Министерство образования Московской области

ГБПОУ МО «Колледж «Коломна»

09.02.07

К защите допускается

зам. директора по УР

\_\_\_\_\_\_\_Э.Б. Ромашкина

«\_\_\_\_»\_\_\_\_\_\_\_ 20\_\_\_г.

ДИПЛОМНЫЙ ПРОЕКТ

Тема: Проектирование и внедрение ИТ-системы для автоматизации продаж музыкального оборудования

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

ККОО.ДП2188.000

|  |  |  |
| --- | --- | --- |
| Студент | «\_\_\_\_»\_\_\_\_\_\_\_\_\_2025 г. | Михеев В. А. |
| Руководитель проекта | «\_\_\_\_»\_\_\_\_\_\_\_\_\_2025 г. | Бунаков П. Ю. |
| Консультант по экономическому разделу | «\_\_\_\_»\_\_\_\_\_\_\_\_\_2025 г. | Караваев А.В. |
| Нормоконтролер | «\_\_\_\_»\_\_\_\_\_\_\_\_\_2025 г. | Грушникова Т.Н. |
| Рецензент | «\_\_\_\_»\_\_\_\_\_\_\_\_\_2025 г. | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

Дата защиты \_\_\_\_\_\_\_\_\_\_\_\_ Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_

2025

Министерство образования Московской области

Государственное бюджетное профессиональное образовательное учреждение Московской области «Колледж «Коломна»

|  |  |
| --- | --- |
|  | «Утверждаю»  Зам. директора по УР  \_\_\_\_\_\_\_\_\_\_ Э.Б. Ромашкина  «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_ 2025 г. |

ЗАДАНИЕ

на дипломный проект по специальности 09.02.07 «Информационные системы и программирование» студенту очного отделения, группы 42112

Михееву Владиславу Артемовичу

Тема проекта Проектирование и внедрение ИТ-системы для автоматизации продаж музыкального оборудования

1. Постановка задачи:

Разработать веб-приложение и веб-сервис API (далее Web API), реализующие концепцию автоматизации операций по покупке музыкальных инструментов.

Web API сервис должен решать следующие задачи:

* Управление ассортиментом приложения, добавление новых позиций и снятия с продаж прошлых, а также внесение корректировок в характеристики товаров;
* Отслеживание выполнения заказов и подтверждение его выполнения;
* Подтверждение актуальности данных пользователей и управление ролями пользователей;
* Отслеживания заказов, совершенных конкретным пользователем;
* Создание новых категорий товаров;

Веб-приложение должно решать следующие задачи:

* Регистрация новых пользователей в системе;
* Возможность оформить заказ;
* Возможность отследить выполнение заказа;

Интерфейс для Веб-приложения и Web API должен быть интуитивно понятным и учитывать уровень развития сервисов конкурентов.

В реализованном проекте должны присутствовать следующие формы для просмотра необходимой информации пользователем:

В Web API:

* Окно редактирования: добавление продукта, исключение продукта, корректировка технических характеристик;
* Окно редактирования: добавление/исключение категорий товаров;
* Окно редактирования: завершения заказа;
* Окно редактирования завершения доставки;

В веб-приложении:

* Информация о товаре;
* Каталог товаров с фильтрацией;
* Информация об аккаунте пользователя;
* Окно регистрации пользователя;

Для организации данных необходимо хранить и обрабатывать следующую информацию:

Об адресах: уникальный идентификатор, город, улица, дом, квартира, дата и время создания записи, дата и время изменения записи, признак удаления.

О категориях: уникальный идентификатор, название, признак удаления, дата и время создания записи, дата и время изменения записи, идентификатор создавшего пользователя, идентификатор изменившего пользователя.

О сотрудниках: уникальный идентификатор, хеш пароля, электронная почта, дата и время создания записи, дата и время изменения записи, признак удаления, номер телефона.

О заказах: уникальный идентификатор, цена, номер телефона клиента, идентификатор клиента, идентификатор адреса, дата и время создания записи, дата и время изменения записи, дата и время завершения заказа, статус заказа, признак удаления, номер заказа.

О продуктах: уникальный идентификатор, название, описание, цена, вес, производитель, количество, изображения, статус, дата и время создания записи, дата и время изменения записи, признак удаления, признак скрытия, идентификатор категории, идентификатор создавшего пользователя, идентификатор изменившего пользователя.

О пользователях: уникальный идентификатор, номер телефона, хеш пароля, электронная почта, дата и время создания записи, дата и время изменения записи, признак удаления.

2. Состав и объем дипломного проекта

2.1 Пояснительная записка, содержащая следующие обязательные разделы:

Введение

1. Основная часть
   1. Разработка системного проекта
      1. Описание предметной области
      2. Анализ предметной области
      3. Техническое задание
      4. Описание используемых технологий
   2. Проектирование информационной системы
      1. Функциональное моделирование
      2. Проектирование интерфейсов приложения
   3. Разработка информационной системы
      1. Модель данных
      2. Структура проекта
   4. Оценка качества программного обеспечения

2 Экономический раздел

3 Охрана труда при работе с вычислительной техникой

Выводы и заключение

Список литературы

Приложения

2.2 Презентация к докладу.

3. Требования к работе:

Пояснительная записка должна содержать не менее 50 листов формата A4. Шрифт оформления пояснительной записки Times New Roman, 14 пт, 1,5 междустрочный интервал.

Презентация должна содержать не более 15 слайдов. На первом слайде указывается название учебного заведения, тема дипломного проекта, фамилия, имя и отчество студента и руководителя, название специальности, город и год. Фон презентации должен оформляться в светлых, нейтральных тонах с контрастным черным текстом. Размер шрифта – не менее 28. Заголовки нужно выделить, шрифт заголовков – не менее 36.

4. Содержание экономического раздела**:**

Экономический раздел включает следующие пункты:

* Понятие себестоимости;
* Затраты в составе себестоимости;
* Расчет себестоимости информационной системы.

5. Содержание раздела по охране труда:

* Охрана труда при работе с вычислительной техникой

Рекомендованная литература:

1. Беляков Г.И. Охрана труда и техника безопасности: учебник для среднего профессионального образования / Беляков Г.И. – 3-е изд., перераб. и доп. – М.: Издательство Юрайт, 2023. – 404 с.
2. Виноградова Н. А. Научно-исследовательская работа студента: Технология написания и оформления доклада, реферата, курсовой и выпускной квалификационной работы: учеб. пособие для студ. Учреждений сред. Проф. Образования / Н.А. Виноградова, Н.В. Микляева. – 13-е изд., стер. – М.: Издательский центр «Академия», 2023;
3. Кудрина, Е. В. Основы алгоритмизации и программирования на языке С#: учеб. пособие для среднего профессионального образования / Е.В. Кудрина, М.В. Огнева. – М.: Издательство Юрайт, 2023;
4. Федорова Г. Н. Разработка модулей программного обеспечения для компьютерных систем: учебник для студ. Учреждений сред. Проф. Образования / Г.Н. Федорова. – 3-е изд., исправленное. – М.: Издательский центр «Академия», 2023;
5. Федорова Г. Н. Осуществление интеграции программных модулей: учебник для студ. учреждений сред. Проф. Образования / Г.Н. Федорова. – 3-е изд., исправленное. – М.: Издательский центр «Академия», 2023;
6. Подбельский В. В. Базовый курс C#: учебник для среднего профессионального образования/ В.В. Подбельский.-М.: Издательство Юрайт, 2023;
7. Рыжко А.Л. Экономика отрасли информационных систем: учебное пособие для среднего профессионального образования / Рыжко А.Л., Рыжко Н.А., Лобанова Н.М., Кучинская Е.О. – 2-е изд., испр. и доп. –М.: Издательство Юрайт, 2024. – 176 с.

Дата выдачи задания « 07 » апреля 2025 г.

Срок окончания дипломного проекта «14» июня 2025 г.

Руководитель проекта \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Руководитель структурного подразделения ­­\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Консультант по экономическому разделу \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Задание утверждено на заседании цикловой комиссии

Протокол № 7 от «12» марта 2025 г.

Председатель цикловой комиссии \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2025 года.

Содержание

|  |  |
| --- | --- |
| Введение | |
| 1 Основная часть | 10 |
| 1.1 Разработка системного проекта |  |
| 1.1.1 Описание предметной области |  |
| 1.1.2 Анализ предметной области | 11 |
| 1.1.3 Техническое задание | 13 |
| 1.1.4 Описание используемых технологий |  |
| 1.2 Проектирование информационной системы | 15 |
| 1.2.1 Функциональное моделирование |  |
| 1.2.2 Проектирование интерфейсов приложения | 18 |
| 1.3 Разработка информационной системы | 27 |
| 1.3.1 Модель данных |  |
| 1.3.2 Структура проекта | 36 |
| 1.4 Оценка качества программного обеспечения | 38 |
| 2 Экономический раздел | 42 |
| 2.1 Понятие себестоимости |  |
| 2.2 Затраты в составе себестоимости | 43 |
| 2.3 Расчет себестоимости информационной системы | 46 |
| 3 Охрана труда при работе с вычислительной техникой | 53 |
| Выводы и заключение | 57 |
| Список литературы | 58 |
| Приложение А – Код программы | 59 |
| Приложение Б – Флэш-накопитель с программой | 62 |

Целью данного проекта является проектирование и внедрение ИТ-системы для автоматизации продаж музыкального оборудования. Система предназначена для оптимизации процессов оформления заказов, управления товарным ассортиментом и взаимодействия с клиентами. Она включает клиентскую часть, позволяющую пользователям просматривать каталог товаров, добавлять их в корзину и оформлять заказы, а также административную панель для управления продуктами, категориями, производителями и обработки заказов.

Клиентская часть системы предоставляет интерфейс с каталогом товаров, разбитым на категории, функционалом корзины для формирования заказа и личным профилем пользователя. Административная часть включает управление заказами, добавление и редактирование товаров, работу с категориями и производителями, а также возможность поиска клиентской информации по номеру телефона. Это обеспечивает полный контроль над процессом продаж и упрощает администрирование интернет-магазина.

В ходе реализации проекта были выполнены следующие задачи: проектирование базы данных для хранения информации о товарах, заказах, клиентах и категориях; разработка клиентского интерфейса с удобной навигацией по каталогу и корзиной заказов; создание административной панели с функциями управления товарами, категориями, производителями и обработки заказов.

Для разработки системы были использованы современные веб-технологии, обеспечивающие стабильную работу, безопасность и удобство взаимодействия пользователей с интерфейсом.

Таким образом, внедрение данной системы позволяет автоматизировать ключевые бизнес-процессы магазина музыкального оборудования: от оформления заказов до управления ассортиментом. Это сокращает время обработки заказов, минимизирует ошибки, связанные с ручным вводом данных, и повышает эффективность работы сотрудников.

1. ОСНОВНАЯ ЧАСТЬ
2. Разработка системного проекта
   * 1. Описание предметной области

Разрабатываемая ИТ-система представляет собой комплексное решение для автоматизации продаж музыкального оборудования, охватывающее ключевые бизнес-процессы от оформления заказов до управления товарным ассортиментом. Современный рынок музыкальной индустрии требует эффективных инструментов, позволяющих оптимизировать взаимодействие с клиентами, упростить управление каталогом продукции и повысить скорость обработки заказов. Автоматизация этих процессов способствует снижению операционных издержек, минимизации ошибок ручного ввода и улучшению качества обслуживания.

Основные сущности системы включают товары, которые характеризуются такими параметрами как название, категория, производитель, цена, наличие на складе и технические характеристики. Клиенты системы имеют персональные профили с контактными данными, историей заказов и предпочтениями, а также возможностью регистрации, авторизации и управления корзиной товаров. Заказы фиксируются системой с детализацией состава товаров, статуса обработки, контактных данных клиента и способа оплаты. Категории и производители организованы в иерархическую структуру каталога с возможностью гибкого управления через административный интерфейс.

Ключевые функциональные возможности системы сосредоточены вокруг удобного клиентского интерфейса для просмотра каталога, фильтрации товаров и оформления заказов. Административная часть предоставляет комплекс инструментов для управления продуктами, категориями и производителями, а также для мониторинга и обработки заказов. Особое внимание уделено функции поиска клиентов по номеру телефона, что значительно упрощает работу с повторными покупками и анализом покупательской активности.

Внедрение системы обеспечивает значительное повышение операционной эффективности за счет сокращения времени обработки заказов и автоматизации рутинных задач. Персонал компании получает в распоряжение мощные инструменты для управления ассортиментом и анализа продаж, тогда как клиенты отмечают удобство и прозрачность процесса оформления заказов. Дополнительным преимуществом становится снижение количества ошибок, связанных с человеческим фактором, и улучшенный контроль товарных остатков.

Перспективы развития системы включают интеграцию с CRM-платформами, внедрение аналитических модулей для прогнозирования спроса и расширение вариантов оплаты. В долгосрочной перспективе рассматривается разработка мобильного приложения для клиентов, внедрение системы персональных рекомендаций на основе истории покупок и подключение к крупным маркетплейсам для увеличения каналов дистрибуции. Эти усовершенствования позволят создать универсальную бизнес-платформу, соответствующую динамике развития рынка музыкального оборудования.

* + 1. Анализ предметной области

Разрабатываемая система автоматизации продаж музыкального оборудования включает двух основных категорий пользователей: клиентов интернет-магазина и администраторов системы. Клиенты взаимодействуют с системой через веб-интерфейс, где могут осуществлять поиск товаров, формировать заказы и управлять своими учетными данными. Администраторы используют специализированный интерфейс для управления товарным ассортиментом, обработки заказов и анализа клиентской базы. Централизованная архитектура системы обеспечивает эффективное взаимодействие между всеми участниками процесса продаж.

Основные функциональные возможности системы охватывают несколько ключевых направлений. В части работы с товарными данными система обеспечивает учет всего ассортимента музыкального оборудования с детализацией по названиям, категориям, производителям, ценам и наличию на складе. Для работы с клиентами реализованы механизмы регистрации и ведения истории заказов. Процесс оформления заказов включает фиксацию состава товаров и отслеживание статусов обработки. Отдельное внимание уделено инструментам управления структурой каталога и анализу покупательской активности.

Взаимодействие пользователей с системой организовано с учетом их ролей и потребностей. После прохождения процедуры регистрации клиенты получают доступ к полному каталогу музыкального оборудования, который поддерживает удобную навигацию по категориям товаров. Процесс формирования заказа реализован через механизм корзины, а в личном кабинете доступна история всех совершенных покупок. Администраторы системы работают через специализированный интерфейс, который предоставляет полный контроль над товарным ассортиментом, включая добавление новых позиций и актуализацию существующих данных. Особое внимание в административном интерфейсе уделено инструментам обработки заказов и аналитическим возможностям по работе с клиентской базой.

