Министерство образования и науки Алтайского края

КГБПОУ «Благовещенский строительный техникум»

09.02.07 Информационные системы и программирование

**КУРСОВАЯ** **РАБОТА**

по МДК.01.01. Разработка программных модулей

ПМ.01 Разработка модулей программного обеспечения для компьютерных систем

**Разработка программного обеспечения**

**«Инвентаризация компьютерной техники организации»**

|  |  |  |
| --- | --- | --- |
| Студент | \_\_\_\_\_\_\_\_\_\_\_\_\_\_ | М.Ю.Кусакин |
| Руководитель работы | \_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Н.А. Марченко |
| Нормоконтролер | \_\_\_\_\_\_\_\_\_\_\_\_\_\_ | С.В. Бобылев |
|  |  |  |
|  |  |  |
|  |  |  |

Степное Озеро

2023

**Содержание**

[Введение 3](#_Toc135128094)

[1 Анализ предметной области 4](#_Toc135128095)

[1.1 Актуальность и практическая значимость 4](#_Toc135128096)

[1.2 Определение требований к системе 6](#_Toc135128097)

[2 Проектирование 7](#_Toc135128098)

[2.1 Архитектура программного обеспечения 7](#_Toc135128099)

[2.2 Моделирование программного обеспечения 8](#_Toc135128100)

[2.3 База данных 11](#_Toc135128102)

[2.4 Пользовательский интерфейс 15](#_Toc135128103)

[3 Разработка 19](#_Toc135128104)

[3.1 Выбор инструментов разработки 19](#_Toc135128105)

[3.2 Основные программные модули 19](#_Toc135128106)

[3.3 Файловая структура проекта 20](#_Toc135128107)

[3.4 Серверная часть программного обеспечения 20](#_Toc135128108)

[3.5 Клиентская часть программного обеспечения 21](#_Toc135128109)

[3.6 Организация данных 21](#_Toc135128110)

[4 Описание результатов разработки 22](#_Toc135128111)

[4.1 Тестирование программы 22](#_Toc135128112)

[4.2 Результат разработки 22](#_Toc135128113)

[4.3 Руководство пользователя 22](#_Toc135128114)

[Заключение 27](#_Toc135128115)

[Список использованных источников и литературы 28](#_Toc135128116)

[Приложение А. Листинги кода PostgreSQL по созданию таблиц 29](#_Toc135128117)

[Приложение Б. Листинги кода на языке C# по созданию пользовательского интерфейса 34](#_Toc135128118)

[Приложение В. Тест-кейсы 35](#_Toc135128119)

[Приложение Г. Таблицы базы данных 36](#_Toc135128120)

# Введение

Сегодня современные условия развития рынка предъявляют организациям новые требования. Все чаще можно услышать о необходимости структурирования и автоматизации бизнес-процессов. Информационные системы повсеместно внедряются, управление автоматизируется. Большинство руководителей организаций и предприятий понимают, внедрение эффективной и современной системы управления необходимо, чтобы: ускорить бизнес-процессы; снизить транзакционные издержки; сбалансировать соотношение полномочий у подчиненных и партнеров; не отвлекаться от стратегических задач для постоянного контроля. Автоматизация организации служит не только инструментом управления бизнесом, но и средством, позволяющим в значительной мере облегчить процесс принятия решения. Система автоматизации сократит время на решение задач, которые постоянно растут [4].

Управление товарными ресурсами в организации предполагает планирование и прогнозирование деятельности, проведение анализа большого количества информации по истории поступлений, поставок, товарных запасов, списаний, расчёт оптимальных размеров товарных запасов и дальнейшего планирования размеров и номенклатуры заказов поставок необходимых для поддержания эффективного функционирования склада. Оптимизация уровня товарных запасов в организации встает в связи проблемой содержания запасов, с одной стороны нужно избежать переполнения товаров на складах, и с другой стороны нежелательно допускать отсутствия необходимого товара длительное время.

В настоящее время компьютеры и другие электронные устройства являются неотъемлемой частью современного бизнеса. Каждая организация обладает компьютерами, принтерами и другим оборудованием, которые используются для выполнения различных задач. Однако, с ростом количества компьютеров в организации, возникают проблемы с их управлением и контролем. В данной курсовой работе будет рассмотрена разработка программного обеспечения для инвентаризации компьютерной техники организации. Это приложение поможет упростить процесс управления компьютерным оборудованием, обеспечивая полный контроль и точную информацию о каждом устройстве. Разработка данного приложения позволит повысить эффективность работы организации, а также сократить расходы на обслуживание и ремонт оборудования.

Актуальность темы обоснована высоким спросом на компьютерную технику в организации, из-за чего в некоторые организации начинают обзаводиться большим количеством техники, после чего становится очень сложно вести подсчёт всей техники.

Объектом автоматизации является организация КГБОУ Алтайский Краевой Педагогический Лицей-интернат - краевое государственное бюджетное общеобразовательное учреждение.

Предмет исследования - процессы управления компьютерной техники организации КГБОУ Алтайский Краевой Педагогический Лицей-интернат

Целью курсовой работы является разработка настольного приложения автоматизированной информационной системы работы склада по поддержке процессов закупа и учёта хранения товаров на складе.

Задачи курсовой работы:

* проведение анализа предметной деятельности;
* определение состава автоматизируемых задач и функционала программного обеспечения;
* определение структуры базы данных системы;
* построение информационных моделей системы;
* построение интерфейса взаимодействия пользователя с системой;
* разработка кода программного обеспечения (ПО) на основе готовой спецификации;
* проведение отладки и тестирования программного обеспечения системы;
* подготовка руководства пользователя.

Для разработки программного обеспечения «Инвентаризация компьютерной техники организации» применима технология Windows Forms, в которой приложение программируется на базе классов .NET Framework class library. В основе технологии производный класс от System.WinForms.Form, имеющий множество свойств и методов с большими возможностями графического интерфейса к формам.

Бэкенд приложения реализован в объектно-реляционной системе управления базами данных PostgreSQL (СУБД).

Настольное приложение создано в соответствии со стандартами проектирования и разработки программного обеспечения [1][2].

## 1 Анализ предметной области

## 1.1 Актуальность и практическая значимость

Склад – это нежилое помещение, предназначенное для хранения сырья, продукции, товаров и прочих грузов, обеспечивающее соблюдение требуемых условий хранения и оснащенное оборудованием для хранения и удобными для разгрузки-погрузки конструкциями и сооружениями.

Структура организации ООО «Стройлидер» представлена на рисунке 1.

Рисунок 1 – Организационная структура

Организация состоит из четырех отделов:

* закупок;
* складской;
* экономический;
* юридический.

Каждый отдел выполняет свою необходимую функцию в организации. Самым функциональным отделом в организации является складской отдел, в котором реализуются основные задачи по хранению и размещению товаров и продукции. Оператор реализует логистические функции, определяет через существующую информационную систему свободные ячейки на стеллажах в зависимости от размеров и максимальной нагрузки на стеллаж в целом. Информация о свободных ячейках учитывается во время погрузки и отгрузки товара или продукции. Кладовщики управляют процессом размещения товаров на стеллажи. Грузчики могут использовать как вилочный погрузчик с противовесом, так и располагать товар вручную, в зависимости от размещения товара. За весь этот процесс отвечает менеджер склада. Также менеджер склада участвует в процессе поступления товара и отгрузки.

Склад организации ООО «Стройлидер» принимает партии товаров от поставщиков и отпускает его клиентам мелкими партиями. Требуется вести учет поступающих и отпускаемых товаров, поставщиков и клиентов, формировать приходные и расходные накладные. Также необходимо формировать отчеты о поступлении и отпуске товаров на складе за произвольный период.

На складе происходит движение материальных и информационных потоков. Первые представлены движением товара от поставщиков на склад или со склада к покупателям, а информационные потоки представлены документацией, необходимой для этих операций.

