**СОДЕРЖАНИЕ**

**Оглавление**

[**ВВЕДЕНИЕ 4**](#_3znysh7)

[**1. ОПИСАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ 6**](#_2et92p0)

[**1.1. Введение в предметную область 6**](#_tyjcwt)

[**1.2. Анализ готовых решений 10**](#_1t3h5sf)

[**2. ПРОЕКТИРОВАНИЕ СИСТЕМЫ 14**](#_4d34og8)

[**2.1. Общее назначение системы 14**](#_2s8eyo1)

[**2.2. Требования к функциям, выполняемым системой 14**](#_17dp8vu)

[**2.3. План тестирования 19**](#_3rdcrjn)

[**3. РЕАЛИЗАЦИЯ ПРОЕКТНОЙ СИСТЕМЫ 21**](#_26in1rg)

[**3.1. Описание среды разработки 21**](#_lnxbz9)

[**3.1.1. Выбор и описание программных инструментов 21**](#_35nkun2)

[**3.1.2. Обоснование выбора инструментария по разработке 22**](#_1ksv4uv)

[**3.2. Разработка программного модуля 23**](#_44sinio)

[**3.2.1. Реализация пользовательского интерфейса программы 23**](#_2jxsxqh)

[**3.2.2. Описание кодом функциональных узлов модуля 26**](#_z337ya)

[**3.2.3. Результат работы и тестирование 28**](#_3j2qqm3)

[**ЗАКЛЮЧЕНИЕ 29**](#_1y810tw)

[**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 30**](#_4i7ojhp)

[**ПРИЛОЖЕНИЕ 1 31**](#_2xcytpi)

[**ПРИЛОЖЕНИЕ 2 32**](#_pog9w2t6txjx)

[**ПРИЛОЖЕНИЕ 3 33**](#_uy5b62o86a3u)

[**ПРИЛОЖЕНИЕ 4 41**](#_w7k6xu7s9yq6)

# ВВЕДЕНИЕ

Современные технологии и высокая конкуренция на рынке требуют от компаний эффективного управления своими ресурсами, включая учет товаров на складе. Система учета товаров на складе играет ключевую роль в успешной деятельности организации, обеспечивая контроль за запасами, оптимизацию закупок, улучшение обслуживания клиентов и снижение издержек. Эффективная система учета является основой для оптимизации операций на складе и минимизации затрат. В условиях быстро меняющегося рынка организации сталкиваются с необходимостью повышения скорости обработки заказов, точности учета и обеспечения доступности товаров для клиентов.

Цель данной курсового проекта заключается в создании системы учета товаров на складе. Для достижения этой цели необходимо изучить существующие методы автоматизации учета товаров на складе, проанализировать их преимущества и недостатки учитывающую специфику предприятия.

Для достижения поставленной цели необходимо решить следующие задачи:

1. Проанализировать предметную область
2. Проанализировать готовое решение
3. Подготовить Техническое Задание
4. Подготовить план тестирования
5. Обосновать выбор инструментов и средств разработки
6. Описать реализацию Техническое Задание
7. Выполнить тестирование

Объектом исследования является процесс автоматизации учета товаров на складе. В рамках исследования будет рассмотрено, как организации осуществляют учет товаров, как осуществляется взаимодействие между различными подразделениями и как осуществляется работа со складами, включая прием, хранение и выдачу товаров.

Предметом исследования являются процессы, используемые для автоматизации создании, хранения и обработке анализа товаров на складе. Особое внимание будет уделено современным подходам к автоматизации этих процессов, а также использованию программного обеспечения для учета товаров и анализа складских остатков. Будут рассмотрены различные системы, как традиционные, так и автоматизированные, с целью выявления их сильных и слабых сторон.

Актуальность работы обусловлена растущей необходимостью в эффективных системах учета товаров в условиях глобализации и увеличения объема бизнеса. Современные предприятия сталкиваются с постоянным потоком товаров, возрастанием требований со стороны клиентов и усиливающейся конкуренцией на рынке. В такой ситуации наличие надежной и удобной системы учета на складе является залогом успешной и прибыльной работы компании. Кроме того, процессы учета и анализа помогут в принятии более обоснованных управленческих решений и оптимизации запасов.

Таким образом, рассматриваемый курсовой проект направлен на решение актуальной задачи по разработке эффективной системы учета товаров на складе. Во время работы будут проанализированы уже существующие подходы, выявлены основные проблемы и предложены рекомендации по их решению. Реализованный модуль обеспечит более высокий уровень точности учета и эффективности работы склада, что в свою очередь приведет к улучшению качества обслуживания клиентов и снижению затрат.

# 1 ОПИСАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ

## 1.1 Введение в предметную область

Система учета товаров на складе представляет собой важный элемент управления запасами, который играет ключевую роль в обеспечении эффективной работы торговых и производственных компаний. В условиях динамично меняющегося рынка, где потребительские предпочтения и объемы продаж могут меняться с высокой скоростью, правильный учет товаров становится необходимым для поддержания конкурентоспособности и повышения уровня обслуживания клиентов.

Учет товаров включает в себя процессы хранение, мониторинг и управление запасами. Эти процессы требуют точности и оперативности, поскольку ошибки в учете могут привести к финансовым потерям, недовольству клиентов и снижению репутации компании. С учетом растущего объема товарооборота и сложности логистических цепочек, традиционные методы учета становятся недостаточно эффективными.

Информатизация системы учета товаров на складе подразумевает внедрение современных информационных технологий, которые позволяют автоматизировать и оптимизировать процессы управления запасами. К основным преимуществам информатизации можно отнести:

1. Увеличение точности учета: Автоматизированные системы минимизируют вероятность ошибок, связанных с бумажным или иным физическим носителем, ввод данных которых требуется вручную, поэтому система позволяет более точно отслеживать движение товаров.
2. Скорость обработки данных: Современные технологии позволяют быстро обрабатывать большие объёмы информации, что значительно ускоряет процессы учета и принятия решений.
3. Улучшение прозрачности: Информатизация обеспечивает доступ к актуальной информации о состоянии запасов в реальном времени, что позволяет лучше контролировать процессы на складе.
4. Оптимизация затрат: Автоматизация помогает сократить затраты на трудозатраты, а также уменьшить издержки, связанные с хранением и управлением запасами.

Интеграция информационных технологий в систему учета товаров на складе основана на нескольких ключевых теоретических концепциях:

1. Системный подход: Учет товаров следует рассматривать как часть более широкой системы управления предприятием. Это требует интеграции различных бизнес-процессов и использования единой информационной платформы.
2. Процессный подход: Учет товаров должен быть организован как последовательность взаимосвязанных процессов, что позволяет оптимизировать каждую стадию — от поступления товара до его отгрузки.
3. Модульность: Современные системы учета часто имеют модульную архитектуру, что позволяет компаниям адаптировать систему под свои специфические нужды и легко добавлять новые функции по мере необходимости.
4. Использование стандартов: Внедрение общепринятых стандартов (например, GS1 для штрих-кодов) облегчает интеграцию различных систем и технологий, что способствует более эффективному учету товаров.
5. Аналитика и прогнозирование: Современные системы учета часто включают инструменты для анализа данных и прогнозирования спроса, что помогает компаниям более эффективно управлять запасами.

Создание системы учета товаров на складе требует учета современных требований и законодательных аспектов, связанных с информационными системами. Рассмотрим основные из них:

1. Требования к функциональности

Учет товаров: Система должна обеспечивать точный учет входящих и исходящих товаров, а также их остатка на складе.