Система разработана с ориентацией на эффективное решение задач онлайн-продаж музыкального оборудования. Простота и интуитивная понятность пользовательских интерфейсов сочетаются с мощным функционалом для администрирования бизнес-процессов. Такое сочетание характеристик позволяет минимизировать временные затраты на обработку заказов и значительно снижает вероятность ошибок, связанных с человеческим фактором, что в целом способствует повышению качества обслуживания клиентов и эффективности работы предприятия.

* + 1. Техническое задание

Требуется разработать программный комплекс для автоматизации продаж музыкального оборудования, состоящий из клиентского веб-приложения и административной панели, взаимодействующих через REST API.

Клиентская часть должна предоставлять пользователям возможность просматривать каталог товаров, распределенных по категориям, а также осуществлять поиск и фильтрацию продукции. Каждая карточка товара должна содержать детальное описание, технические характеристики, фотографии и актуальную цену. Пользователи должны иметь возможность добавлять товары в корзину, оформлять заказы и просматривать историю покупок через личный профиль.

Административная панель должна обеспечивать управление товарным ассортиментом, включая добавление, редактирование и удаление позиций, а также контроль категорий и производителей оборудования. Необходимо реализовать функционал для обработки заказов с возможностью изменения их статусов и просмотра деталей по клиентам через номер телефона.

Серверная часть должна быть построена на основе REST API, обеспечивающего стабильное взаимодействие между клиентским и административным интерфейсами. Система должна быть масштабируемой для возможного расширения функционала, такого как интеграция с платежными системами, CRM (Customer Relationship Management) или сервисами аналитики.

* + 1. Описание используемых технологий

В проекте применяется трехуровневая архитектура, обеспечивающая четкое разделение компонентов на слои представления, бизнес-логики и работы с данными. Такой подход значительно упрощает разработку и дальнейшее сопровождение системы за счет модульности и независимого развития отдельных компонентов. Архитектура способствует повышению тестируемости, безопасности и масштабируемости решения, позволяя оптимизировать каждый уровень отдельно в соответствии с изменяющимися требованиями.

Трехуровневая структура включает:

* **Слой представления** (React) - отвечает за взаимодействие с пользователем и отображение данных;
* **Слой бизнес-логики** (C#) - реализует ключевые функции системы, включая валидацию, расчеты и алгоритмы работы с заказами;
* **Слой данных** (PostgreSQL) - обеспечивает хранение, извлечение и преобразование информации.

В качестве системы управления базами данных выбрана PostgreSQL, что обусловлено ее надежностью, производительностью и способностью эффективно обрабатывать значительные объемы информации. СУБД обеспечивает целостность данных о клиентах, товарах и заказах, поддерживая высокую доступность даже при интенсивной нагрузке. Для работы с базой данных используется инструмент DBeaver, предоставляющий удобный интерфейс для администрирования и выполнения запросов.

Серверная часть системы разрабатывается на платформе .NET с использованием языка C# в среде Microsoft Visual Studio 2022. Этот выбор обусловлен богатыми возможностями фреймворка ASP.NET Core для создания Web API, а также наличием опыта работы с данной технологией, что ускорило процесс разработки.

Клиентская часть реализована на React с использованием TypeScript в редакторе Visual Studio Code. Такое сочетание технологий обеспечивает создание производительного и надежного интерфейса с строгой типизацией, что способствует выявлению ошибок на ранних этапах разработки. React был выбран благодаря своей гибкости, богатой экосистеме и возможности создания переиспользуемых компонентов, что особенно важно для интернет-магазина с каталогом товаров.

Использование TypeScript в сочетании с React значительно повышает качество кода за счет статической типизации, улучшает его читаемость и упрощает поддержку, особенно в командной разработке. Это решение также позволяет эффективно применять современные подходы, такие как функциональные компоненты и хуки.

Интеграция всех компонентов системы осуществляется через REST API, что обеспечивает гибкость и возможность дальнейшего расширения функциональности, включая подключение дополнительных сервисов и платежных систем.

1. Проектирование информационной системы
   * 1. Функциональное моделирование

Перед разработкой проекта были спроектированы следующие диаграммы, каждая из которых играет важную роль в описании функциональности и структуры системы.

Диаграмма вариантов использования (прецедентов) помогла определить основные действия, которые пользователи могут выполнять в системе. Это включает в себя описание функциональности системы с точки зрения ее пользователей, а также админов и предоставляет общее представление о взаимодействии между системой и ее окружением. Спроектированная диаграмма представлена на рисунке 1.

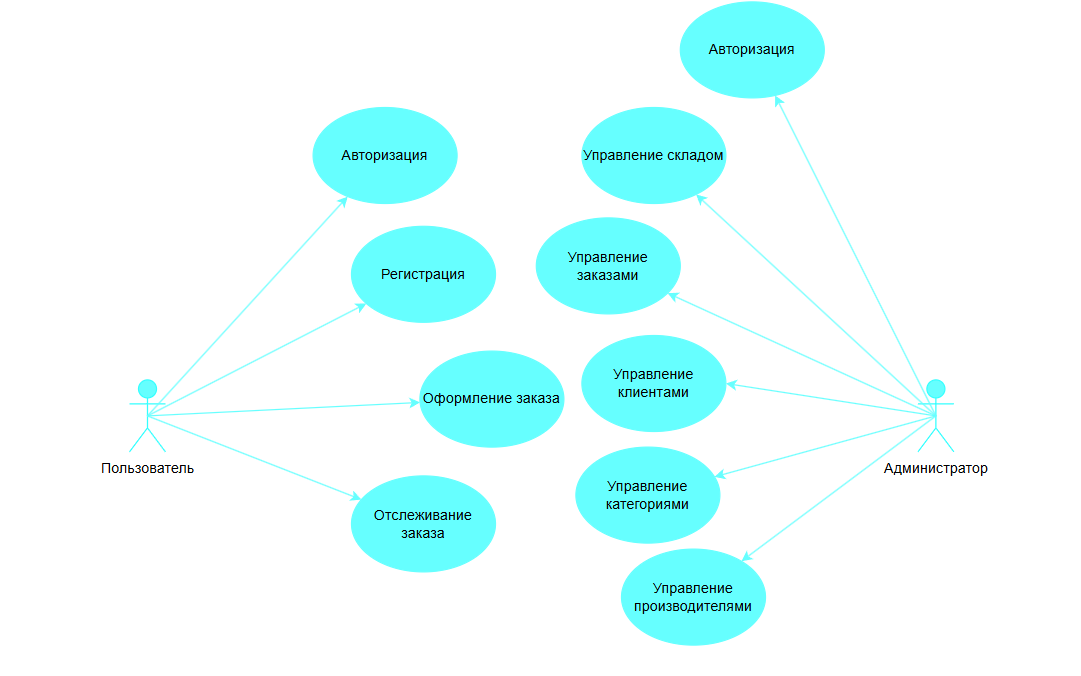
Рисунок 1 – Диаграмма вариантов использования

Диаграмма последовательности разработана для иллюстрации последовательности взаимодействия между объектами в системе в конкретном сценарии использования. Данная диаграмма помогает описать, как объекты взаимодействуют друг с другом во времени и какие сообщения или операции они обмениваются в рамках конкретного сценария. Разработанная диаграмма представлена на рисунке 2.

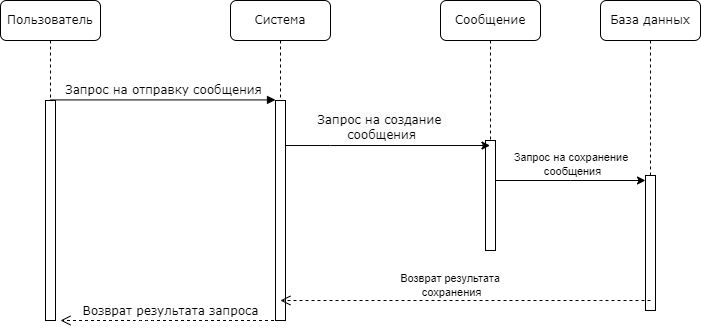
Рисунок 2 – Диаграмма последовательности

Диаграмма деятельности была использована для визуализации последовательности действий или процессов в системе. Это позволило описать порядок выполнения операций или задач в системе и выделить основные этапы взаимодействия пользователей с системой. Разработанная диаграмма представлена на рисунке 3.

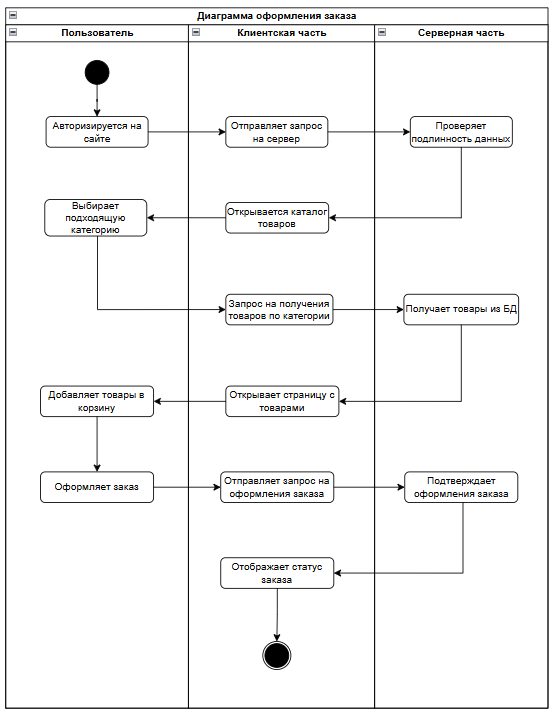


Рисунок 3 – Диаграмма деятельности

* + 1. Функциональное моделирование

Проектирование интерфейсов системы автоматизации продаж музыкального оборудования ориентировано на создание удобного и интуитивно понятного взаимодействия между пользователями и системой. Клиентский веб-интерфейс позволяет легко находить товары в каталоге и оформлять заказы, в то время как административная панель предоставляет полный контроль над ассортиментом и продажами.

Оба интерфейса выполнены в единой бело-синей цветовой гамме, создающей ощущение чистоты и технологичности. Основной акцент сделан на удобной навигации и простоте выполнения ключевых операций. Синий цвет (#1976d2) используется для акцентных элементов и кнопок действий, подчеркивая важные функции системы, в то время как белый фон обеспечивает нейтральное и ненавязчивое окружение.

При разработке интерфейсов были применены современные принципы юзабилити. Минималистичный дизайн исключает лишние элементы, оставляя только необходимый функционал. Все компоненты интерфейса выполнены в едином стиле, что обеспечивает последовательность взаимодействия. Важные действия сопровождаются визуальной обратной связью, а адаптивный дизайн гарантирует корректное отображение на любых устройствах.

Клиентская часть включает каталог товаров с удобной системой фильтрации, корзину с пошаговым оформлением заказа и личный кабинет пользователя. Административный интерфейс содержит инструменты управления товарами, обработки заказов, работы с категориями и анализа продаж. Каждый экран спроектирован таким образом, чтобы выполнение задач требовало минимального количества действий.

Визуальное оформление построено на контрасте белого фона и синих акцентных элементов. Четкая типографика и продуманная иерархия информации делают интерфейс удобочитаемым. Все интерактивные элементы имеют достаточные размеры для комфортного использования, а линейные иконки в едином стиле дополняют общий минималистичный дизайн. Такое оформление сочетает профессиональный внешний вид с высокой функциональностью.

Для начала работы с моим Web Api необходимо открыть стартовую страницу авторизации, изображенную на рисунке 4

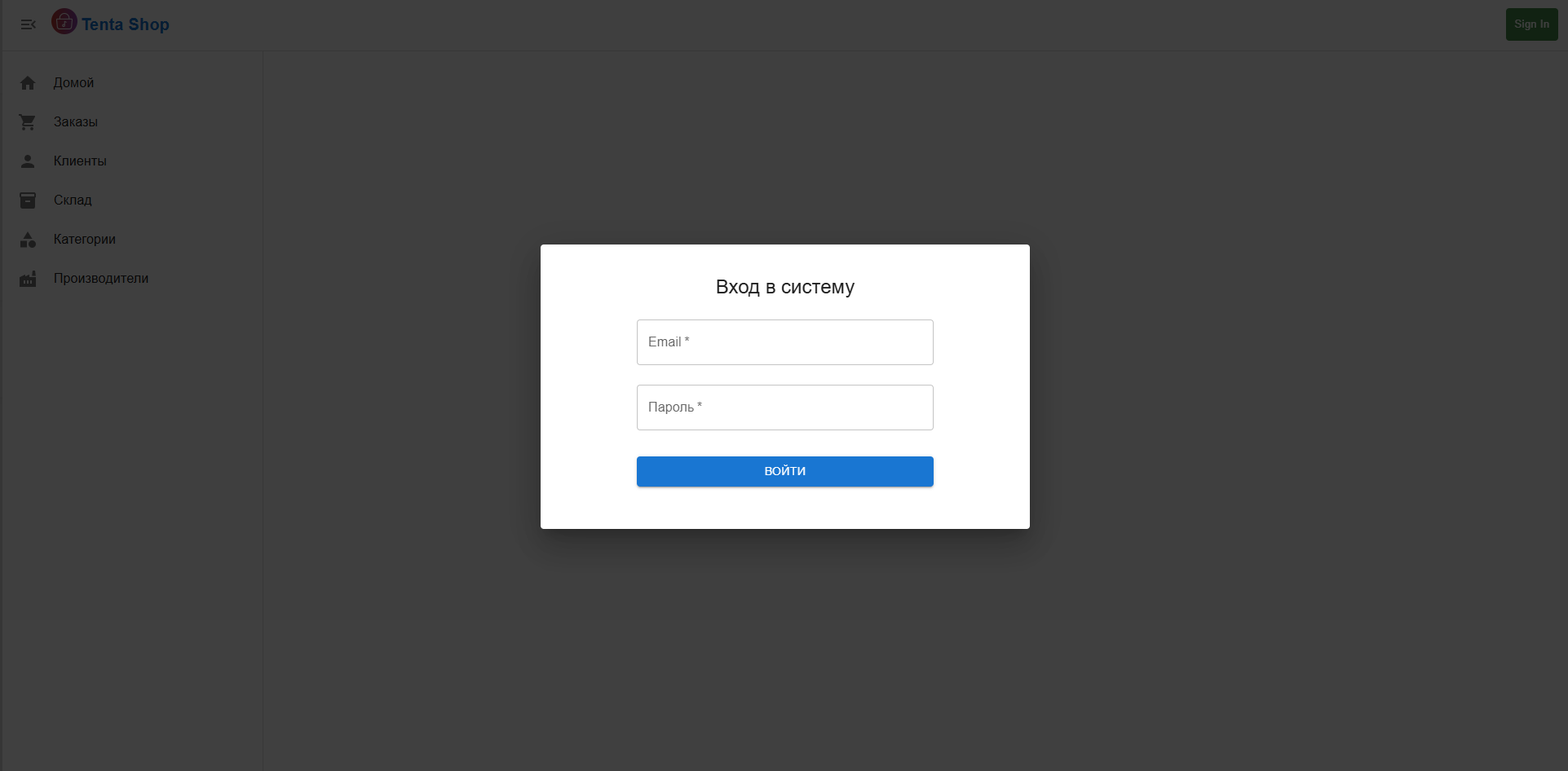


Рисунок 4 – Главная страница авторизации

После прохождение авторизации админ будет перенаправлен на начальную страницу. (Рисунок 5)

