи гитар, а также сопутствующих товаров. Это приложение призвано упростить и оптимизировать операции, связанные с заказом и изготовлением гитар, а также управлением их продажей.

Автоматизированное веб-приложение разработано для удобного управления и контроля процесса изготовления и продажи гитар. Оно включает в себя набор инструментов и функций, которые помогают производителям, продавцам и клиентам эффективно справляться с задачами, связанными с заказом, производством, инвентаризацией и реализацией гитар и сопутствующих товаров.

Работа веб-приложения охватывает полный цикл - от приема заказа на изготовление гитары до её продажи клиенту. Это включает в себя получение заказа, выбор комплектующих, планирование производства, отслеживание статуса изготовления, управление складскими запасами, ценообразование и организацию продаж. Важной задачей является обеспечение прозрачности процессов для всех участников - администраторов, мастеров и покупателей.

Автоматизация производства и продаж гитар позволяет значительно повысить эффективность и гибкость бизнеса. Это обеспечивает удобство для клиентов, экономит время и ресурсы производителей и продавцов.

Веб-приложение предлагает удобный пользовательский интерфейс и широкую функциональность для управления заказами, производством, инвентарем, ценообразованием и продажами в интегрированной базе данных.

Потенциальными пользователями веб-приложения являются производители гитар, музыкальные магазины и онлайн-площадки по продаже музыкальных инструментов, а также индивидуальные мастера, которые заинтересованы в автоматизации и оптимизации своих бизнес-процессов.

## Постановка задачи

Функции, которые должны быть автоматизированы в соответствии с проанализированными бизнес-процессами и задачами интернет-магазина:

* веб-приложение должно позволять пользователю вводить данные о клиенте, такие как ФИО, номер телефона, адрес проживания;
* ввод данных о музыкальных инсрументах, таких как название товара, тип, описание, цена, фото;
* отображение выбранных данных на пользовательском интерфейсе;
* сохранение введенных данных в таблице базы данных;
* автоматическое формирование заказа с участием выбранного пользователя, с возможностью выбора даты и времени готовности, а также даты и времени создания заказа;
* составление заказа в формате PDF для печати;
* интерфейс веб-приложения должен быть максимально прост и интуитивно понятен любому пользователю, что обеспечивает комфортное взаимодействие пациентов с программным средством.

Существующие аналоги с указанием отличий, которые будут реализованы в разрабатываемом программном средстве:

* «MusicMarket» – ­­­онлайн-сервис для автоматизации продажи музыкальных инструмнтов в магазинах. Сайт предлагает доставку по всей Беларуси, возможность оплаты частями. На сайте есть отзывы покупателей, информация о программе лояльности и самовывозе. В отличие от разрабатываемого приложения не предоставляет функцию изготовления индивидуальных инструментов.

# Проектирование веб-приложения

## Проектирование модели

Исходя из исследования предметной области можно выделить следующие сущности проекта.

Каждый клиент, который был зарегистрирован в интернет-магазине, характеризуется следующими данными:

* ФИО клиента;
* адрес проживания;
* номер телефона.

Каждый товар, предоставляемая в интернет-магазине, характеризуется следующими данными:

* название;
* категория;
* описание;
* цена;
* фото;
* наличие.

Каждый мастер, который изготавливает музыкальные инструменты, характеризуется следующими данными:

* ФИО мастера.

Каждая запись в корзину клиента характеризуется следующими данными:

* данные клиента;
* данные продукта;
* количество.

Каждый медицинский работник, который работает в медицинской клинике, характеризуется следующими данными:

* ФИО персонала.

Каждый товар, который заказывает клиент характеризуется следующими данными:

* данные заказа;
* данные товара;
* количество
* сумма.

Каждый заказ характеризуется следующими данными:

* данные клиента;
* данные мастера;
* сумма;
* дата и время оформления;
* дата и время завершения заказа.

Каждый индивидуальный заказ характеризуется следующими данными:

* данные клиента;
* данные мастера;
* тип;
* верхняя дека;
* нижняя дека;
* цвет.

При работе с клиентами, мастер должен иметь возможность решать следующие задачи:

* добавление новых записей в индивидуальные заказы клиентов;
* добавление новых заказов клиентов;
* просмотр заказов клиентов.

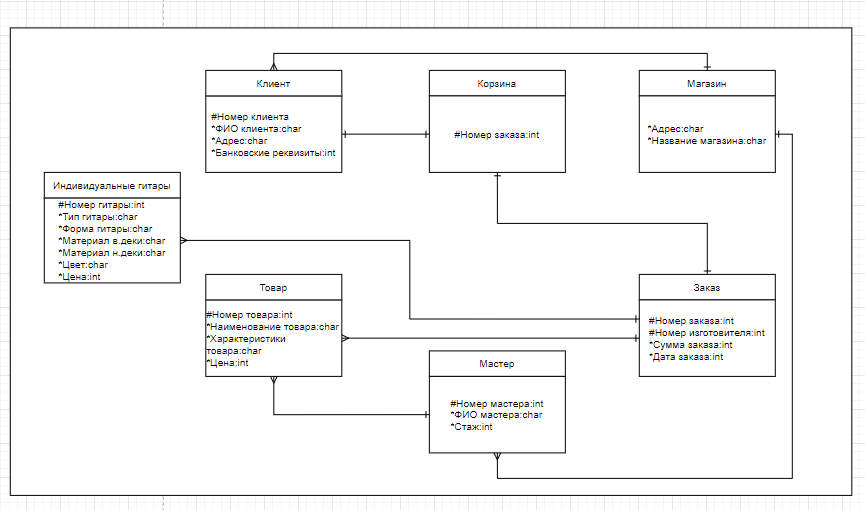
Диаграмма «Сущность-связь» представлена на рисунке 2.1.1.

Рисунок 2.1.1 – Диаграмма «Сущность-связь» в нотации Мартина

Диаграмма вариантов использования (ДВИ) – это инструмент, используемый для моделирования функциональных требований системы. Они описывают взаимодействие внешних пользователей или других систем (актеров) с системой для достижения конкретных целей (вариантов использования). Данная диаграмма для разрабатываемого средства представлена на первом листе графической части.

Диаграмма классов – это диаграмма, которая используется для визуализации структуры системы путем отображения ее классов, атрибутов, методов и отношений между классами. Диаграмма классов для разрабатываемого средства представлена на втором листе графической части.

Диаграмма последовательности – это еще один из типов диаграмм UML, который используется для представления взаимодействий между объектами в системе в хронологическом порядке, показывая, как объекты обмениваются сообщениями для выполнения функциональности. Диаграмма последовательности для функции составления заказа представлена на третьем листе графической части.

Диаграмма состояний — это графическое представление поведения системы, которое моделирует различные состояния, переходы между ними и связи между состояниями. Она позволяет описать, как система изменяет свое состояние в ответ на внешние события или внутренние условия. Диаграмма состояний для функции составления заказа представлена на четвертом листе графической части.

Диаграмма развертывания — это графическое представление архитектуры системы, которое показывает размещение компонентов на клиентской и серверной частях системы, а также связи между ними. Она предоставляет обзор структуры системы и помогает визуализировать развертывание компонентов на физических устройствах. Диаграмма развертывания представлена на пятом листе графической части.

Алгоритмы, используемые в этом программном средстве, включают следующие шаги:

* ведение и редактирование базы данных клиентов: добавление, редактирование и удаление информации о клиентах, включая их персональные данные, историю заказов, текущие заказы и индивидуальные заказы.
* учет записей на покупку товаров: возможность клиентам заказывать музыкальные инструменты и их комплектующие.

## Требования к веб-приложению

Требования к стилистическому оформлению:

* веб-приложение должно иметь профессиональный и современный внешний вид, соответствующий тематике интернет-магазина;
* цветовая палитра должна быть спокойной и гармоничной, с использованием нейтральных и пастельных оттенков;
* использование иконок и изображений будет соответствовать контексту и улучшит восприятие пользователей;
* интерфейс должен быть минималистичным и чистым, избегая излишней сложности и перегруженности элементами;
* стили и элементы должны быть единообразными на всех страницах приложения, чтобы обеспечить целостность и легкость восприятия.

Требования к графическому дизайну:

* графический дизайн должен быть согласован с общим стилем и цветовой палитрой приложения;
* использование простых и интуитивно понятных иконок и элементов управления поможет пользователям быстро ориентироваться;
* графические элементы должны быть оптимизированы для быстрой загрузки страниц и оптимального использования ресурсов.

Требования к шрифтовому оформлению:

* шрифты должны быть читабельными даже при малом размере и различных устройствах;
* рекомендуется использовать шрифты без засечек для лучшей читаемости;
* шрифты должны быть согласованы с общим стилем и дизайном приложения.

Требования к средствам просмотра:

* веб-приложение должно быть адаптивным и отзывчивым, чтобы корректно отображаться на различных устройствах, таких как компьютеры и ноутбуки;
* должна быть обеспечена возможность просмотра и использования приложения в разных браузерах, таких как Chrome, Firefox, Safari и т.д. Требования к контенту (наполнению):
* веб-приложение должно предоставлять пользователю возможность вести и редактировать базу данных клиентов, включая их персональные данные, историю заказов, текущие заказы и индивидуальные заказы;
* приложение должно предоставлять возможность клиентам заказывать музыкальные инструменты и их комплектующие, заказывать индивидуальные инструменты;
* приложение должно предоставлять возможность просмотра и управления заказами клиентов.

Требования к системе управления контентом:

* приложение должно предоставлять возможность просмотра и управления заказами клиентов;
* система управления контентом должна обеспечивать безопасность данных и разграничение прав доступа к базе данных.

Требования к компоновке страниц:

* страницы приложения должны быть организованы логично и интуитивно понятно для пользователя;
* шапка, кнопки, формы и панели инструментов должны быть разработаны таким образом, чтобы обеспечить легкий и быстрый доступ к основным функциям приложения;
* размещение элементов интерфейса должно быть продуманным и удобным для использования на различных устройствах.

## Структура веб-приложения

Ниже приведено графическое представление логической структуры веб-приложения для автоматизации работы интернет-магазина:

* главная страница
* каталог гитар и комплектующих
* конструктор для создания прототипа гитары
* корзина
* оформление заказа
* профиль покупателя
* просмотр индивидуальных заказов мастером

Ниже приведена организация лаконичного интерфейса программного средства:

* создание меню
* разработка кнопочных форм
* разработка панелей инструментов

Структурная схема веб-приложения в виде диаграммы fire flow представлена на рисунке 2.3.1

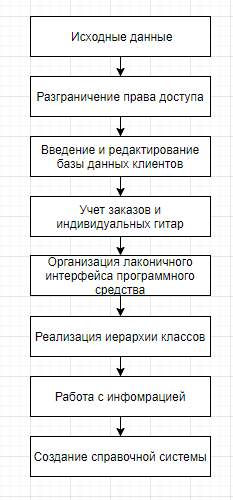


Рисунок 2.3.1 – Структурная схема

## Проектирование макета веб-приложения

Для разработки веб-приложения для автоматизации изготовления и продажи гитар и сопутствующих товаров рекомендуется использовать блочный тип дизайна. Блочный дизайн предоставляет гибкость и адаптивность, что является важным для веб-приложений, которые будут использоваться на различных устройствах и экранах. Блочный дизайн позволит легко управлять размещением элементов на странице, а также обеспечит хорошую читабельность контента и удобство использования.

Содержимое страницы авторизации:

* текстовое поле «Логин» будет позволять ввести логин аккаунта;
* текстовое поле «Пароль» будет позволять ввести пароль от аккаунта;
* кнопка «Войти» будет осуществлять проверку введенных данных и вход в аккаунт (переход в профиль пользователя);
* кнопка «Создать аккаунт»

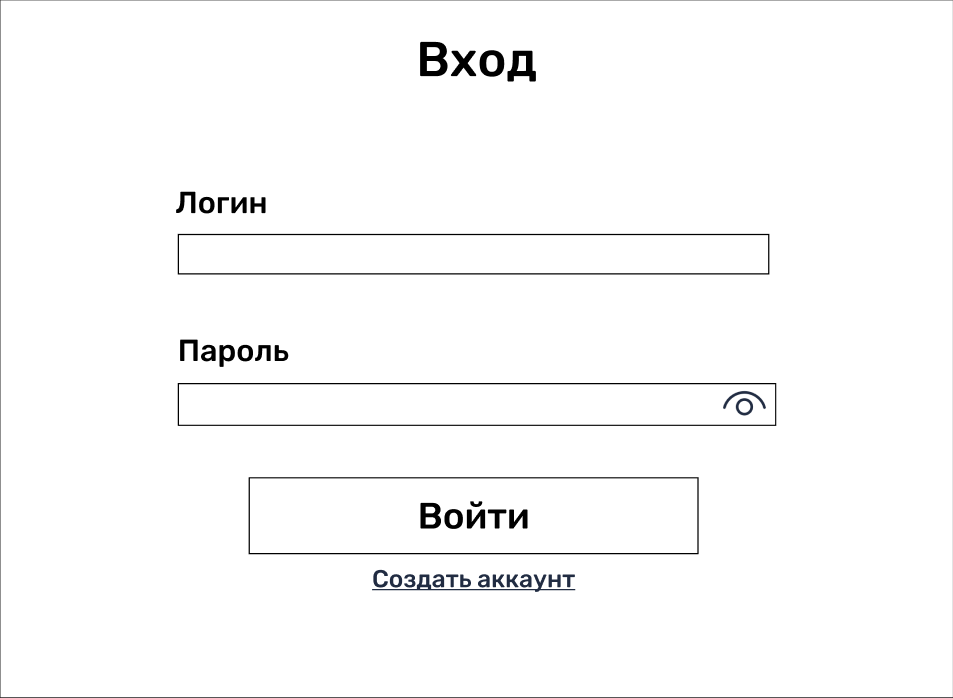
На рисунке 2.4.1 представлен прототип окна авторизации.