К основным функциям склада относят [3]:

* формирование нужного набора в согласовании с заказом потребителя;
* складирование и хранение товаров;
* консолидация грузов (объединение незначительных грузов в более большую партию);
* оказание услуг клиенту: расфасовка продукции; наполнение и распаковка груза; придание товарного вида; транспортно-экспедиционные услуги; изначальная обработка груза.

Матрица ответственности – это таблица, которая регламентирует уровень несения ответственности сотрудника в организации при выполнении задач.

Менеджер проекта несет ответственность за все происходящее в нем. Поскольку заданий много, то руководитель имеет право делегировать выполнение заданий другим сотрудникам.

Матрица ответственности на складе отображена в таблице 1.

Таблица 1 – Матрица ответственности склада

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Обязанности  Должность | Контроль над отделами | Закупка товара и продукции | Управление складом | Управление размещением грузов | Управление логистикой | Размещение товаров на складе | Ведение бухгалтерии организации | Юридическое сопровождение |
| Директор | Х |  |  |  |  |  |  |  |
| Менеджер по закупкам |  | Х |  |  |  |  |  |  |
| Менеджер склада |  |  | Х |  |  |  |  |  |
| Кладовщик |  |  |  | Х |  |  |  |  |
| Оператор |  |  |  |  | Х |  |  |  |
| Грузчик |  |  |  |  |  | Х |  |  |
| Заместитель директора по экономике |  |  |  |  |  |  | Х |  |
| Юрист |  |  |  |  |  |  |  | Х |

Склад принимает и складирует готовую продукцию, эта операция сопровождается формированием приходной накладной. Она состоит из двух частей: общей (в которую входят номер накладной, наименование поставщика и дата сдачи продукции на склад) и спецификации (в нее входят наименования и количество передаваемой продукции).

Продукция со складов направляется заказчикам в соответствии с заключенными договорами. Отправляется продукция на основании товарно-транспортной накладной. Товарно-транспортная накладная состоит из общей части (номер накладной, реквизиты грузоотправителя, грузополучателя и грузоперевозчика, дата отгрузки) и спецификации (вид и количество отгружаемой продукции).

После получения продукции заказчик должен произвести оплату, которая оформляется платежным поручением, выписанным на основании товарно-транспортной накладной. Общая часть платежного поручения включает номера товарно-транспортной накладной, платежного поручения и дату оплаты. Счет-фактура – это документ, выдаваемый поставщиком покупателю или предоставляемые поставщиком банку для подтверждения платежа покупателя. Общая часть включает реквизиты поставщика и покупателя, а в спецификации указываются наименование товара, единицы измерения, цена и сумма.

Они предназначены для хранения систематизированной информации о работе оптового склада. Пользователю должны быть предоставлены удобные средства для отбора данных по запросам. Проанализировав основные операции на оптовом складе, можно сделать вывод, что потребуется выполнение следующих запросов: определить дату продажи товара, определить дату поставки товара, выбрать товары по определенной цене, сумма налога поставленных товаров, выбрать количество товаров в наличие. Для решения поставленной цели, было решено создать специализированную программу, рассчитанную на решение определенного и ограниченного круга задач и управление заранее определенной структурой информации. Разрабатываемая база данных должна выдавать однозначные требования на поставленные запросы пользователей, должна иметь читабельный и удобный интерфейс. Конечными пользователями информационной системы являются заведующий складом, старший кладовщик, кладовщик-комплектовщик, водитель погрузчика, которые относятся к категории пользователей не искушенных в вопросах ведения, администрирования баз данных и поддержании их в актуальном состоянии.

## 

## 1.2 Определение требований к системе

Разрабатываемая система должна обеспечивать создание, систематизацию, накопление, хранение и обработку необходимой информации о прошлой, текущей и будущей деятельности организации.

Анализируя деятельность склада организации, были выявлены следующие бизнес-процессы:

* приемка товара;
* размещение товара;
* комплектация заказов;
* учёт и хранение;
* отгрузка.

Функциональные требования к системе:

В связи с полученными данными о существующей информационной системы и анализом деятельности организации выделим следующие функциональные требования:

1. Учёт строительных материалов.
2. Обеспечение проверки вводимых данных в систему.
3. Ведение в систему различных справочников для полноценной работы программы и сокращения вводимых данных путем их выбора.
4. Ведение базы документов поставки и отгрузки товара.
5. Получение информации о движении товара или продукта.
6. Получение информации о сводных местах на стеллажах.
7. Разграничение прав доступа в системе.
8. Учёт стеллажей в системе.

Основные нефункциональные требования:

1. Возможность расширения системы с учетом потенциального роста групп товаров (для сохранения на складах).
2. Удобный и простой пользовательский интерфейс (расположение компонентов управления должны быть в зоне видимости основной задачи).
3. Использование современного аппаратного обеспечения (с учётом роста технических характеристик различных устройств система должна работать при минимальных требованиях).
4. Поддержка больших объемов, хранимых данных по каталогу оборудования.
5. Обеспечение защиты информации от неразрешенного доступа и изменений (обеспечить ролевой доступ к системе, а также защитить информацию от изменений).
6. Обеспечение возможности проверки введённых данных в систему.

## 2 Проектирование

## 2.1 Архитектура программного обеспечения

Архитектура информационной системы – концепция, определяющая модель, структуру, выполняемые функции и взаимосвязь компонентов информационной системы [5].

Клиент-сервер – это архитектура приложения, в которой его функциональные части взаимодействуют по схеме запрос-ответ. Двумя взаимодействующими частями такой архитектуры являются клиент (выполняет активную функцию, т.е. инициирует запросы) и сервер (отвечает на поступающие запросы). Классическая архитектура клиент-сервер подразумевает распределение трёх основных частей настольного приложения (интерфейс взаимодействия с пользователем, серверную логику и хранилище данных) по двум физическим модулям.

Архитектура представлена визуально на рисунке 2.

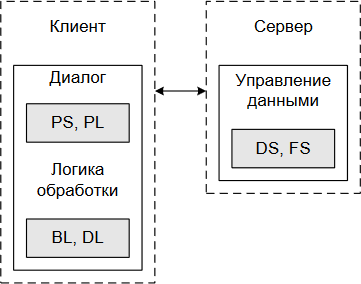


Рисунок 2 – Архитектура клиент-сервер

Многоуровневая архитектура клиент-сервер – это разновидность описанной выше двухуровневой архитектуры, где PS – средства представления (англ. Presentation Services), PL – логика представления (англ. Presentation Logic), BL –прикладная логика (англ. Business Logic), DL – логика управления данными (англ. Data Logic), DS – операции с базой данных (англ. Data Services) и FS – файловые операции (англ. File Services), в которой обработка данных распределена между одним или несколькими серверами, что позволяет эффективно разделить функции хранения, обработки и представления данных. Данная архитектура реализует более эффективный подход к использованию возможностей как серверов, так и клиентов [5].

Одной из разновидностей многоуровневой архитектуры является трёхуровневая архитектура.

Трёхуровневая архитектура – это архитектура приложения, разделённая на три основ-ных компонента: клиент, сервер приложений и сервер баз данных. Архитектура представлена визуально на рисунке 3.

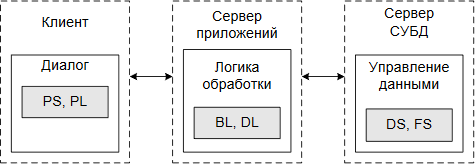


Рисунок 3 – Трёхуровневая архитектура

В рамках разработки было принято решение разработать платформу в соответствии с принципами трёхуровневой архитектуры по следующим причинам [7]: упрощение масштабируемости; упрощение реализации полномочий пользователей за счёт переноса прикладной логики на серверную сторону; более высокий уровень безопасности по сравнению с двухуровневой архитектурой.

