**Содержание**

[**Введение** 3](#_Toc93651448)

[**Приложение А** 10](#_Toc93651449)

# **Введение**

Для человека, живущего в социуме, достаточно важным и значимым является вопрос внешнего вида. Все люди знают поговорку "встречают по одёжке…" и учитывают, что именно одежда является важнейшей составляющей первого впечатления.

Темой данного курсового проекта стала разработка информационной системы для магазина одежды, так как актуальность этой темы в данное время велика, так как появляется необходимость эффективной работы с товаром. Чтобы облегчить задачу учета продаваемой и поступающей одежды работникам магазина будет разработана и спроектирована данная автоматизированная система.

Цель работы – разработать приложение для учета товаров в магазине одежды "Samurai".

Исходя из поставленной цели, были решены задачи:

* обоснование необходимости создания информационной системы;
* проведение анализа и выбор технологий для разработки проекта;
* разработка проекта, взаимодействующего с базой данных.

Пояснительная записка курсового проекта содержит описание всех этапов разработки информационной системы: техническое задание, анализ предметной области, проектирование и реализацию базы данных.

1 Основные понятия. Описание предметной области

* 1. Общие принципы построения информационных систем

Для начала нам надо разобраться в основных понятиях разработки компьютерных игр и машинном обучении:

Цель. Первое, что нам понадобится – это определиться с целью проекта, что мы ждём и чего хотим от этого проекта. Для своей программы, я поставил цель, создать информационную систему для принятия заявок клиента и обработки учета товара для оптимизации работы в магазине одежды.

Средство. Мы обозначили цель, теперь осталось выбрать нам средства для достижения поставленной цели. Основным средством – является программный код, с помощью него мы будем обучать и развивать наш искусственный интеллект

Программный код. Необходимо выбрать язык программирования, который мы будем использовать для написания кода, для нейроэволюции не так важен язык программирования, как правильное его использование. Я буду использовать язык программирования С#, потому что он содержит в себе полезные библиотеки и алгоритмы для более простой реализации задуманного, а так-же очень приятен синтаксисом.

Numpy - эта библиотека которая всегда используется в программах нейроэволюции, так как она создана для того, чтобы проводить различные манипуляции с большими массивами и матрицами

SQL Server Management Studio (SSMS) — это интегрированная среда для управления любой инфраструктурой SQL, от SQL Server до баз данных SQL Azure. SSMS предоставляет средства для настройки, наблюдения и администрирования экземпляров SQL Server и баз данных. С помощью SSMS можно развертывать, отслеживать и обновлять компоненты уровня данных, используемые вашими приложениями, а также создавать запросы и скрипты.

Visual studio code – удобный текстовый редактор от компании Microsoft.

1.2 Описание предметной области. Постановка задачи

1.2.1 Описание предметной области

База данных создается для упрощения работы сети магазинов.

1.2.2 Постановка задачи

Для более удобного просмотра товара в магазины, в современном мире, появились приложения, отвечающие за дистанционную отправку товаров со склада. Для более удобного пользования этими приложениями, происходит взаимодействие с базами данных. План по созданию программы выглядит так:

1. Создание и заполнение базы данных.
2. Построение и оформление интерфейса приложения.
3. Подключение БД к интерфейсу.
4. Реализация функции регистрации и авторизации.
5. Создание дополнительных интерфейсов для каждой отдельной функции.
6. Тестирование и редактирование приложения.
7. Релиз проекта.

Разрабатываемое приложение должно обеспечивать для рабочего склада:

* доступ к полному перечню товаров;
* возможность всячески просмотр списка товаров сотруднику;
* возможность авторизации;

Для Сотрудника:

* доступ к просмотру каталога товаров;
* возможность редактировать товар;
* возможность Добавления товара;
* возможность авторизации;
* возможность обновления товара;
* возможность поиска товара;
* доступ вызова склада, когда не хватка товара в магазине.

1.3 Анализ подобных программ

Проведем анализ несколько программ по учету клиентов:

Стоматология Здоровые зубы – программа для учета клиентов в стоматологии. Выбор данной программы для анализа основан на схожем функционале по учету клиентов. В обоих программах клиенту назначают разные процедуры в определенные даты и время, которые он проходит. Отличия этой программы от разрабатываемой программы заключается в наличие поставки предметов на склад стоматологии. Пример работы показан на рисунке 1.

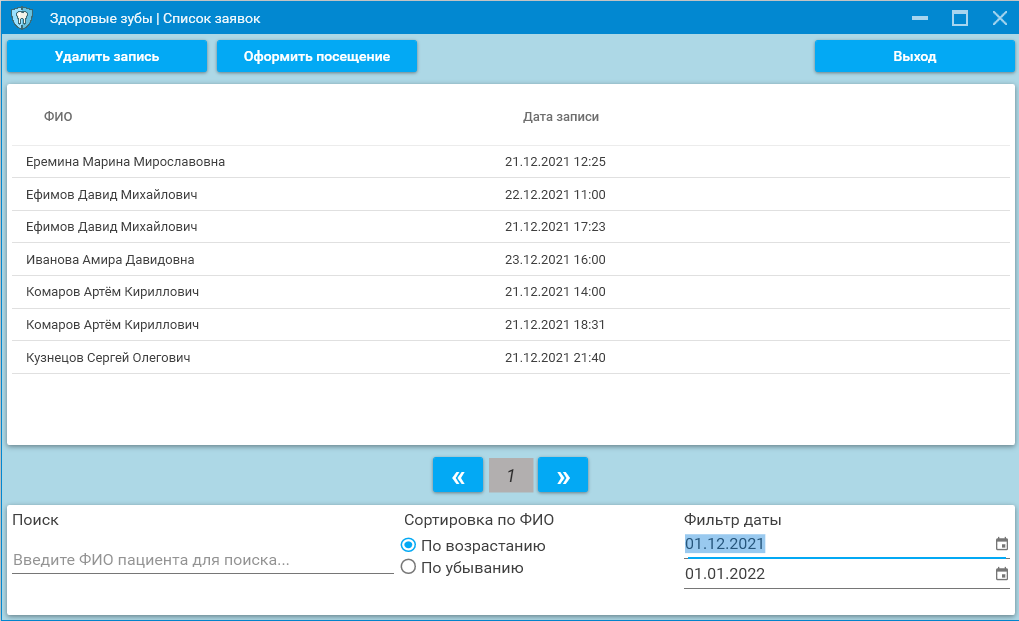


Рисунок 1- Пример Здоровые зубы

2 Этапы разработки программы

2.1 Техническое задание

2.1.1 Назначение разработки

Назначение программы состоит в том, чтобы улучшить и обеспечить работу с клиентами с помощью информационной системы.

2.1.2 Область применения

Область применения информационной системы по учету клиентов обширна, начиная с частных заведений заканчивая всемирными компаниями.