Рисунок 2.4.1 – Макет страницы авторизации

Содержимое главной страницы:

* шапка страницы с кнопками для перехода по основным страницам;
* блок с описанием интернет-магазина;
* блок с описанием преимуществ интернет-магазина;
* подвал страницы с контактной информацией, временем работы.

На рисунке 2.4.2 представлен макет главной страницы веб-приложения.

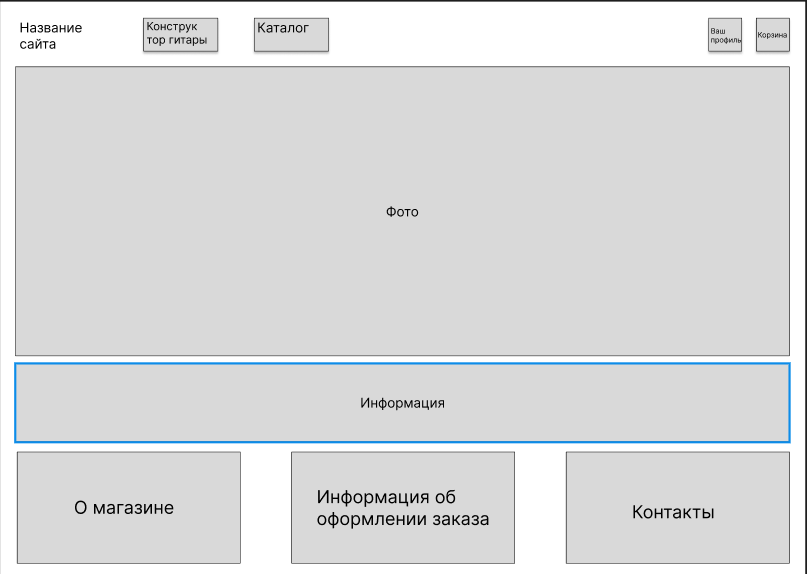


Рисунок 2.4.2 – Макет главной страницы

Содержимое страницы профиля пациента:

* шапка страницы с кнопками для перехода по основным страницам;
* текстовое поле «Логин», которое будет выводить логин зарегистрированного пользователя;
* текстовое поле «Пароль», которое будет выводить пароль зарегистрированного пользователя;
* текстовое поле «Имя», которое будет выводить имя зарегистрированного пользователя;
* текстовое поле «Фамилия», которое будет выводить фамилию зарегистрированного пользователя;
* текстовое поле «Отчество», которое будет выводить отчество зарегистрированного пользователя;
* текстовое поле «Номер телефона», которое будет выводить номер телефона зарегистрированного пользователя;
* текстовое поле «Адрес проживания», которое будет выводить адрес зарегистрированного пользователя;
* кнопка для перехода к информации о заказах пользователя;
* кнопка для выхода из аккаунта пользователя.

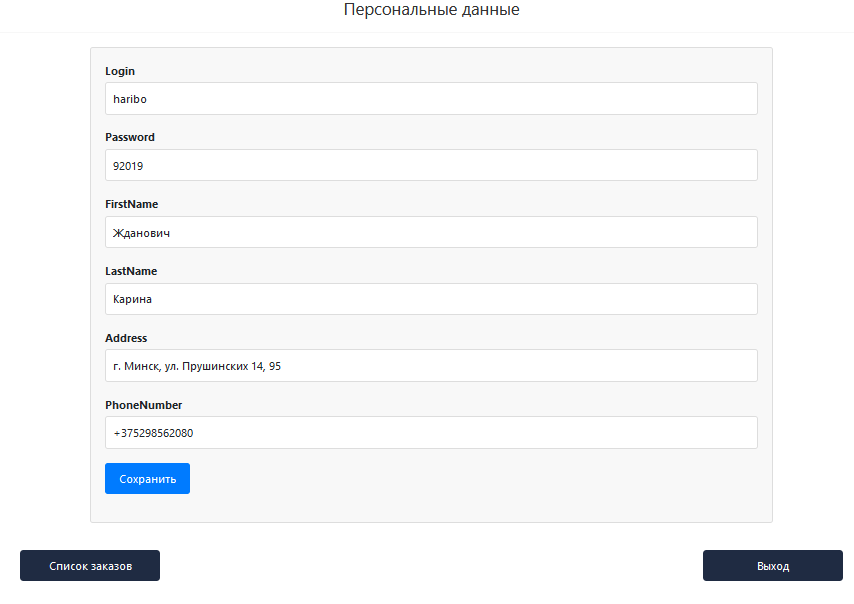
На рисунке 2.4.3 представлен макет страницы профиля клиента.

Рисунок 2.4.3 – Макет страницы профиля клиента

Содержимое страницы товаров:

* шапка страницы с кнопками для перехода по основным страницам;
* блок «Товары», который будет выводить карточки товаров.

На рисунке 2.4.4 представлен макет страницы товаров. 

Рисунок 2.4.4 – Макет страницы выбранной услуги

Содержимое страницы профиля мастера:

* шапка страницы с кнопками для перехода по основным страницам;
* блок «Заказы», который будет выводить заказы мастера.

На рисунке 2.4.5 представлен макет страницы профиля специалиста. 

Рисунок 2.4.5 – Макет страницы профиля специалиста

## Программно-технические средства, необходимые для разработки приложения

Анализируя предметную область и рассмотрев функциональные возможности приложения, было принято решение использовать следующие инструменты разработки.

В качестве интегрированной среды разработки будет использоваться Microsoft Visual Studio. Это позволит эффективно создавать, отлаживать и собирать приложение, а также использовать широкий спектр инструментов и функций, доступных в Visual Studio.

Для разработки приложения будет использоваться язык C# с помощью ASP.NET Core и Razor Pages. ASP.NET Core предоставляет мощный фреймворк для создания веб-приложений, а Razor Pages обеспечивает удобную модель программирования для создания интерфейса.

Для работы с базой данных будет использоваться SQL Server, а для управления базой данных — SQL Server Management Studio. SQL Server обладает мощными возможностями хранения и управления данными, а SQL Server Management Studio предоставляет удобный интерфейс для администрирования базы данных.

Для разработки веб-интерфейса приложения будут использоваться HTML, CSS и JavaScript. HTML обеспечивает структуру страницы, CSS — визуальное оформление, а JavaScript — динамическое взаимодействие с пользователем.

Для работы с данными в приложении будет использоваться Entity Framework, который обеспечивает удобное взаимодействие с базой данных SQL Server. Кроме того, для выполнения запросов к данным будет использоваться язык запросов LINQ.

Применение этих технологий обусловлено их совместимостью, эффективностью и удобством использования при разработке веб-приложений, требующих работы с данными и поддержки сети. Использование этих инструментов позволит создать функциональное и удобное в использовании веб-приложение с сетевой поддержкой и минимальными затратами времени и ресурсов.

## Защита и сохранность данных

В контексте разработки веб-приложения для автоматизации изготовления и продажи гитар и сопутствующих товаров, обеспечение защиты и сохранности данных является критически важным аспектом. Ниже представлено описание мер, которые можно принять для обеспечения безопасности информации.

Ограничение доступа к данным:

* разграничение прав доступа: веб-приложение должно иметь механизм аутентификации и авторизации, который позволяет управлять доступом пользователей к различным частям приложения и данным. Разные роли пользователей (администраторы, мастера, клиенты) должны иметь различные уровни доступа и функциональные возможности;
* управление доступом на уровне базы данных: должны быть применены механизмы управления доступом на уровне базы данных, такие как роли и разрешения, чтобы ограничить доступ к конкретным таблицам и данным.

Защита информации от несанкционированного использования:

* шифрование данных: критически важные данные, такие как персональные данные клиентов и история заказоы, должны быть зашифрованы в базе данных и при передаче по сети. Это поможет предотвратить несанкционированный доступ к данным, даже если злоумышленник получит физический или сетевой доступ к системе;
* защита от SQL-инъекций: необходимо применять параметризованные запросы или использовать ORM (Object-Relational Mapping) для взаимодействия с базой данных, чтобы предотвратить атаки SQL-инъекций.

Механизмы защиты:

* резервное копирование данных: регулярное создание резервных копий базы данных поможет защитить данные от потери в случае сбоев или чрезвычайных ситуаций;
* защита от SQL-инъекций: необходимо применять параметризованные запросы или использовать ORM (Object-Relational Mapping) для взаимодействия с базой данных, чтобы предотвратить атаки SQL-инъекций.
* мониторинг и журналирование: важно осуществлять мониторинг и журналирование активности пользователей, чтобы обнаруживать подозрительную активность или нарушения безопасности;
* обновление и патчи: регулярное обновление операционной системы, фреймворка ASP.NET Core и других компонентов приложения позволит исправить уязвимости безопасности и предотвратить эксплуатацию известных уязвимостей.

## Организация и ведение информационной базы (модели)

Организация данных подразумевает создание модели данных, главными элементами которой являются сущности и их связи.

Реляционная модель основана на математическом понятии отношения, представлением которого является таблица. В реляционной модели отношения используются для хранения информации об объектах, представленных в базе данных. Отношение имеет вид двухмерной таблицы, в которой строки соответствуют записям, а столбцы атрибутам. Каждая запись должна однозначно характеризоваться в таблице. Для этого используют первичные и вторичные ключи. Достоинством реляционной модели является простота и удобство физической реализации.

Реляционная модель базы данных подразумевает нормализацию всех таблиц данных. Нормализация – это формальный метод анализа отношений на основе их первичного ключа и функциональных зависимостей, существующих между их атрибутами. Структура базы данных разрабатываемого программного средства включает девять таблиц.

Структура данных таблиц и их краткое описание приводится в таблицах 2.1-2.9.

Таблица «Administrators» хранит информацию о всех администраторах. Структура приведена в таблице 2.1.

Таблица 2.1 – Структура таблицы «Administrators»

|  |  |  |  |
| --- | --- | --- | --- |
| Имя поля | Тип данных | Размер, байт | Описание |
| Id | INTEGER | 8 | Идентификационный номер |
| Login | nvarchar(60) | 120 | Логин пользователя |
| Password | nvarchar(100) | 200 | Пароль пользователя |

Таблица «Status» хранит информацию о статусах заказа. Структура приведена в таблице 2.2.

Таблица 2.2 – Структура таблицы «Status»

|  |  |  |  |
| --- | --- | --- | --- |
| Имя поля | Тип данных | Размер, байт | Описание |
| Id | int | 4 | Идентификационный номер |
| Name | nvarchar(60) | 120 | Название статуса |

Таблица «Masters» хранит информацию о мастерах, структура приведена в таблице 2.3.

Таблица 2.3 – Структура таблицы «Masters»

|  |  |  |  |
| --- | --- | --- | --- |
| Имя поля | Тип данных | Размер, байт | Описание |
| Id | int | 4 | Идентификационный номер |
| Login | nvarchar(60) | 120 | Логин мастера |
| Password | nvarchar(100) | 200 | Пароль мастера |
| FirstName | nvarchar(60) | 120 | Фамилия мастера |
| MiddleName | nvarchar(60) | 120 | Имя мастера |
| LastName | nvarchar(60) | 120 | Отчество мастера |

Таблица «CustomGuitars» хранит информацию индивидуальных гитар, структура приведена в таблице 2.4.

Таблица 2.4 – Структура таблицы «CustomGuitars»

|  |  |  |  |
| --- | --- | --- | --- |
| Имя поля | Тип данных | Размер, байт | Описание |
| Id | int | 4 | Идентификационный номер индивидуальной гитары |
| Type | nvarchar(60) | 120 | Тип гитары |
| TopDeck | nvarchar(60) | 120 | Материал верхней деки |
| BottomDeck | nvarchar(60) | 120 | Материал нижней деки |
| Color | nvarchar(60) | 120 | Цвет гитары |
| MastersId | int | 4 | Идентификационный номер мастера |
| ClientsId | int | 4 | Идентификационный номер клиента |

Таблица «Clients» хранит информацию о клиентах, структура приведена в таблице 2.5.