Интерфейс пользователя: Удобный и интуитивно понятный интерфейс для пользователей с различными уровнями подготовки.

Интеграция: Система должна поддерживать интеграцию с другими системами (например, бухгалтерскими или ERP-системами).

2. Безопасность данных

Защита информации: Необходимо обеспечить защиту данных от несанкционированного доступа, утечек и потерь.

Резервное копирование: Регулярное резервное копирование данных для предотвращения их потери.

3. Законодательные аспекты.

Федеральный закон № 149-ФЗ "Об информации, информационных технологиях и о защите информации": Определяет основные требования к защите информации и ее обработке. [Интернет источник 11]

Федеральный закон № 152-ФЗ "О персональных данных": Регулирует обработку персональных данных, что актуально, если в системе будут храниться данные о сотрудниках или клиентах. [Интернет источник 12]

Федеральный закон № 187-ФЗ «О безопасности критической информационной инфраструктуры Российской Федерации» — закон, который описывает правила защиты IT-инфраструктуры на предприятиях, работающих в сферах, критически важных для государства. [Интернет источник 13]

Федеральный закон № 184-ФЗ «О техническом регулировании» — основной закон, который регулирует отношения, возникающие при разработке, принятии, применении и исполнении обязательных и добровольных требований к продукции или к связанным с ними процессам, а также при оценке соответствия. [Интернет источник 14]

ЕСПД (Единая система проектной документации): Установленные требования к проектированию и разработке программных продуктов, в том числе к документации. [Интернет источник 15]

4. Требования к доступности

Мобильность: Возможность доступа к системе с различных устройств (ПК, планшеты, смартфоны).

Кроссплатформенность: Поддержка различных операционных систем и браузеров.

5. Требования к производительности

Скорость обработки данных: Система должна обеспечивать быструю обработку запросов и операций.

Масштабируемость: Возможность увеличения объема обрабатываемых данных без значительного ухудшения производительности.

6. Поддержка стандартов

Использование стандартов обмена данными: Например, XML или JSON для интеграции с другими системами.

Стандарты кодирования: Применение общепринятых стандартов программирования для обеспечения качества кода.

7. Тестирование и качество

Тестирование: Проведение тестирования системы на всех этапах разработки для выявления и устранения ошибок.

Документация: Подробная документация по разработке и эксплуатации системы

При создании системы была выбрана методология разработки Waterfall (каскадная или водопадная модель). Классическая поэтапная методология, в которой каждый следующий шаг начинается только после завершения предыдущего.

Разработка системы учета товаров на складе включает несколько ключевых этапов:

1. Анализ текущих процессов: на первом этапе необходимо провести анализ существующих бизнес-процессов учета товаров, выявить их недостатки и определить области для улучшения.
2. Определение требований: на основе анализа формулируются требования к новой системе учета, включая функциональные и нефункциональные характеристики.
3. Проектирование системы: на этом этапе разрабатывается архитектура системы, выбираются технологии и инструменты для реализации проекта.
4. Разработка и тестирование: создание программного обеспечения с последующим тестированием для выявления и устранения ошибок
5. Внедрение системы: Обучение сотрудников, переход на новую систему учета и интеграция с другими бизнес-процессами компании.
6. Мониторинг и оптимизация: после внедрения важно постоянно отслеживать эффективность работы системы и вносить необходимые коррективы для ее оптимизации.

## 1.2 Анализ готовых решений

На современном рынке существует несколько информационных систем, которые активно используются в сфере учетов товаров на складе. Рассмотрим наиболее популярные решения:

1С: Управление торговлей — это комплексное решение для автоматизации учета и управления торговыми процессами в малом и среднем бизнесе. Система позволяет вести учет товаров на складе, управлять продажами и закупками, а также анализировать финансовые результаты.

Функциональные возможности:

* Учет товаров: Подробный учет остатков, движений и стоимости товаров.
* Инвентаризация: Возможность проведения инвентаризаций с формированием отчетов.
* Отчеты: Генерация аналитических отчетов по остаткам, продажам и закупкам.
* Интеграция: Возможность интеграции с другими модулями 1С (бухгалтерия, CRM).

Преимущества:

* Широкая функциональность: Подходит для различных отраслей.  
  Локальная поддержка: Доступность специалистов для консультаций и обучения.
* Гибкость настройки: Возможность адаптации под специфические потребности бизнеса.

Недостатки:

* Сложность освоения: Требует обучения для эффективного использования.
* Стоимость: Лицензии и обслуживание могут быть дорогими для малых предприятий.

SAP Business One — это ERP-система, разработанная для малых и средних предприятий. Она охватывает все аспекты управления бизнесом, включая учет товаров на складе, финансы и продажи.  
Функциональные возможности:

* Управление запасами: Отслеживание остатков и автоматизация процессов пополнения.
* Аналитика: Инструменты для анализа данных и прогнозирования спроса.
* Интеграция: Возможность интеграции с другими решениями SAP и сторонними приложениями.
* Мобильный доступ: Поддержка мобильных устройств для управления складом на ходу.

Преимущества:

* Гибкость и масштабируемость: Подходит как для небольших, так и для крупных компаний.
* Полноценная ERP-система: Объединяет все бизнес-процессы в одном решении.
* Международный опыт: Используется во многих странах, что подтверждает надежность.

Недостатки

* Высокая стоимость внедрения: Может быть недоступна для малых предприятий.
* Сложность настройки: Требует значительных усилий для настройки под специфические нужды.

# 2 ПРОЕКТИРОВАНИЕ СИСТЕМЫ

## 2.1 Общее назначение системы

Создание системы учета товаров на складе направлено на оптимизацию управления запасами и улучшение процессов, связанных с хранением и управлением товарами. Основная цель данной системы заключается в автоматизации учета и контроля за движением товаров. Это обеспечивает получение точной и актуальной информации о количестве и состоянии запасов в реальном времени, что критически важно для оперативного принятия решений.

Ключевым аспектом системы является мониторинг запасов, который позволяет более эффективно управлять ими. Это включает отслеживание уровня запасов, что помогает избежать как избытка, так и нехватки товаров.

Оптимизация хранения является еще одной важной задачей системы учета. Она способствует более рациональному размещению товаров на складе, что повышает эффективность использования складских площадей.

Кроме того, система учета товаров на складе включает управление сроками хранения, что особенно важно для товаров с ограниченным сроком годности. Это позволяет контролировать их движение и предотвращать потери.

Созданный программный продукт предназначен для внедрения в организации различного масштаба — от малых предприятий до крупных корпораций, работающих в различных секторах, связанных с товарооборотом. Важно отметить, что данное программное обеспечение обладает высокой степенью масштабируемости.

## 2.2. Требования к функциям, выполняемым системой

1. Для системы учета товаров на складе, улучшение взаимодействия с клиентом может включать в себя следующие функциональные задачи:
   * + 1. Управление товарами:

Добавление нового товара: Ввод информации о товаре (название, описание, дата добавления, количество, категория).

Редактирование товара: Обновление данных о существующем товаре.

Удаление товара: Удаление товара из системы.

* + - 1. Управление поставщиками:

Управление данными о поставщиках: добавление и удаление их из системы. Мониторинг количества.

* + - 1. Управление пользователями и правами доступа:

Создание и редактирование учетных записей пользователей: Управление доступом к системе для различных ролей.

Настройка прав доступа: Определение, какие функции доступны различным пользователям.

Каждая функция должна выполняться быстро и без ошибок, обеспечивая надежное и удобное использование системы пользователем.