2.1.3 Основания для разработки

Основанием для разработки послужило то, что эта тема всегда актуальна из-за постоянно появляющихся компаний, работающих с клиентами

2.1.4 Описание проекта

Проект направлен на создание информационной системы для магазина. От системы требуется проводить учет товаров, поставку товара, а также хранить информацию о текущем количестве продукции в магазине, необходимые для проведения процедур.

2.1.5 Требования к информационной и программной совместимости

Для корректной работы приложения у пользователя должен быть ПК обладающий следующими минимальными параметрами системы:

* Процессор Intel Pentium или AMD Athlon
* Объем оперативной памяти 4 Гб
* Объем видеопамяти 512 Мб
* Интернет-соединение не ниже 128 кб/сек
* Свободное место на жестком диске не менее 1 Гб

Программа должна функционировать под управлением ОС семейства Windows следующих версий: Windows 7, 8, 10, 11.

2.1.6 Требования к входным данным

База данных должна содержать данные о товарах их наличие в магазине и на складе.

2.1.7 Требования к выходным данным

Выходными данными являются любые данные, полученные из запроса к базе данных, например, рассмотрим получение информации о товаре в магазине после авторизации сотрудника.

Выходные данные:

* Название продукта;
* Категория;
* Цена;
* Размер;
* Изображение.

2.2 Анализ предметной области. Выбор метода решения

Выбрав необходимые товары или услуги, пользователь обычно имеет возможность тут же на сайте выбрать метод оплаты и доставки. Совокупность отобранных товаров, способ оплаты и доставки представляют собой законченный заказ, который оформляется на сайте путем сообщения минимально необходимой информации о покупателе. Информация о покупателе может храниться в базе данных магазина если бизнес-модель магазина рассчитана на повторные покупки, или же отправляться разово. В интернет-магазинах, рассчитанных на повторные покупки, также ведется отслеживание возвратов посетителя и история покупок. Часто при оформлении заказа предусматривается возможность сообщить некоторые дополнительные пожелания от покупателя продавцу.

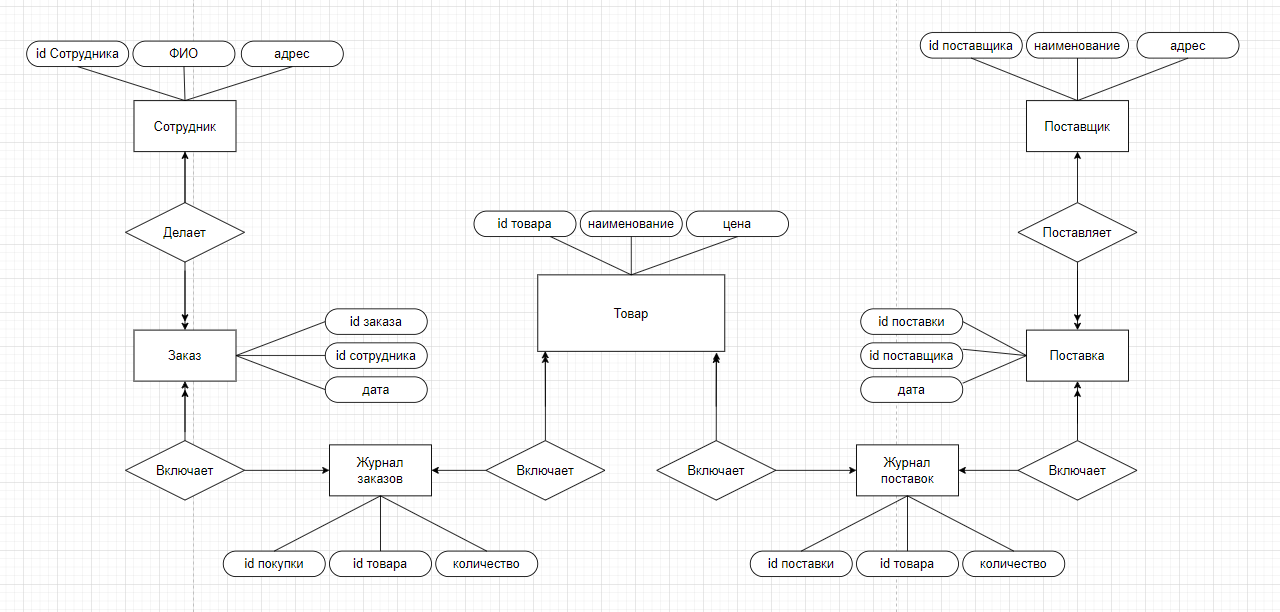


Рисунок 2 – Концептуальная модель

В логической модели представлены 12 таблиц.

Таблица «Сотрудник» содержит данные о сотрудниках магазинов.

Таблица «Магазины» хранит данные о магазинах.

Таблица «Товары в магазине» описывает товары, которые находятся в магазинах.

Таблица «Товары» содержит информацию о товарах.

Таблица «Категория товаров» содержит информацию о категории товара.

Таблица «Отгруженные товары» описывает товары, которые находятся на складе.

Таблица «Поступления на складе» хранит данные о складе.

Таблица «Кладовщик» содержит данные о сотрудниках склада.

Таблица «Торговая марка» хранит данные о производителе товара.

Таблица «Заявка» содержит информацию о поставке.

Таблица «Товары в заявках» содержит информацию о количестве товара в магазин.

Таблица «Отгрузка» описывает товары, которые находятся в поставке.