## 

## 2.2 Моделирование программного обеспечения

Анализ предметной области показывает, что логистический процесс на складе можно разделить на две основные группы:

* операции, связанные с функцией логистической координации снабжения (закупки) и продаж. Так как данные операции не относятся к сфере оперативной складской логистики;
* операции, связанные с перемещением материального потока на складе и соответствующим документооборотом.

В основу построения функциональной модели были выбраны четыре базовые функции получение товара, хранение, комплектация, отгрузка.

Для формального представления функциональной модели бизнес-процессов современного склада использована методология функционального моделирования IDEF0, являющуюся составной частью SADT, а в качестве модели декомпозиции – «модель жизненного цикла» [9].

На первом уровне модели выделены следующие основные бизнес-процессы, через которые проходит материальный поток на складе (рисунок 4):

1) Получение/прием товара – прием, проверка соответствия поставки сопроводительным документам и целостности товара.

2) Хранение товара – определение мест хранения поступающих на склад товарно-материальных ценностей для товара, сортировка, построение оптимальных маршрутов, размещение грузов в зоне хранения.

3) Комплектация товара – отбор товара из зоны хранения, комплектация и упаковка.

4) Отгрузка товара – формирование партий отгрузки и отгрузки товара клиентам.

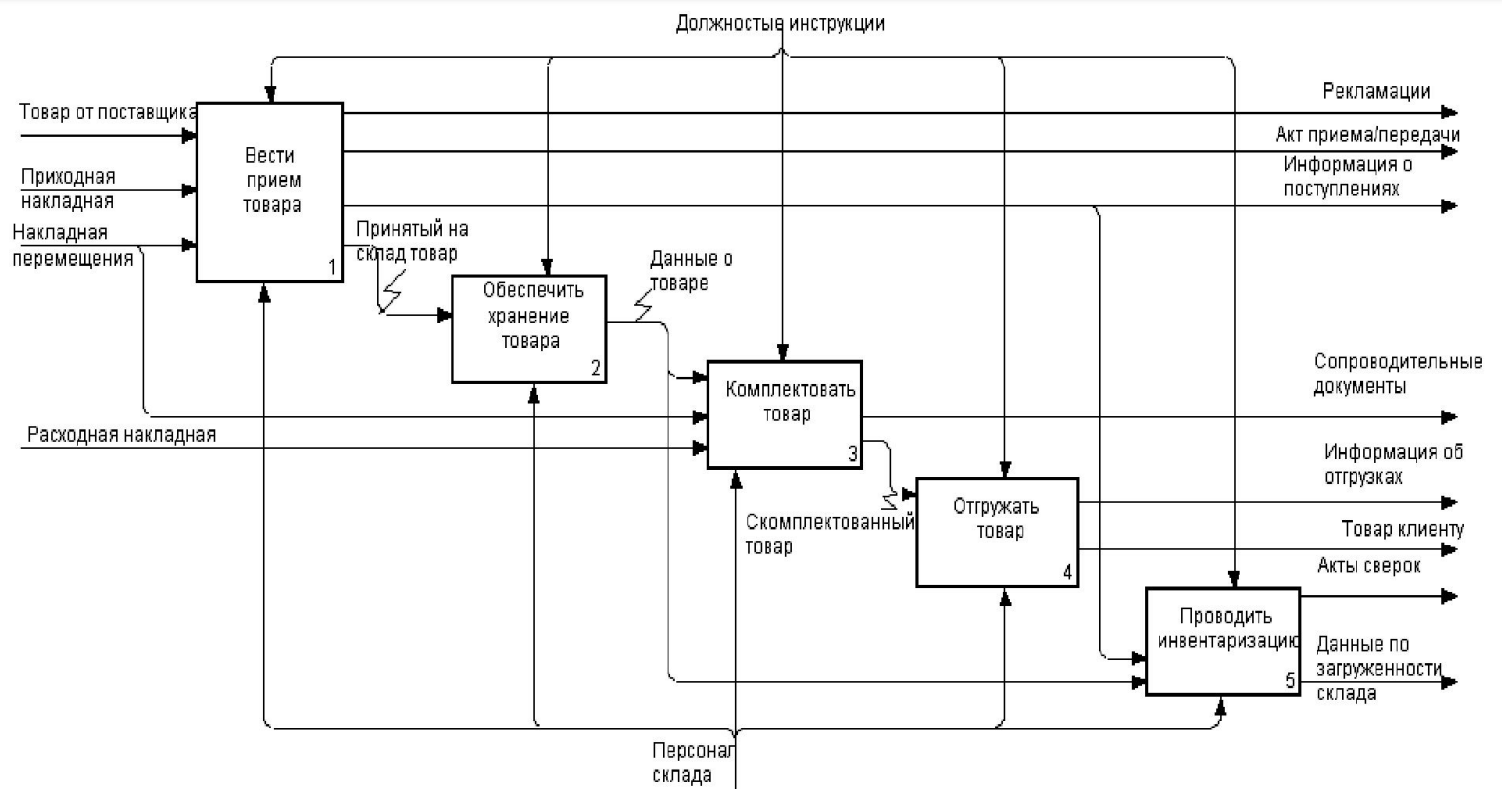


Рисунок 4 – Основные бизнес-процессы склада

Дальнейшая декомпозиция функциональной модели будет производиться с учетом модели жизненного цикла перемещения материального и информационного потоков по технологическим операциям склада.

Бизнес-процесс приемки товара на склад приведен на рисунке 5.

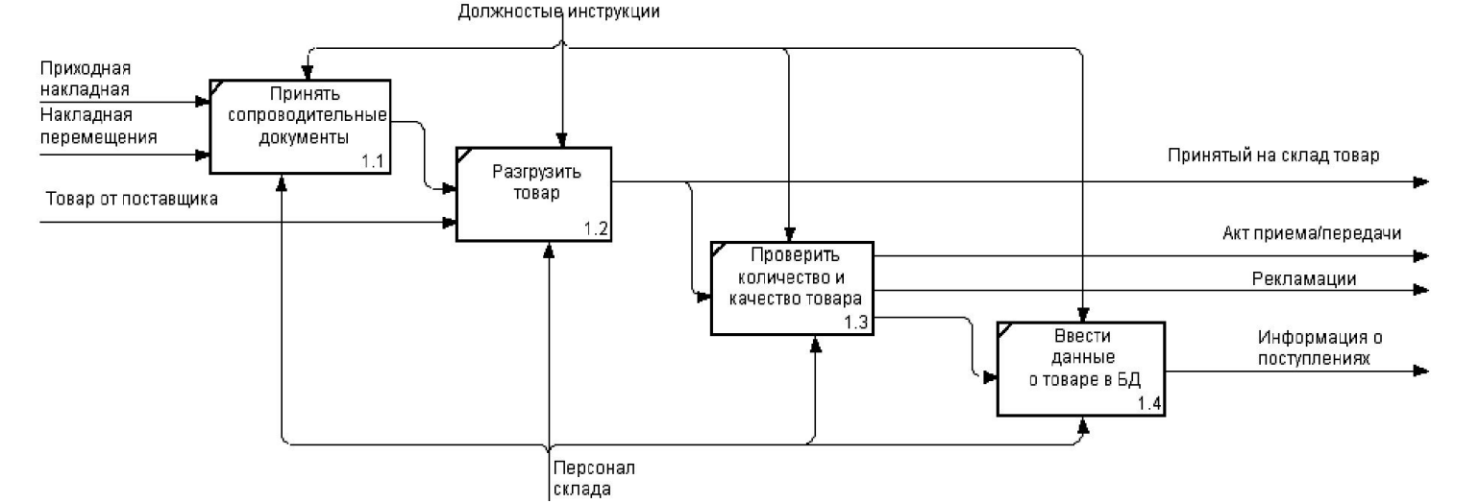


Рисунок 5 – Процесс приемки товара

Бизнес-процесс хранения товара представлен на рисунке 6. Для более эффективного использования складского пространства и оптимизации размещения товар на складе предлагаются алгоритмы размещения, основанные на следующих правилах:

* + размещение товара в свободные ячейки;
  + размещение в соответствии с АВС-классификацией;
  + размещение товара в занятые ячейки к такому же или похожему товару (если это возможно). При размещении к похожему товару контролируется совместимость;
  + размещение товара в занятые ячейки к любому товару.