Таблица 1 – Объекты предметной области

|  |  |
| --- | --- |
| № | Наименование объекта |
| 1 | Goods |
| 2 | Goods\_type |
| 3 | Suppliers |
| 4 | Roles |
| 5 | Accounts |

После того как мы описали объекты нашей предметной области, рассмотрим атрибуты, присущие каждому из них.

Таблица 2 – Сущность Goods

| Атрибут | Тип | Описание |
| --- | --- | --- |
| Goods\_id | INTEGER NOT NULL | Уникальный идентификатор товара |
| Title | VARCHAR NOT NULL | Название товара |
| Quantity | INTEGER NOT NULL | Количество товара на складе |
| Type\_id | INTEGER NOT NULL | Идентификатор типа товара |
| Add\_date | TEXT NOT NULL | Дата добавления товара в систему |
| Comment | TEXT NOT NULL | Дополнительные комментарии или примечания к товару |

Таблица 3 – Сущность Godds\_type

|  |  |  |
| --- | --- | --- |
| Атрибут | Тип | Описание |
| Goods\_type\_id | INTEGER NOT NULL | Уникальный идентификатор типа товара |
| Type\_name | VARCHAR NOT NULL | Название типа товара |
| Comment | TEXT NOT NULL | Дополнительные комментарии или описания типа товара |

Таблица 4 – Сущность Suppliers

|  |  |  |
| --- | --- | --- |
| Атрибут | Тип | Описание |
| Id | INTEGER NOT NULL | Уникальный идентификатор поставщика |
| Name | VARCHAR NOT NULL | Название поставщика |
| Address | TEXT NOT NULL | Адрес поставщика |
| Phone\_number | VARCHAR NOT NULL | Номер поставщика |

Таблица 5 – Сущность Roles

|  |  |  |
| --- | --- | --- |
| Атрибут | Тип | Описание |
| Id | INTEGER NOT NULL | Уникальный идентификатор роли |
| Role\_name | VARCHAR NOT NULL | Название роли |

Таблица 6 – Сущность Accounts

| Атрибут | Тип | Описание |
| --- | --- | --- |
| Id | INTEGER NOT NULL | Уникальный идентификатор аккаунта пользователя |
| Login | VARCHAR NOT NULL | Уникальное имя пользователя для входа в систему |
| Password | VARCHAR NOT NULL | Пароль пользователя для аутентификации |
| Role\_id | INTEGER NOT NULL | Идентификатор роли пользователя |

После того как все таблицы были созданы, нужно установить связи между ними.

Связь Goods - Goods\_type (Товар – Тип товара)

У одного Типа товара может быть привязано множество товаров. Связь один-ко-многим.



рисунок 1 – Связь «Тип товара – Товар»

Связь Role - Accounts (Роль – Пользователь)

У одной роли может быть множество пользователей. Связь один-ко-многим.



рисунок 2 – Связь «Роль – Пользователь»

На основе выявленных сущностей построим ER-диаграмму базы данных системы учета товаров на складе. Она показана на рисунке 3.

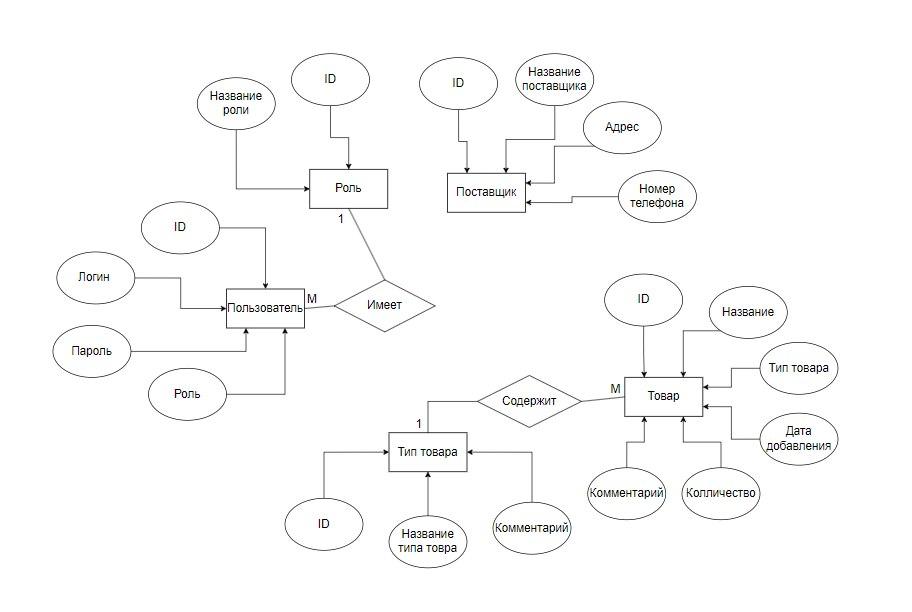


рисунок 3 – ER-диаграмма базы данных системы учета товаров на складе

На рисунке 4 представлена диаграмма IDEF1X, которая используется для создания концептуальных, логических и физических моделей данных

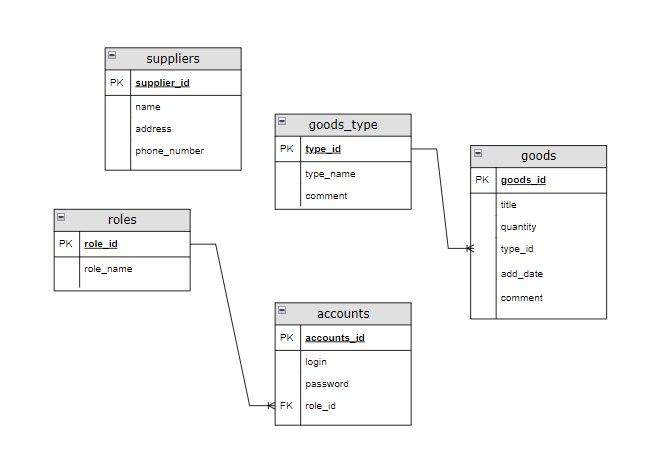


Рисунок 4 – IDEF1X диаграмма базы данных системы учета товаров на складе

## 2.3. План тестирования

Цель тестирования модуля «Система учета товаров на складе» — подтвердить его качество и соответствие функциональным и нефункциональным требованиям. Тестирование направлено на выявление ошибок, проверку правильности выполнения функций, а также оценку удобства, производительности, безопасности и удовлетворения потребностей пользователей.

Задачи тестирования включают:

1. Оценка производительности и стабильности системы:

* Тестирование поведения системы при длительной непрерывной работе.
* Проверка стабильности работы интерфейса системы.
* Нагрузочное тестирование системы, при большом количестве пользователей.

1. Тестирования удобства системы:

* Тестирование удобства интерфейса, как для новых пользователей, так и для старых.
* Тестирование информативности системы, наличие всплывающих подсказок.

1. Тестирование на совместимость с различными средами запуска:

* Проверка корректного отображения интерфейса модуля с различными разрешениями экранов.

1. Тестирование корректности работы системы в соответствии с заявленными спецификациями:

* Управление товарами: добавление, удаление и изменение.
* Управление данными поставщиков их добавление и удаление.
* Также управление пользователями, взаимодействие с ними.

# 3. РЕАЛИЗАЦИЯ ПРОЕКТНОЙ СИСТЕМЫ

## 3.1. Описание среды разработки

Visual Studio Code (VS Code) — это универсальная и востребованная среда разработки, созданная компанией Microsoft. Она поддерживает работу с разными языками программирования и предоставляет множество инструментов, которые делают процесс создания программ более комфортным и результативным. Вот некоторые ключевые особенности VS Code:

1. Легковесность и производительность: VS Code является легковесным редактором, который загружается быстро и работает плавно, даже на менее мощных машинах.