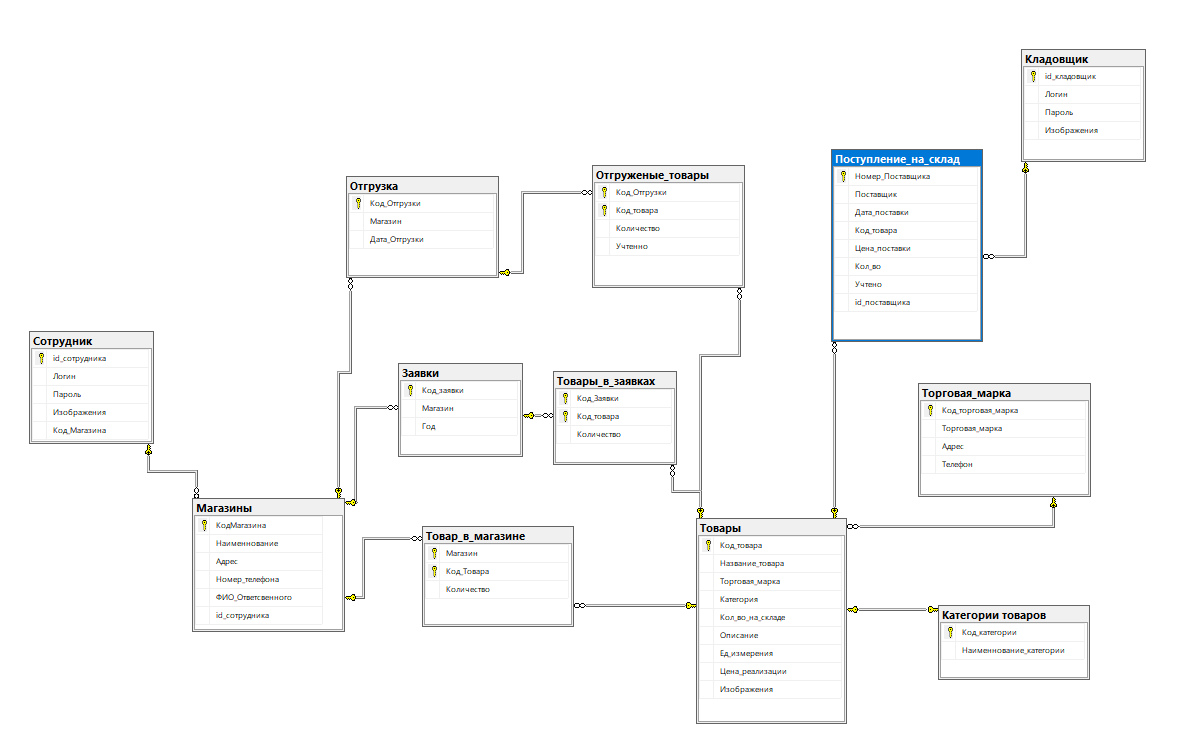


Рисунок 3 – ERD-Диаграмма базы данных

2.3 Разработка структуры программы

При разработке приложения «Информационная система по учету клиентов в магазине одежды» будет использоваться подход Database First, который позволяет реконструировать модель из существующей базы данных. Модель хранится в EDMX-файле (расширение EDMX) и может быть просмотрена и изменена в Entity Framework Designer. Классы, с которыми взаимодействует приложение, автоматически создаются из файла EDMX.

Доступ к данным будет осуществляется с помощью технологии .NET, состоящей из:

* ADO.NET – основная модель доступа к данным для приложений, основанных на Microsoft .NET
* ADO.NET Entity Framework – объектно-ориентированная технология доступа к данным, является ORM-решением для .NET Framework 4.8

Для хранения данных используется Microsoft SQL Server – система управления реляционными базами данных (СУБД), разработанная корпорацией Microsoft

Организация взаимодействия с БД в ADO.NET

* Устанавливается соединение, открывается подключение к базе данных
* Выполняется один или несколько запросов
* Осуществляется отключение от источника данных
* При необходимости переноса изменений, из отсоединенного набора данных в БД, а также при необходимости просмотра изменений, внесенных в БД другими пользователями, осуществляется подключение к источнику данных, выполняются необходимые действия, после чего производится отключение от БД



Рисунок 4 – Схема взаимодействия приложения с базой данных

2.4 Проектирование интерфейса пользователя

Данное ПО состоит из множества форм и для перемещения между ними необходимо реализовать перемещение по формам следующим образом:



Рисунок 5 – Отношения между формами

Интерфейс пользователя состоит из панели управления в левой части экрана, названия окна сверху и таблицы по центру. Пример интерфейса пользователя приведен на рисунке 6 (список товара).

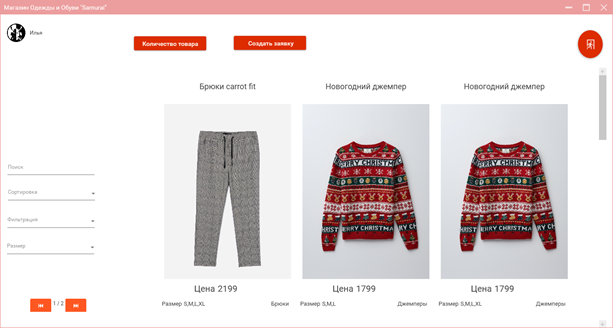


Рисунок 6 – Окно списка товара

## **2.5 Выбор средств разработки**

При выборе, я остановился на СУБД MS SQL Server и Windows Presentation Foundation C#.

Microsoft SQL Server — система управления реляционными базами данных (СУБД), разработанная корпорацией Microsoft. Основной используемый язык запросов — Transact-SQL, создан совместно [Microsoft](https://www.tadviser.ru/index.php/%D0%9A%D0%BE%D0%BC%D0%BF%D0%B0%D0%BD%D0%B8%D1%8F:Microsoft) и [Sybase](https://www.tadviser.ru/index.php/%D0%9A%D0%BE%D0%BC%D0%BF%D0%B0%D0%BD%D0%B8%D1%8F:Sybase" \o "Sybase). Transact-SQL является реализацией стандарта ANSI/ISO по структурированному языку запросов (SQL) с расширениями. Используется для работы с небольшими и средними по размеру базами данных до крупных баз данных масштаба предприятия; конкурирует с другими СУБД в этом сегменте рынка.

SQL является общепринятым интерфейсом к базам данных. «Все промышленные базы — [Oracle](https://www.tadviser.ru/index.php/%D0%9A%D0%BE%D0%BC%D0%BF%D0%B0%D0%BD%D0%B8%D1%8F:Oracle), [Microsoft SQL Server](https://www.tadviser.ru/index.php/%D0%9F%D1%80%D0%BE%D0%B4%D1%83%D0%BA%D1%82:Microsoft_SQL_Server), [PostgreSQL](https://www.tadviser.ru/index.php/PostgreSQL" \o "PostgreSQL), [MySQL](https://www.tadviser.ru/index.php/%D0%9F%D1%80%D0%BE%D0%B4%D1%83%D0%BA%D1%82:MySQL) — работают на SQL.

Рассмотрим, какие у Microsoft SQL Server преимущества и недостатки. Это нужно, чтобы вы понимали, насколько данная СУБД подходит для ваших целей.

Основные достоинства:

Масштабирование системы. Взаимодействовать с ней можно как на простых ноутбуках, так и на ПК с мощным процессором, который способен обрабатывать большой объем запросов;

Размер страниц – до 8 Кб. Данные извлекаются быстро, а сложную информацию удобнее хранить. Система обрабатывает транзакции в интерактивном режиме, есть динамическая блокировка;

Автоматизация рутинных административных задач. Например, управление блокировками и памятью, редактура размеров файлов. В программе продуманы настройки, можно создавать профили пользователей;

Удобный поиск. Его можно осуществлять по фразам, словам, тексту либо создавать ключевые индексы.

Поддержка работы с другими решениями Майкрософт, в том числе с Excel, Access.

Также в программе предусмотрена синхронизация, есть репликации через интернет, службы преобразования информации и полноценный web-ассистент для форматирования страниц. Дополнительно в нее интегрирован сервис интерактивного анализа (можно принимать решения, создавать корпоративные отчеты).