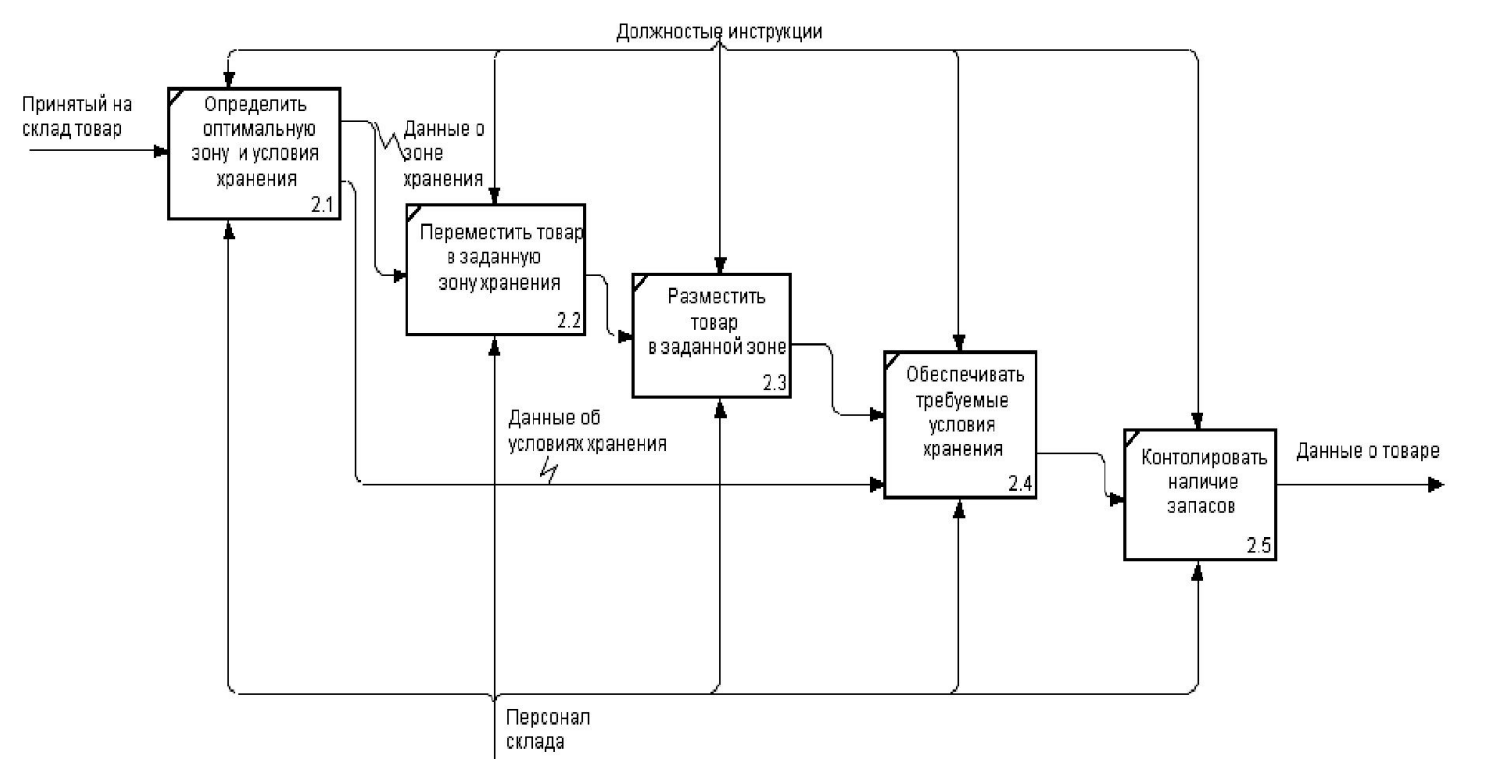


Рисунок 6 – Процесс хранений товара

Бизнес-процесс комплектации представлен на рисунке 7.

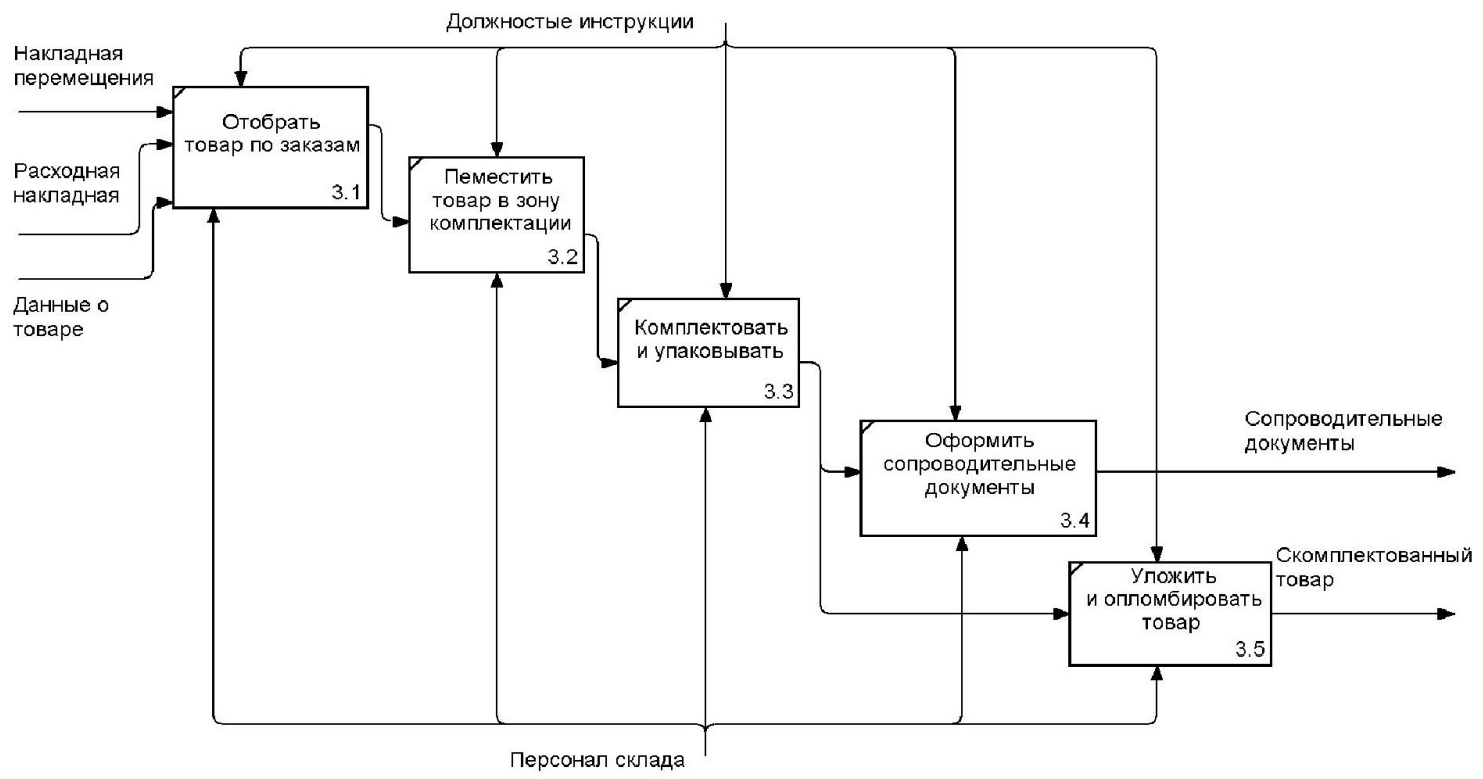


Рисунок 7 – Процесс комплектации товара

Бизнес-процесс отгрузки товара представлен на рисунке 8.