2. Поддержка множества языков: VS Code поддерживает множество языков программирования, таких как JavaScript, TypeScript, Python, Java, C#, PHP, Go и многие другие. Дополнительные языки можно подключить через расширения.

3. Расширяемость: Одним из преимуществ VS Code является его система расширений. Пользователи могут устанавливать дополнительные модули, которые расширяют функциональность редактора, добавляют поддержку языков, меняют оформление и предоставляют новые инструменты.

4. Интеграция с системами контроля версий: VS Code имеет встроенную поддержку Git, что позволяет пользователям легко управлять версиями кода, просматривать изменения и выполнять коммиты прямо из редактора.

5. Интерфейс: Который легко освоить, включает в себя панель с боковой стороны для перемещения по файлам, встроенный терминал и панель инструментов, которая позволяет быстро получить доступ к функциям.

## 3.1.1. Выбор и описание программных инструментов

Программные инструменты, выбранные для разработки модуля:

Golang (GO) - это язык программирования, разработанный в Google в 2007 году и официально выпущенный в 2009 году. Он был создан с целью улучшения разработки программного обеспечения, особенно в контексте больших систем и распределенных приложений.

Fyne - это библиотека для разработки графических интерфейсов пользователя (GUI) на языке Golang. Она позволяет создавать кроссплатформенные приложения, которые могут работать на различных операционных системах, включая Windows, macOS, Linux, а также на мобильных платформах Android и iOS.

SQLStudio — это инструмент для работы с базами данных, который предоставляет графический интерфейс для выполнения SQL-запросов и управления базами данных. Он позволяет пользователям визуально взаимодействовать с базами данных, упрощая процесс создания, редактирования и удаления данных.

SQL - язык запросов, который применяется для создания, обработки и хранения информации в реляционных базах данных.

## 3.1.2. Обоснование выбора инструментария по разработке

Выбор языка программирования Golang основан на следующих преимуществах:

1. Простота: Go язык программирования с простым и понятным синтаксисом, что делает его удобным для изучения и применения. Благодаря этому разработчики могут быстро освоить язык и писать чистый и понятный код.
2. Статическая типизация: Go является статически типизированным языком, что позволяет выявлять ошибки на этапе компиляции. Это повышает надежность кода.
3. Параллелизм: Одной из ключевых особенностей Go является встроенная поддержка параллелизма через грутины и каналы. Горутины (goroutines) позволяют легко запускать функции параллельно, а каналы обеспечивают безопасный обмен данными между ними.
4. Сборка мусора: Go включает автоматическую сборку мусора, что упрощает управление памятью и снижает вероятность утечек памяти.
5. Стандартная библиотека: Go имеет богатую стандартную библиотеку, которая предоставляет множество инструментов для работы с сетями, файловыми системами, криптографией и многим другим.
6. Кроссплатформенность: Go поддерживает кроссплатформенную разработку, позволяя компилировать программы для различных операционных систем (Windows, macOS, Linux) из одной кодовой базы.

## 3.2. Разработка программного модуля

В этой части курсового проекта представлен анализ процесса создания программного модуля «Система учета товаров на складе». Разработка включает в себя техническую реализацию требований проекта с применением выбранных программных средств и технологий. В данном разделе рассматриваются ключевые аспекты создания программного обеспечения: проектирование пользовательского интерфейса (UI), взаимодействие кода с базой данных, а также реализацию основных функций.

## 3.2.1. Реализация пользовательского интерфейса программы

Чтобы взаимодействие с пользователя с модулем системы учета товаров на складе было продуктивным, был разработан и внедрен интуитивно понятный и удобный интерфейс.

Структура интерфейса содержит:

* + - 1. Меню авторизации и регистрации пользователя.
      2. Окно просмотра товаров, содержащихся на складе.
      3. Окно редактирования товаров.
      4. Окно добавления товаров.
      5. Окно просмотра поставщиков.
      6. Окно просмотра и редактирования данных о поставщиках
      7. Окно просмотра пользователя, под которым была авторизация.
      8. Окно где можно просматривать остальных пользователей, а также удалять и добавлять новых

Проектирование пользовательского интерфейса включает в себя не только корректное расположение элементов в окне, но и грамотное представление необходимой информации, а также учет пользовательских действий. Это, в свою очередь, способствует оптимизации пользовательского опыта.

Разработанный интерфейс пользователя, реализованный на языке программирования Golang, показан на “Рисунок 5”.

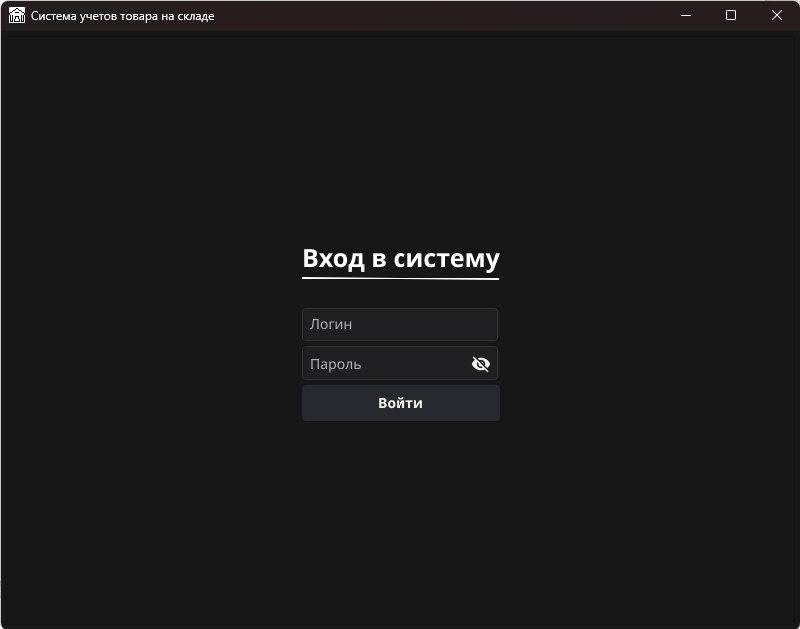


Рисунок 5 – Окно аутентификации пользователя

* В верхней части окна расположена надпись, которая указывает на необходимую аутентификацию в системе.
* Ниже расположены поля Логин и Пароль, которые необходимо заполнить чтобы войти в систему.
* Под полями расположена кнопка «Войти», которая реализует вход в систему.

В модуле предусмотрены и другие окна, каждое из которых предназначено для определенных задач.

Окно главного меню (ПРИЛОЖЕНИЕ 3, рисунок 6):

* В главном меню представлены функциональные поля с информацией о товаре, включая его ID, название и количество. Нажав на поле, вы сможете получить более подробную информацию о товаре, а также иметь возможность удалить его или внести изменения.
* Ниже полей расположена кнопка, перейдя по которой можно добавить товар.
* Слева под надписью «Вкладки» расположены кнопки, с помощью которых можно переключать между вкладками «Склад» и «Поставщики».
* Справа, в верхнем углу расположена кнопка, которая реализует переход в окно с управлением пользователем.

Окно добавить товар (ПРИЛОЖЕНИЕ 3, рисунок 7)

* В окне «Добавить товар» расположены поля для заполнения: Название товара, количество, id-категория, дата добавления, комментарий.
* Ниже представлены кнопки «Cancel», чтобы отменить и «Submit», чтобы подтвердить добавление.