Основные минусы:

* Зависимость от ОС. Система работает только с Windows;
* Высокая стоимость.

Windows Presentation Foundation (WPF) представляет собой обширный API-интерфейс для создания настольных графических программ имеющих насыщенный дизайн и интерактивность. В отличие от устаревшей технологии Windows Forms, WPF включает новую модель построения пользовательских приложений (в основе WPF лежит мощная инфраструктура, основанная на DirectX).

Это означает возможность применения развитых графических эффектов, не платя за это производительностью, как это было в Windows Forms. Фактически даже становятся доступными такие расширенные средства, как поддержка видеофайлов и трехмерное содержимое. Используя эти средства (при наличии хорошего инструмента графического дизайна), можно создавать бросающиеся в глаза пользовательские интерфейсы и визуальные эффекты, которые были просто невозможны в Windows Forms.

Язык C#, разработанный компанией Майкрософт, один из самых популярных современных языков программирования. Он востребован на рынке разработки в различных странах, C# применяют при работе с программами для ПК, создании сложных веб-сервисов или мобильных приложений. Появившийся как язык для собственных нужд платформы Microsoft .NET, постепенно этот язык стал очень популярным. А потому мы решили сделать небольшой обзор для тех, кто выбирает, какие инструменты разработки стоит освоить в ближайшее время.

Итак, разработка языка началась в 1998 году, а первая версия увидела свет в 2001. Группой разработчиков руководил известный в профессиональных кругах специалист Андерс Хейлсберг. Новые версии C# выходят сравнительно часто, а текущие доработки, исправление багов и расширение библиотек ведется практически на постоянной основе.

В результате язык получился крайне гибкий, мощный и универсальный. На нем пишут практически все, что угодно, от небольших веб-приложений до мощных программных систем, объединяющих в себе веб-структуры, приложения для десктопов и мобильных устройств. Все это стало возможным благодаря удобному Си-подобному синтаксису, строгому структурированию, огромному количеству фреймворков и библиотек (их число достигает нескольких сотен).

Долгое время платформа .NET поставлялась с закрытым ядром, что создавало определенные сложности в разработке и снижало популярность C# в профессиональной среде. Но в ноябре 2014 Майкрософт радикально изменила подход и стала выдавать бесплатные лицензии для Visual Studio уже с открытым исходным кодом для всех наборов инструментов.

## **2.6 Тестирование программы**

Для тестирования функционала программы были проведены следующие тесты:

Таблица 1 – AuthoriationTest

|  |  |
| --- | --- |
| Test Case # | AuthoriationTest |
| Наименование проекта | Информационная система «Samurai» |
| Номер версии | 1.0 |
| Имя тестировщика | Шурхавецкий Олег |
| Даты тестирования | 22.02.2022 |
| Приоритет тестирования  *(*Малый/Средний/высокий*)* | Высокий |
| Название тестирования/Имя | Проверка окна авторизации на вхождение в систему |
| Резюме испытания | Ошибка авторизации |
| Шаги тестирования | Ввести данные на окне “Логин” и “Пароль”, и нажать кнопку войти. |
| Данные тестирования | Символы на окне авторизации. |
| Ожидаемый результат | При вводе правильного логина и пароля, пользователь заходит в свой личный кабинет. |
| Фактический результат | Пользователь вошел в личный кабинет |
| Предпосылки | После регистрации пользователь заходит на главное окно программы. |
| Постусловия |  |
| Статус  (Pass/Fail) | Pass |

Таблица 2 – Добавление заказа на товар

|  |  |
| --- | --- |
| Test Case # | AddTovarTest |
| Наименование проекта | Информационная система «Samurai» |
| Номер версии | 1.0 |
| Имя тестировщика | Шурхавецкий Олег |
| Даты тестирования | 22.02.2022 |
| Приоритет тестирования  *(*Малый/Средний/высокий*)* | Средний |
| Название тестирования/Имя | Проверка на добавление нового заказа в систему |
| Резюме испытания | Ошибка добавления |
| Шаги тестирования | Ввести данные на окне добавления и нажать кнопку принять. |
| Данные тестирования | Символы на окне добавления. |
| Ожидаемый результат | При вводе всех данных, клиент добавиться в систему. |
| Фактический результат | Клиент добавился в систему |
| Предпосылки |  |
| Постусловия |  |
| Статус  (Pass/Fail) | Pass |

Таблица 3 – Выход из учетной записи

|  |  |
| --- | --- |
| Test Case # | ExitToAutotivication |
| Наименование проекта | Информационная система «Samurai» |
| Номер версии | 1.0 |
| Имя тестировщика | Шурхавецкий Олег |
| Даты тестирования | 22.02.2022 |
| Приоритет тестирования  *(*Малый/Средний/высокий*)* | Средний |
| Название тестирования/Имя | Проверка на выход пользователя в системе. |
| Резюме испытания | Ошибка выхода . |
| Шаги тестирования | на окне выхода нажать на кнопку ок. |
| Данные тестирования | Символы на окне выхода. |
| Ожидаемый результат | При вводе всех данных, информация о клиенте изменилась. |
| Фактический результат | Информация о клиенте изменилась. |
| Предпосылки |  |
| Постусловия |  |
| Статус  (Pass/Fail) | Pass |

Таблица 4 – Загрузка списка Товара

|  |  |
| --- | --- |
| Test Case # | LoadTovar |
| Наименование проекта | Информационная система «Samurai» |
| Номер версии | 1.0 |
| Имя тестировщика | Шурхавецкий Олег |
| Даты тестирования | 22.02.2022 |
| Приоритет тестирования  *(*Малый/Средний/высокий*)* | Высокий |
| Название тестирования/Имя | Проверка на загрузку списка товара. |
| Резюме испытания | Ошибка загрузки списка товара из системы. |
| Шаги тестирования | Открыть форму списка товара. |
| Данные тестирования | Данные товара. |
| Ожидаемый результат | При открытии формы появился список товара. |
| Фактический результат | Появился список товара. |
| Предпосылки |  |
| Постусловия |  |
| Статус  (Pass/Fail) | Pass |

Таблица 5 – Загрузка списка поставщиков

|  |  |
| --- | --- |
| Test Case # | ProviderLoad |
| Наименование проекта | Информационная система «Samurai» |
| Номер версии | 1.0 |
| Имя тестировщика | Шурхавецкий Олег |
| Даты тестирования | 13.02.2022 |
| Приоритет тестирования  *(*Малый/Средний/высокий*)* | Высокий |
| Название тестирования/Имя | Проверка на вывод списка поставщика |
| Резюме испытания | Ошибка вывода. |
| Шаги тестирования | Открыть форму списка поставщиков. |
| Данные тестирования | Символы на окне поставщики сеансов. |
| Ожидаемый результат | При входе на форму выводится список поставщиков. |
| Фактический результат | Сеанс выводится в систему. |
| Предпосылки |  |
| Постусловия |  |
| Статус  (Pass/Fail) | Pass |