## 

Рисунок 8 – Процесс отгрузки товара

UML (Unified Modeling Language) – унифицированный язык моделирования, который используется для объектно-ориентированного анализа и проектирования. Основным применением является для визуализации программных систем [9].

Согласно сценарной модели взаимодействия UML склада, пользователи программного продукта обращаются к различным инцидентам (рисунок 9).

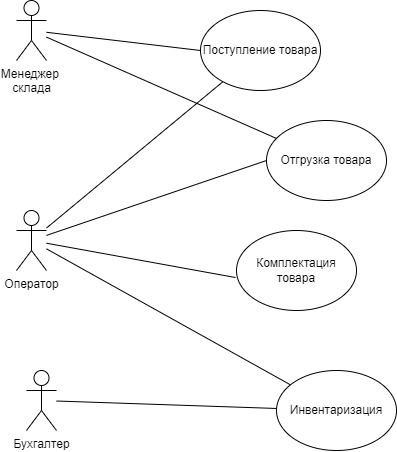


Рисунок 9 – Диаграмма вариантов использования для проектирования

Согласно сценарной модели взаимодействия UML склада, пользователи программного продукта обращаются к различным инцидентам.

## 2.3 База данных

База данных склада Warehouse должна содержать следующие таблицы: «users», «role», «delivery\_goods», «consumption\_goods», «goods», «unit», «inventory», «equipment», «rack», «rack\_category», «suppliers». Атрибуты таблиц отображены в таблицах 2-12.

Таблица «users» предназначена для хранения информации о пользователях приложения, в частности их логины и пароли.

Таблица 2 – Таблица «users»

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Наименование  атрибутов | Физический  тип | Ограничение  целостности | | Уникальность ключа | Размер | Обязательность заполнения |
| Формат | Ключ |
| КодПользователи | Число | integer | PK | + |  | + |
| ФИО | Текст | character varying | - | - | 100 | + |
| Логин | Текст | character varying | - | - | 25 | + |
| Пароль | Текст | character varying | - | - | 25 | + |
| Роль | Число | integer | FK | - |  | + |

В таблице «role» хранится информация о ролях пользователей.

Таблица 3 – Таблица «role»

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Наименование  атрибутов | Физический  тип | Ограничение  целостности | | Уникальность ключа | Размер | Обязательность заполнения |
| Формат | Ключ |
| КодРоли | Число | integer | PK | + |  | + |
| Название | Текст | character varying | - | - | 50 | + |

Таблица «unit» предназначена для хранения информации о единицах измерения товара.

Таблица 4 – Таблица «unit»

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Наименование  атрибутов | Физический  тип | Ограничение  целостности | | Уникальность ключа | Размер | Обязательность заполнения |
| Формат | Ключ |
| КодЕдиницы | Число | integer | PK | + |  | + |
| Название | Текст | character varying | - | - | 50 | + |

Таблица «rack» содержит информацию о вместимости, категории, ячейках стеллажей.

Таблица 5 – Таблица «rack»

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Наименование  атрибутов | Физический  тип | Ограничение  целостности | | Уникальность ключа | Размер | Обязательность заполнения |
| формат | ключ |
| КодСтеллажи | Число | integer | PK | + |  | + |
| Название | Текст | character varying | - | - | 100 | + |
| Вместимость | Число | numeric | - | - |  | + |
| Категория | Число | integer | FK | - |  | + |
| Ячейки | Число | integer | - | - |  | + |

Таблица «rack\_category» включает информацию о секторах и стеллажах.

Таблица 6 – Таблица «rack\_category»

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Наименование  атрибутов | Физический  тип | Ограничение  целостности | | Уникальность ключа | Размер | Обязательность заполнения |
| формат | ключ |
| КодКатегория | Число | integer | PK | + |  | + |
| Название | Текст | character varying | - | - | 100 | + |

Таблица «suppliers» предназначена для хранения информации о поставщиках склада: название, адрес и телефон поставщика.

Таблица 7 – Таблица «suppliers»

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Наименование  атрибутов | Физический  тип | Ограничение  целостности | | Уникальность ключа | Размер | Обязательность заполнения |
| формат | ключ |
| КодПоставщики | Число | integer | PK | + |  | + |
| Название | Текст | character varying | - | - | 250 | + |
| Адрес | Текст | character varying | - | - | 250 | + |
| Телефон | Текст | character varying | - | - | 12 | + |

В таблице «equipment» хранится информация о комплектации стеллажа.

Таблица 8 – Таблица «equipment»

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Наименование  атрибутов | Физический  тип | Ограничение  целостности | | Уникальность ключа | Размер | Обязательность заполнения |
| Формат | Ключ |
| КодКомплектация | Число | integer | PK | + |  | + |
| Стеллаж | Число | integer | FK | - |  | + |
| Ячейка | Текст | character varying | - | - | 50 | + |
| Время | Дата/Время | timestamp with time zone | - | - |  | + |

В таблице «inventory» содержится информация о времени, пользователе, проводившем инвентаризацию, и описании процедуры инвентаризации.

Таблица 9 – Таблица «inventory»

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Наименование  атрибутов | Физический  тип | Ограничение  целостности | | Уникальность ключа | Размер | Обязательность заполнения |
| Формат | Ключ |
| КодИнвентариза-ция | Число | integer | PK | + |  | + |
| Время | Дата/Время | timestamp with time zone | - | - |  | + |
| Описание | Текст | text | - | - |  | + |
| КодПользователи | Число | integer | FK | - |  | + |

Таблица «goods» содержит информацию о товарах, хранящихся на складе.

Таблица 10 – Таблица «goods»

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Наименование  атрибутов | Физический  тип | Ограничение  целостности | | Уникальность ключа | Размер | Обязательность заполнения |
| Формат | Ключ |
| КодТовар | Число | integer | PK | + | - | + |
| Наименование товара | Текст | character varying | - | - | 250 | + |
| Единица измерения | Число | integer | FK | - |  | + |

В таблице «delivery\_goods» хранится информация о товарах, поставленных на склад.

Таблица 11 – Таблица «delivery\_goods»

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Наименование  атрибутов | Физический  тип | Ограничение  целостности | | Уникальность ключа | Размер | Обязательность заполнения |
| Формат | Ключ |
| КодПоставка | Число | integer | PK | + |  | + |
| КодПоставщики | Число | integer | FK | - |  | + |
| КодМенеджер | Число | integer | FK | - |  | + |
| Код документа | Число | integer |  | + |  | + |
| Дата поставки | Дата/время | timestamp with time zone |  | - |  | + |
| Товары | Текст | character varying |  | - | 250 | - |

Таблица «consumption\_goods» предназначена для хранения информации о расходах.

Таблица 12 – Таблица «consumption\_goods»

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Наименование  атрибутов | Физический  тип | Ограничение  целостности | | Уникальность ключа | Размер | Обязательность заполнения |
| Формат | Ключ |
| КодРасхода | Число | integer | PK | + |  | + |
| КодПоставщики | Число | integer | FK | - |  | + |
| КодМенеджер | Число | integer | FK | - |  | + |
| Код документа | Число | integer |  | + |  | + |
| Дата расхода | Дата/время | timestamp with time zone |  | - |  | + |
| Товары | Текст | character varying |  | - | 250 | - |

Определены сущности и атрибуты информационной системы, а также установлены связи между ними. База данных имеет 11 таблиц. Каждый атрибут имеет свой тип данных и его ограничение (при необходимости).