Окно подробнее о товаре (ПРИЛОЖЕНИЕ 3, рисунок 8)

* В данном окне расположена подробная информация о товаре. Его id, название, количество, тип товара, дата добавления и описание.
* Ниже расположены две кнопки «Изменить» и «Удалить» функционал которых соответствует их названию.
* И кнопка «Назад» в углу слева, чтобы вернуться назад.

Окно поставщики (ПРИЛОЖЕНИЕ 3, рисунок 9)

* В окне "Поставщики" отображаются поля с информацией о поставщиках, включая их названия, телефоны и адреса. Каждое из этих полей имеет кнопку "Удалить", которая позволяет легко удалять ненужные записи.
* В нижней части окна находится кнопка «Добавить поставщика». Нажав на нее, вы сможете легко добавить нового поставщика в систему.
* В левой части также расположены вкладки, между которыми можно переключаться.

Окно добавить поставщика (ПРИЛОЖЕНИЕ 3, рисунок 10)

* В данном окне расположены поля для ввода данных о поставщике: название, адрес, телефон.
* Ниже расположена кнопка «Submit»

Окно профиль (ПРИЛОЖЕНИЕ 3, рисунок 11)

* В данном окне расположены данные об аккаунте, под которым был совершен вход в систему: ID, логин, права доступа.
* Ниже расположена кнопка «Выйти» для выхода из системы.
* Слева вкладки между аккаунтом и пользователями, которые можно переключать.

Окно пользователи (ПРИЛОЖЕНИЕ 3, рисунок 12)

* По центру в данном окне показаны все пользователи, зарегистрированные в системе и кнопка «Удалить» под каждым из них для удаления.
* Ниже расположена функциональная кнопка «Добавить пользователя» с помощью которой можно добавлять пользователей в систему.

## 3.2.2. Описание кодом функциональных узлов модуля

При запуске программы первое что видит пользователь — это окно аутентификации. Пользователь должен зайти под своими данными. Функция для входа показана в «ПРИЛОЖЕНИЕ 3, рисунок 13». Окно для входа в систему показано на «Рисунок 5».

С помощью функции NewGoodForm пользователь может добавить товар в систему. Код с данной функцией показана в «ПРИЛОЖЕНИЕ 3, рисунок 14».

Функция SuppliersWindow отвечающая за переключение между вкладками «Склад» и «Поставщики» показана в «ПРИЛОЖЕНИЕ 3, рисунок 15».

Функция AddSuppliers отвечает за добавление поставщика, часть кода которой показана в «ПРИЛОЖЕНИЕ 3, рисунок 16»

Кнопка профиль, отвечающая за переход на новое окно, работает при помощи функции ProfileWindow часть кода, которой изображена в «ПРИЛОЖЕНИЕ 3, рисунок 17».

Функция ProfileToolBar отвечает за переключение между вкладками «Аккаунт» и «Пользователи». Фрагмент данной функции изображен в «ПРИЛОЖЕНИЕ 3, рисунок 18».

Функция NewUserWindow предназначена для добавления нового пользователя в систему код, который изображен в «ПРИЛОЖЕНИЕ 3, рисунок 19».

Функция ProfileWindow также отвечает за выход из системы, часть кода, отвечающая за выход, показана в «ПРИЛОЖЕНИЕ 3, рисунок 20».

## 3.2.3. Результат работы и тестирование

В данном разделе будут описаны тест кейсы, с которыми можно будет подробнее ознакомиться в “ПРИЛОЖЕНИЕ 2”.

Кейс "Вход в систему": Проверяет успешный вход пользователя. При вводе корректного логина и пароля предоставляется доступ к интерфейсу. В случае ошибки выводится сообщение о необходимости проверки данных.

Кейс "Добавление товара на склад": Проверяет добавление нового товара. Пользователь заполняет поля: название, количество, дату и комментарий. После нажатия "Добавить" система подтверждает успешное добавление. При пустых обязательных полях или некорректных данных выводится уведомление об ошибке.

Кейс "Добавление поставщиков": Проверяет процесс добавления нового поставщика. Пользователь вводит название, адрес и телефон. После нажатия "Добавить" система подтверждает успешное добавление. При пустых полях или неверных данных выводится сообщение об ошибке.

Кейс "Добавление пользователя": Проверяет добавление нового пользователя администратором. Вводятся логин, пароль и роль. После нажатия "Добавить" система создает пользователя и отображает сообщение об успехе. Если логин уже существует или обязательные поля не заполнены, выводится уведомление об ошибках.

# ЗАКЛЮЧЕНИЕ

В ходе выполнения курсовой работы на тему "Система учета товаров на складе" была разработана система, обеспечивающая эффективный учет и управление товарными запасами, поставщиками и пользователями. Основные функциональные модули системы включают мониторинг и добавление товаров, мониторинг и добавление поставщиков, а также управление пользователями.

Для реализации поставленных задач был проведен анализ существующих решений, определены основные требования к системе, а также разработана её структура. Процесс разработки включал в себя создание технического задания, разработку пользовательского интерфейса, реализацию функционала системы и последующее его тестирование.

Тестирование показало, что разработанный модуль соответствует всем заявленным требованиям к системе.

Стоит отметить, что для корректной работы программы ее необходимо постоянно поддерживать обновлениями, чтобы соответствовать меняющимся стандартам и требованиям к программному обеспечению.

Анализ результатов работы показал, что автоматизация учета товаров на складе значительно повышает эффективность управления запасами, снижает затраты на трудозатраты сотрудников. Внедрение данной системы может способствовать оптимизации бизнес-процессов и повышению общей производительности компании.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

Стандарты и законодательные материалы

1. Федеральный закон от 27 июля 2006 года № 152-ФЗ. Закон регулирует обработку и защиту персональных данных граждан. При разработке модуля продажи авиабилетов необходимо учитывать требования этого закона для обеспечения конфиденциальности и безопасности данных пользователей, таких как имя, контактная информация и данные платежных карт.
2. ГОСТ Р 53325-2012. Стандарт устанавливает требования к качеству и безопасности услуг общественного транспорта, включая авиаперевозки. Он охватывает аспекты, связанные с комфортом пассажиров, точностью выполнения расписания и безопасностью перевозок.
3. ГОСТ Р ИСО/МЭК 27001-2012. Стандарт определяет требования к системам управления информационной безопасностью, включая оценку рисков и управление ими.