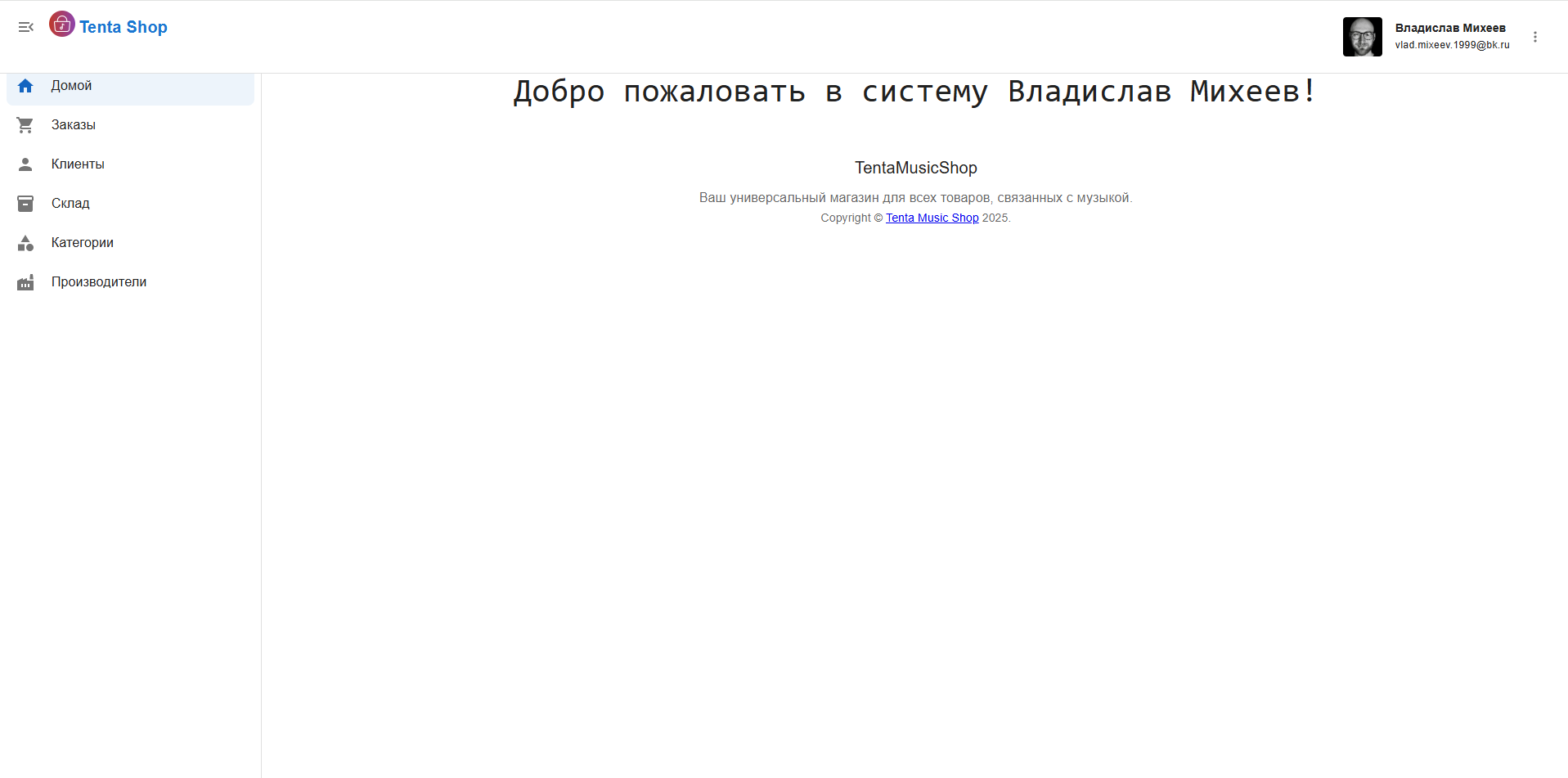


Рисунок 5 – Домашняя страница

Админ может перейти на вкладку Заказы где может посмотреть все активные заказы в текущий момент, посмотреть полные данные или отредактировать существующие, или завершить/доставить заказ. А также админ может добавить новый автомобиль в систему. (Рисунок 6)

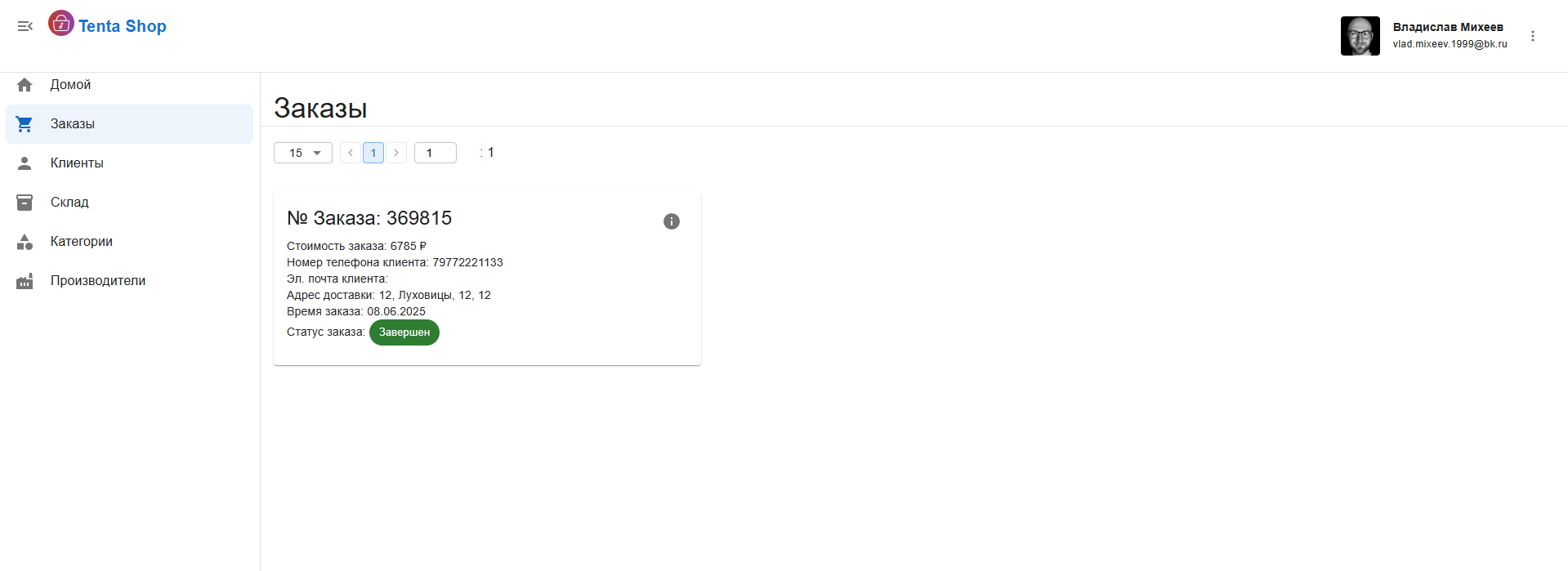


Рисунок 6 – Страница заказов

Админ может перейти на вкладку Клиенты где может посмотреть всю статистику по номеру телефона. (Рисунок 7).

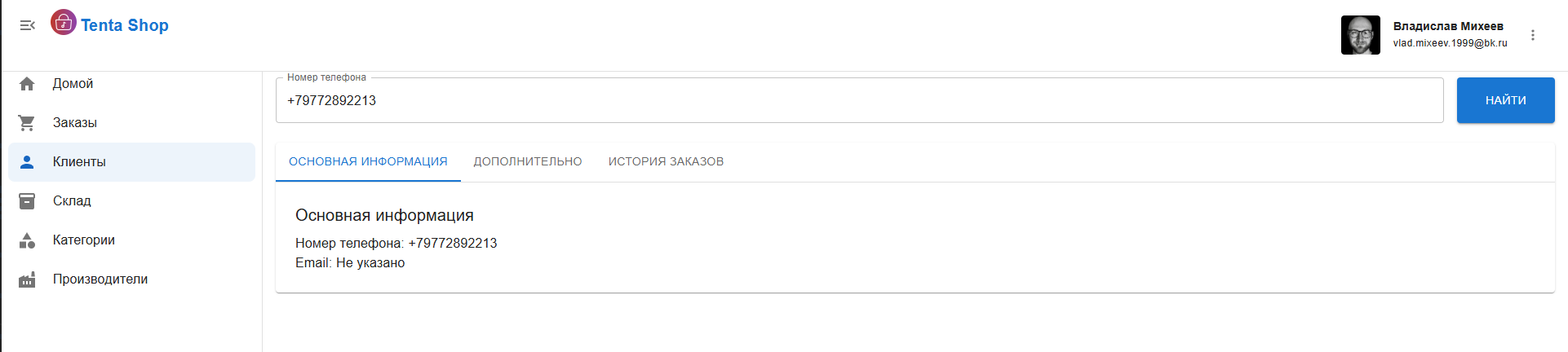


Рисунок 7 – Страница клиентов

Админ может перейти на вкладку Склад где может посмотреть все товары, посмотреть полные данные или отредактировать существующие, или удалить товар. А также админ может добавить новый товар в систему. (Рисунок 8)

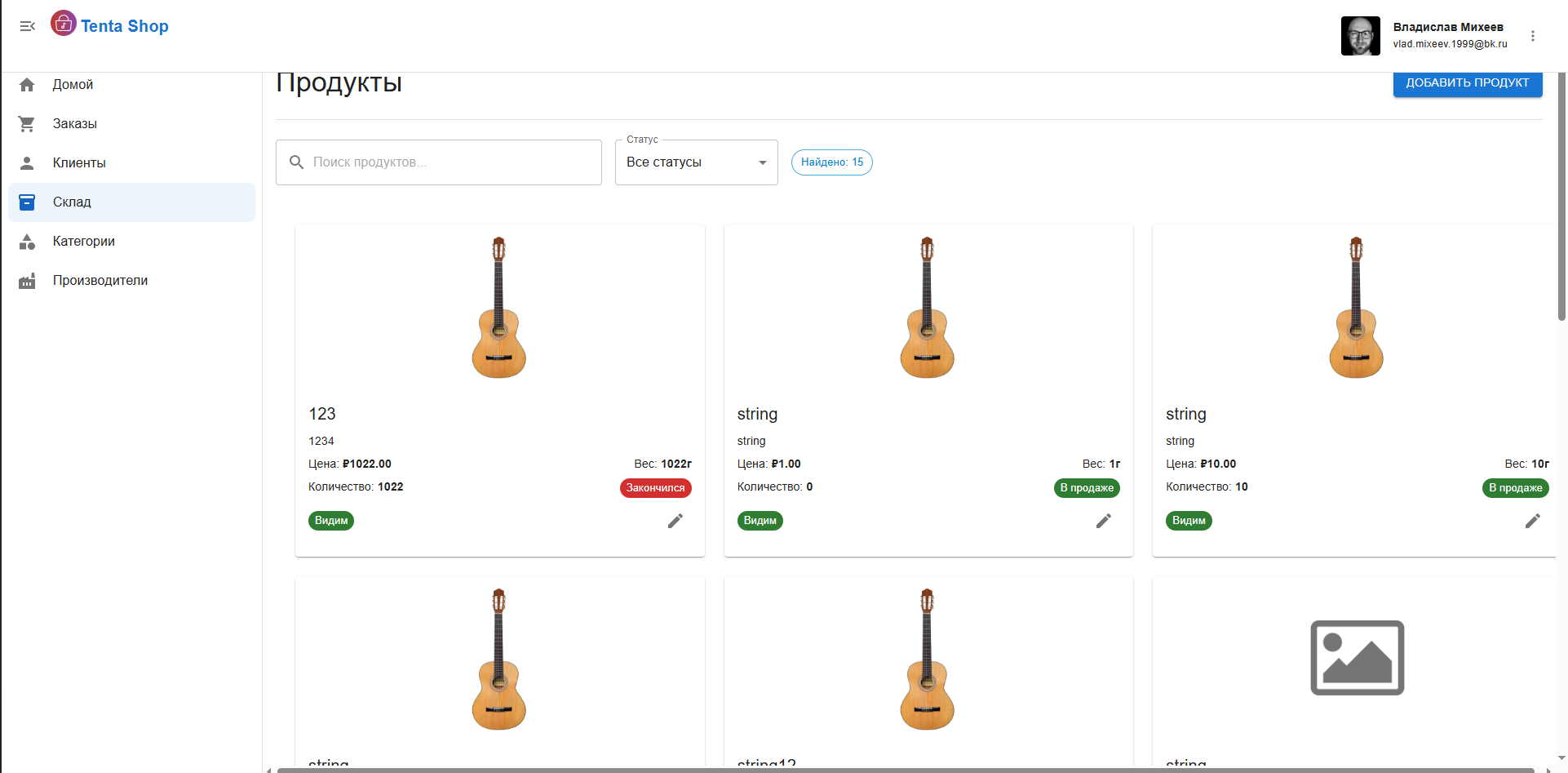


Рисунок 8 – Страница товаров

Админ может перейти на вкладку Категории где может посмотреть все категории и создать новые, а также удалить существующие. (Рисунок 9)