Таблица 2.5 – Структура таблицы «Clients»

|  |  |  |  |
| --- | --- | --- | --- |
| Имя поля | Тип данных | Размер, байт | Описание |
| Id | int | 4 | Идентификационный номер |
| Login | nvarchar(60) | 120 | Логин пользователя |
| Password | nvarchar(100) | 200 | Пароль пользователя |
| FirstName | nvarchar(60) | 120 | Фамилия пользователя |
| LastName | nvarchar(60) | 120 | Имя пользователя |
| Address | nvarchar(100) | 200 | Адрес пользователя |
| PhoneNumber | nvarchar(MAX) | n\*2 | Номер телефона пользователя |

Таблица «Orders» хранит информацию о всех заказах. Структура приведена в таблице 2.6.

Таблица 2.6 – Структура таблицы «Orders»

|  |  |  |  |
| --- | --- | --- | --- |
| Имя поля | Тип данных | Размер, байт | Описание |
| Id | int | 4 | Идентификационный номер заказа |
| ClientsId | int | 4 | Идентификационный номер клиента |
| MastersId | int | 4 | Идентификационный номер мастера |
| Amount | double | 8 | Сумма заказа |
| OrderDate | datetime2(7) | 8 | Дата оформления |
| DeliveryDate | datetime2(7) | 8 | Дата завершения |

Таблица «Products» хранит информацию о всех товарах. Структура приведена в таблице 2.7.

Таблица 2.7 – Структура таблицы «Products»

|  |  |  |  |
| --- | --- | --- | --- |
| Имя поля | Тип данных | Размер, байт | Описание |
| Id | int | 4 | Идентификационный номер товара |
| Name | nvarchar(60) | 120 | Название товара |
| Description | nvarchar(MAX) | n\*2 | Описание товара |
| Price | double | 8 | Цена товара |
| Image | nvarchar(MAX) | n\*2 | Фотография товара |
| Availability | int | 4 | Наличие товара |
| Category | nvarchar(60) | 120 | Категория товара |

Таблица «Carts» хранит информацию о товарах в корзине клиента, структура приведена в таблице 2.8.

Таблица 2.8 – Структура таблицы «Carts»

|  |  |  |  |
| --- | --- | --- | --- |
| Имя поля | Тип данных | Размер, байт | Описание |
| Id | int | 4 | Идентификационный номер |
| ProductsId | int | 4 | Идентификационный номер товара |
| ClientsId | int | 4 | Идентификационный номер клиента |
| Quantity | int | 4 | Количество товара |

Таблица «OrderItems» хранит информацию о заказанных товарах, структура приведена в таблице 2.9.

Таблица 2.9 – Структура таблицы «OrderItems»

|  |  |  |  |
| --- | --- | --- | --- |
| Имя поля | Тип данных | Размер, байт | Описание |
| Id | int | 4 | Идентификационный номер |
| OrdersId | int | 4 | Идентификационный номер заказа |
| ProductsId | int | 4 | Идентификационный номер товара |
| Quantity | int | 4 | Количество товара |
| Price | double | 8 | Стоимость |

# Реализация веб-приложения

## Описание разделов веб-приложения

Разработанное веб-приложение для автоматизации изготовления и продажи гитар и сопутствующих товаров будет поддерживать различные профили пользователей, включая администратора, мастера и клиента. Каждый профиль будет иметь свой функционал и доступ к определенным разделам приложения. Ниже приведено описание разделов веб-приложения, главной страницы и навигационного меню, учитывая специфику каждого профиля.

Главная страница административной части делится на 2 блока: блок «Редактирование» и блок «Создание». Блок «Редактирование» включает в себя 7 кнопок отвечающих за редактирование всех данных приложения. Блок «Создание» включает в себя 7 кнопок отвечающих за создание данных для приложения.

Главная страница мастера тоже делится на 2 блока. Первый блок содержит заказы. Второй блок содержит индивидуальные заказы.

Главная страница клиента также делится на 2 блока. Первый блок содержит текстовые поля для вывода и редактирования персональных данных пациента. Второй блок содержит 2 кнопки. Первая кнопка для просмотра заказов. Вторая кнопка для выхода из аккаунта.

Навигационное меню в веб-приложении представляет собой шапку, присутствующую на каждой странице. На шапке расположены такие элементы, как:

* кнопка «Главная», при нажатии на которую открывается главная страница;
* кнопка «Каталог», при нажатии на которую открывается страница товаров;
* кнопка «Профиль» позволяет пользователям просматривать их профиль, если они авторизованы, или перенаправляет их на страницу входа, если они не авторизованы;
* кнопка «Конструктор» при нажатии на которую открывается страница конструктора, которая позволяет пользователям создать прототип индивидуальной гитары;
* кнопка «Корзины» при нажатии на которую открывается страница корзины, которая позволяет пользователям просмотреть товары в корзине.

## Разработка административной части приложения

Для разработки административной части веб-приложения интернет-магазина необходимо учесть логическую (карту) и физическую структуру (каталог) серверной части приложения.

Логическая структура серверной части веб-приложения будет включать следующие компоненты:

* справочник моделей, содержащий информацию о моделях данных, используемых в приложении, такие как информация о клиентах, история заказов;
* клиентская база данных, где будет храниться информация о клиентах, включая их персональные данные и историю заказов.

Физическая структура серверной части веб-приложения будет организована в виде соответствующего каталога, где будут размещены файлы и компоненты, необходимые для работы приложения и обработки запросов.

Одним из важных аспектов административной части приложения является навигация, которая будет обеспечивать удобное перемещение по различным разделам и функциям. Для этого будут использоваться меню и гиперссылки.

На главной странице административной части приложения будет разделена на два блока: блок «Редактирование» и блок «Создание». Блок «Редактирование» будет содержать 7 кнопок, позволяющих редактировать различные данные приложения:

* кнопка «Редактирование мастера» позволяет редактировать персональные данные мастеров;
* кнопка «Редактирование клиентов» позволяет редактировать персональную информацию клиентов;
* кнопка «Редактирование товаров» позволяет редактировать информацию о товаре;
* кнопка «Редактирование заказов» позволяет редактировать данные заказов;
* кнопка «Редактирование администратора» позволяет редактировать персональную информацию администраторов;
* кнопка «Редактирование заказанных предметов» позволяет редактировать заказанные товары;
* кнопка «Редактирование индивидуальных гитар» позволяет редактировать индивидуальные заказы.

Блок «Создание» включает в себя 7 кнопок отвечающих за создание данных для приложения:

* кнопка «Добавление мастера» позволяет добавить мастера;
* кнопка «Добавление клиента» позволяет добавить клиента;
* кнопка «Добавление товара» позволяет добавить товар;
* кнопка «Добавление заказа» позволяет добавить заказ;
* кнопка «Добавление администратора» позволяет добавить администратора;
* кнопка «Добавление заказанного товара» позволяет добавить заказанный товар;
* кнопка «Добавление индивидуальной гитары» позволяет добавить индивидуальную гитару.

Таким образом, административная часть приложения будет обеспечивать удобное редактирование и создание данных, а также навигацию с помощью меню и кнопок, обеспечивая пользователю лаконичный и понятный интерфейс для работы с приложением.

## Разработка клиентской части приложения

Для разработки клиентской части веб-приложения интернет-магазина, которая будет предназначена для мастеров и клиентов, необходимо учесть логическую (карту) и физическую структуру (каталог) клиентской части приложения.

Логическая структура клиентской части приложения для мастера будет включать следующие компоненты:

* таблица «Заказы» которая выводит заказы, которые назначены мастеру;
* таблица «Индивидуальные заказы» которая выводит индивидуальные заказы, которые назначены мастеру;
* кнопка «Выход» позволяет мастеру выйти из аккаунта.

Логическая структура клиентской части приложения для клиента будет включать следующие компоненты:

* текстовое поле «Имя», которое будет выводить имя зарегистрированного пользователя;
* текстовое поле «Фамилия», которое будет выводить фамилию зарегистрированного пользователя;
* текстовое поле «Отчество», которое будет выводить отчество зарегистрированного пользователя;
* текстовое поле «Номер телефона», которое будет выводить номер телефона зарегистрированного пользователя;
* текстовое поле «Адрес проживания», которое будет выводить адрес зарегистрированного пользователя;
* кнопка «Текущие заказы» позволяет клиенту просмотреть заказы;
* кнопка «Выход» позволяет клиенту выйти из аккаунта.

## Описание используемых функций и процедур

Описание функций и процедур в рамках среды разрабатываемого веб-приложения может быть следующим:

1. Функция "Регистрация пользователя":

- Описание: Позволяет пользователям создать новый аккаунт в системе.

- Взаимосвязи: Может быть связана с процедурой "Аутентификация пользователя" для входа в систему после успешной регистрации.

- Элементы управления: Форма регистрации со следующими полями: логин, пароль и кнопка "Зарегистрироваться".

2. Процедура "Аутентификация пользователя":

- Описание: Проверяет учетные данные пользователя и предоставляет доступ к системе.

- Взаимосвязи: Связана с функцией "Регистрация пользователя" и процедурой "Восстановление пароля".

- Элементы управления: Форма входа со следующими полями: адрес электронной почты, пароль и кнопка "Войти".

3. Процедура "Просмотр товаров":

- Описание: Позволяет клиентам просмотреть доступные товары.

- Взаимосвязи: Может быть связана с функцией "Добавить в корзину".

- Элементы управления: Список для выбора товара.

4. Функция "Добавить в корзину":

- Описание: Позволяет клиентам добавить товар в корзину.

- Взаимосвязи: Связана с функцией "Оформление заказа".

- Элементы управления: Кнопка "Добавить в корзину".

5. Функция "Оформление заказа":

- Описание: Позволяет клиентам оформить новый заказ.

- Взаимосвязи: Может быть связана с функцией "Текущие заказы".

- Элементы управления: Форма с полями для ввода данных о доставке, оплате и кнопка "Оформить".

6. Процедура "Текущие заказы":

- Описание: Позволяет клиентам просматривать содержимое заказов.

- Взаимосвязи: Связана с функцией "Оформление заказа".

- Элементы управления: Кнопка "Текущие заказы" в профиле каждого клиента.

Листинги модулей с комментариями и указанием элементов управления, которые инициируют их выполнение, должны быть представлены в документации или коде разрабатываемого веб-приложения. Это поможет разработчикам и пользователям лучше понять логику и взаимодействие между функциями и процедурами в приложении. На диаграмме вариантов использования представлен весь функционал пользователя и администратора.

Полный код программы, вместе с функциями, которые в данном подпункте были опущены в связи с аналогичностью или иными причинами, представлен в приложении А.

## Функциональное тестирование

Функциональное тестирование является важной частью процесса обеспечения качества программного обеспечения. Оно направлено на проверку функциональности системы с целью убедиться, что она соответствует требованиям и ожиданиям пользователей. Прежде чем перейти к разработке тест-кейсов, необходимо провести анализ требований, понять функциональность веб-приложения и определить основные сценарии использования. Это позволит сосредоточиться на наиболее важных и критических аспектах веб-приложения. Подходящие тест-кейсы должны быть разработаны для каждого сценария использования, чтобы проверить, что функции веб-приложения работают правильно и соответствуют ожиданиям пользователей. На таблицах 3.5.1-3.5.5 представлен набор тест-кейсов для программного средства.