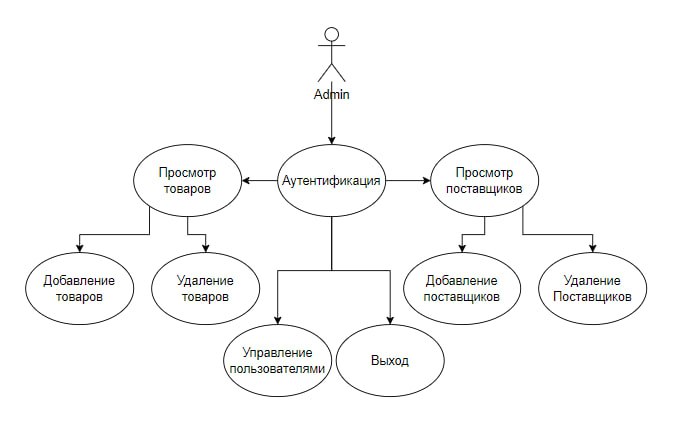
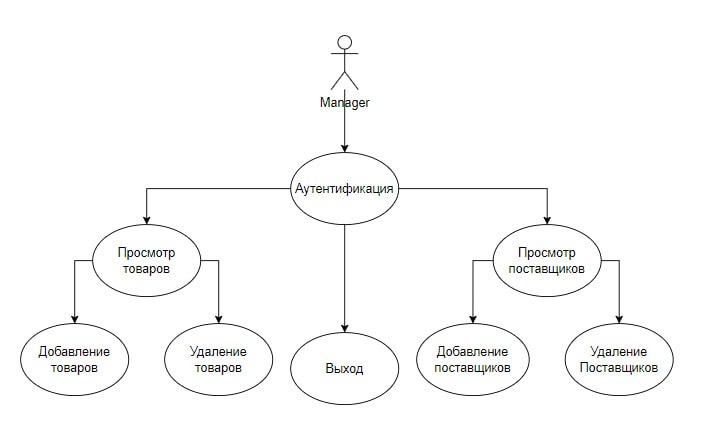
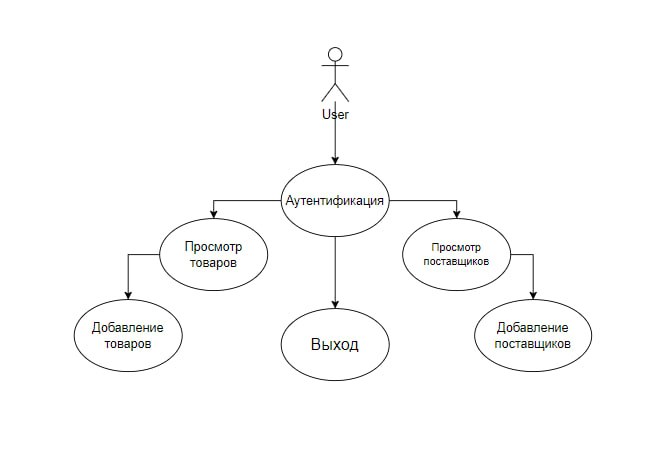
Учебники и учебные пособия

1. Kernighan B. W., Pike R. - The Go Programming Language, 2018.
2. Громов П. Эффективное программирование на Go. Издательство “БХВ-Петербург”, 2018.
3. Сидоров Е. - Основы разработки GUI-приложений на Go с использованием Fyne, 2019.
4. Соловьев Р. - Интерфейсы в Go: как их использовать и создавать. Издательство “Диалектика”, 2019.
5. Курочкин В. - Программирование на Go: от новичка до профессионала. Издательство “БХВ-Петербург”, 2020.
6. Кузнецов И. - Создание приложений на Go: от идеи до реализации. Издательство “БХВ-Петербург”, 2021.

Интернет источники

1. Документация по Go - <https://golang.org/doc/>, 2018
2. Документация Fyne - <https://fyne.io/develop/>, 2019
3. Документация по работа с СУБД - <https://metanit.com/go/tutorial/10.1.php>
4. Работа с sqlx в Go - <https://habr.com/ru/companies/vk/articles/692062/>
5. Golang: Основы - <https://tproger.ru/translations/golang-basics>
6. Создание desktop приложения с помощью Golang и Fyne - <https://habr.com/ru/articles/497148/>
7. Идеальная настройка вашего Golang проекта - <https://nuancesprog.ru/p/5966/>
8. Работа с SQL в Go - <https://www.freecodecamp.org/news/how-to-work-with-sql-databases-in-go/>
9. Принципы SOLID - <https://habr.com/ru/companies/ruvds/articles/426413/>
10. ООП в Go - <https://habr.com/ru/articles/243593/>
11. 149-ФЗ <https://www.consultant.ru/document/cons_doc_LAW_61798/>
12. 152-ФЗ <https://www.consultant.ru/document/cons_doc_LAW_61801/>
13. 187-ФЗ <https://www.consultant.ru/document/cons_doc_LAW_220885/>
14. 184-ФЗ <https://www.consultant.ru/document/cons_doc_LAW_40241/>
15. ЕСПД - <https://www.swrit.ru/gost-espd.html>

# ПРИЛОЖЕНИЕ 1

1. 
2. Рисунок 21 – Диаграмма прецедентности для пользователя Admin
3. 
4. Рисунок 22 – Диаграмма прецедентности для пользователя Manager
5. 
6. Рисунок 22 – Диаграмма прецедентности для пользователя Manager

# ПРИЛОЖЕНИЕ 2

**План-тест**

| № | Наименование функциональности | Наименование поля | Тестовый набор | Ожидаемый результат |
| --- | --- | --- | --- | --- |
| 1 | Авторизация | Логин | Ввести корректный логин:vlad | Успешный вход в систему |
| Ввести некорректный логин:vlad\_99 | Ошибка: Такого аккаунта не существует |
| Оставить поле пустым: | Ошибка: Заполните все поля |
| Пароль | Ввести корректный пароль | Успешный вход в систему |
| Ввести некорректный пароль | Ошибка: Такого аккаунта не существует |
| Оставить поле пустым | Ошибка: Заполните все поля |
|  | | | | |
| 2 | Добавление товара | Название товара | Ввести корректное название товара: Молоко | Поле принимает текст |
| Ввести некорректное название товара: Молоко\_3 | Ошибка: Ошибка преобразования |
| Оставить поле пустым: | Ошибка: Не все поля заполнены |
| Количество | Ввести корректное название товара: 46 | Поле принимает текст |
| Ввести некорректное название товара: как– | Ошибка: Ошибка преобразования |
| Оставить поле пустым: | Ошибка: Не все поля заполнены |
| id категории | Ввести корректное название товара: 1:Продукты.СК | Поле принимает текст |
| Оставить поле пустым: | Ошибка: Не все поля заполнены |
| Дата добавления | Ввести корректное название товара: 09.12.2024 | Поле принимает текст |
| Ввести некорректное название товара: | При разработке программы была допущена ошибка при которой программа пропускала ошибку: ввод букв в поле дата. |
| Оставить поле пустым: | Ошибка: Не все поля заполнены |
|  | | | | |
| 3 | Добавление поставщиков | Название | Ввести корректное название поставщика: ООО “СтройМетал” | Поле принимает текст |
| Ввести некорректное название товара: 1447- | Ошибка: Ошибка преобразования Но при разработке программы была допущена ошибка при которой численные значения можно было ввести в поле название |
| Оставить поле пустым: | Ошибка: Не все поля заполнены |
| Адрес | Ввести корректный адрес: Г. Москва, Ул. Пушкина 25 | Поле принимает текст |
| Ввести некорректный адрес: 200391 | Ошибка: Ошибка преобразования Но при разработке программы была допущена ошибка при которой численные значения можно было ввести в поле Адрес |
| Оставить поле пустым: | Ошибка: Не все поля заполнены |
| Телефон | Ввести корректный номер телефона: 8(977)753-43-43 | Поле принимает текст |
| Ввести некорректный номер телефона: привет | Ошибка: Ошибка преобразования Но при разработке программы была допущена ошибка при которой буквенные значения можно было ввести в поле телефон |
| Оставить поле пустым: | Ошибка: Не все поля заполнены |
| Поле ввода данных | Правильное отображение полей для ввода данных | При разработке была допущена ошибка при которой некорректно отображаются поля для ввода информации |
|  | | | | |
| 4 | Добавление пользователя | Логин | Ввести корректный логин: igor12345 | Поле принимает текст |
| Оставить поле пустым: | Ошибка: Заполните все поля |
| Пароль | Ввести корректный пароль: qwerty123 | Поле принимает текст |
| Оставить поле пустым: | Ошибка: Заполните все поля |
| Роль | Ввести корректную роль: 1:admin | Поле принимает текст |
| Оставить поле пустым: | Ошибка: Заполните все поля |
| Пользователь | Ввести одинаковые с существующем пользователем Логин и Роль | Ошибка: Пользователь не может быть внесен в бд |
|
|  | | | | |