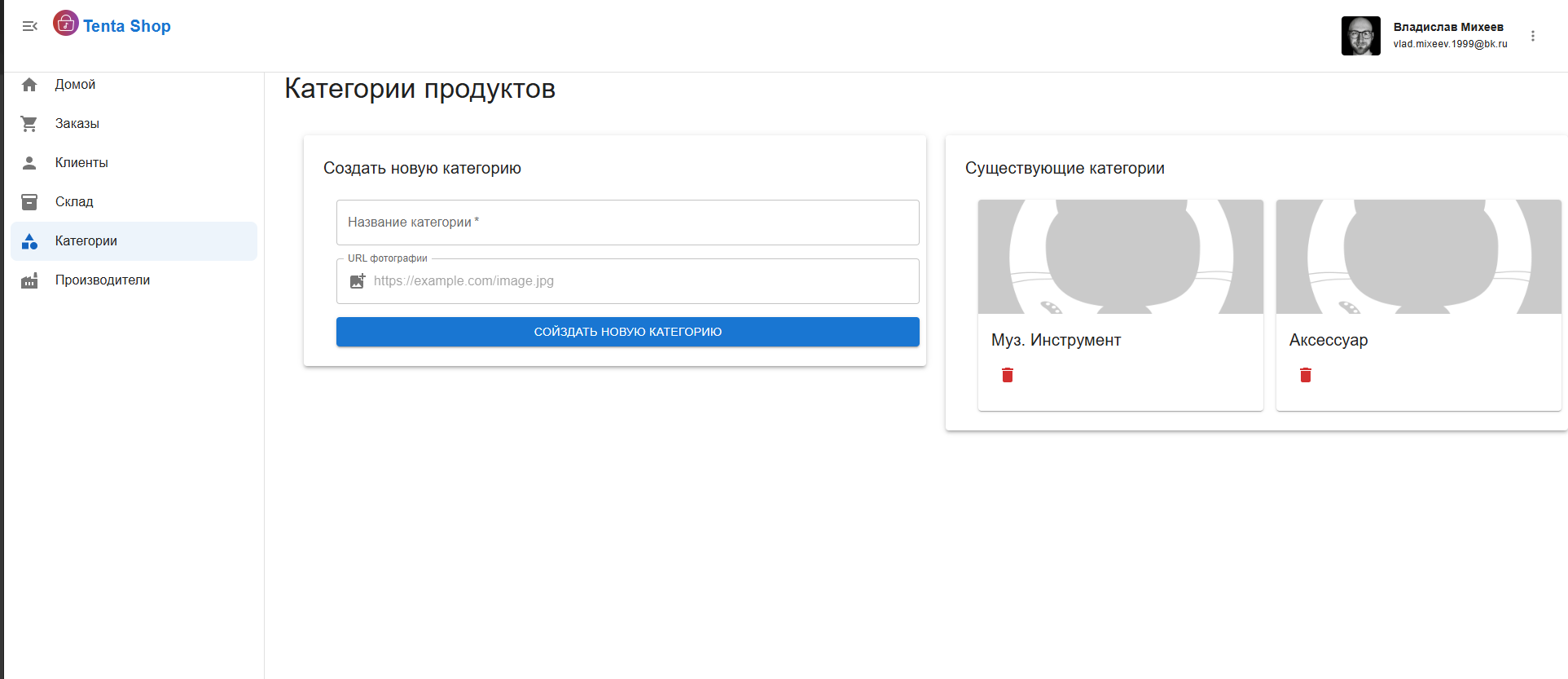


Рисунок 9 – Страница категорий

Админ может перейти на вкладку Производители где может посмотреть всех производителей товаров. А также админ может сформировать отчет от даты до даты по завершенным сделкам. (Рисунок 10)

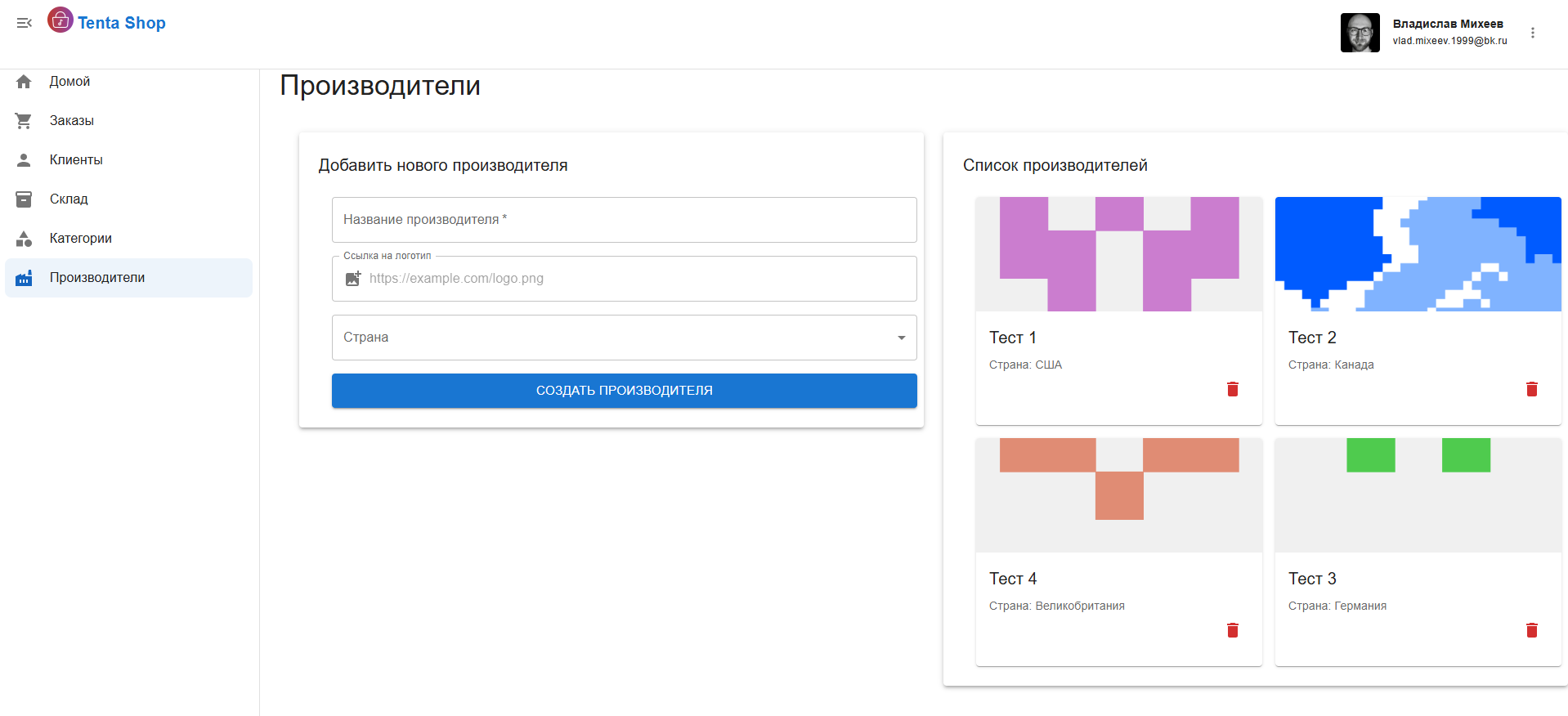


Рисунок 10 – Страница производителей

Для начала работы с клиентским приложение нужно открыть стартовую страницу. (Рисунок 11)

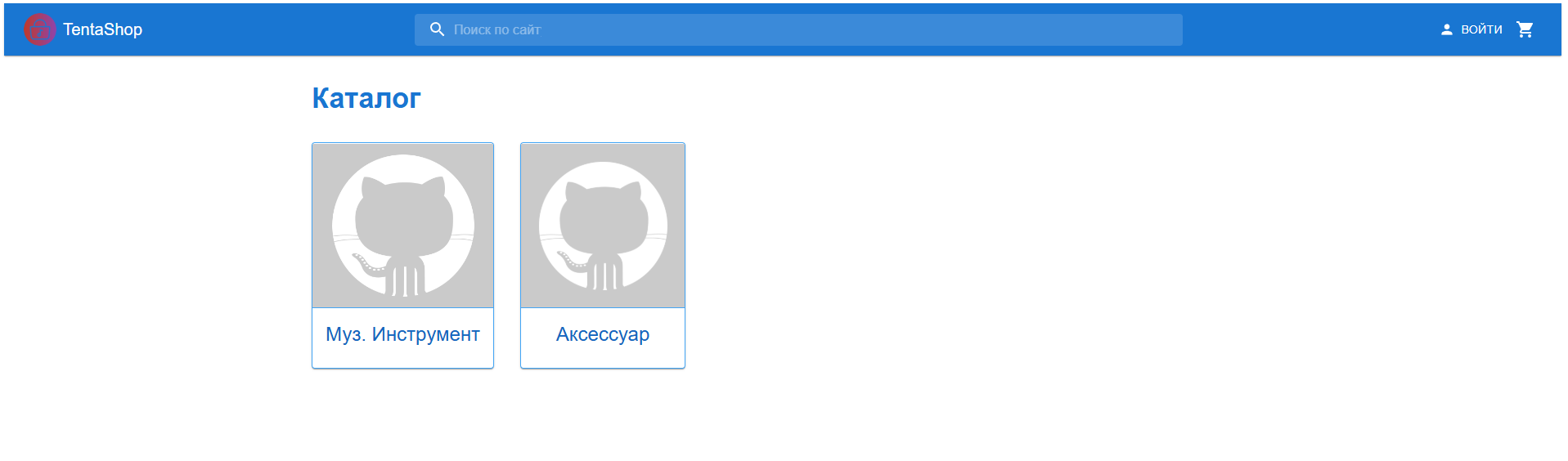


Рисунок 11 – Стартовая страница

Если у пользователя нет аккаунт, он должен перейти на страницу регистрации. (Рисунок 12)

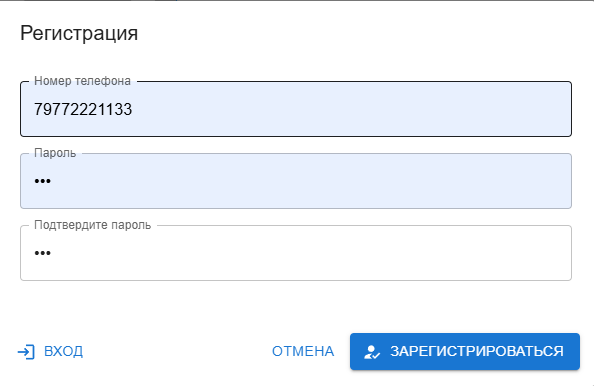


Рисунок 12 – Страница регистрации

После регистрации пользователь попадает на страницу каталога товара. (Рисунок 13)

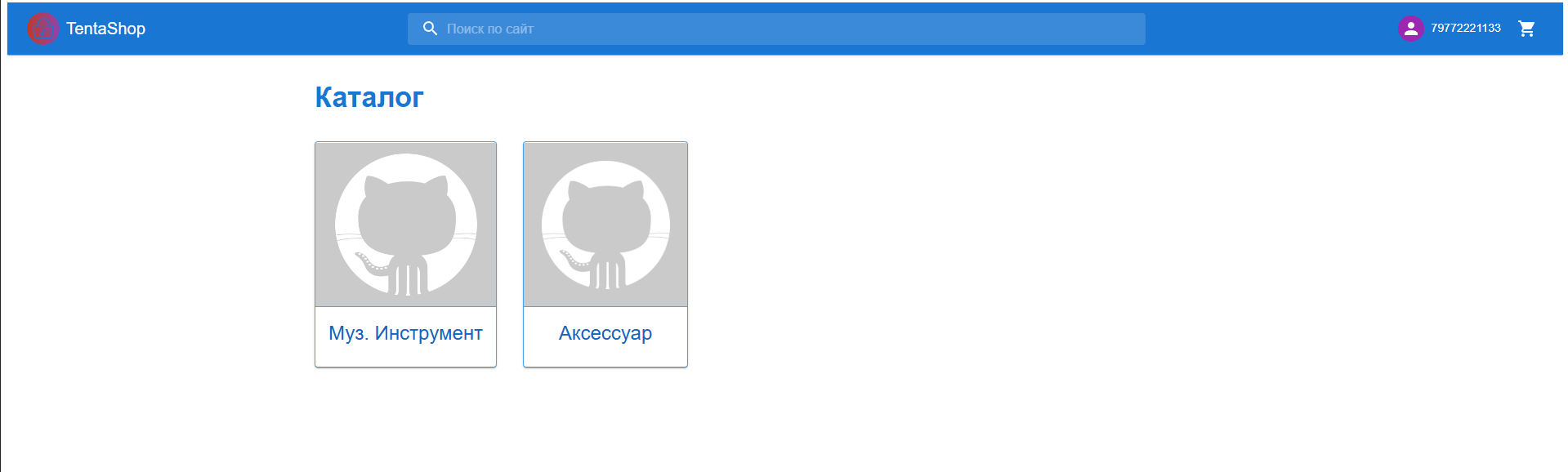


Рисунок 13 – Страница каталога

На странице каталога пользователь должен выбрать интересующую категорию товара и перейти на страницу категории (Рисунок 14)