Таблица 3.5.1 – Тест-кейс для функции авторизации пользователя в веб-приложении

|  |  |  |
| --- | --- | --- |
| Действие | Ожидаемый результат | Результат теста |
| Открыть веб-приложение «GuitarExtasy» | Веб-приложение «GuitarExtasy» открыто и доступно | Пройден |
| Нажать на кнопку «Профиль» | Отображается окно авторизации | Пройден |
| Ввести в поле «Логин» и «Пароль» | Отображение введенного логина и пароля в поле | Пройден |
| Нажать на кнопку «Войти» | Отображение профиля пользователя | Пройден |

Таблица 3.5.2 – Тест-кейс для функции добавления мастера

|  |  |  |
| --- | --- | --- |
| Действие | Ожидаемый результат | Результат теста |
| 1 | 2 | 3 |
| Открыть приложение «GuitarExtasy» | Веб-приложение «GuitarExtasy» открыто и доступно | Пройден |
| Авторизоваться в веб-приложении(роль администратора) | Пользователь авторизован, открыт профиль администратора | Пройден |
| Нажать на кнопку «Добавление мастера» | Отображение окна с формой | Пройден |
| Ввести в поле «Логин» данные «master\_002»,в поле «Пароль» данные «master\_002», в поле «ФИО» данные «Тихомиров Егор Витальевич» | Отображение введенного логина, пароля и ФИО в поле | Пройден |
| Нажать на кнопку «Добавить» | Отображение таблицы мастеров | Пройден |

Таблица 3.5.3 – Тест-кейс для функции добавления товара в корзину

|  |  |  |
| --- | --- | --- |
| Действие | Ожидаемый результат | Результат теста |
| Открыть приложение «GuitarExtasy» | Веб-приложение «GuitarExtasy» открыто и доступно | Пройден |
| Нажать на кнопку «Профиль» | Отображается окно авторизации | Пройден |
| Ввести в поле «Логин» данные «haribo» и в поле «Пароль» данные «92019» | Отображение введенного логина и пароля в поле | Пройден |
| Нажать на кнопку «Войти» | Отображение профиля клиента | Пройден |
| Нажать на кнопку «Каталог» | Отображение окна с каталогом товаров | Пройден |
| Выбрать товар из предложенных | Отображение информации товара | Пройден |
| Нажать на кнопку «Добавить в корзину» | Добавление товара в корзину | Пройден |

Таблица 3.5.4 – Тест-кейс для функции оформления заказа

|  |  |  |
| --- | --- | --- |
| Действие | Ожидаемый результат | Результат теста |
| Открыть приложение «GuitarExtasy» | Веб-приложение «GuitarExtasy» открыто и доступно | Пройден |
| Нажать на кнопку «Профиль» | Отображается окно авторизации | Пройден |
| Ввести в поле «Логин» данные «haribo» и в поле «Пароль» данные «92019» | Отображение введенного логина и пароля в поле | Пройден |
| Нажать на кнопку «Войти» | Отображение профиля клиента | Пройден |
| Нажать на кнопку «Каталог» | Отображение окна с каталогом товаров | Пройден |
| Выбрать товар из предложенных | Отображение информации товара | Пройден |
| Нажать на кнопку «Добавить в корзину» | Добавление товара в корзину | Пройден |
| Нажать на кнопку «Корзина» | Отображение корзины с товарами | Пройден |
| Нажать на кнопку «Оформить заказ» | Отображение модального окна с вводом доставки и оплаты | Пройден |

Таблица 3.5.5 – Тест-кейс для функции индивидуальная гитара

|  |  |  |
| --- | --- | --- |
| Действие | Ожидаемый результат | Результат теста |
| Открыть приложение «GuitarExtasy» | Веб-приложение «GuitarExtasy» открыто и доступно | Пройден |
| Нажать на кнопку «Профиль» | Отображается окно авторизации | Пройден |
| Ввести в поле «Логин» данные «haribo» и в поле «Пароль» данные «92019» | Отображение введенного логина и пароля в поле | Пройден |
| Нажать на кнопку «Войти» | Отображение профиля клиента | Пройден |
| Нажать на кнопку «Конструктор» | Отображение окна конструктора | Пройден |
| Выбрать из групп радиокнопки типа, материала верхней деки, материала нижней деки, цвета | Выбранные радиокнопки выделяются | Пройден |

# Применение

## Назначение веб-приложения

Название разработанного программного средства – «GuitarExtasy».

Назначение разрабатываемого веб-приложения для автоматизации процесса изготовления и продажи гитар заключается в облегчении и оптимизации процессов управления данными о клиентах, оформление заказов. Оно предоставляет мастерам и клиентам удобный доступ к необходимой информации и функциональности через веб-интерфейс.

Целью разработки данного веб-приложения является создание эффективного инструмента для интернет-магазина, который позволит автоматизировать и упростить процессы управления базой данных клиентов, оформлении заказов. Главной задачей является повышение эффективности работы магазина, улучшение обслуживания клиентов и уменьшение ручной работы персонала.

Для установки веб-приложения используется установщик «GuitarExtasy.exe».

Для запуска веб-приложение необходимо предусмотреть наличие SQL Server для базы данных. Затем пользователь может открыть веб-браузер и ввести URL-адрес, указывающий на главную страницу веб-приложения, чтобы получить доступ к функциональности и начать его использование.

Разработанное веб-приложение предназначено для автоматизации процесса изготовления и продажи гитар. Оно позволяет управлять базой данных клиентов, оформление заказов. Веб-приложение облегчает процессы ведения документации магазина, позволяет клиентам удобно заказывать товары, а мастерам эффективно организовывать рабочие процессы.

Класс задач, решаемых данной системой, включает:

* управление базой данных клиентов: добавление, редактирование и удаление информации о клиентах;
* ведение истории заказов клиентов: сохранение и обновление информации о заказах;
* разграничение прав доступа: обеспечение безопасности данных и ограничение доступа к конфиденциальной информации только авторизованным пользователям в соответствии с их ролями и полномочиями.

Область применения данного веб-приложения ограничена сферой продаж и предназначена для использования в магазинах. Оно предоставляет инструменты для автоматизации работы персонала и облегчения взаимодействия с клиентами. Оно служит вспомогательным средством для упрощения процессов управления данными.

## Программно-аппаратное обеспечение сервера и клиента

Для корректной работы веб-приложения для автоматизации процесса изготовления и продажи гитар, требуется следующее программно-аппаратное обеспечение для сервера и клиента.

Сервер:

* операционная система. Рекомендуется использовать Windows Server для обеспечения стабильности и безопасности работы сервера;
* процессор. Мощный многоядерный процессор для обработки большого количества запросов от клиентов;
* память. Достаточно оперативной памяти для обработки и хранения данных, рекомендуется минимум 8 ГБ;
* хранилище. Достаточное пространство для хранения базы данных, файлов и резервных копий;
* база данных. Установленный и настроенный SQL Server для хранения и управления данными клиентов, заказами и другой необходимой информацией;
* интернет-соединение. Стабильное и быстрое соединение для обеспечения доступа клиентов к приложению.

Клиент:

* операционная система. Может быть Windows, macOS или Linux, совместимая с браузером, который будет использоваться для доступа к веб-приложению;
* браузер. Рекомендуется использовать последнюю версию современного браузера, такого как Google Chrome, Mozilla Firefox или Microsoft Edge;
* процессор и память. Достаточно мощный процессор и оперативная память для обеспечения плавной работы веб-приложения;
* интернет-соединение. Стабильное и быстрое соединение для обеспечения доступа к серверу и передачи данных.

Требования к программно-аппаратному обеспечению обусловлены необходимостью обработки и хранения большого объема данных, обеспечения безопасности и стабильности работы приложения, а также удобства использования для клиентов. Правильная конфигурация сервера и клиента позволит обеспечить эффективность и надежность работы веб-приложения для медицинской клиники.

# Заключение

Веб-приложение для автоматизации процесса изготовления и продажи гитар было разработано с использованием современных веб-технологий, таких как HTML, CSS, JavaScript и C#. Для разработки могли быть использованы такие инструменты и фреймворки, как ASP.NET, Entity Framework и SQL Server.

Веб-приложение выполняет все поставленные задачи. Оно предоставляет функционал для управления базой данных клиентов, включая добавление, редактирование и удаление информации о клиентах, их персональных данных и истории заказов. Также реализована возможность клиентам заказывать индивидуальные гитары. Приложение также позволяет назначать мастерам заказы, просматривать и управлять заказами клиентов.

Преимущества разработанного веб-приложения:

* Автоматизация и упрощение работы магазина: Веб-приложение позволяет управлять и хранить информацию о клиентах и заказах, сокращая необходимость вручную вести и обрабатывать бумажные документы;
* Улучшение доступности и удобства для клиентов: Клиенты могут оформлять заказы и приобретать индивидуальные гитары;
* Разграничение прав доступа: Приложение предоставляет возможность управлять доступом к базе данных, обеспечивая безопасность и конфиденциальность информации.

Недостатки разработанного веб-приложения:

* Зависимость от интернет-соединения: для использования веб-приложения требуется стабильное и быстрое интернет-соединение, что может быть проблематично в некоторых областях или в случае сбоев в сети.

В целом, разработанное веб-приложение предоставляет значительные преимущества для магазина, автоматизируя и упрощая процессы работы. Однако, необходимо учитывать некоторые недостатки и затраты на техническую поддержку и обеспечение стабильного интернет-соединения.

# Список использованных источников

1. Общие сведения о C# [Электронный ресурс]. – Режим доступа: <https://learn.microsoft.com/ru-ru/dotnet/csharp/>. - Дата доступа: 26.05.2024.
2. Общие сведения об ASP.NET Core Razor Pages [Электронный ресурс]. – Режим доступа: <https://learn.microsoft.com/ru-ru/aspnet/core/razor-pages/>. - Дата доступа: 10.06.2024.
3. Общие сведения о Entity Framework Core [Электронный ресурс]. – Режим доступа: <https://learn.microsoft.com/ru-ru/ef/core/>. – Дата доступа: 28.05.2024.
4. Общие сведения об Identity в ASP.NET Core [Электронный ресурс]. – Режим доступа: <https://learn.microsoft.com/ru-ru/aspnet/core/security/authentication/identity?view=aspnetcore-6.0>. – Дата доступа: 28.05.2024.
5. Общие сведения о Bootstrap для ASP.NET Core [Электронный ресурс]. – Режим доступа: <https://learn.microsoft.com/ru-ru/aspnet/core/client-side/bootstrap?view=aspnetcore-6.0>. - Дата доступа: 26.05.2024.
6. Общие сведения о SQL Server для веб-приложений [Электронный ресурс]. – Режим доступа: <https://learn.microsoft.com/ru-ru/sql/connect/ado-net/sql/>. – Дата доступа: 28.05.2024.
7. Общие сведения о Visual Studio для веб-разработки [Электронный ресурс]. – Режим доступа: <https://visualstudio.microsoft.com/ru/vs/features/>. – Дата доступа: 10.06.2024.
8. Общие сведения о развертывании веб-приложений на Azure [Электронный ресурс]. – Режим доступа: <https://learn.microsoft.com/ru-ru/aspnet/core/host-and-deploy/azure-apps/>. – Дата доступа: 28.05.2024.

# Приложение А

**(обязательное)  
Текст программных модулей**

//код страницы корзины

//ClientBascket.cshtml

@page

@model GuitarExtasy.Pages.ClientBascketModel

@{

ViewData["Title"] = "Корзина";

double? totalPrice = 0;

}

<style>

.card {

border: none;

border-radius: 8px;

box-shadow: 0 0.5rem 1rem rgba(0, 0, 0, 0.15);

}

.card-img-top {

border-top-left-radius: 8px;

border-top-right-radius: 8px;

height: 200px;

object-fit: cover;

}

.card-title {

font-size: 1.1rem;

font-weight: bold;

.card-text {

font-size: 0.9rem;

color: #555;

}

.btn-outline-secondary {

border-color: #ccc;

color: #555;

}

.btn-outline-secondary:hover {

background-color: #f5f5f5;

}

.btn-danger {

background-color: #dc3545;

border-color: #dc3545;}

.btn-danger:hover {

background-color: #c82333;

border-color: #bd2130;

}

.btn-primary {

background-color: #007bff;

border-color: #007bff;

justify-content: center;

}

.btn-primary:hover {

background-color: #0056b3;

border-color: #004a99;

}

.row.justify-content-start {

margin-left: 5%;

width: 90%;

}

.row.justify-content-center {

margin-left: 30%;

width: 40%;

.h5 {

font-size: 1.25rem;

font-weight: 500;

}

</style>

<h1>Корзина</h1>

@if (Model.Carts == null)

{

<div class="row justify-content-center">

<div class="col-md-6">

<div class="card shadow-sm">

<div class="card-body">

<p class="h5 text-center">Корзина пуста</p>

</div>

</div>

</div>

</div>

}

else

{

<div class="row justify-content-start">

@foreach (var item in Model.Carts)

{

<div class="col-md-3 mb-4">

<div class="card shadow-sm">

<img src="@item.Products.Image" class="card-img-top" alt="@item.Products.Name">

<div class="card-body">

<h5 class="card-title">@item.Products.Name</h5>

<p class="card-text">Количество: @item.Quantity</p>

<p class="card-text">Цена: @item.Products.Price Br</p>

<form method="post" asp-page-handler="RemoveFromCart">

<input type="hidden" name="id" value="@item.Id" />

<input type="submit" value="Удалить" class="btn btn-danger" />

</form>

</div>

</div>

</div>

totalPrice += item.Products.Price \* item.Quantity;

}

</div>

<div class="row justify-content-center">

<div class="col-md-6">

<div class="card shadow-sm">

<div class="card-body">

<dl class="row">

<dt class="col-sm-6 h5">Итого:</dt>

<dd class="col-sm-6 h5">@totalPrice Br</dd>

</dl>

<form method="post" asp-page-handler="Checkout">

<button type="button" class="btn btn-primary" data-toggle="modal" data-target="#checkoutModal">Оформить заказ</button>

</form>

</div>

</div>

</div>

</div>

}

<div class="modal fade" id="checkoutModal" tabindex="-1" aria-labelledby="checkoutModalLabel" aria-hidden="true">