В базе данных реализована связь «один-ко-многим», что позволяет сделать вывод о том, что база данных «Склад товаров» приведена к третьей нормальной форме (3НФ). На рисунке 10 отображена логическая модель базы данных.

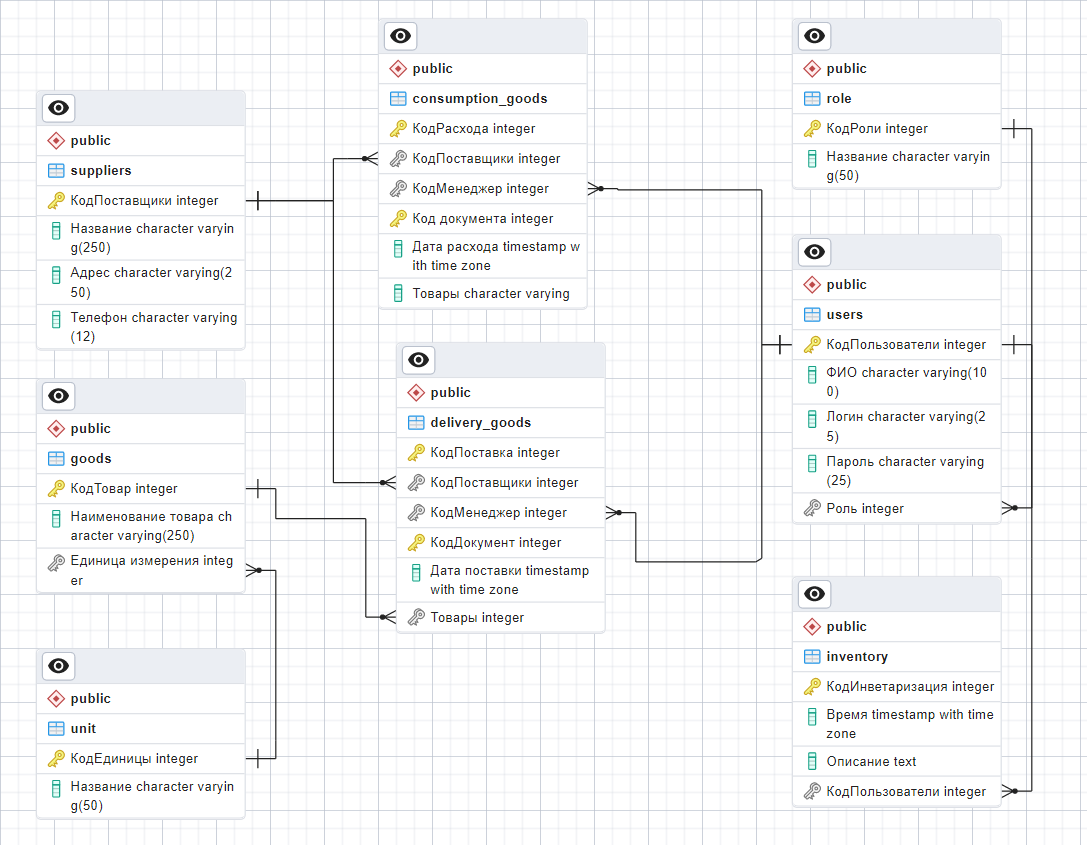
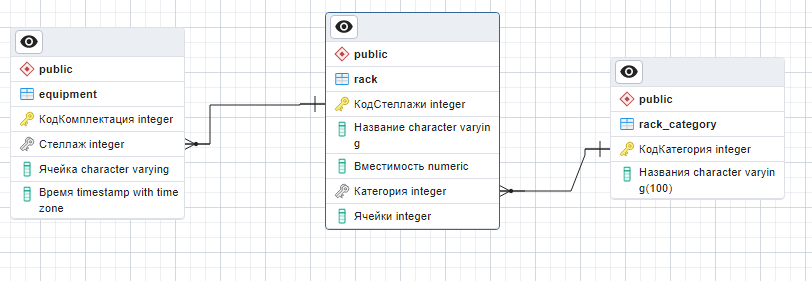


Рисунок 10 – Логическая модель информационной системы

Создание базы данных реализовано с помощью SQL-запроса к базе данных. Дальнейшее обращение к базе данных будет реализовано с клиентской части.

## 2.4 Пользовательский интерфейс

Для авторизации пользователю предлагается окно «Вход в систему» (рисунок 11). На данной форме размещены два поля для ввода, две метки для подписи полей, кнопки для входа и выхода. На кнопку «Войти» установлен обработчик для входа в систему, если пароль верный, то система запустит главное окно, если нет, то сообщит об ошибке.

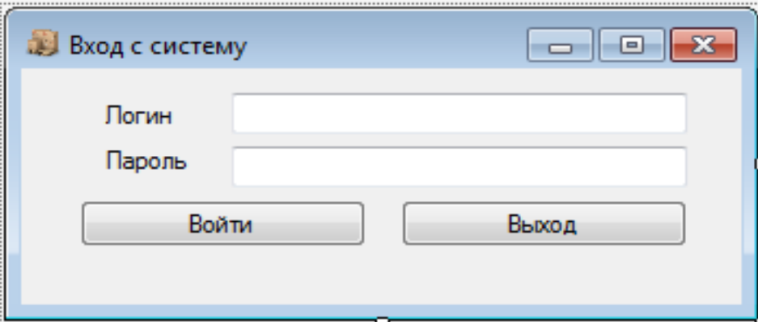


Рисунок 11 – Окно «Вход в систему»

Основное окно, в котором пользователь выбирает необходимые действия отображено на рисунке 12. В верхней части окна размещены – меню, табличная часть и несколько кнопок, по нажатию на которые открывается окно «Расход» или «Приход», также добавлена строка статуса.

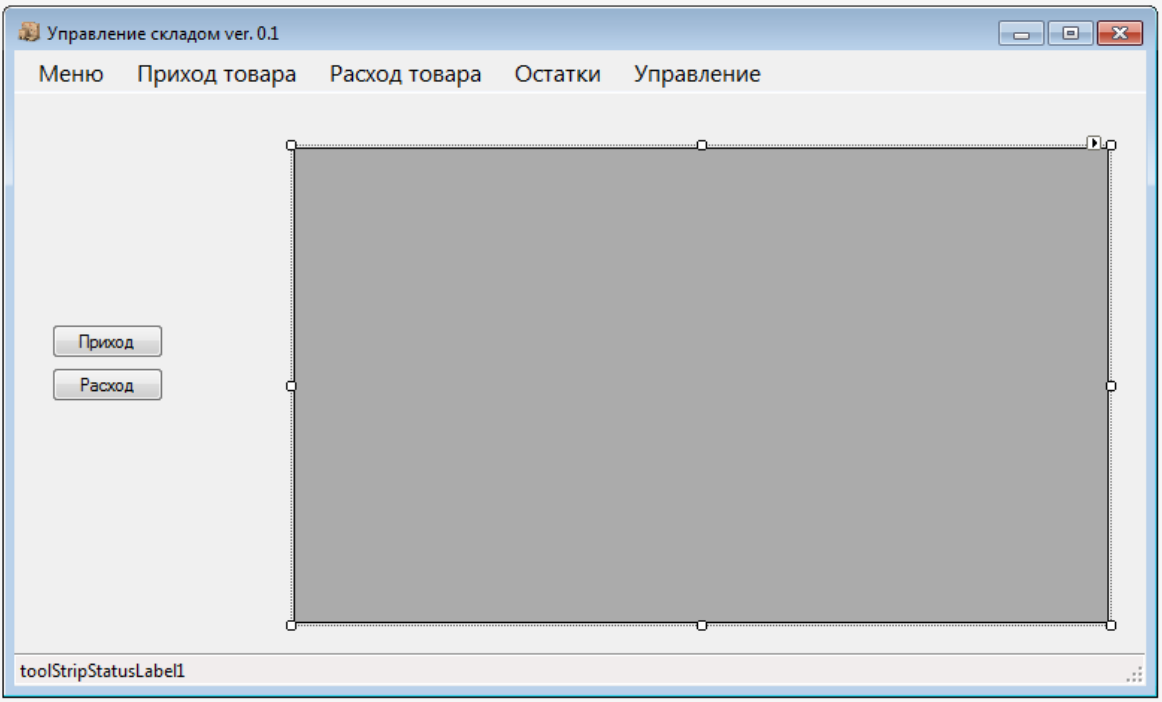


Рисунок 12 – Главное окно «Управление складом»

При поставке необходимо выбрать все товары, что можно сделать вручную или сканером штрих-кода. На форме поставки размещены табличная часть, поля для ввода и кнопка для выполнения действий (рисунок 13).