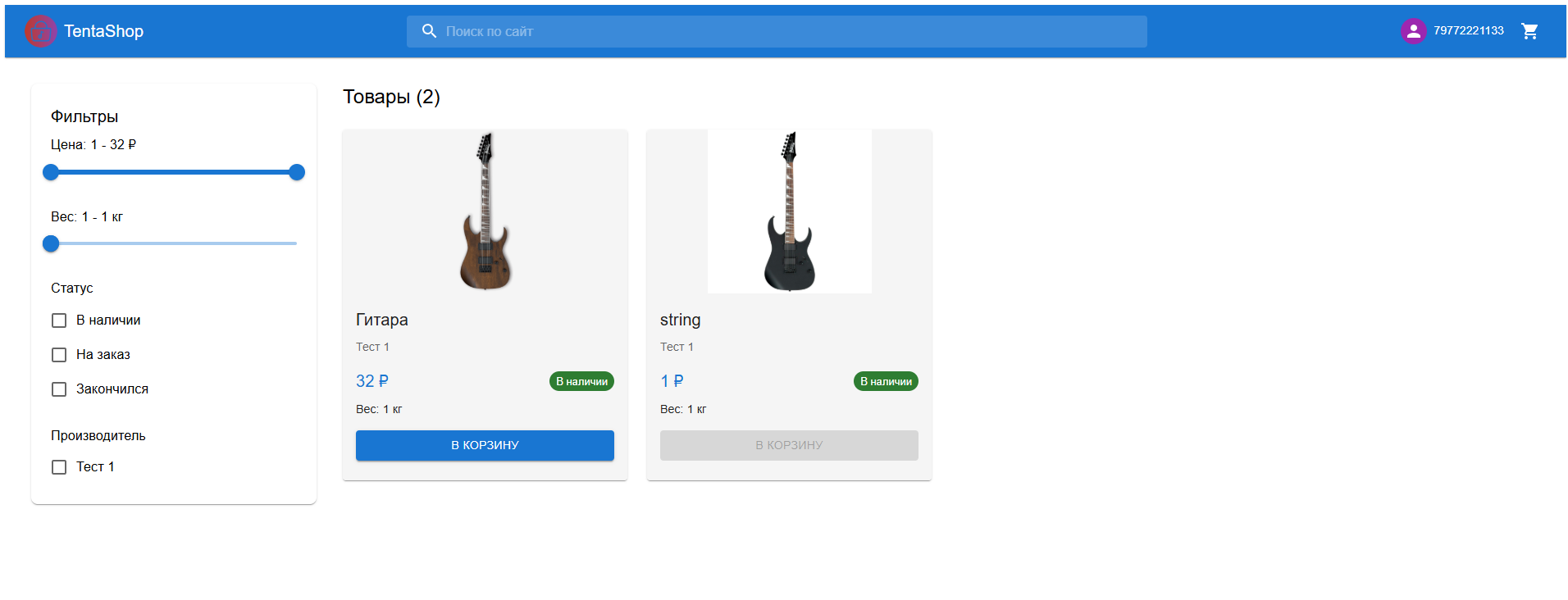


Рисунок 14 – Окно авторизации

Пользователь, выбрав необходимые товары он переходит на страницу оформления заказа (Рисунок 15)

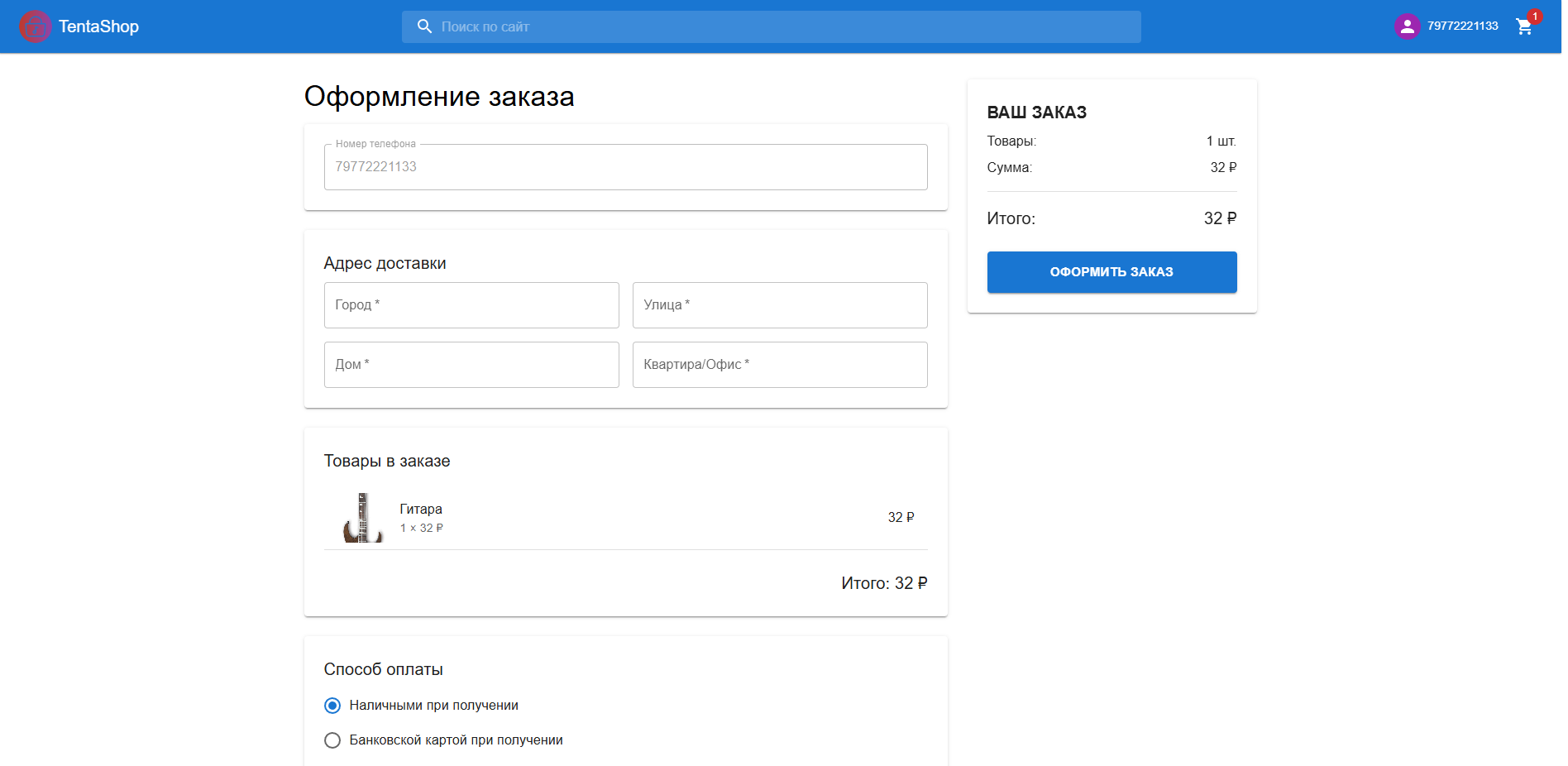


Рисунок 15 – Страница оформления заказа

После заполнения адреса доставки и нажатия кнопки Оформить заказ, пользователь переходит в свой профиль, где он может отслеживать изменения статуса заказа (Рисунок 16)

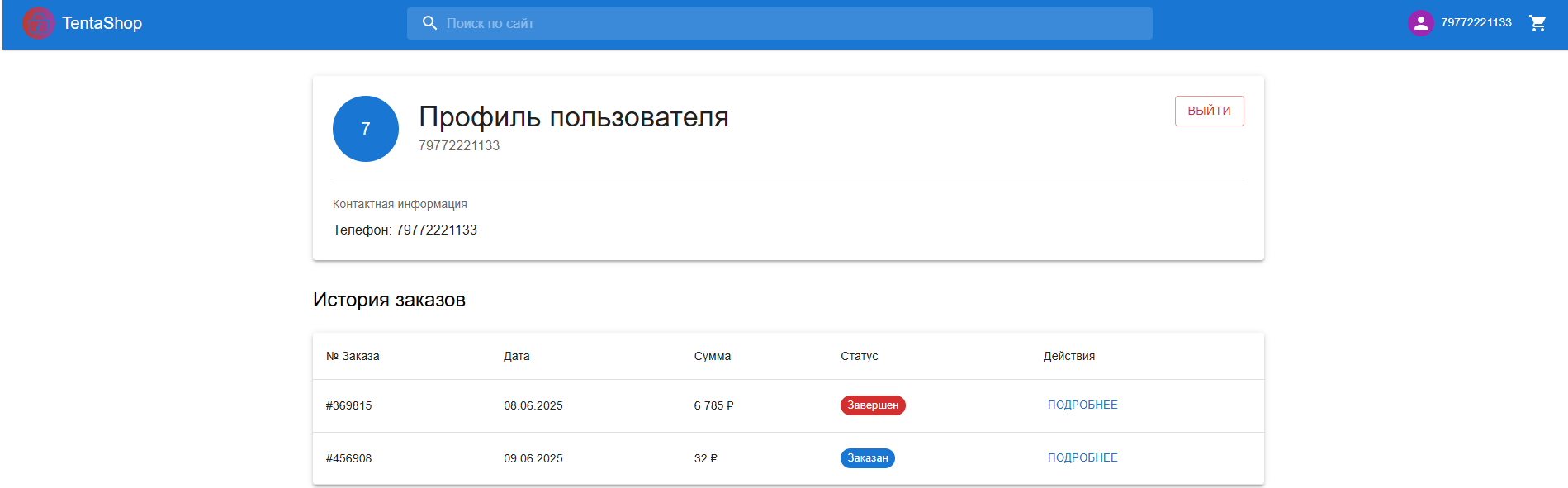


Рисунок 16 – страница профиля пользователя

1. Разработка информационной системы
   * 1. Модель данных

Главной целью проектирования базы данных для системы автоматизации продаж музыкального оборудования было создание гибкой и эффективной структуры для хранения всей необходимой информации. Разработанная модель данных представляет собой систему взаимосвязанных таблиц, учитывающих все бизнес-процессы интернет-магазина. Структура базы данных обеспечивает надежное хранение информации о товарах, категориях, производителях, клиентах и заказах, что позволяет эффективно управлять продажами и обеспечивать удобство работы с системой.

Модель «Категория»:

Модель «Категория»" представляет собой ключевую сущность для организации каталога товаров. Каждая категория имеет уникальный идентификатор и название, которое должно быть уникальным в системе. Для визуального представления категория содержит ссылку на фотографию. В модели хранятся метаданные о создании и изменении записи, включая идентификаторы пользователей, внесших изменения. Флаг isremoved позволяет скрывать категорию без физического удаления из базы данных.

Модель «Производитель»:

Модель «Производитель» описывает компании, выпускающие музыкальное оборудование. Каждый производитель имеет уникальный идентификатор, название и логотип. Дополнительно сохраняется информация о стране происхождения. Как и для других сущностей, предусмотрены поля для отслеживания времени создания и изменения, а также флаг активности записи в системе.

Модель «Товар»:

Модель «Товар» является центральной сущностью системы. Каждый товар содержит уникальный идентификатор, название, подробное описание и цену. Для управления ассортиментом хранится информация о количестве на складе и текущем статусе товара. Товар связан с производителем и категорией, содержит технические характеристики (например, вес) и может иметь несколько изображений. Особые флаги позволяют временно скрывать товар из каталога без удаления.

Модель «Клиент»:

Модель «Клиент» хранит информацию о пользователях системы. Основными данными являются номер телефона (используется как логин), электронная почта и хэш пароля для аутентификации. Для всех записей фиксируются даты создания и изменения, а также флаг активности. Эта модель обеспечивает безопасный доступ к системе и персонализацию взаимодействия.

Модель «Заказ»:

Модель «Заказ» фиксирует факт покупки товаров. Каждый заказ имеет уникальный номер и связывается с клиентом. Хранится полная стоимость заказа, статус выполнения и адрес доставки (связь с отдельной моделью "Адрес"). Важные временные метки включают дату создания, изменения и завершения заказа. Флаг isremoved позволяет архивировать заказы без их удаления.

Модель «Позиция заказа»:

Модель «Позиция заказа» содержит информацию о конкретных товарах в заказе. Для каждой позиции сохраняется идентификатор товара, его цена на момент заказа (защита от изменений в каталоге) и принадлежность к категории. Эта модель обеспечивает детализацию заказов и точный учет продаж.

Модель «Адрес»:

Модель «Адрес» хранит информацию о местах доставки. Содержит стандартные адресные компоненты: город, улицу, дом и квартиру. Как и другие модели, включает метки времени создания/изменения и флаг активности.

Модель «Сотрудник»:

Модель «Сотрудник» предназначена для учета персонала с доступом к административной панели. Содержит учетные данные (email и хэш пароля), контактный телефон и стандартные метаданные. Отделена от модели клиента для обеспечения различных уровней доступа.

Все поля спроектированной базы данных описаны в таблице 1 – в словаре данных.

Словарь данных – структурированный набор терминов и их определений, используемых для описания данных, используемых в системе или проекте.

Таблица 1 – Словарь данных

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Orders | | | | |
| Ключ | Поле | Тип данных | Обязательное | Примечание |
| Первичный | Id | Uuid | Да | Уникальный идентификатор |
|  | Price | Numeric(12,4) | Да | Сумма заказа |
|  | ClientPhoneNumber | VarChar(20) | Да | Номер телефона клиента |
|  | ClientId | Uuid | Да | Идентификатор клиента |
|  | AddressId | Uuid | Да | Идентификатор адреса |
|  | CreatedDateTimeUtc | TimeStamp | Да | Дата создания |
|  | ModifiedDateTimeUtc | TimeStamp | Нет | Дата изменения |
|  | CompetedDateTimeUtc | TimeStamp | Нет | Дата завершения заказа |
|  | State | Int4 | Да | Статус заказа |
|  | IsRemoved | Bool | Да | Статус удаления |
|  | OrderNumber | Int4 | Да | Номер заказа |
| Address | | | | |
| Ключ | Поле | Тип данных | Обязательное | Примечание |
| Первичный | Id | Uuid | Да | Уникальный идентификатор |
|  | City | VarChar | Да | Город |
|  | Street | VarChar | Да | Улица |
|  | Home | VarChar | Да | Дом |
|  | Apartment | VarChar | Нет | Квартира |
|  | CreatedDateTimeUtc | TimeStamp | Да | Дата создания |
|  | ModifiedDateTimeUtc | VarChar | Нет | Дата изменения |
|  | IsRemoved | Bool | Да | Статус удаления |

Продолжение таблицы 1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Categories | | | | | |
| Ключ | Поле | Тип данных | Обязательное | Примечание | |
| Первичный | Id | Uuid | Да | Уникальный идентификатор | |
|  | Name | VarChar | Да | Название | |
|  | IsRemoved | Bool | Да | Статус Удаления | |
|  | CreatedDateTimeUtc | TimeStamp | Да | Дата создания | |
|  | ModifiedDateTimeUtc | TimeStamp | Нет | Дата изменения | |
|  | CreatedUserId | Uuid | Да | Идентификатор пользователя | |
|  | ModifiedUserId | Uuid | Да | Идентификатор | |
|  | Photo | VarChar | Да | Фото | |
| Employees | | | | | |
| Ключ | Поле | Тип данных | Обязательное | | Примечание |
| Первичный | Id | Uuid | Да | | Уникальный идентификатор |
|  | PasswordHash | VarChar | Да | | Хэш пароля |
|  | Email | VarChar | Да | | Почта |
|  | CreatedDateTimeUtc | TimeStamp | Да | | Дата создания |
|  | ModifiedDateTimeUtc | TimeStamp | Нет | | Дата изменения |
|  | IsRemoved | Bool | Да | | Статус удаления |
|  | PhoneNumber | VarChar | Нет | | Номер телефона |
| Manufacturers | | | | | |
| Ключ | Поле | Тип данных | Обязательное | | Примечание |
| Первичный | Id | Uuid | Да | | Уникальный идентификатор |