# ПРИЛОЖЕНИЕ 3

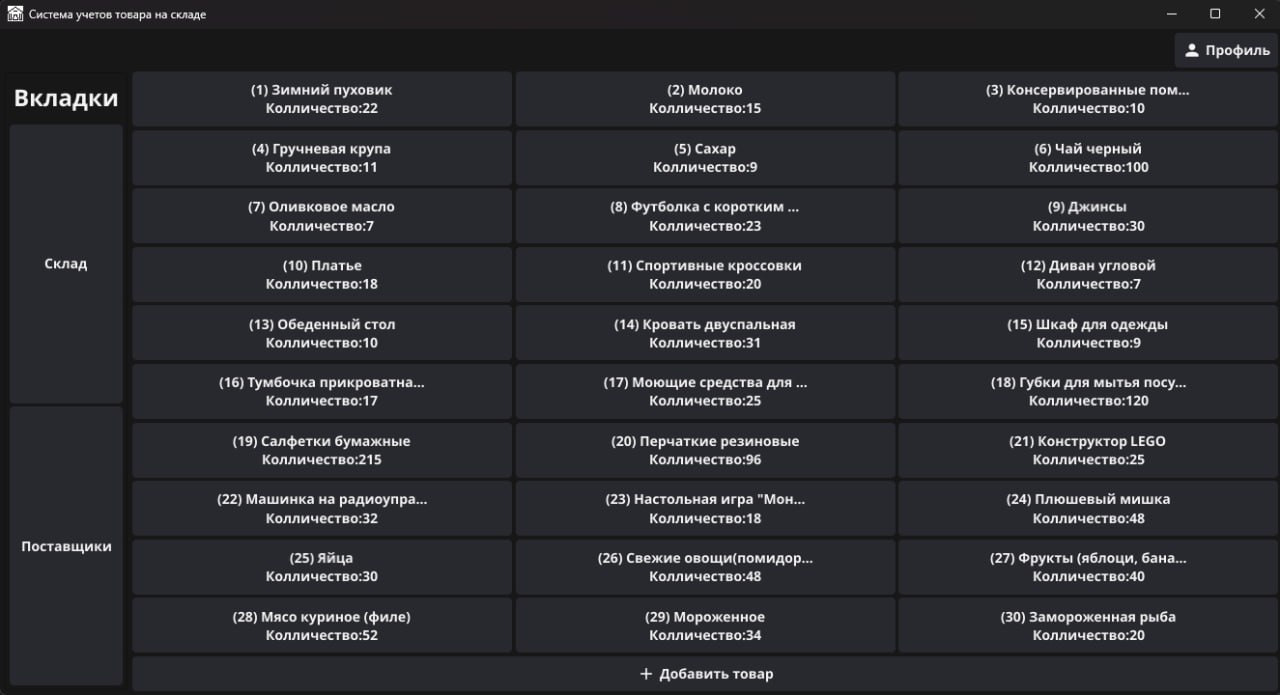
1. 