<div class="modal-dialog">

<div class="modal-content">

<div class="modal-header">

<h5 class="modal-title" id="checkoutModalLabel">Оформление заказа</h5>

<button type="button" class="close" data-dismiss="modal" aria-label="Close">

<span aria-hidden="true">&times;</span>

</button>

</div>

<div class="modal-body">

<form id="checkoutForm" method="post" asp-page-handler="Checkout">

<input type="hidden" name="totalPrice" value="@totalPrice" /

<div class="form-group">

<label for="deliveryOption">Выберите способ получения:</label>

<select class="form-control" id="deliveryOption" name="deliveryOption" onchange="toggleDeliveryInput()">

<option value="pickup">Самовывоз</option>

<option value="delivery">Доставка</option>

</select>

</div>

<div class="form-group" id="deliveryAddressGroup" style="display: none;">

<label for="deliveryAddress">Адрес доставки:</label>

<input type="text" class="form-control" id="deliveryAddress" name="deliveryAddress" />

</div>

<div class="form-group">

<label for="paymentMethod">Выберите способ оплаты:</label>

<select class="form-control" id="paymentMethod" name="paymentMethod" onchange="togglePaymentInput()">

<option value="cash">Наличные</option>

<option value="card">Карта</option>

</select>

</div>

<div class="form-group" id="cardDetailsGroup" style="display: none;">

<label for="cardNumber">Номер карты:</label>

<input type="text" class="form-control" id="cardNumber" name="cardNumber" />

<label for="cardExpiryDate">Срок действия:</label>

<input type="text" class="form-control" id="cardExpiryDate" name="cardExpiryDate" />

<label for="cardCvc">CVC:</label>

<input type="text" class="form-control" id="cardCvc" name="cardCvc" />

</div>

<button type="submit" class="btn btn-success">Оплатить</button>

</form>

</div>

</div>

</div>

</div>

<script>

function toggleDeliveryInput() {

var deliveryOption = document.getElementById("deliveryOption").value;

var deliveryAddressGroup = document.getElementById("deliveryAddressGroup")

if (deliveryOption === "delivery") {

deliveryAddressGroup.style.display = "block";

} else {

deliveryAddressGroup.style.display = "none";

}

}

function togglePaymentInput() {

var paymentMethod = document.getElementById("paymentMethod").value;

var cardDetailsGroup = document.getElementById("cardDetailsGroup");

if (paymentMethod === "card") {

cardDetailsGroup.style.display = "block";

} else {

cardDetailsGroup.style.display = "none";

}

}

document.getElementById("checkoutForm").addEventListener("submit", function(event) {

event.preventDefault();

alert("Заказ успешно оформлен!");

event.target.submit();

});

</script>

//ClientBascket.cshtml.cs

using GuitarExtasy.Models;

using GuitarExtasy.Data;

using Microsoft.AspNetCore.Mvc;

using Microsoft.AspNetCore.Mvc.RazorPages;

using System.Collections.Generic;

using System.Linq;

using Microsoft.EntityFrameworkCore;

using System;

using System.Threading.Tasks;

using System.Security.Claims;

namespace GuitarExtasy.Pages

{

public class ClientBascketModel : PageModel

{

private readonly GuitarExtasyContext \_context;

public ClientBascketModel(GuitarExtasyContext context)

{

\_context = context;

}

public List<GuitarExtasy.Models.OrderItems> OrderItems { get; set; }

public GuitarExtasy.Models.Orders Order { get; set; }

public GuitarExtasy.Models.Clients Client { get; set; }

public List<GuitarExtasy.Models.Carts> Carts { get; set; }

public List<GuitarExtasy.Models.Products> Product { get; set; }

public async Task OnGetAsync()

{

var userId = User.FindFirstValue(ClaimTypes.Name);

var client = await \_context.Clients.FirstOrDefaultAsync(c => c.Login == userId);

Carts = await \_context.Carts

.Where(c => c.Clients.Id == client.Id)

.Include(c => c.Clients)

.Include(c => c.Products)

.ToListAsync();

}

public async Task<IActionResult> OnPostRemoveFromCartAsync(int id)

{

var cartItem = await \_context.Carts.FindAsync(id);

if (cartItem != null)

{

\_context.Carts.Remove(cartItem);

await \_context.SaveChangesAsync();

}

return RedirectToPage();

}

public async Task<IActionResult> OnPostCheckoutAsync(double totalPrice)

{

var userId = User.FindFirstValue(ClaimTypes.Name);

if (userId == null)

{

return BadRequest("Пользователь не найден.");

}

var client = await \_context.Clients.FirstOrDefaultAsync(c => c.Login == userId);

if (client == null)

{

return NotFound("Клиент не найден.");

}

var newOrder = new GuitarExtasy.Models.Orders

{

ClientsId = client.Id,

Amount = totalPrice,

OrderDate = DateTime.Now,

StatusId = 1

};

\_context.Orders.Add(newOrder);

await \_context.SaveChangesAsync();

var orderId = newOrder.Id;

var cartItems = await \_context.Carts

.Where(c => c.ClientsId == client.Id)

.Include(c => c.Products)

.ToListAsync();

if (cartItems.Count == 0)

{

return BadRequest("Корзина пуста.");

}

foreach (var cartItem in cartItems)

{

if (cartItem.Products == null)

{

Console.WriteLine($"Продукт с ID {cartItem.ProductsId} не найден в базе данных.");

return BadRequest("Некоторые продукты не найдены.");

}

var orderItem = new GuitarExtasy.Models.OrderItems

{

OrdersId = orderId,

ProductsId = cartItem.ProductsId,

Quantity = cartItem.Quantity,

Price = cartItem.Products.Price \* cartItem.Quantity

};

\_context.OrderItems.Add(orderItem);

var product = await \_context.Products.FindAsync(cartItem.ProductsId);

if (product != null)

{

product.Availability -= cartItem.Quantity;

\_context.Products.Update(product);

}

}

\_context.Carts.RemoveRange(cartItems);

await \_context.SaveChangesAsync();

return RedirectToPage("/ClientBascket");