Продолжение таблицы 1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Ключ | Поле | Тип данных | Обязательное | Примечание |
|  | Name | VarChar | Да | Название |
|  | Logo | VarChar | Да | Логотип |
|  | CreatedDateTimeUtc | TimeStamp | Да | Дата создания |
|  | ModifiedDateTimeUtc | TimeStamp | Нет | Дата изменения |
|  | IsRemoved | Bool | Да | Статус удаления |
|  | Country | Int4 | Да | Страна |
| OrderItems | | | | |
| Ключ | Поле | Тип данных | Обязательное | Примечание |
| Первичный | Id | Uuid | Да | Уникальный идентификатор |
|  | OrderId | Uuid | Да | Идентификатор заказа |
|  | ProductId | Uuid | Да | Идентификатор продукта |
|  | ProductPrice | Numeric(12,4) | Да | Цена продукта |
|  | IsRemoved | Bool | Да | Статус удаления |
|  | CreatedDateTimeUtc | TimeStamp | Да | Дата создания |
|  | ModifiedDateTimeUtc | TimeStamp | Нет | Дата изменения |
|  | ProductCategoryId | Uuid | Да | Идентификатор категории |
| Products | | | | |
| Ключ | Поле | Тип данных | Обязательное | Примечание |
| Первичный | Id | Uuid | Да | Уникальный идентификатор |
|  | Name | VarChar | Да | Имя |
|  | Description | VarChar | Да | Описание |
|  | Price | Numeric(12,4) | Да | Цена продукта |

Продолжение таблицы 1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Ключ | Поле | Тип данных | Обязательное | Примечание |
|  | Weight | Numeric | Да | Вес |
|  | ManufacturerId | Uuid | Да | Идентификатор производителя |
|  | Quantity | Int4 | Да | Количество |
|  | Images | VarChar[] | Да | Фотографии |
|  | Status | Int4 | Да | Статус |
|  | CreatedDateTimeUtc | TimeStamp | Да | Дата создания |
|  | ModifiedDateTimeUtc | TimeStamp | Нет | Дата изменения |
|  | IsRemoved | Bool | Да | Статус удаления |
|  | IsHidden | Bool | Да | Статус скрытия |
|  | CategoryId | Uuid | Да | Идентификатор категории |
|  | CreatedUserId | Uuid | Да | Идентификатор создателя |
|  | ModifiedUserId | Uuid | Нет | Идентификатор изменителя |
| Users | | | | |
| Ключ | Поле | Тип данных | Обязательное | Примечание |
| Первичный | Id | Uuid | Да | Уникальный идентификатор |
|  | PhoneNumber | VarChar | Да | Номер телефона |
|  | PasswordHash | VarChar | Да | Хэш пароля |
|  | Email | VarChar | Нет | Почта |
|  | CreatedDateTimeUtc | TimeStamp | Да | Дата создания |
|  | ModifiedDateTimeUtc | TimeStamp | Нет | Дата Изменения |
|  | IsRemoved | Bool | Да | Статус |

Исходя из предметной области была построена ER-диаграмма для базы данных, предназначенной для эффективного управления арендами автомобилей.

ER-диаграмма (сокращение от Entity-Relationship diagram) – это графическое представление структуры базы данных, которое помогает моделировать и анализировать взаимосвязи между сущностями в системе. Этот метод дает возможность визуализировать основные сущности, их атрибуты и связи между ними. На рисунке 17 изображена ER-диаграмма для базы данных.

База данных спроектирована с использованием первичных ключей (Primary Key), но без явного объявления внешних ключей (Foreign Key) на уровне схемы. Связи между таблицами реализованы через логические поля (например, ClientId в таблице User), что позволяет гибко управлять данными, но требует дополнительного контроля целостности на уровне приложения. На ER-диаграмме связи между сущностями не отображены, так как они не являются частью DDL-определения базы данных, однако в SQL-запросах используются JOIN-операции для объединения данных по этим полям.

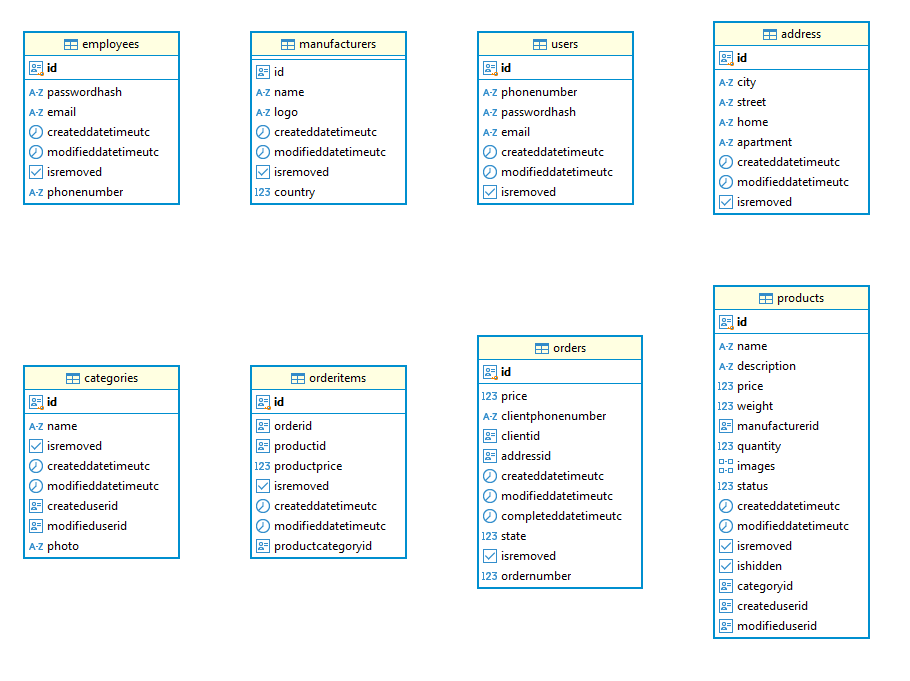


Рисунок 17 – ER диаграмма

* + 1. Структура проекта

Структура проекта построена с учетом принципов модульности и четкого разделения ответственности между компонентами. Каждая директория содержит логически связанные элементы системы, что обеспечивает удобство поддержки и развития приложения.

В корневой директории находятся основные конфигурационные файлы, включая настройки сборки и запуска приложений. Здесь же могут располагаться глобальные ресурсы, используемые во всех частях системы.

Директория Controllers содержит классы контроллеров, которые обрабатывают входящие HTTP-запросы, управляют маршрутизацией и координируют работу сервисов. Каждый контроллер отвечает за определенную группу endpoint'ов API.

В Model определены модели представления данных, используемые для передачи информации между клиентом и сервером. Эти структуры оптимизированы под нужды API и могут включать валидацию входных данных.

Директория Service включает сервисные классы, реализующие основную бизнес-логику приложения. Сервисы обрабатывают данные, выполняют сложные вычисления и координируют работу репозиториев.

Repository содержит классы для работы с данными, абстрагирующие доступ к хранилищу. Репозитории инкапсулируют запросы к базе данных или внешним API, обеспечивая чистоту бизнес-логики.

В Tools собраны вспомогательные утилиты и общие инструменты: хелперы, расширения, middleware и другие повторно используемые компоненты, упрощающие разработку.

TentaShop представляет клиентское React-приложение для конечных пользователей. Оно включает компоненты интерфейса, страницы, маршрутизацию и логику взаимодействия с бэкендом.

AdminApp — это отдельное React-приложение для административной панели, содержащее специализированные компоненты для управления системой, недоступные обычным пользователям.

Такая организация позволяет легко масштабировать проект, добавлять новые функции и вносить изменения, не нарушая работу других компонентов. Четкое разделение слоев приложения способствует поддержанию чистоты кода и упрощает командную разработку.

Структура проекта «MusicShopProject» представлена на рисунке 18.

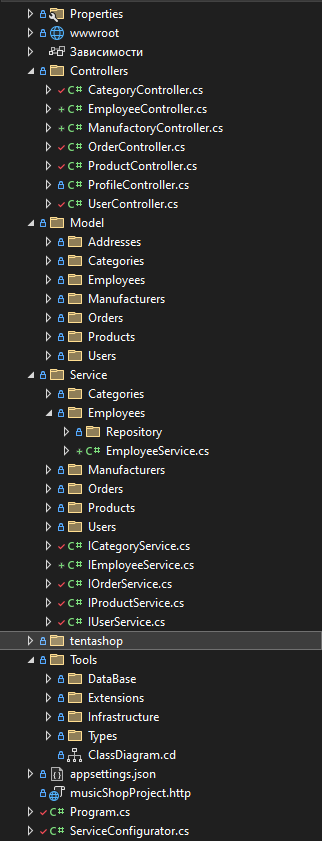


Рисунок 18 – Структура разработанного проекта

* 1. Оценка качества программного обеспечения

Оценка качества программного обеспечения является важным этапом проектирования и внедрения ИТ-системы для автоматизации продаж музыкального оборудования. Основная цель этого процесса — убедиться, что система соответствует заявленным требованиям, функционирует стабильно и обеспечивает надежную работу всех модулей в различных сценариях использования.

Тест-кейсы играют ключевую роль в процессе оценки качества, так как представляют собой детализированные сценарии проверки функциональности системы. Они позволяют выявить возможные ошибки на этапе разработки и внедрения, что способствует повышению надежности и удобства использования системы.

Процесс добавления нового клиента является одним из ключевых функционалов информационной системы управления арендами и пользователями.

Тест добавления нового клиента направлен на проверку корректности работы модуля управления клиентами, который позволяет администратору создавать новые записи о клиентах, включая их контактные данные и адреса. Успешное выполнение данного теста гарантирует, что система способна эффективно расширять базу данных клиентов, обеспечивая удобство работы с информацией о пользователях. Тест-кейс представлен в таблице 3.

Таблица 3 – Тест-кейс добавления нового клиента

|  |  |
| --- | --- |
| Статус | Пройден |
| Название | Регистрация нового клиента |
| Цель | Убедиться, что система корректно добавляет нового клиента в базу данных. |
| Функция | AuthContext.tsx |
| Предусловия | * Страница «Регистрация» открыта. |
| Входные данные | Телефон: «+71234567890»  Пароль: «TestPassword»  Подтверждение пароля: «TestPassword» |

Продолжение таблицы 3

|  |  |  |
| --- | --- | --- |
| Шаги выполнения | Ожидаемый результат | Фактический результат |
| 1. Нажать кнопку «Регистрация»; 2. Заполнить формы; 3. Нажать кнопку «Зарегистрироваться»; 4. Проверить, что вы вошли в систему. | * Клиент успешно добавлен в базу данных; * Данные клиента отображаются на странице «Клиенты»; | * Клиент успешно добавлен в базу данных; * Данные клиента отображаются на странице «Клиенты»; |
| Комментарий | Тест выполнен успешно. Все шаги выполнены в соответствии с ожидаемым результатом. | |

Таким образом, тестирование процесса добавления нового клиента подтвердило корректность работы модуля управления клиентами. Все шаги выполнения теста были успешно реализованы, а фактические результаты совпали с ожидаемыми. Это гарантирует, что система способна эффективно обрабатывать новые записи о клиентах, обеспечивая точность и удобство работы с базой данных.

Тест процесса добавления нового товара направлен на проверку корректности работы модуля товаров, который позволяет администратору добавлять товары. Особое внимание уделяется обновлению товаров после добавления. Тест-кейс представлен в таблице 4.

Таблица 4 – Тест-кейс добавления нового товара

|  |  |
| --- | --- |
| Статус | Пройден |
| Название | Добавление нового товара |
| Цель | Убедиться, что система корректно добавляет товар. |
| Функция | AddProductPage.tsx |
| Предусловия | * Администратор авторизован в системе; * Страница «Склад» открыта; * Страница «Создания» открыта |

Продолжение таблицы 4

|  |  |  |
| --- | --- | --- |
| Входные данные | - Название: "Гитара" (ID: 2);  - Описание: "Тест" (ID: 3);  - Производитель: "Тест3" (ID: 6fa85f34-5717-4532-b2fc-2c963f66afa1);  - Категория: "Москва" (ID: 3fa85f34-5717-4532-b1fc-2c963f66afa6);  - Цена: "1000" (ID: b2c3d4e5-f6g7-8901-h2i3-j4k5l6m7n8o9);  - Вес: 1000 (ID: 7);  - Статус: 0 (ID: 8);  - Скрытый: false (ID: 9);  - Количество: 10. (ID: 10);  - Ссылка на фотографию: "https://avatars.githubusercontent.com/u/1221211" (ID: 11); | |
| Шаги выполнения | Ожидаемый результат | Фактический результат |
| 1. Добавить новый товар; 2. Заполнить данные; 3. Нажать кнопку «Сохранить»; 4. Проверить, что товар появился в списке товаров. | * Товар успешно отображается в таблице; * Товар отображается в базе данных; | * Товар успешно отображается в таблице; * Товар отображается в базе данных; |
| Комментарий | Тест выполнен успешно. Все шаги выполнены в соответствии с ожидаемым результатом. | |

Таким образом, тестирование процесса включения нового товара подтвердило корректность работы системы в части работы с данными. Все шаги выполнения теста соответствуют ожидаемым результатам, что свидетельствует о надежности и точности данного функционала.

Оценка качества программного обеспечения является неотъемлемой частью процесса разработки информационной системы управления заказами в магазине музыкальных инструментов.

Тестирование проводилось с использованием тест-кейсов, которые позволили систематизировать проверку операций системы. Результаты тестирования показали, что система корректно выполняет заявленные функции: клиенты успешно добавляются в базу данных, товары успешно добавляются в базу данных. Это подтверждает, что система соответствует требованиям технического задания и готова к использованию в реальных условиях.

В целом, проведенная оценка качества программного обеспечения подтвердила, что информационная система для автоматизации операций по оформлению заказов является надежным, функциональным и удобным инструментом для автоматизации бизнес-процессов. Система готова к внедрению и способна обеспечить хороший уровень комфорта для администраторов и клиентов, а также стать основой для дальнейшего развития и расширения функционала.