В поле со списком добавлен перечень поставщиков из таблицы поставщиков, в поле со стеллажами – стеллаж для размещения товаров. При выборе ручным способом указываются товары, при сканировании штрихкодом товар добавляется автоматически.

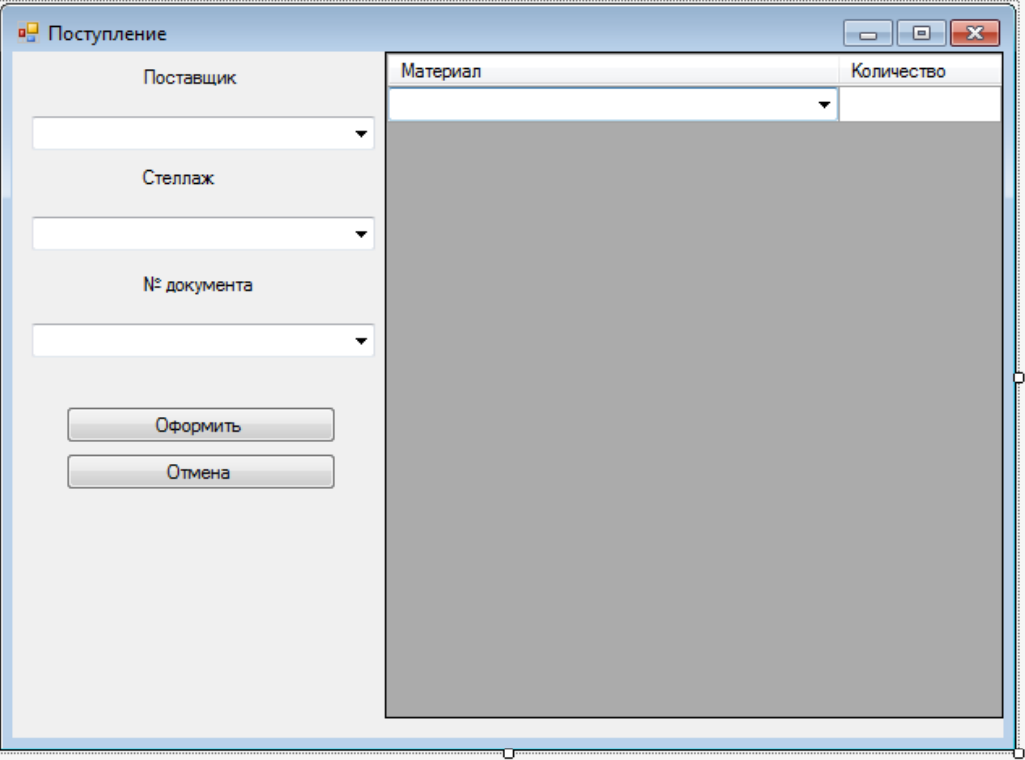


Рисунок 13 – Форма поступления товара

Стеллаж – складское специализированное мебельное оборудование для хранения предметов/грузов, состоящее из металлических стоек, балок и поперечных балок либо многоярусных настилов, закрепленных на балках, либо состоящее из закреплённых на стойках консолей.

Форма расхода товара реализована аналогичным образом, выбор стеллажа при этом убран.

Форма остатков для просмотра остатков любой товарной продукции на текущую дату представлена на рисунке 14.

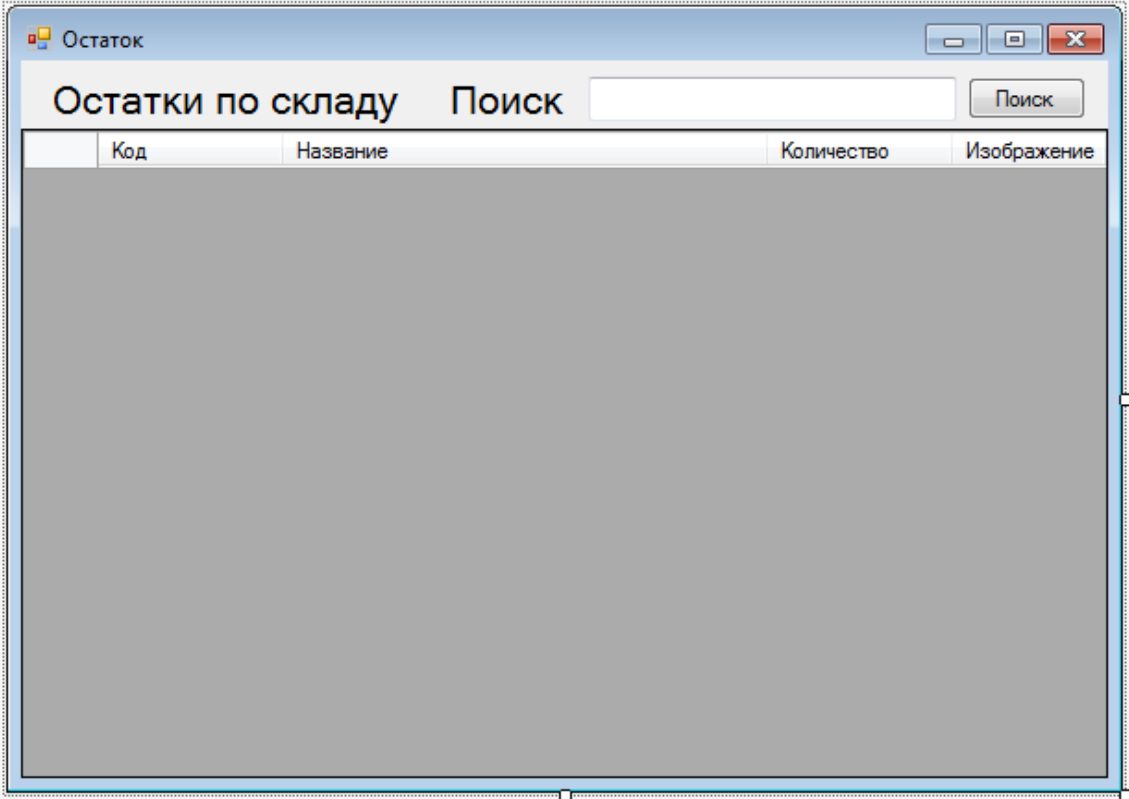


Рисунок 14 – Форма проверки товара на складе

Форма для управления поставщиками с табличной частью и полями для ввода приведена на рисунке 15.