Таблица 6 – Количество товара

|  |  |
| --- | --- |
| Test Case # | TovarCoilchesvto |
| Наименование проекта | Информационная система «Samurai» |
| Номер версии | 1.0 |
| Имя тестировщика | Шурхавецкий Олег |
| Даты тестирования | 22.02.2022 |
| Приоритет тестирования  *(*Малый/Средний/высокий*)* | Высокий |
| Название тестирования/Имя | Проверка на вывод количества товара |
| Резюме испытания | Количество не было выведено в системе. |
| Шаги тестирования | Зайти на окно количество для отображения списка количество товара в системе. |
| Данные тестирования | Данные количество товара на складе. |
| Ожидаемый результат | Сколько было приобретено товара в магазине были потрачены. |
| Фактический результат | Товары были потрачены. |
| Предпосылки |  |
| Постусловия |  |
| Статус  (Pass/Fail) | Pass |

## **2.7 Внедрение программы**

Внедрение программного обеспечения — процесс настройки программного обеспечения под определённые условия использования, а также обучения пользователей работе с программным продуктом.

Для внедрения программы можно вынести несколько способов:

1. Разработка продукта и инсталляция – завершение разработки программы, затем подготовка к инсталляции продукта на ПК.

2. Выпуск программы – официальный выпуск программы на платформу ПК.

3. Обновление программы – выпуск новых версий, устраняющие имеющиеся проблемы и добавляющие новые функции.

4. Прекращение поддержки – выход последнего обновления для программы, устраняющие большинство проблем.

Основные этапы внедрения ПО:

1. Обследование
2. Разработка технического задания
3. Настройка системы
4. Тестирование системы
5. Опытная эксплуатация
6. Промышленная эксплуатация

Первый этап проекта – диагностика предприятия или его обследование.

В ходе первого этапа необходимо оценить возможность размещения ПО на оборудовании предприятия, исследовать сеть и рабочие места

Второй этап проекта внедрения программного продукта – разработка технического задания. Которая включает в себя описание всей хранимой информации, разграничения прав доступа пользователей и необходимые корректировки ПО под нужды предприятия.

Третий этап проекта – настройка системы (автоматизация). Настройка системы включает в себя формирование в программе всех справочников системы, настройка всех алгоритмов расчета, форм ввода и отчетных форм, ввод пользователей системы и настройка прав доступа. Продолжительность данного этапа напрямую зависит от квалификации специалистов и от уровня сложности поставленной задачи.

Четвертый этап проекта – тестирование программного продукта (системы). Тестирование системы включает в себя подготовку демонстрационного примера, внесение тестовых данных, проверку алгоритмов расчета и исправление обнаруженных ошибок.

Пятый этап проекта – опытная эксплуатация системы. Опытная эксплуатация системы включает в себя работу с реальными данными, но при этом параллельно используется прежняя старая система либо те электронные таблицы, в которых предприятия до настоящего момента осуществляла свою работу. Этот этап необходим для того, чтобы можно было сопоставить результаты работы в новой системе с результатами, которые получены были прежним способом.

Шестой этап проекта – промышленная эксплуатация системы. Промышленная эксплуатация системы подразумевает переход предприятия на новый программный продукт и отказ от всех альтернативных способов работы за рамками данной системы.

## **2.8 Программная документация**

### **2.8.1 Руководства программиста**

Рассмотрим структуру программы:

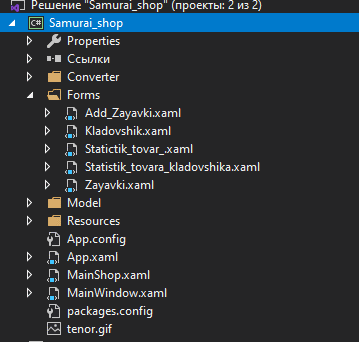


Рисунок 4 – Структура программы

* «Converters» папка, содержащая класс для отображения картинок в программе;
* «Model» папка, содержащая модель базы данных;
* «Forms» папка, содержащая формы программы;
* «Resourses» папка, содержащая логотип программы;

### **2.8.2 Руководство пользователя**

Приложение написано на языке C#, на платформе Windows Presentation Foundation (WPF), при запуске приложения открывается форма для авторизации сотрудников сети магазинов «Samurai»

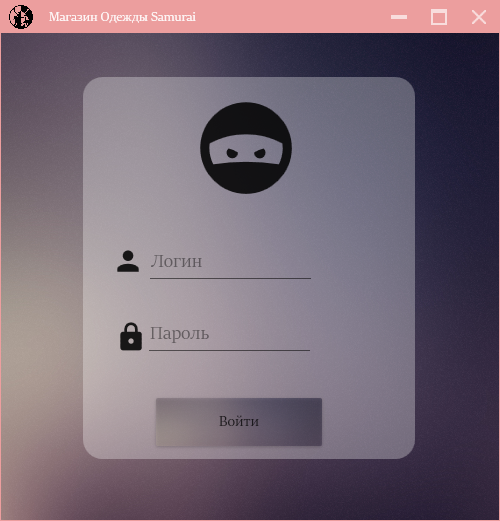


Рисунок 5 – Форма авторизации

Если сотрудник введет логин или пароль не верно ему вылезет ошибка «Неверный логин или пароль».

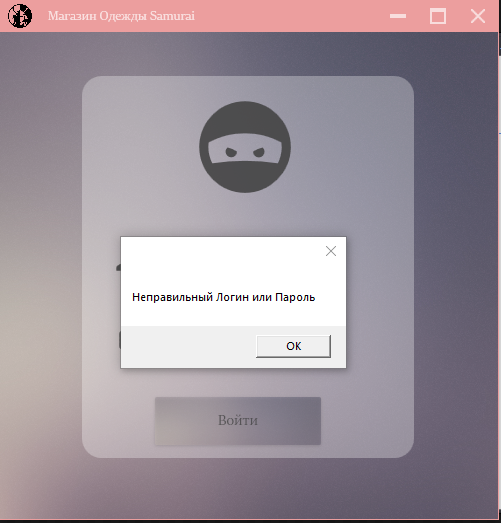


Рисунок 6 – Ошибка авторизации

После авторизации, открывается форма в зависимости от роли сотрудника, рассмотрим форму после авторизации сотрудника магазина. Интерфейс сотрудника магазина включает:

* Просмотр продуктов находящиеся в магазине;
* Поиск товара по наименованию;
* Сортировка по названию и стоимости товара;
* Фильтрация по типу товара;
* Фильтрация по размеру одежды.