}

}

}

# Приложение Б

**(справочное)  
Результаты работы программы**

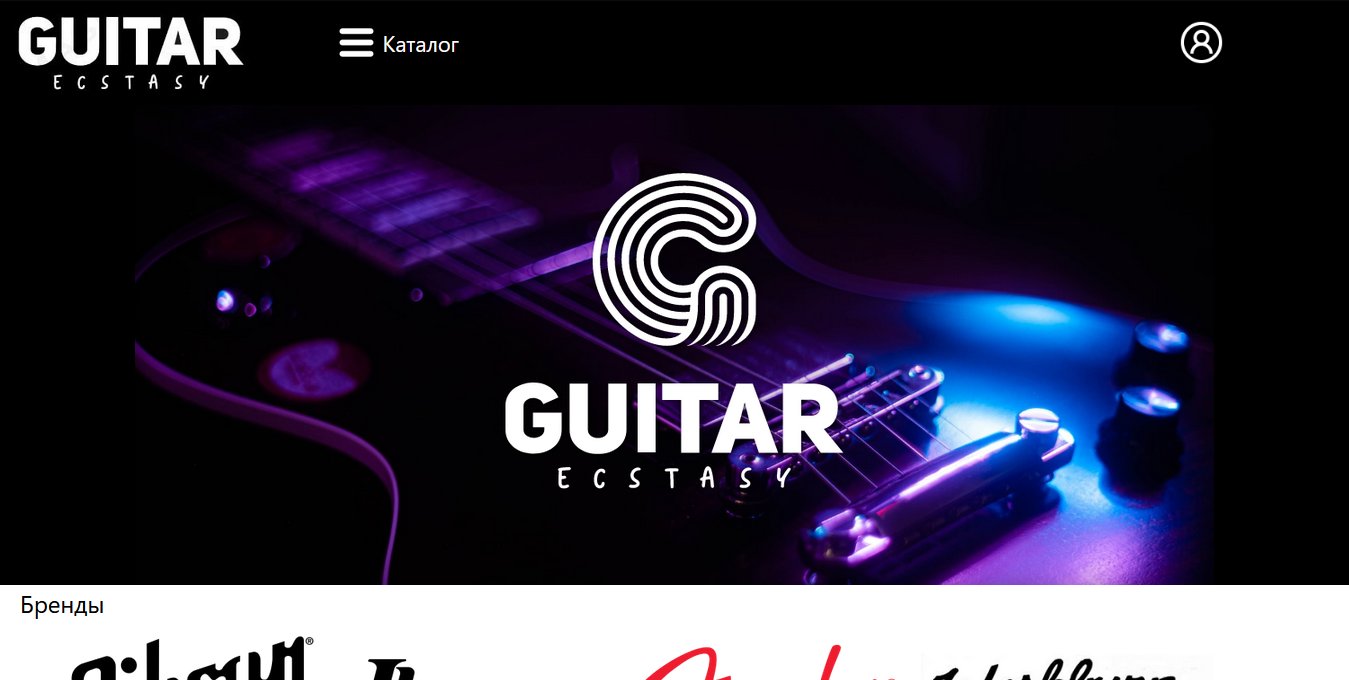


Рисунок Б.1 –Главное окно до авторизации

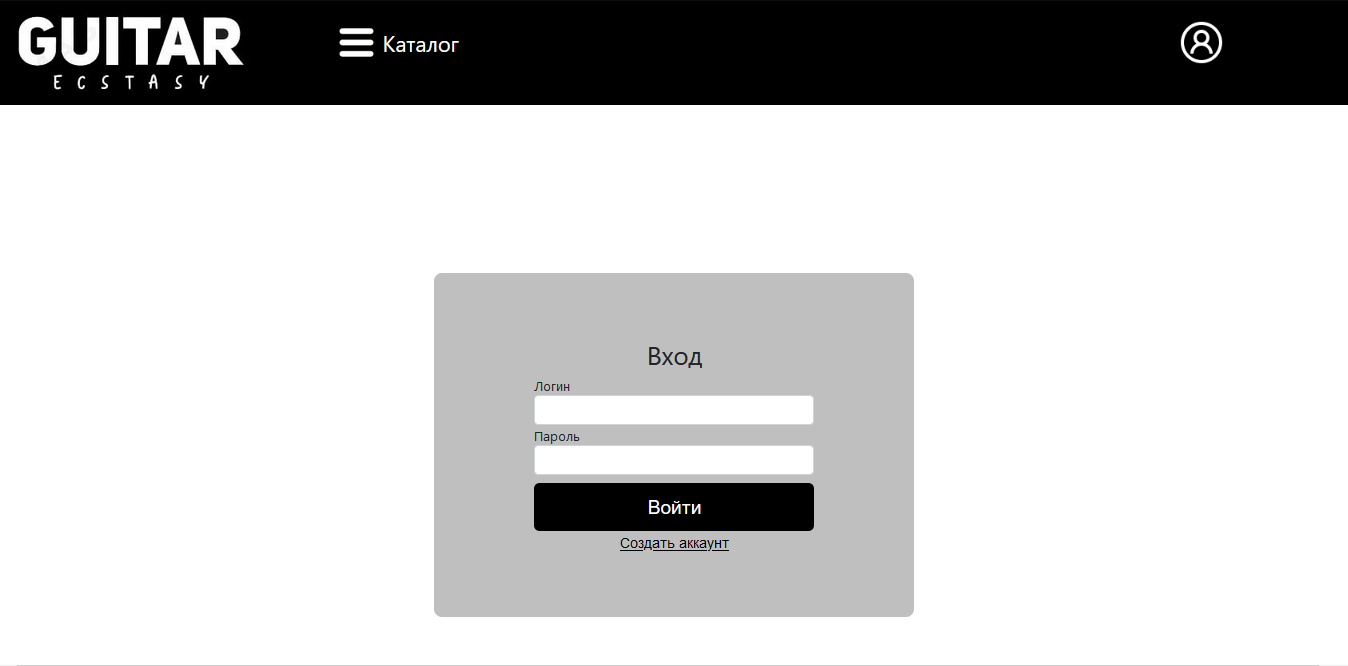


Рисунок Б.2 –Окно авторизации

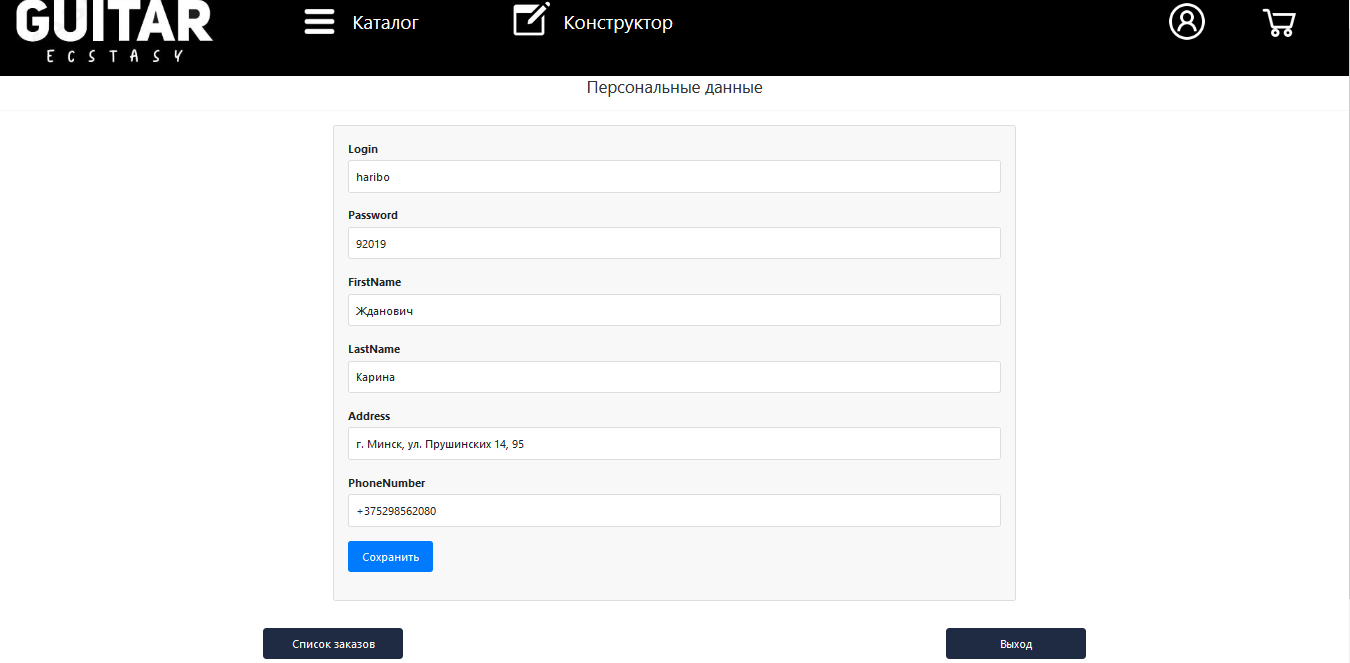


Рисунок Б.3 – Профиль клиента

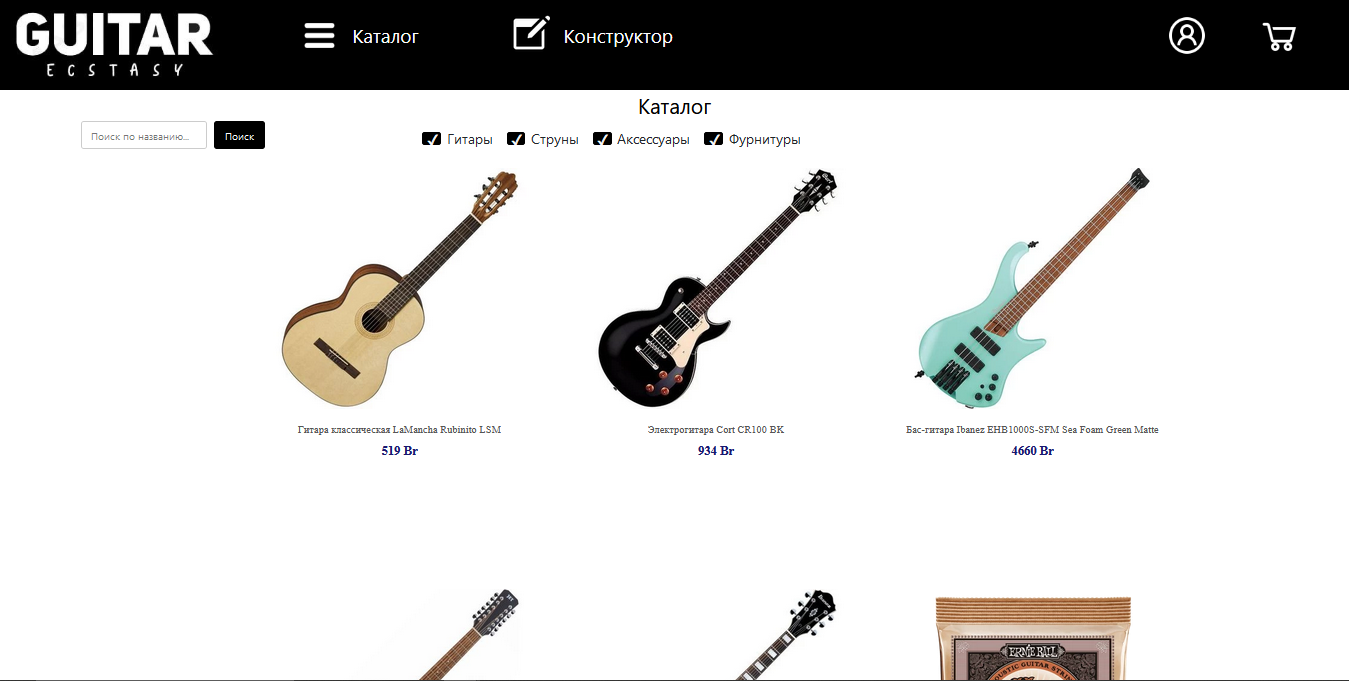


Рисунок Б.4 – Окно каталога



Рисунок Б.5 – Окно конструктора

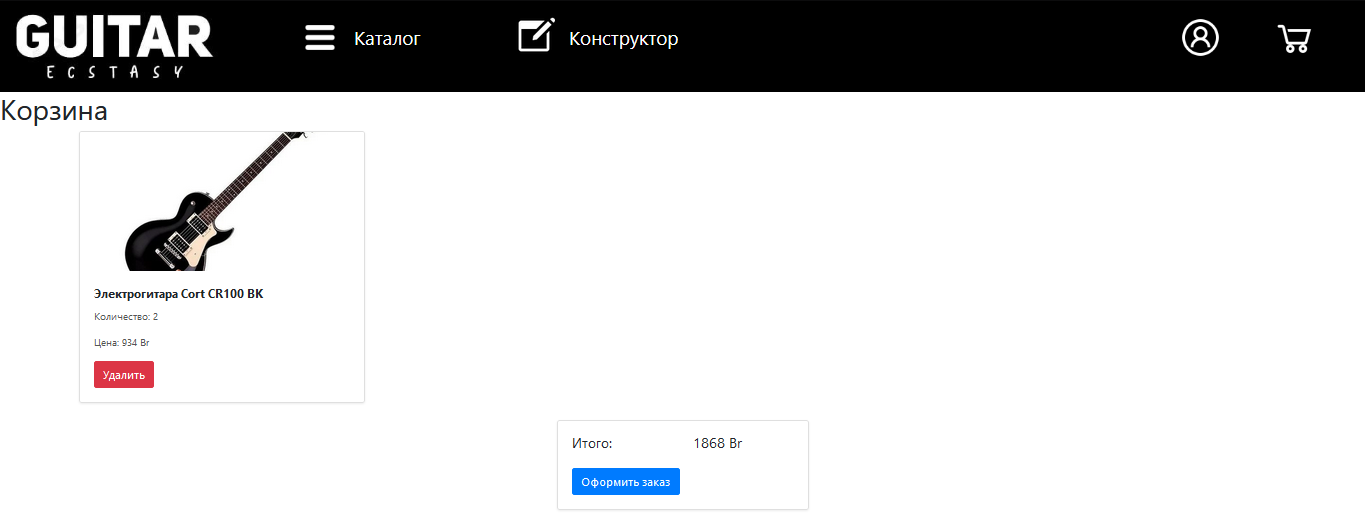


Рисунок Б.6 – Окно корзины клиента

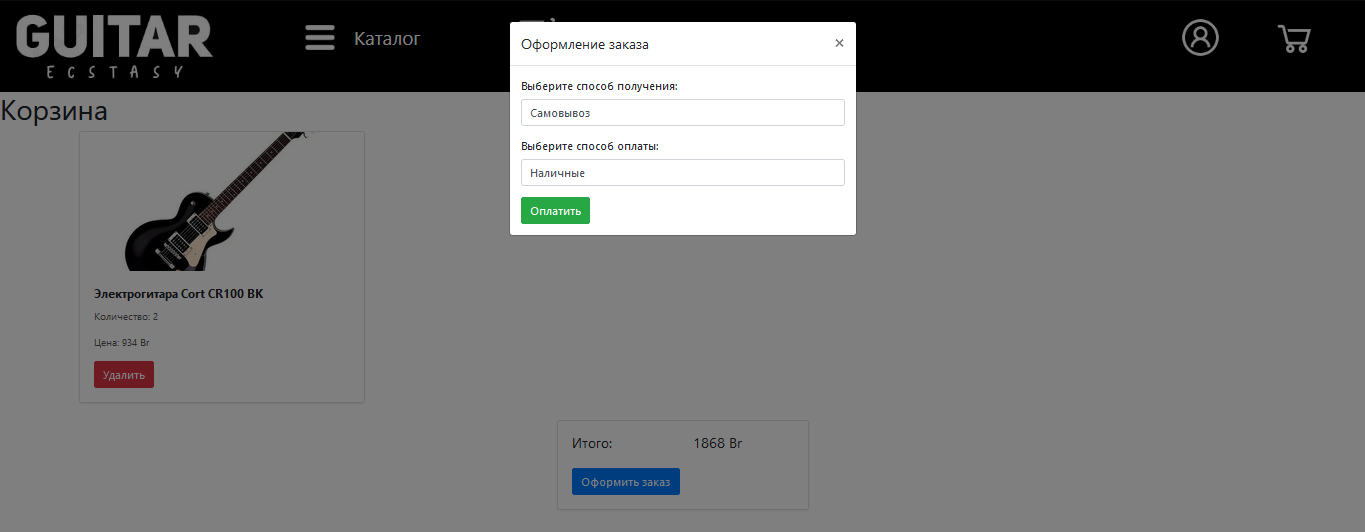


Рисунок Б.7 – Модальное окно оформления заказа

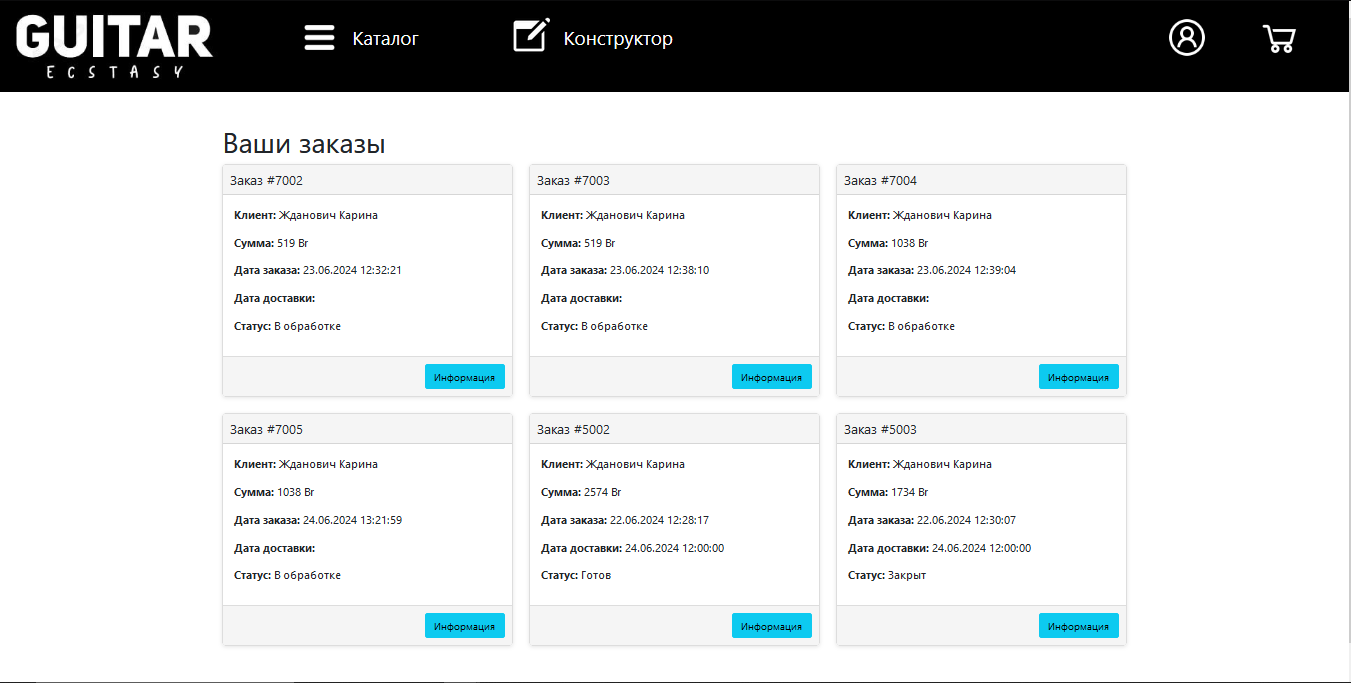


Рисунок Б.8 – Окно текущих заказов клиента

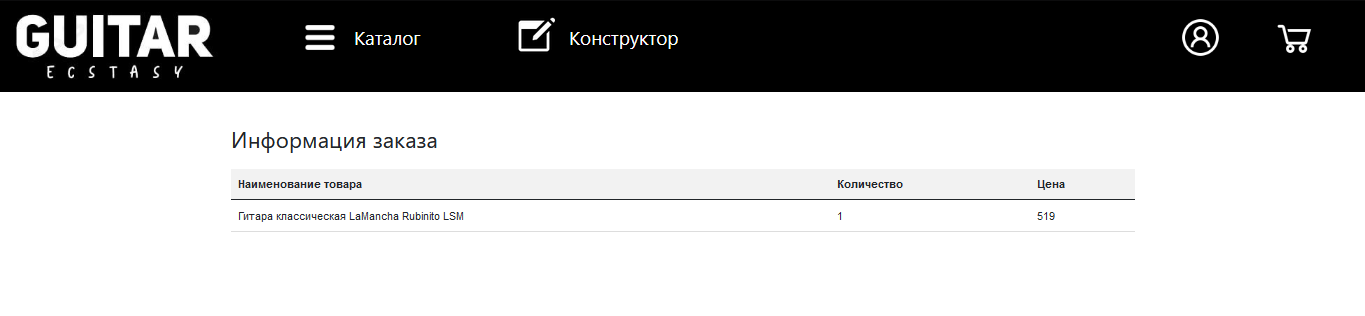


Рисунок Б.9 – Окно с информацией заказа

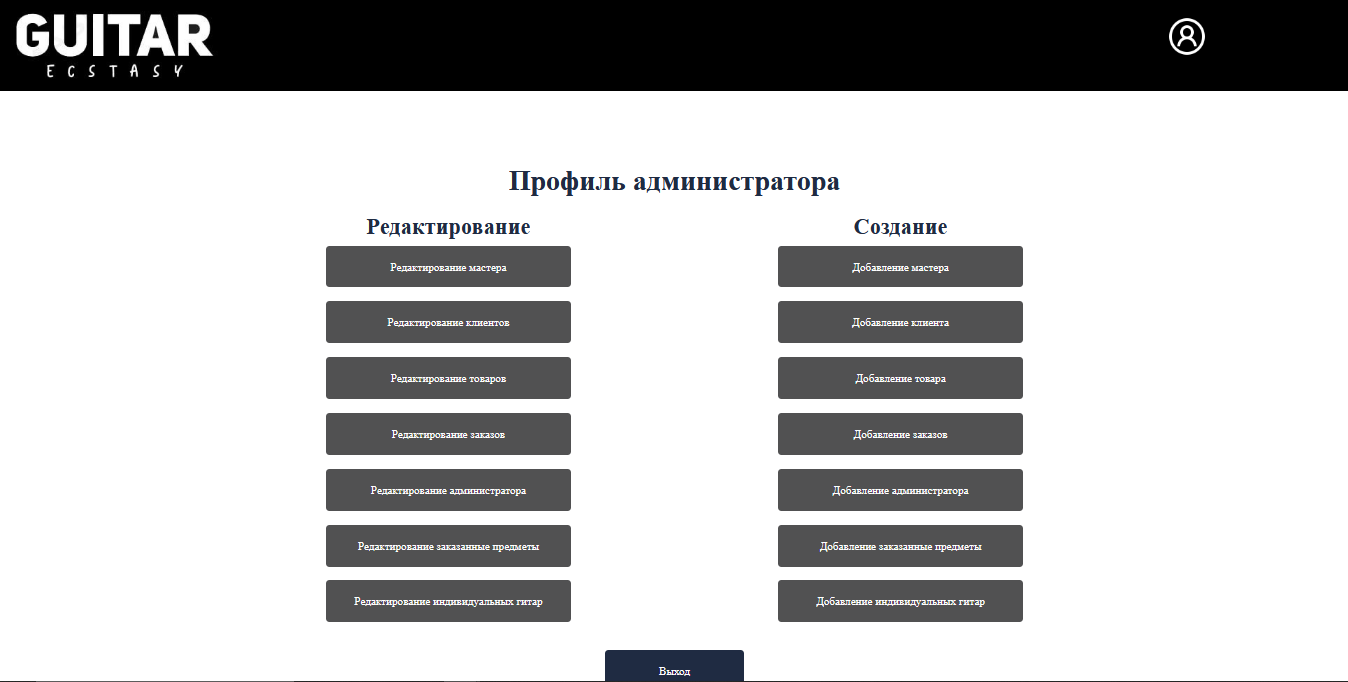


Рисунок Б.10 – Профиль администратора

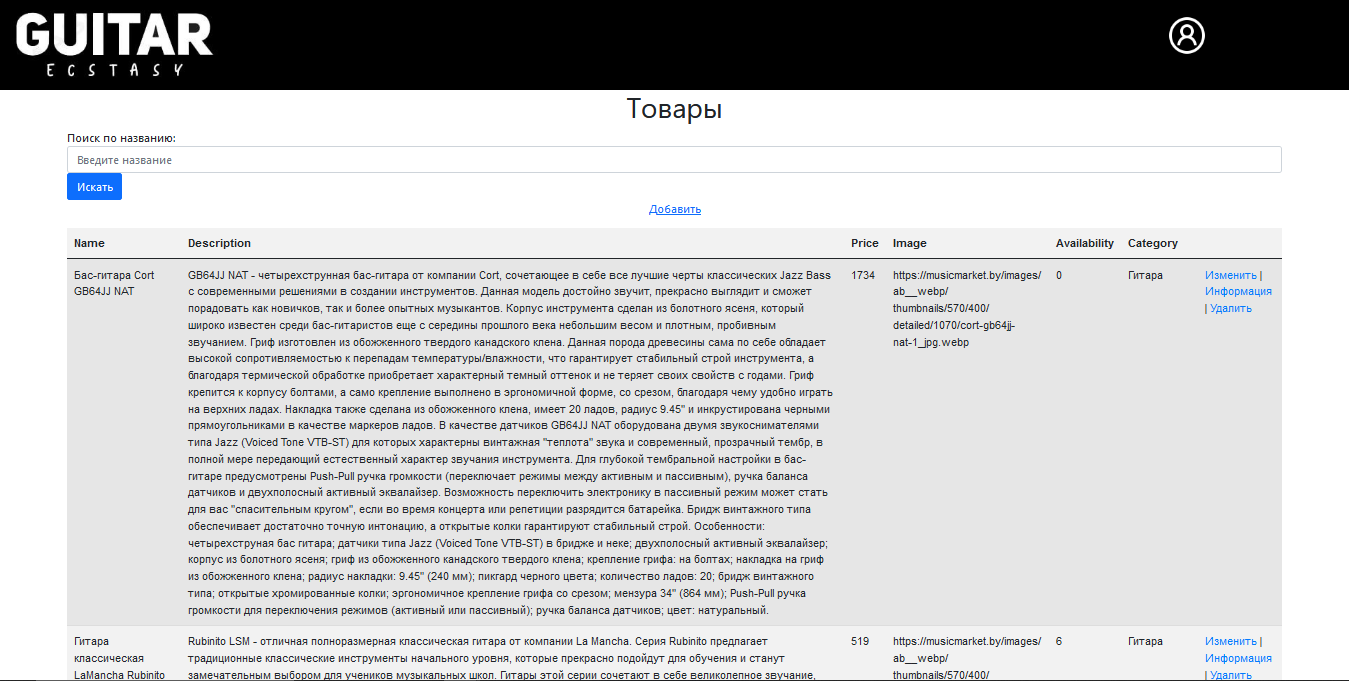


Рисунок Б.11 – Окно с таблицей товаров

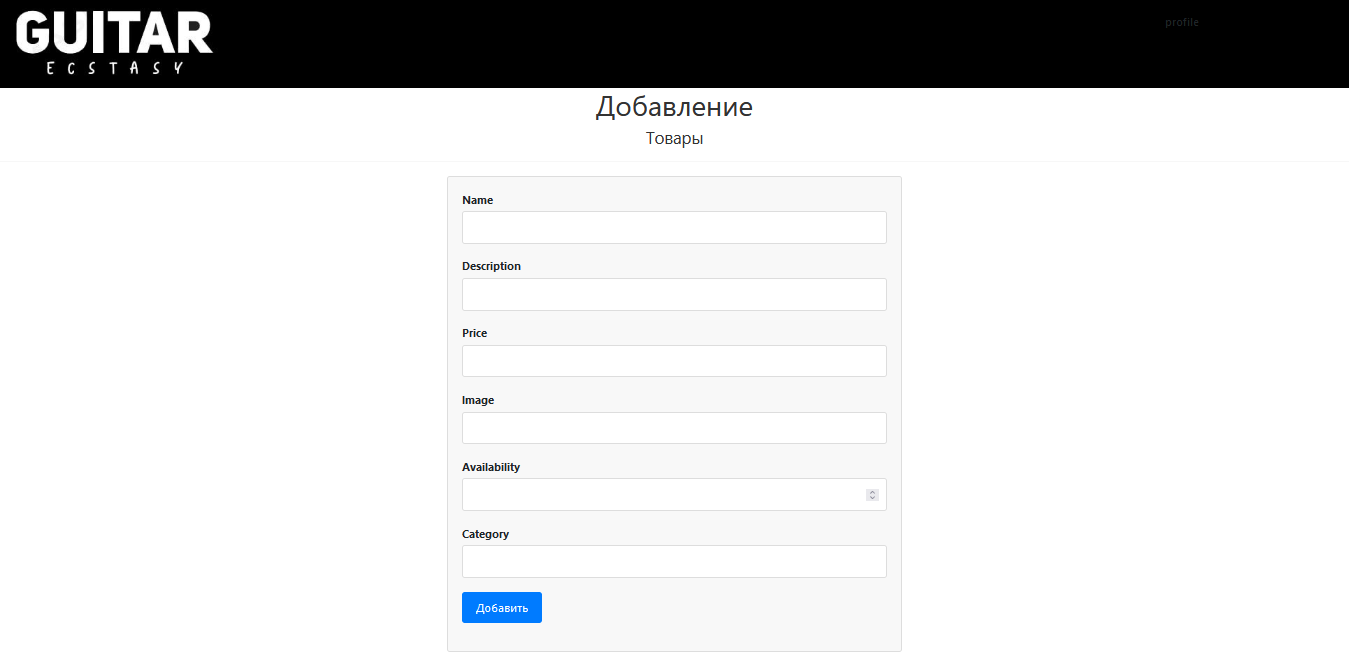


Рисунок Б.12 – Окно для добавления товара

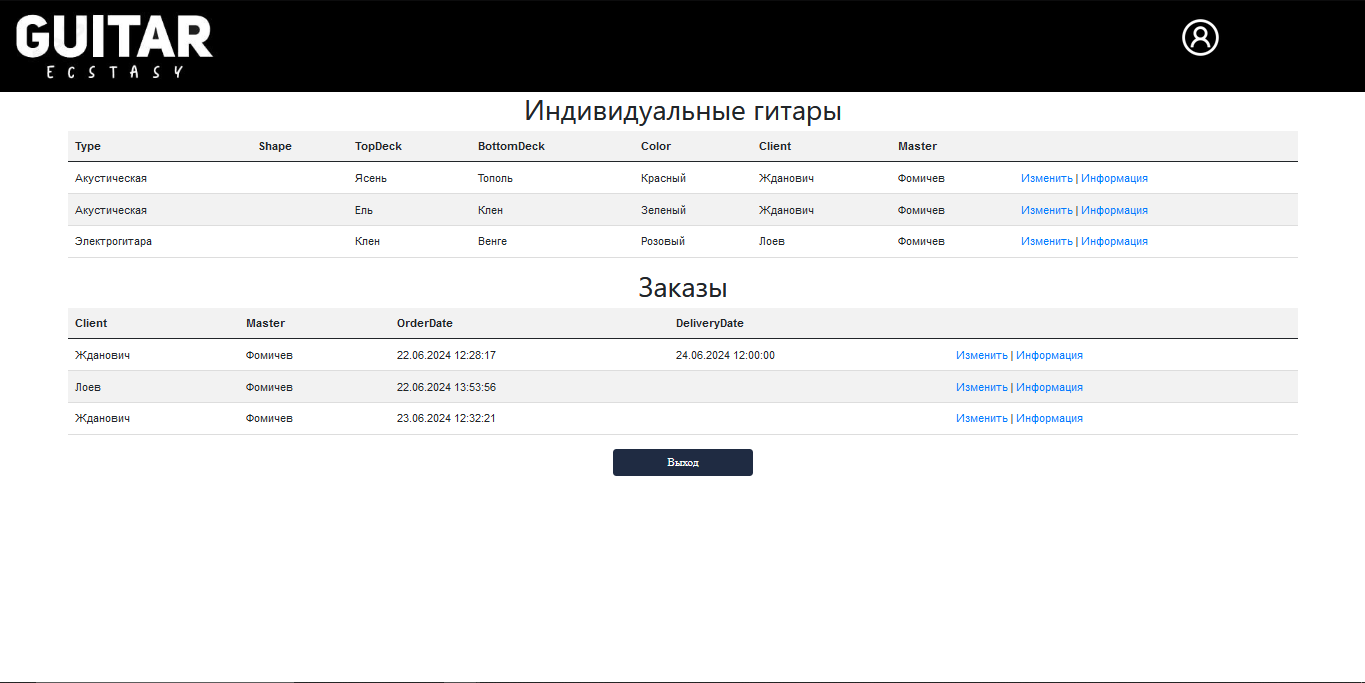
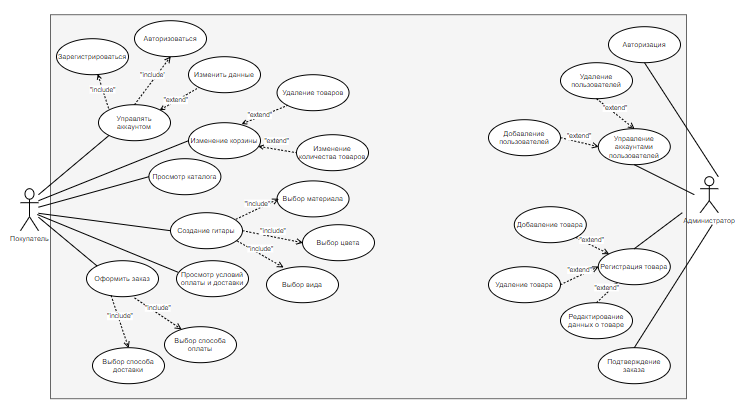


Рисунок Б.13 – Профиль мастера



КБП

У

Т. Контр.

Разраб. Разраб.

Жданович К.С.

Провер. Провер.

Кривошеина А.В.

Реценз.

Утверд.

Н. Контр.

№ докум.

Подпись

Дата

Изм.

Лист

Масса

Лит.

Масштаб

Лист 1

Листов 5

Инв.№подл.

Подп. и дата

Взам.инв.№

Инв.№дубл.

Подп. и дата

КП Т.197005.401 ГЧ Т.ХХХХХХ.401 ГЧ

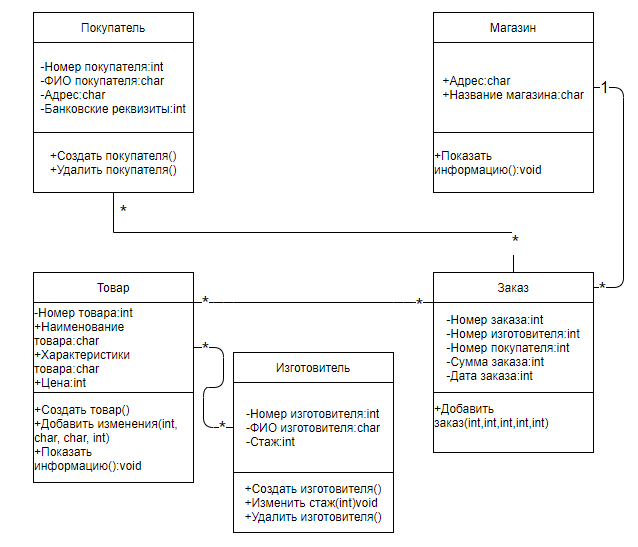
*КП Т.197005.401 ГЧ*

*Диаграмма вариантов использования*

*Создание веб-приложения для автоматизации изготовления и продажи гитар и сопутствующих товаров*

*Создание веб-приложения для автоматизации изготовления и продажи гитар и сопутствующих товаров*

*Диаграмма вариантов использования*



*Создание веб-приложения для автоматизации изготовления и продажи гитар и сопутствующих товаров*