Рисунок 6 – Окно главного меню

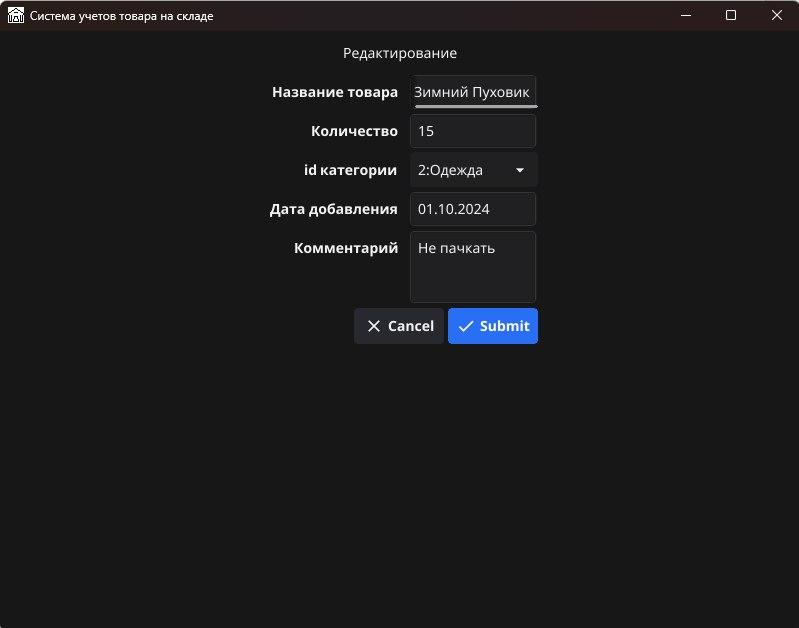


Рисунок 7 – Окно добавить товар

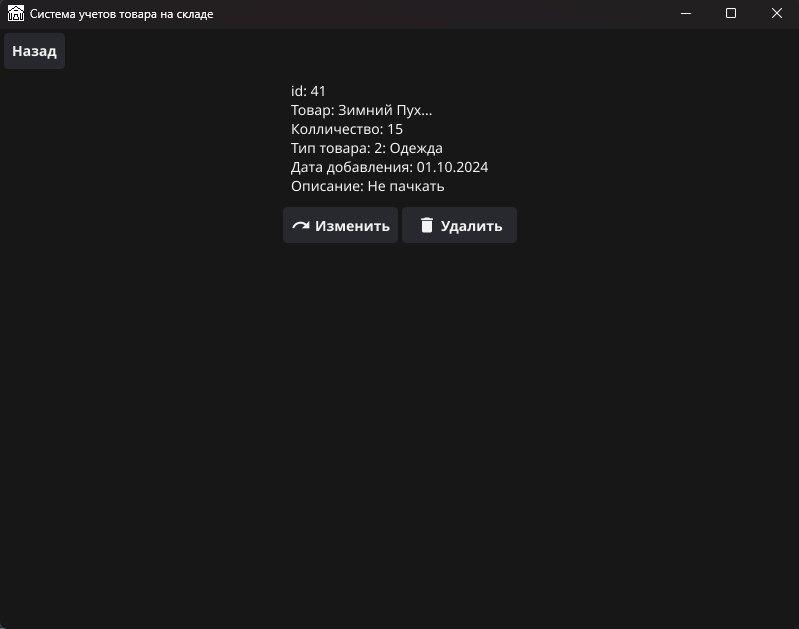


Рисунок 8 – Окно подробнее о товаре

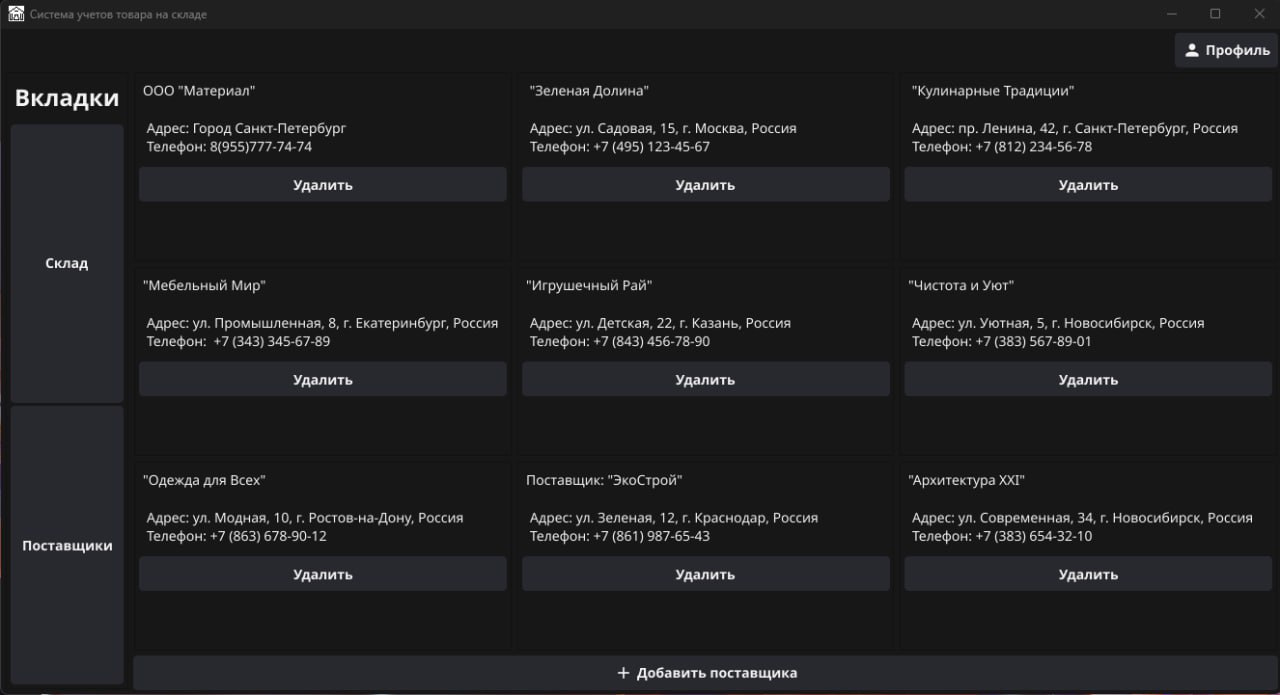


Рисунок 9 – Окно поставщики

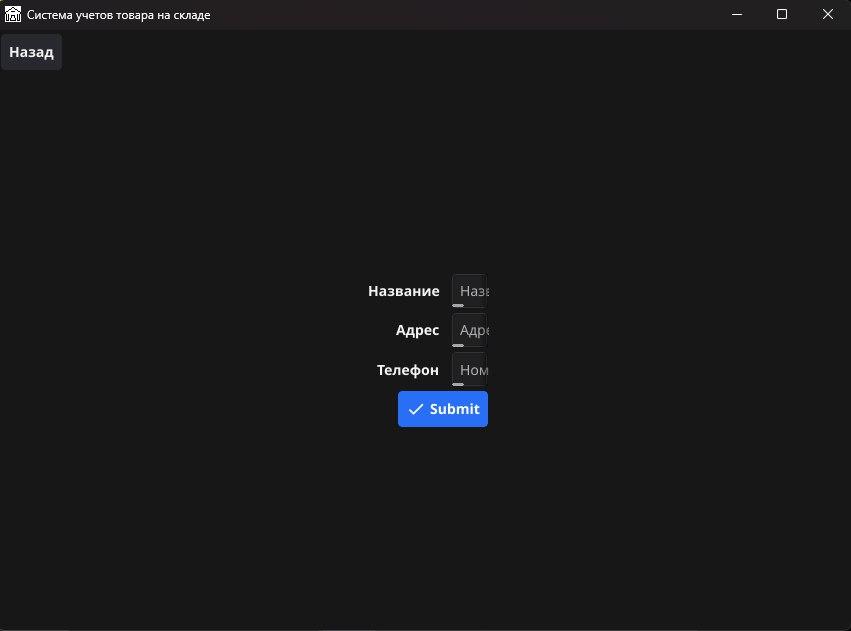


Рисунок 10 – Окно добавить поставщика

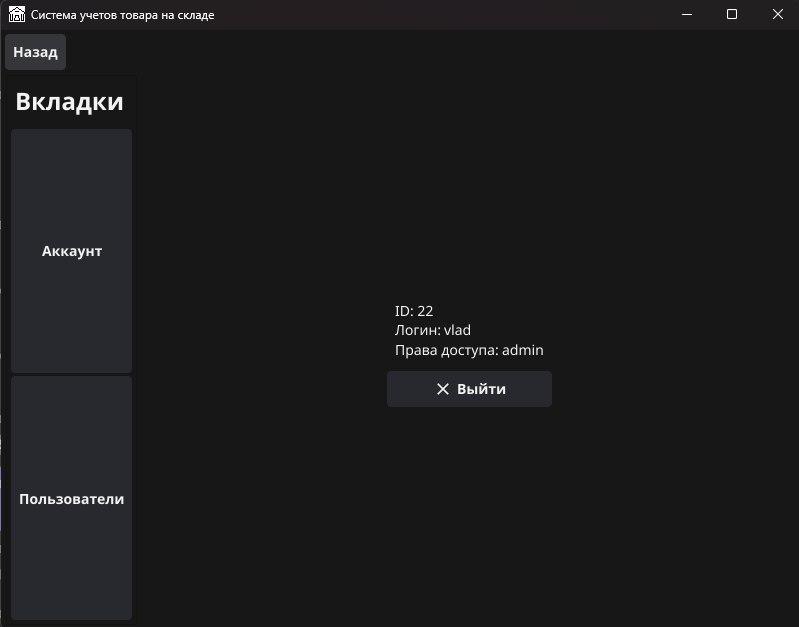


Рисунок 11 – Окно профиль

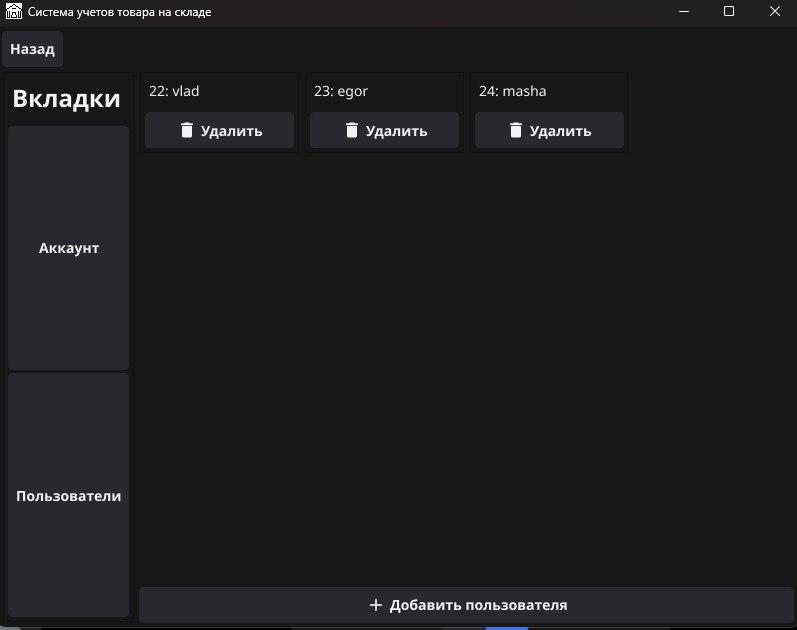


Рисунок 12 – Окно пользователи



Рисунок 13 – Функция входа в систему



Рисунок 14 — Функция добавления товара



Рисунок 15 — Функция переключения между вкладками Склад и Поставщики



Рисунок 16 — Функция добавления поставщика



Рисунок 17 — Функция перехода на окно профиль



Рисунок 18 — Функция переключения между вкладками Аккаунт и Пользователи



Рисунок 19 — Функция добавление нового пользователя



Рисунок 20 — Функция выхода из системы