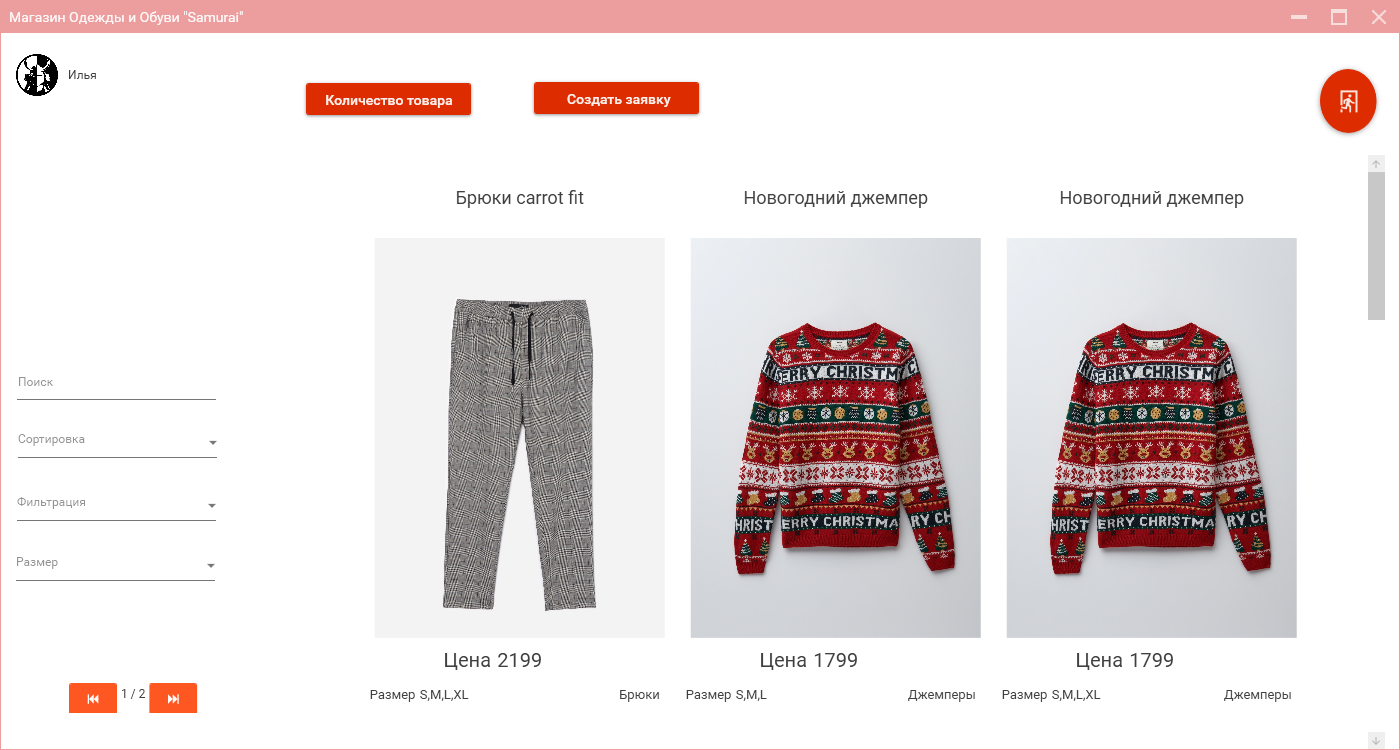


Рисунок 7 – Интерфейс сотрудника магазина

Сотрудник магазина может создать заявку для добавления нового товара в магазин или увеличить количество имеющегося. При нажатии на кнопку «Создать заявку» открывается форма создания заявки. В ней расположены два списка, первый список заполнен всеми товарами, которые находятся в сети магазинов. При введении количества товара и нажатии на кнопку «Добавить» товар добавляется во второй список.

У добавленного продукта есть кнопка «Удалить», которая удаляет выбранный продукт (Рисунок 9). При нажатии на кнопку «Количество товаров» открывается форма количество товаров. В ней расположен список с количеством товаров в магазинах (Рисунок 8).

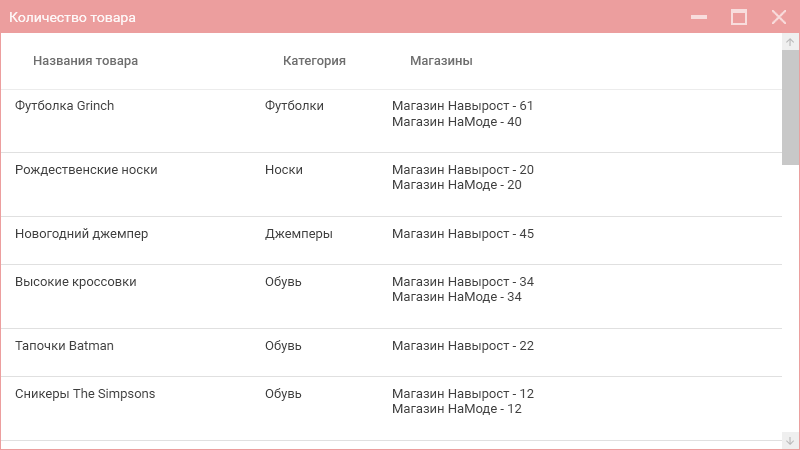


Рисунок 8 – Список с количеством товаров

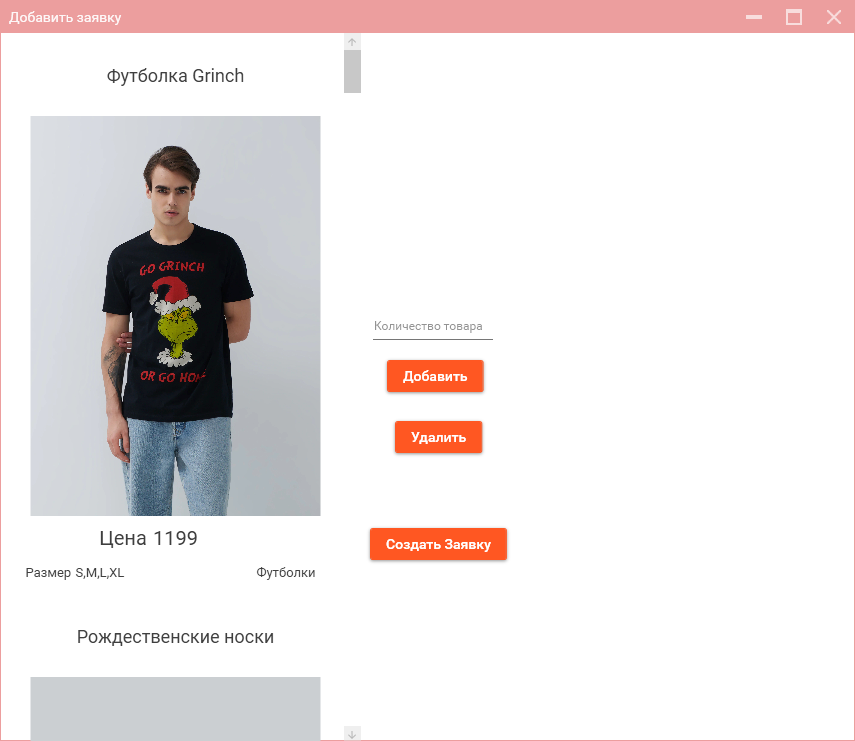


Рисунок 9 – Добавление продуктов для оформления заявки

При нажатии кнопки «Создать Заявку» формируется номер заявки, дата заявки и номер сотрудника, который создал заявку.