КБП

У

Т. Контр.

Разраб. Разраб.

Жданович К.С.

Провер. Провер.

Кривошеина А.В.

Реценз.

Утверд.

Н. Контр.

№ докум.

Подпись

Дата

Изм.

Лист

Масса

Лит.

Масштаб

Лист 2

Листов 5

Инв.№подл.

Подп. и дата

Взам.инв.№

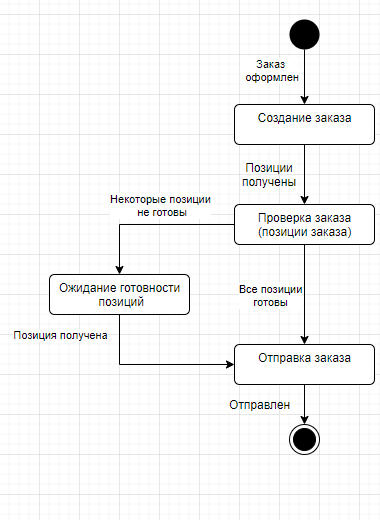
Инв.№дубл.

Подп. и дата

КП Т.197005.401 ГЧ Т.ХХХХХХ.401 ГЧ

*КП Т.197005.401 ГЧ*

*Диаграмма классов*



КБП

У

Т. Контр.

Разраб. Разраб.

Жданович К.С.

Провер. Провер.

Кривошеина А.В.

Реценз.

Утверд.

Н. Контр.

№ докум.

Подпись

Дата

Изм.

Лист

Масса

Лит.

Масштаб

Лист 3

Листов 5

Инв.№подл.

Подп. и дата

Взам.инв.№

Инв.№дубл.

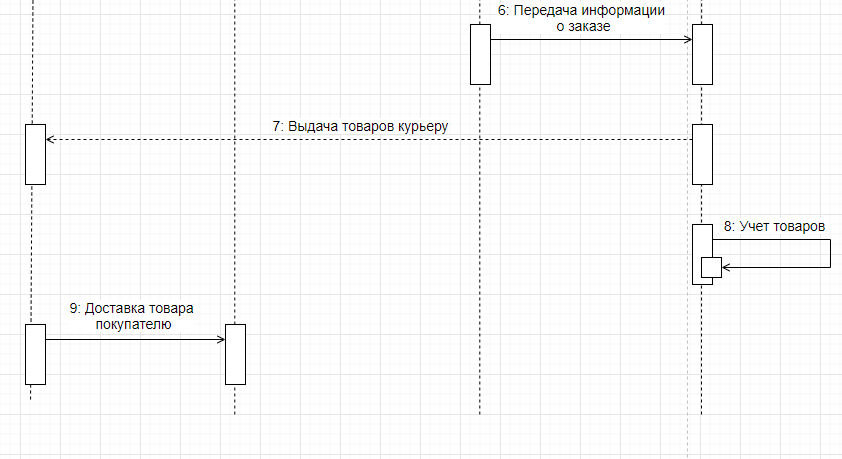
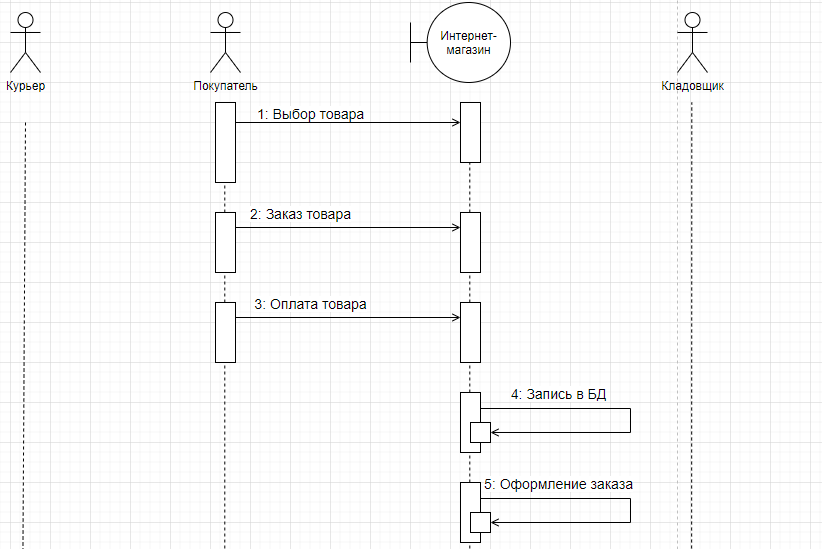
Подп. и дата

КП Т.197005.401 ГЧ Т.ХХХХХХ.401 ГЧ

*КП Т.197005.401 ГЧ*

*Диаграмма состояний (для функции оформления заказа)*

*Создание веб-приложения для автоматизации изготовления и продажи гитар и сопутствующих товаров*



*Диаграмма последовательности*

*Создание веб-приложения для автоматизации изготовления и продажи гитар и сопутствующих товаров*

КБП

У

Т. Контр.

Разраб. Разраб.

Жданович К.С.

Провер. Провер.

Кривошеина А.В.

Реценз.

Утверд.

Н. Контр.

№ докум.

Подпись

Дата

Изм.

Лист

Масса

Лит.

Масштаб

Лист 4

Листов 5

Инв.№подл.

Подп. и дата

Взам.инв.№

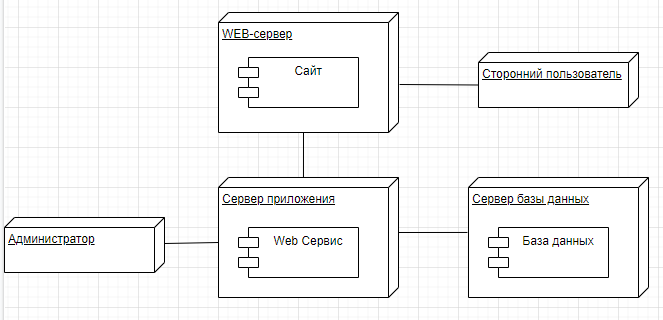
Инв.№дубл.

Подп. и дата

КП Т.197005.401 ГЧ Т.ХХХХХХ.401 ГЧ

*КП Т.197005.401 ГЧ*

*КП Т.197005.401 ГЧ*



*Создание веб-приложения для автоматизации изготовления и продажи гитар и сопутствующих товаров*

КБП

У

Т. Контр.

Разраб. Разраб.

Жданович К.С.

Провер. Провер.

Кривошеина А.В.

Реценз.

Утверд.

Н. Контр.

№ докум.

Подпись

Дата

Изм.

Лист

Масса

Лит.

Масштаб

Лист 5

Листов 5

Инв.№подл.

Подп. и дата

Взам.инв.№

Инв.№дубл.

Подп. и дата

КП Т.197005.401 ГЧ Т.ХХХХХХ.401 ГЧ