2 ЭКОНОМИЧЕСКИЙ РАЗДЕЛ

2.1 Понятие себестоимости

Себестоимость (товаров, работ, услуг) — стоимостная оценка используемых в процессе производства природных ресурсов, сырья, материалов, основных фондов, трудовых ресурсов и других затрат на ее производство и реализацию.

В себестоимости находят отражение стоимостная оценка природных ресурсов, сырья, материалов, топлива, энергии, трудовых ресурсов.

В себестоимость включается часть стоимости основных фондов, используемых для производства продукции, а также расходы, связанные с обслуживанием и управлением производством.

Себестоимость продукции образует основу цены товаров и является одним из качественных показателей работы предприятия.

От уровня себестоимости зависят прибыль, получаемая предприятиями и рентабельность продукции.

Себестоимость может быть фактической и нормативной.

При расчете фактической себестоимости берут реальные данные, т.е. исходя из фактических издержек, формируется цена товара, работы или услуги. Производить такой расчет очень неудобно, т.к. зачастую узнать себестоимость необходимо раньше, чем производится конкретное производительное действие. От этого зависит рентабельность бизнеса.

Рассчитывая нормативную себестоимость, данные берут согласно производственных норм.

Благодаря этому можно жестко контролировать расход материалов, что минимизирует возникновение неоправданных расходов.

В следующем вопросе конкретизируем затраты, включаемые в состав себестоимости.

2.2 Затраты в составе себестоимости

Перечень расходов, включаемых в себестоимость продукции (работ, услуг) в целях расчета налогооблагаемой прибыли, определяется в соответствии с законодательством Российской Федерации, в частности с главой 25 НК РФ «Налог на прибыль организаций».

Расходами признаются обоснованные и документально подтвержденные затраты, осуществленные предприятием.

Под обоснованными расходами понимаются экономически оправданные затраты, оценка которых выражена в денежной форме.

Под документально подтвержденными расходами понимаются затраты, подтвержденные документами, оформленными в соответствии с законодательством Российской Федерации. Расходами признаются любые затраты при условии, что они произведены для осуществления деятельности, направленной на получение дохода.

Расходы в зависимости от их характера, а также условий осуществления и направлений деятельности предприятия подразделяются на расходы, связанные с производством и реализацией, и внереализационные расходы.

Расходы, связанные с производством и реализацией, включают в себя:

* расходы, связанные с изготовлением (производством)
* (или) реализацией товаров (работ, услуг, имущественных прав);
* расходы на содержание и эксплуатацию, ремонт и техническое обслуживание основных средств и иного имущества, а также на поддержание их в исправном состоянии;
* расходы на обязательное и добровольное страхование;
* прочие расходы, связанные с производством и (или) реализацией.
* Расходы, связанные с производством и реализацией и образующие
* себестоимость продукции, группируются в соответствии с их экономическим содержанием по следующим элементам:
* материальные расходы;
* расходы на оплату труда;
* суммы начисленной амортизации;
* прочие расходы.

К материальным расходам, в частности, относятся следующие затраты предприятия:

* на приобретение сырья и (или) материалов, которые входят в состав вырабатываемой продукции (основные материалы);
* приобретение материалов для обеспечения технологического процесса и для упаковки продукции или расходуемых на другие производственные и хозяйственные нужды;
* приобретение запасных частей и расходных материалов, используемых для ремонта оборудования, инструментов, приспособлений и др.;
* приобретение топлива, воды и энергии всех видов, расходуемых на технологические цели, выработку всех видов энергии, отопление зданий, а также расходы на трансформацию и передачу энергии;
* расходы, связанные с содержанием и эксплуатацией фондов природоохранного назначения, а также платежи за предельно допустимые выбросы загрязняющих веществ в природную среду.

Под возвратными отходами понимаются остатки сырья, материалов, полуфабрикатов, теплоносителей и других видов материальных ресурсов, полностью или частично утратившие потребительские качества исходного ресурса и в силу этого используемые с повышенными затратами (с понижением выхода продукции) или вовсе не используемые по прямому назначению.

К расходам на оплату труда относятся затраты предприятия на оплату труда основного производственного персонала, а также не состоящих в штате работников, участвующих в основной деятельности предприятия.

Затраты на оплату труда включают в себя заработную плату, начисляемую по сдельным расценкам, тарифным ставкам и должностным окладам в соответствии с системами оплаты труда, принятыми на предприятии; надбавки о доплаты; премии за производственные результаты; оплату очередных и дополнительных отпусков; единовременные вознаграждения за выслугу лет и другие расходы, предусмотренные законодательством Российской Федерации.

К суммам начисленной амортизации относятся суммы амортизационных отчислений по амортизируемому имуществу (основным средствам и нематериальным активам).

К прочим расходам относятся:

* расходы на сертификацию продукции и услуг;
* расходы на оплату услуг по охране имущества, обслуживанию охранной пожарной сигнализации, услуг пожарной охраны и иных услуг охранной деятельности;
* расходы по обеспечению нормальных условий труда и техники безопасности, предусмотренных законодательством РФ;
* арендные (лизинговые) платежи за арендуемое имущество;
* расходы на содержание служебного автотранспорта;
* расходы на командировки в пределах норм, утвержденных
* правительством РФ;
* расходы на канцелярские товары;
* расходы на подготовку и освоение новых производств, цехов и агрегатов;
* расходы на ремонт основных средств.
* В состав внереализационных расходов включают:
* расходы на содержание переданного по договору аренды имущества;
* проценты по долговым обязательствам любого вида, включая проценты по кредитам и ценные бумаги, выпущенным предприятием;
* расходы, связанные с организацией выпуска и обслуживанием собственных ценных бумаг;
* судебные расходы и арбитражные сборы и др.

К внереализационным расходам в целях определения налогооблагаемой прибыли приравниваются убытки, связанные с потерями от брака, потерями от стихийных бедствий, пожаров, аварий и других чрезвычайных ситуаций, а также убытки прошлых периодов, выявленные в текущем периоде.

2.3 Расчет себестоимости информационной системы

Расчет себестоимости разрабатываемой информационной системы включает определенную последовательность действий. Покажем их содержание, применяя расчетные действия. Их результаты приведем в таблицах, используя при необходимости соответствующие формулы.

Расчет трудоемкости произведен в таблице 5.

Таблица 5 – Расчет трудоёмкости

|  |  |
| --- | --- |
| Виды работ | Трудоёмкость, дн. |
| Получение ТЗ | 1 |
| Сбор информации и ознакомление с предметной | 2 |
| областью |  |
| Разработка основного алгоритма | 2 |
| Разработка программы | 7 |
| Отладка | 1 |
| Подготовка технической документации | 2 |

Продолжение таблицы 5

|  |  |
| --- | --- |
| Виды работ | Трудоёмкость, дн. |
| Сдача продукта | 1 |
| Итого | 16 |

Расчет затрат на разработку информационной системы.

Величина себестоимости разрабатываемого программного продукта рассчитывается по формуле 1.

С = Сзп + Сэл + Сам + См + Сн + Ссоц.ф , где: (1)

С – величина себестоимости разрабатываемого программного продукта;

Сзп – заработная плата;

СЭл – расходы на электроэнергию;

САм – сумма амортизационных отчислений;

См – затраты на расходные материалы;

Сн – накладные расходы;

Ссоц.ф – отчисления в социальные фонды;

Численность – 2 человека (студент-разработчик и преподаватель).

Зарплата персонала (ЗП) показана в таблице 6. За основу заработной платы студента, разрабатывающего программный продукт, взята величина его заработной платы на должности «Программист».

Таблица 6 – Расчет основной заработной платы

|  |  |  |
| --- | --- | --- |
| Персонал | Количество  (чел.) | ЗП за время разработки  (руб.) |
| Руководитель  диплома | 1 | 3 200 |
| Студент | 1 | 13 936 |
| Всего | 2 | 17 136 |

Расчет заработной платы.

Заработная плата руководителя диплома за час = 400 руб.

Время работы руководителя диплома = 8 ч.

Заработная плата руководителя диплома на время разработки дипломного проекта = 8 \* 400 = 3 200 руб.

Величина оплаты труда студента на своей рабочей должности составит 27 000 руб. В расчетах исходим из того, что заработная плата начисляется за 31 день. Посуточная заработная плата студента за один месяц работы в мае составит 27 000 / 31 = 871 руб.

Время работы студента – 128 часов за 16 дней (с учетом 8 часов работы в день) осуществления проекта. Тогда зарплата за 16 дней работы составит 871 \* 16 = 13 936 рублей.

Затраты на электроэнергию:

Сэл=Р\*Т\* Z, где: (2)

Р – общая мощность оборудования (КВт);

Т – общее время работы оборудования (р.ч.);

Z – цена одного КВт (В 2025 г. – 7 руб. 33 коп).

Затраты на электроэнергию показаны в таблице 7.

Таблица 7 – Расчет затрат на электроэнергию

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Оборудование | Количество,  шт. | Общая  стоимость,  руб. | Общая  мощность,  КВт | Количество часов работы, р.ч. |
| Персональный компьютер | 1 | 750,59 | 0,8 | 128 |
| Лазерный принтер | 1 | 15,39 | 0,7 | 3 |
| Итого | 2 | 765,98 | 1,5 | 131 |

Амортизационные отчисления.

Расчет амортизационных отчислений на рабочий период производится по формуле 3.

Сам = (Спер\*Н a) / (365/16), где (3)

Спер – первоначальная стоимость оборудования;

Нa – годовая норма амортизации (20%);

365 – количество дней в году;

16 – количество дней на разработку проекта.

Амортизационные отчисления рассчитываются на основании таблицы 8.

Таблица 8 – Амортизационные отчисления

|  |  |  |  |
| --- | --- | --- | --- |
| Оборудование | Количество,  шт. | Первоначальная стоимость оборудования, руб. | Стоимость амортизационных отчислений, руб. |
| Ноутбук | 1 | 115 000 | 1 008,22 |
| Лазерный  принтер | 1 | 13 800 | 120,99 |
| Итого | 2 | 128 800 | 1 129,21 |

Затраты на материалы.

Расчет затрат на материалы осуществляется согласно фактическим данным в год, которые представлены в таблице 9.

Таблица 9 – Расчет затрат на материалы

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование  материала | Количество шт. | Стоимость за 1 шт., руб. | Общая стоимость, руб. |
| Бумага | 1 | 319 | 319 |
| Картридж для принтера | 1 | 3 900 | 3 900 |
| Флеш-накопитель | 1 | 332 | 332 |
| Итого | | | 4 551 |

Носителем информации был выбран Флеш-накопитель, так как этот вариант является наиболее экономичным.

Произведем расчет накладных расходов для выбранного проекта и покажем его в табл. 10.

В нашей стране законодательно закреплены накладные расходы только в таких сферах как строительство и медицина и в ряде других видах деятельности. Остальные предприниматели и компании устанавливают самостоятельно перечень подобных затрат. Установим величину накладных расходов на уровне 31%.

Накладные расходы (Общепроизводственные и общехозяйственные расходы):

Сн = ЗПосн\*% накл. расх. /100 (4)

% накл. расх. – процент накладных расходов – 31.

Таблица 10 – Расчет накладных расходов

|  |  |  |  |
| --- | --- | --- | --- |
| Персонал | Количество | Заработная | Накладные |
|  | (чел) | плата, руб. | расходы, руб. |
| Руководитель диплома | 1 | 3 200 | 992 |
| Студент | 1 | 13 936 | 4 320,16 |
| Итого | | | 5 312,16 |

Расчет отчислений в социальные фонды (таблица 11):

Ссоцф = ЗПосн\* % отч. / 100, где: (5)

% отч. – процент отчислений (В 2025 г. – 30).

Изменения в тарифах в 2025 г. касаются части субъектов малого бизнеса. Для субъектов малого предпринимательства в обрабатывающей промышленности тарифы снижены: 30% с суммы 1,5 МРОТ и 7,6% с суммы, превышающей 1,5 МРОТ. В наших расчетах изменений в соответствии с формулой 5 не будет, так как 1.5 МРОТ составляет в 2025 году 33 660 рублей, а обозначенная заработная плата – 13 936 рублей. Кроме того, фирма работодателя не относится к обрабатывающему сектору экономики страны. Итоги расчетов представлены в таблице 11.

Таблица 11 – Расчет отчислений в социальный фонд

|  |  |  |  |
| --- | --- | --- | --- |
| Персонал | Количество  (чел) | Заработная плата, руб. | Сумма  отчислений, руб. |
| Руководитель дплома | 1 | 3 200 | 960 |
| Студент | 1 | 13 936 | 4 180,8 |
| Итого  1  23 800 | | | 5 140,8 |

Затраты на разработку программного продукта рассчитываются по формуле (6),

С = 17 136 + 766 + 1 129 + 4 551 + 5 312 + 5 141 = 34 035 руб. (6)

Удельный вес по элементам затрат представлен в таблице 12.

Таблица 12 – Удельный вес по элементам затрат

|  |  |  |
| --- | --- | --- |
| Элементы затрат | Сумма (руб.) | Удельный вес (%) |
| Зарплата персонала | 17 136 | 50,3 |
| Расходы на электроэнергию | 766 | 2,3 |
| Сумма амортизационных отчислений | 1 129 | 3,3 |
| Сумма затрат на материалы | 4 551 | 13,4 |
| Накладные расходы | 5 312 | 15,6 |
| Отчисления в соц. фонды | 5 141 | 15,1 |
| Итого | 34 035 | 100 |

На рисунке 19 представлена диаграмма себестоимости программного продукта.

Из диаграммы следует, что наибольший удельный вес поэлементных затрат характерен для зарплаты персонала, накладных расходов и отчислений в социальные фонды.



Рисунок 19 – Себестоимость программного продукта, в процентах (%)

В ходе выполнения экономических расчетов был проведен анализ издержек, возникающих при разработке информационной системы. В результате проведенного анализа удалось выявить, что основная сумма себестоимости складывается из заработной платы (50,3%), расходов на электроэнергию (2,3%), отчислений в социальные фонды (15,1%), амортизационных отчислений (3,3%), накладных расходов (15,6%) и затрат на материалы (13,4%). Общая сумма себестоимости разработанной информационной системы составляет 34 035 рублей.

Окупаемость полученной разработки возможна только в случае ее коммерческого применения или реализации на основе соответствующего бюджетного заказа с учетом ожидаемой нормы прибыли.

Проведенное исследование позволило получить навык рассчитывать затраты на разработку информационной системы в целях последующего внедрения в системы автоматизации процессов на предприятии в соответствии с освоением профильной специальности.