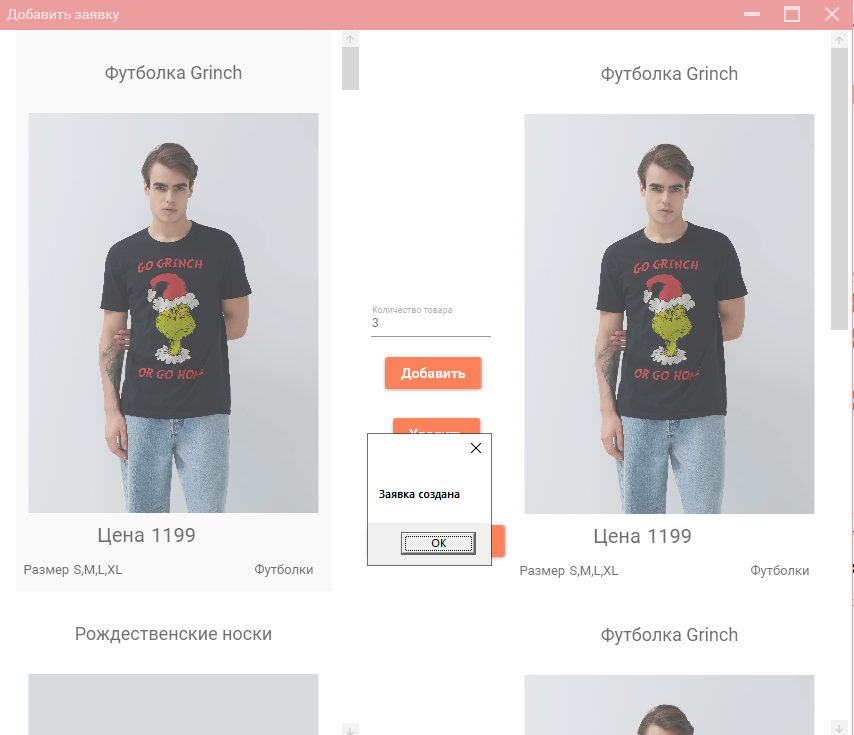


Рисунок 10 – Создание заявки

После создание заявки, кладовщик склада может её принять или отклонить.

Интерфейс кладовщик склада включает(Рисунок 11):

* Просмотр товара, находящегося на складе;
* Просмотр количества товара, находящегося на складе;
* Просмотр заявок, созданные сотрудниками магазина.
* Просмотр Статистики товара находящегося в магазине
* Поиск по Дате или Стоимости поставки
* Фильтрация по поставщикам

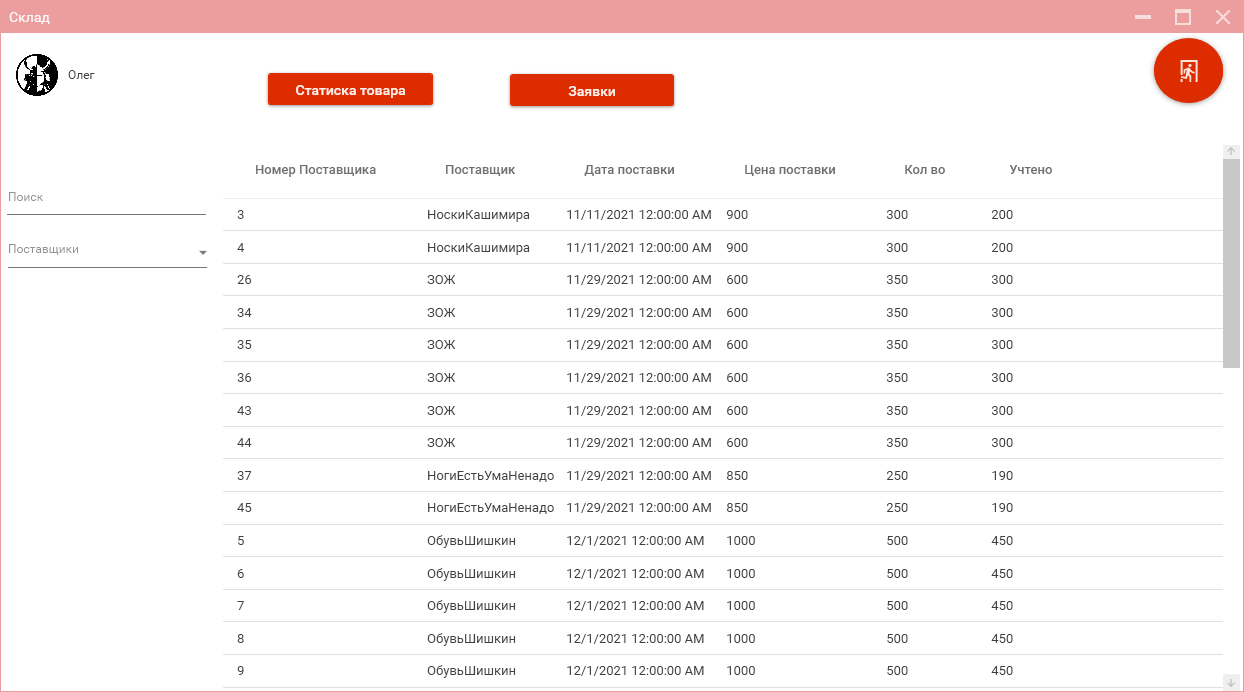


Рисунок 11 – Интерфейс сотрудника склада.

При нажатии кнопки «Заявки» открывается форма со всеми заявками сделанные сотрудниками магазинов (Рисунок 13). При нажатии на кнопку «Статистика товаров» открывается форма со статисткой отгрузки товара со склада. В ней расположены два списка, первый заполнен информацией о магазине. Во втором – общая информация о товарах (Рисунок 12).