*Диаграмма развёртывания*

**Удостоверяющий лист**

электронного документа – курсовой проект

Создание веб-приложения для автоматизации изготовления и продажи гитар

и сопутствующих товаров

Тема

Обозначение \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_КП Т.197005.401\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Кривошеина А.В.

Жданович К.С.

Разработчик \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Руководитель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Ф.И.О.) (Ф.И.О.)

Подписи лиц, ответственных за разработку электронного документа

|  |  |  |
| --- | --- | --- |
| Состав электронного документа | Разработчик | Руководитель |
| Пояснительная записка (на бумажном носителе формата А4), Жданович\_Т-197\_ПЗ.docх |  |  |
| ГЧ, ДВИ.docx |  |  |
| ГЧ, Классов.docx |  |  |
| ГЧ, Последовательности.docx |  |  |
| ГЧ, Состояний.docx |  |  |
| ГЧ, Развёртывания.docx |  |  |
| Папка с проектом «GuitarExtasy» |  |  |
| Установочный пакет программного средства «GuitarExtasy.exe» |  |  |
| Тип носителя: диск |  |  |

**Этикетка**

для курсовых проектов

**Курсовой проект**

Создание веб-приложения для автоматизации изготовления и продажи гитар и сопутствующих товаров»

Тема «

КП Т.197005.401

Разработан

Утвержден \_\_\_\_\_

Руководитель: Кривошеина А.В.

Технические средства: Компьютер с процессором от 1 ГГц, ОЗУ от 2 ГБ, наличие интернет-соединения, монитор, мышь, клавиатура

Программные средства: EntityFramework.dll Microsoft Office, Mozilla Firefox, .NET Framework 4.8

**Состав документа:**

Пояснительная записка –Жданович\_Т-197\_ПЗ.docх

Графическая часть – ДВИ.docx, Последовательности.docx, Классов.docx, Состояний.docx, Развёртывания.docx

Папка с проектом – GuitarExtasy

Установочный пакет программного средства – GuitarExtasy.exe

Сведения о защите информации: пароль отсутствует