3 ТРЕБОВАНИЯ ПО ТЕХНИКЕ БЕЗОПАСНОСТИ ПРИ РАБОТЕ С ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКОЙ

Только длительная работа за компьютером может оказать существенное влияние на здоровье человека.

Рассмотрим основные аспекты длительной работы за компьютером:

* 1. Работающий за компьютером человек длительное время должен сохранять относительно неподвижное положение, что негативно сказывается на позвоночнике и циркуляции крови во всем организме (застой крови). При длительных нарушениях циркуляции крови нарушается питание тканей и повреждаются стенки сосудов, что в свою очередь приводит к их необратимому расширению. Такое расширение сосудов наблюдается, например, при геморрое.
  2. Чтение информации с монитора вызывает перенапряжение глаз. Возникает это главным образом потому, что во время чтения с монитора расстояние от текста до глаз постоянно остается одним и тем же, из-за этого мышцы глаз, регулирующие аккомодацию, находятся в постоянном напряжении. Со временем это может привести к нарушению аккомодативной способности глаз и, следовательно, к нарушениям зрения.

Дисплейная болезнь, характеризуется нарушением аккомодации глаз из-за длительного перенапряжения ресничного тела. Синдром сухого глаза – собирательное название заболевания, вызванного нарушением увлажнения передней поверхности глаза (роговицы) слезной жидкостью. Также длительная работа за компьютером может увеличить риск таких глазных заболеваний как миопия (близорукость), дальнозоркость, глаукома.

* 1. Длительная работа на клавиатуре приводит к перенапряжению суставов кисти и мышц предплечья, что вызывает развитие туннельного синдрома запястья.
  2. Работа за компьютером предполагает переработку большого массива информации и постоянную концентрацию внимания, поэтому при длительной работе за компьютером нередко развивается умственная усталость и нарушение внимания.
  3. Человек, работающий за компьютером, вынужден все время принимать решения, от которых зависит эффективность его работы. Поэтому, длительная работа за компьютером, часто является причиной хронического стресса.
  4. Все чаще появляются сообщения о возникновении компьютерной зависимости. Действительно, длительная работа за компьютером, работа в Интернете и компьютерные игры могут вызвать подобные психические расстройства.
  5. Работа за компьютером нередко поглощает все внимание работающего человека и потому, такие люди часто пренебрегают нормальным питанием и работают впроголодь весь день. Неправильное питание приводит к снижению интеллектуальных способностей человека.
  6. Гиподинамия, стресс, вредные привычки и неправильное питание являются основными причинами сердечно-сосудистых заболеваний и диабета. Таким образом, человек длительное время, работающий за компьютером подвергается реальному риску сердечно-сосудистых заболеваний, различных заболеваний глаз, двигательного аппарата, органов желудочно-кишечного тракта, психических расстройств.

Лечебная гимнастика.

Для развития мышц спины существует множество упражнений, в основном это наклоны в разные стороны. Очень полезно плавание и упражнения на турнике. В том случае, если вы работаете в офисе, старайтесь почаще потягивайтесь и вообще, делайте как можно больше движений, даже без надобности.

Упражнение для рук.

Чем чаще вы будете прерываться для выполнения упражнений, тем больше они принесут пользы.

* Встряхните руки;
* Сжимайте пальцы в кулаки (10 раз);
* Вращайте кулаки вокруг своей оси;
* Надавливая одной рукой на пальцы другой руки со стороны ладони, как бы выворачивая ладонь и запястье наружу.

Положения рук при работе за компьютером.

В процессе работы старайтесь удерживать туловище в вертикальном положении, опираясь на спинку стула или кресла. Ступни ног должны прочно стоять на полу или на специальной подставке, плечи следует держать в расслабленном состоянии. В оптимальной позиции руки должны быть согнуты в локте под углом 90°. Локти должны располагаться вблизи туловища.

Профилактика заболеваний органов дыхания.

* Как можно чаще делайте влажную уборку помещения и проветривайте его;
* Для увеличения влажности можно ставить открытую емкость с водой.

Профилактика нервных расстройств.

* Постарайтесь сделать так, чтобы при работе компьютер давал как можно меньше сбоев и меньше раздражал вас;
* Как можно чаще прерывайте работу с компьютером.

Рабочее место и правильная посадка.

Перед тем как приступить к работе, уделите немного времени организации своего рабочего места. Клавиатура и экран монитора должны быть расположены прямо перед вами. Если конструкция стула позволяет регулировать высоту сиденья, настройте ее таким образом, чтобы в положении сидя ваши ноги прочно опирались на пол или на специальную подставку, а поверхность, на которой расположены клавиатура и мышь, находились на уровне локтя или немного ниже. Разместите клавиатуру и мышь таким образом, чтобы при работе с ними не возникало необходимости отдалять локти от туловища.

Экран монитора должен располагаться на расстоянии вытянутой руки сидящего за столом пользователя. Отрегулируйте высоту подставки монитора таким образом, чтобы центр экрана находился чуть ниже уровня глаз. При отсутствии такой возможности желательно установить монитор на специальную полку или подставку.

При работе с бумажными документами рекомендуется установить на корпус монитора специальный держатель для бумаг.

Упражнения для глаз.

* Зажмурьте глаза на ~ 10 секунд;
* Быстро моргайте в течение ~5-10 сек;
* Сделайте комплекс упражнений:

ВЫВОДЫ И ЗАКЛЮЧЕНИЕ

В рамках данного проекта была разработана ИТ-система для автоматизации продаж музыкального оборудования, включающая клиентскую и административную части. Клиентский сайт позволяет пользователям просматривать каталог товаров с категориями, добавлять товары в корзину, оформлять заказы и управлять профилем. Административный сайт предоставляет функционал для управления заказами, продуктами, категориями и производителями, а также позволяет просматривать информацию о клиентах по номеру телефона.

Целью проекта являлась разработка удобного и функционального решения для автоматизации процессов продаж музыкального оборудования. В результате было создано работоспособное приложение, упрощающее взаимодействие между клиентами и администраторами, а также повышающее эффективность управления товарами и заказами.

Для достижения поставленной цели были выполнены ключевые задачи, включающие проектирование и разработку базы данных, создание пользовательских интерфейсов для клиентов и администраторов, а также обеспечение безопасности системы. Приложение предоставляет удобный инструмент для управления продажами, ускоряет обработку заказов и улучшает контроль за ассортиментом.

Были учтены требования по безопасности и надежности системы, а также проведен анализ затрат на разработку. Данное решение может быть успешно внедрено в работу магазина музыкального оборудования для оптимизации бизнес-процессов и повышения качества обслуживания клиентов.

СПИСОК ЛИТЕРАТУРЫ

1. Беляков Г. И. Охрана труда и техника безопасности: учебник для среднего профессионального образования / Беляков Г.И. — 3-е изд., перераб. и доп. — М: Издательство Юрайт, 2022. — 404 с.
2. Ганенко А.П. Оформление текстовых и графических материалов при подготовке дипломных проектов, курсовых и письменных экзаменационных работ (требования ЕСКД) : учебно-метод. Пособие для студ. Учреждения сред. Проф. Образования / Ганенко А.П., Лапсарь М.И. – 9-е изд., стер. – М.: Издательский центр «Академия», 2022.
3. Федорова Г.Н. Разработка, администрирование и защита баз данных: учебник для студ. Учреждений сред. проф. образования / Федорова Г.Н. – 2-е изд., стер. – М. Петкович Д., 2022

Электронные ресурсы:

1. MSDN – сеть разработчиков Microsoft [Электронный ресурс] Режим доступа: свободный <https://msdn.microsoft.com/ru-ru/dn308572.aspx> (Дата обращения 29.05.2024)
2. Metanit – язык программирования С# и платформа .NET [Электронный ресурс] Режим доступа: свободный <https://metanit.com/sharp/> (Дата обращения 25.05.2025)

ПРИЛОЖЕНИЕ А – Код программы

Весь проект содержит много строчек кода, поэтому прикрепить весь код не представляется возможным. Для этого приложение загружено в публичный репозиторий системы GitHub.

Ссылка на GitHub: https://github.com/tentassill64/musicShopProject

Обработка данных:

* Сохранение продукта

public void SaveProduct(ProductBlank.Validated blank, Guid requestedUserId)

{

String query = @"INSERT INTO products (

id, name, description,

price, weight, manufacturerid, quantity,

categoryid, images, status, createddatetimeutc, modifieddatetimeutc,

isremoved, ishidden, createduserid, modifieduserid)

VALUES (

@p\_id, @p\_name, @p\_description, @p\_price, @p\_weight,

@p\_manufacturerid, @p\_quantity, @p\_category,

@p\_images, @p\_status, @p\_datetimeutc, null, false,

false, @p\_requesteduserid, null

)

ON CONFLICT (id)

DO UPDATE SET

name = @p\_name,

description = @p\_description,

price = @p\_price,

weight = @p\_weight,

manufacturerid = @p\_manufacturerid,

quantity = @p\_quantity,

categoryid = @p\_category,

images = @p\_images,

status = @p\_status,

modifieddatetimeutc = @p\_datetimeutc,

modifieduserid = @p\_requesteduserid";

NpgsqlParameter[] parameters =

{

new("p\_id", blank.Id),

new("p\_name", blank.Name),

new("p\_description", blank.Description),

new("p\_datetimeutc", DateTime.UtcNow),

new("p\_price", blank.Price),

new("p\_weight", blank.Weight),

new("p\_manufacturerid", blank.ManufacturerId),

new("p\_quantity", blank.Quantity),

new("p\_category", blank.CategoryId),

new("p\_images", blank.Images),

new("p\_status", blank.Status),

new("p\_requesteduserid", requestedUserId)

};

\_mainConnector.ExecuteNonQuery(query, parameters);

}

* Сохранение адреса

public void SaveAddress(AddressBlank.Validated addressBlank)

{

String query = "Insert into address (id, city, street, home, apartment, createddatetimeutc, isremoved)" +

"values(@p\_id, @p\_city, @p\_street, @p\_home, @p\_apartment, @p\_timeNow, false) ON CONFLICT(id) DO UPDATE SET " +

"city = @p\_city, street = @p\_street, home = @p\_home, apartment = @p\_apartment, modifieddatetimeutc = @p\_timeNow";

NpgsqlParameter[] parameters =

{

new("p\_id", addressBlank.Id),

new("p\_city", addressBlank.City),

new("p\_street", addressBlank.Street),

new("p\_home", addressBlank.Home),

new("p\_apartment", addressBlank.Apartment),

new("p\_timeNow", DateTime.UtcNow)

};

\_mainConnector.ExecuteNonQuery(query, parameters);

}

* Сохранение заказа

public void SaveOrder(OrderBlank.Validated validatedOrderBlank)

{

String orderQuery = @"Insert into orders (id, price, clientphonenumber, clientid, addressid, createddatetimeutc, completeddatetimeutc, state,

isremoved, ordernumber) values(@p\_id, @p\_price, @p\_clientphonenumber, @p\_clientid, @p\_addressid, @p\_timeNow, @p\_completeddatetimeutc,

@p\_state, false, @o\_ordernumber) ON CONFLICT(id) DO update set price = @p\_price, clientphonenumber = @p\_clientphonenumber, clientid = @p\_clientid,

addressid = @p\_addressid, completeddatetimeutc = @p\_completeddatetimeutc, modifieddatetimeutc = @p\_timeNow";

NpgsqlParameter[] orderParameters =

{

new("p\_id", validatedOrderBlank.Id),

new("p\_price", validatedOrderBlank.Price),

new("p\_clientphonenumber", validatedOrderBlank.ClientPhoneNumber),

new("p\_clientid", validatedOrderBlank.ClientId),

new("p\_addressid", validatedOrderBlank.AddressId),

new("p\_timeNow", DateTime.UtcNow),

new("p\_completeddatetimeutc", validatedOrderBlank.CompletedDateTimeUtc),

new NpgsqlParameter("p\_state", NpgsqlTypes.NpgsqlDbType.Integer) { Value = (int)validatedOrderBlank.State },

new("o\_ordernumber", validatedOrderBlank.OrderNumber)

};

\_mainConnector.ExecuteNonQuery(orderQuery, orderParameters);

String queryOrderItem = @"Insert into orderitems (id, orderId, productId, productprice, isremoved, createddatetimeutc, productcategoryid) values(

@p\_id, @p\_orderId, @p\_productId, @p\_productprice, false, @p\_timeNow, @p\_productcategoryid) ON CONFLICT(id) DO UPDATE SET

orderId = @p\_orderId, productId = @p\_productId, productprice = @p\_productprice,

productcategoryid = @p\_productcategoryid, modifieddatetimeutc = @p\_timeNow";

List<NpgsqlParameter> orderItemsParameters = new();

validatedOrderBlank.ValidatedOrderItemBlanks.ToList().ForEach(validatedOrderItemBlank =>

{

orderItemsParameters.Add(new("p\_id", validatedOrderItemBlank.Id));

orderItemsParameters.Add(new("p\_orderId", validatedOrderItemBlank.OrderId));

orderItemsParameters.Add(new("p\_productId", validatedOrderItemBlank.ProductId));

orderItemsParameters.Add(new("p\_productprice", validatedOrderItemBlank.ProductPrice));

orderItemsParameters.Add(new("p\_productcategoryid", validatedOrderItemBlank.ProductCategoryId));

orderItemsParameters.Add(new("p\_timeNow", DateTime.UtcNow));

\_mainConnector.ExecuteNonQuery(queryOrderItem, orderItemsParameters.ToArray());

orderItemsParameters.Clear();

});

* Сохранение категории

public void SaveCategory(CategoryBlank.Validated validatedBlank)

{

String query = @"INSERT INTO categories (

id, name, isremoved, createddatetimeutc,

createduserid, photo

)

VALUES ( @p\_id, @p\_name, false, @p\_datetime, @p\_createdUserId, @p\_photo)

ON CONFLICT (id)

DO UPDATE SET

name = @p\_name,

modifieddatetimeutc = @p\_datetime,

photo = @p\_photo,

modifieduserid = @p\_modifiedUserId";

NpgsqlParameter[] parameters =

{

new("p\_id", validatedBlank.Id),

new("p\_name", validatedBlank.Name),

new("p\_datetime", DateTime.UtcNow),

new("p\_createdUserId", validatedBlank.CreatedUserId),

new("p\_modifiedUserId", validatedBlank.ModifiedUserId),

new("p\_photo", validatedBlank.Photo)

};

\_mainConnector.ExecuteNonQuery(query, parameters);

}

ПРИЛОЖЕНИЕ Б – Флэш-накопитель с программой