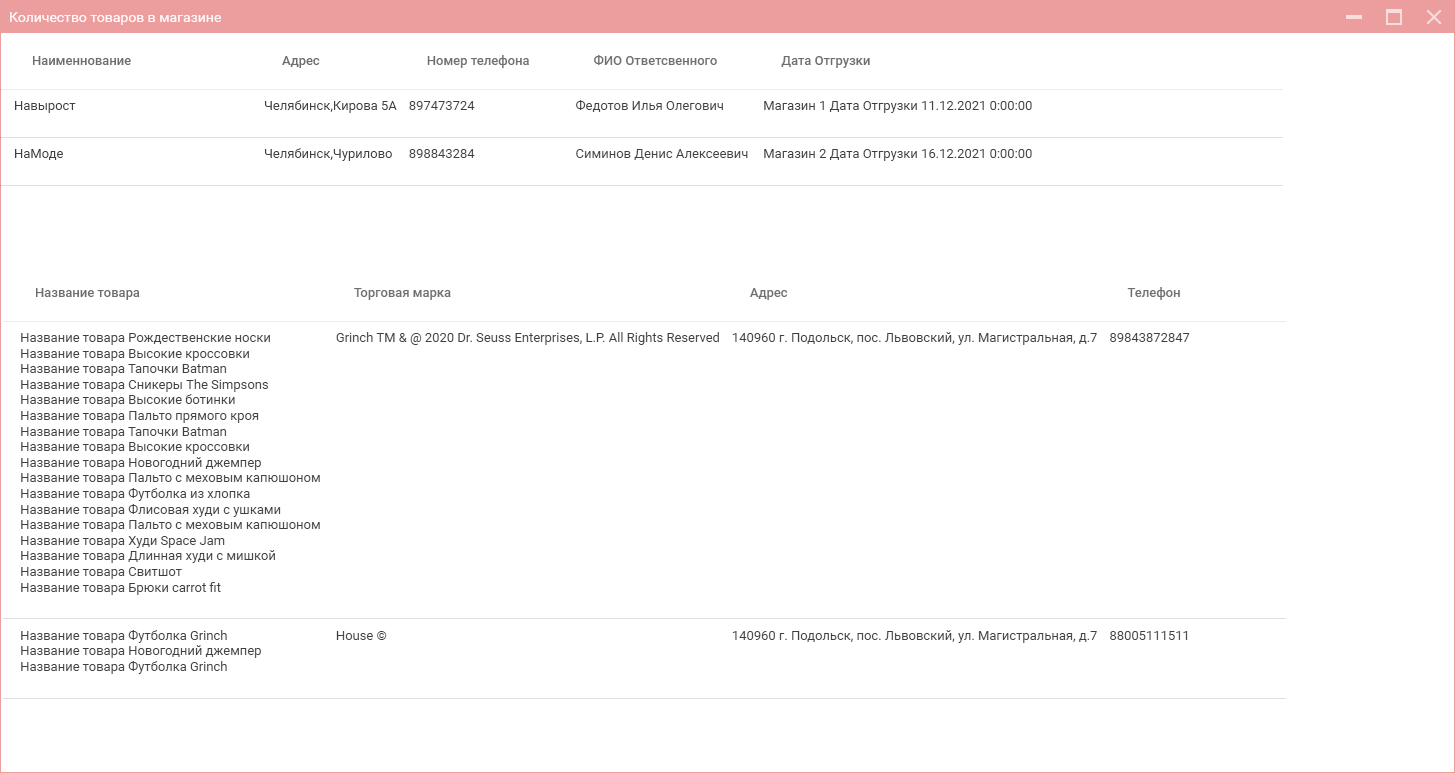


Рисунок 12 – Просмотр заявок.

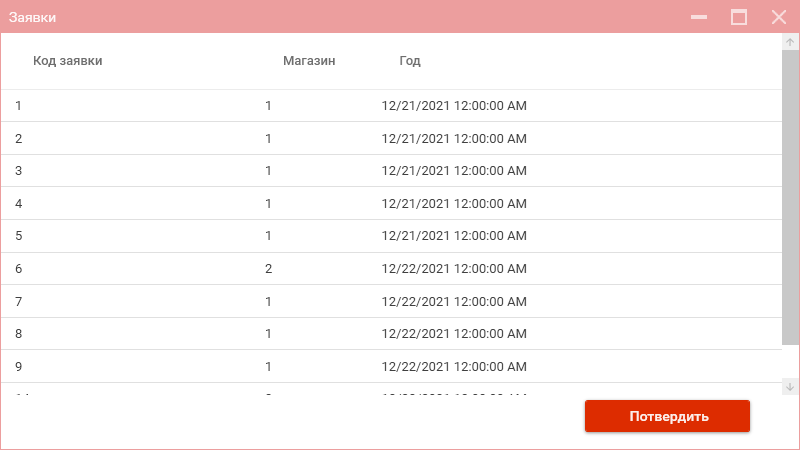


Рисунок 13– Просмотр заявок.

При нажатии кнопки «Подтвердить» выбранная заявка удаляется, а продукты, которые хранились на складе отправляются в магазин.

# **Заключение**

При работе над курсовым проектом были закреплены знания, полученные при написании приложения и документации.

Темой для курсового проекта была выбрана сеть «Магазина Одежды и обуви «Samurai»». Данная тема была выбрана для облегчения работы сотрудников со складом товаров и улучшения принятие заявки на складе для кладовщика товара.

Цель проекта была получение дополнительных знаний в сфере Разработки Баз Данных и приложений, связанных с ней.

Результаты проделанной работы в дальнейшем можно использовать при написании Дипломной работы. В приложение было добавлено множество функций, которые помогут пользователям удобно взаимодействовать с количеством товара в магазине и складом, а Кладовщик легко отслеживать товар в магазине и принимает заявки при необходимости товара.

# **Библиографический список**

1. Б. Гамильтон. ADO.NET сборник рецептов. Для профессионалов. — СПб.: Питер, 2005. — 576 с. — [ISBN 5-469-00239-X](https://ru.wikipedia.org/wiki/%D0%A1%D0%BB%D1%83%D0%B6%D0%B5%D0%B1%D0%BD%D0%B0%D1%8F:%D0%98%D1%81%D1%82%D0%BE%D1%87%D0%BD%D0%B8%D0%BA%D0%B8_%D0%BA%D0%BD%D0%B8%D0%B3/546900239X).
2. Роберт Э. Уолтерс, Майкл Коулс. SQL Server 2008: ускоренный курс для профессионалов = Accelerated SQL Server 2008. — М.: «Вильямс», 2008. — С. 768. — ISBN 978-5-8459-1481-1.
3. House: научная электронная библиотека: сайт. — Подольск,2021—.—URL: https://www.housebrand.com/ru/ru/(дата обращения 02.12.2021).
4. Convertion: научная электронная библиотека: сайт. — Athienou,2014—.—URL: https://convertio.co/ru/webp-jpg/(дата обращения 05.12.2021).
5. habr.com: Блог в сфере IT. URL - https://habr.com/ (дата обращения 09.12.2021)