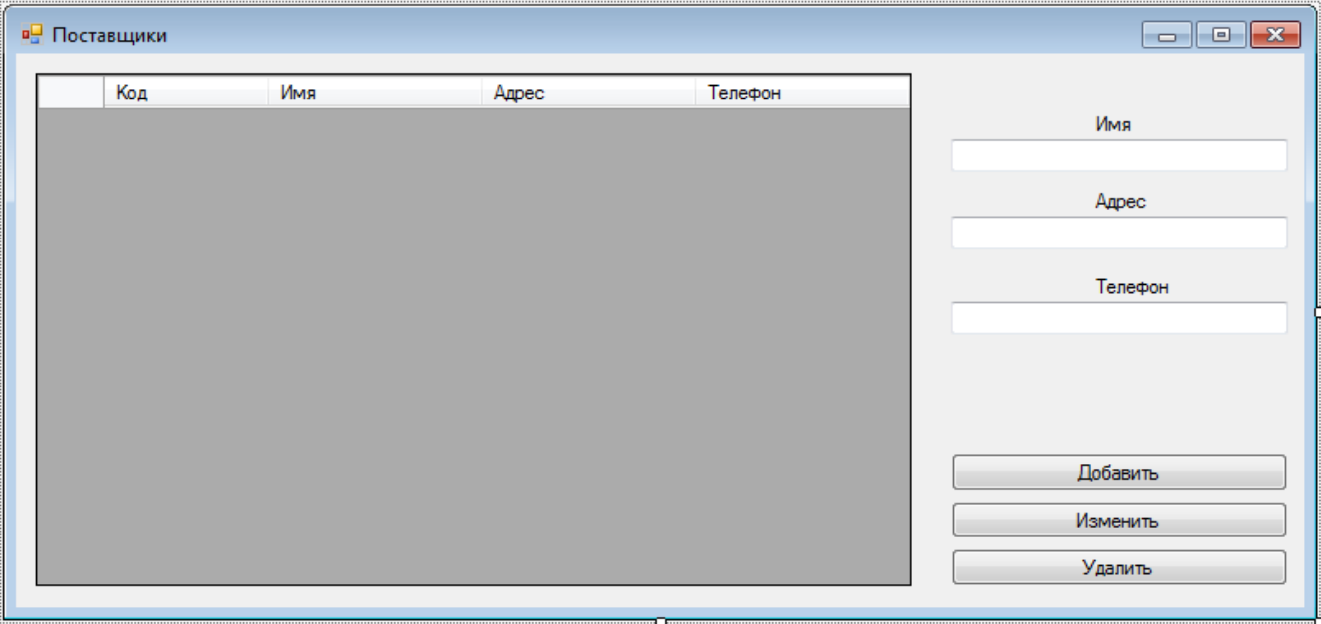


Рисунок 15 – Форма для работы с поставщиками

Аналогичным образом, как и форма для работы с поставщиками, создаются следующие формы: «Товар», «Единицы измерения», «Комплектация», «Стеллажи», «Комплектация».

## 3 Разработка

## 3.1 Выбор инструментов разработки

Разрабатываемый программный продукт написан на языке программирования высокого уровня С# в среде разработки Visual Studio 2022 Community. С# является полностью объектно-ориентированным языком с возможностью наследования.

Visual Studio 2022 Community – это бесплатная полнофункциональная расширяемая среда IDE для создания современных приложений Android, iOS и Windows, а также веб-приложений и облачных служб. Visual Studio 2022 обеспечивает высокое качество кода на протяжении всего цикла жизни программного продукта от проектирования до внедрения. Сегодня это основная и самая эффективная среда разработки, которая олицетворяет собой представление корпорации Microsoft об интеллектуальных клиентских приложениях и позволяет быстро создавать подключаемые к базам данных приложения, способные обеспечить широчайшие возможности для работы пользователей. Данная среда продолжает традиции, заложенные предыдущими версиями продукта, и содержит множество новинок, предназначенных как для упрощения создания приложений для различных платформ отдельными, так и для повышения производительности всей проектной команды [8].

Программный продукт создан на платформе Windows Forms, в которой приложение программируется на базе классов .NET Framework class library. В основе технологии производный класс от System.WinForms.Form, имеющий множество свойств и методов с большими возможностями графического интерфейса к формам [10].

Бэкенд приложения реализован в СУБД PostgreSQL.

PostgreSQL – это передовая открытая система управления объектно-реляционными базами данных. В основном она используется на предприятиях и поддерживает запросы SQL и JSON [6]. У пользователя, есть два способа администрирования СУБД:

* писать запросы через интерфейс командной строки;
* использовать графический пользовательский интерфейс Postgres.

Для создания БД использован кроссплатформенный графический инструмент с открытым исходным кодом графический интерфейс pgAdmin [7].

Преимущества pgAdmin:

* совместим с Linux, Windows, macOS;
* позволяет работать с несколькими серверами одновременно;
* экспорт в файлы типа csv;
* планирование запросов;
* возможность отслеживать ваши сеансы, блокировки БД с помощью панели мониторинга;
* ярлыки в редакторе SQL для более удобной работы;
* встроенный отладчик процедурного языка;
* тщательная документация и активное сообщество.

## 3.2 Основные программные модули

В состав программы входят следующие составные части:

* 1. Модуль авторизации пользователей.
  2. Модуль работы по движению товара.
  3. Модуль управления поставщиками.

Модуль авторизации пользователей обеспечивает:

1. Авторизацию пользователей в приложении.
2. Хранение паролей и логина в базе данных.

Модуль работы по движению товара обеспечивает:

1. Добавление, редактирование, удаление товара.
2. Учёт приема товара
3. Учёт отгрузки товара
4. Поиск необходимого товара

Модуль управления поставщиками обеспечивает:

1. Добавление, редактирование, удаление поставщика.

## 3.3 Файловая структура проекта

Проект НАЗВАНИЕ.sln состоит из 9 форм, файла …..

Сделать скрин обозревателя решений(РИСУНОК)

Спецификация структуры проекта программы приведена в таблице 12.

Таблица 12 – Спецификация структуры проекта программы

|  |  |  |
| --- | --- | --- |
| Наименование | Обозначения | Примечания |
| warehouse.exe | Файл проекта | Связывает все файлы приложения |
| warehouse.cfig | Файл конфигурации | Конфигурация |
| warehouse.dof | Файл параметров | Текущие установки проекта |
| warehouse.res | Файл ресурсов | Содержит пиктограммы, версию и т.д. |
| Unit1.pas | Программный модуль для Form1 | Определяет функциональность формы |
| Unit1.dcu | Объектный файл для Unit1.pas | Откомпилированная версия Unit1.pas |

## 3.4 Серверная часть программного обеспечения

После создания всех необходимых форм была подключена база данных к проекту, строка подключения хранится в конфигурационном файле (рисунок 16) и написаны обработчики на выполнение того или иного действия на нажатие кнопок.

Рисунок 16 – Конфигурационный файл

Листинг кода для создания всех таблиц на языке структурированных запросов PostgreSQL содержится в приложении А.

## 3.5 Клиентская часть программного обеспечения

ТУТ СДЕЛАТЬ ОПИСАНИЕ КОДА

Приложение Б

## 

## 3.6 Организация данных

Входными данными являются:

* информация о товаре;
* информация о пользователях;
* информация о поставщиках;
* информация о количестве пропусков студентов;

При работе программы могут быть получены следующие выходные данные:

* отчет о движении товара;
* отчет об инвентаризации.

## 4 Описание результатов разработки

## 4.1 Тестирование программы

Для тестирования программного обеспечения были составлены тест-кейсы (приложение В), Unit-тест

## Результат разработки

ОПИСАНИЕ СДЕЛАТЬ Приложение Г

## Руководство пользователя

* + 1. Назначение системы.

Программное обеспечение «Склад товаров» – это настольное приложение автоматизированной информационной системы работы склада по поддержке процессов закупа и учёта хранения товаров ООО «Стройлидер».

4.3.2 Условия применения программного продукта.

Для эксплуатации разрабатываемой системы предъявляются следующие минимальные требования к техническому оснащению:

* объем оперативной памяти 4 Гб;
* объем жесткого диска/SSD 256 Гб;
* частота процессора 3,60 ГГц.

Разрабатываемая система должна обеспечивать создание, систематизацию, накопление, хранение и обработку необходимой информации о прошлой, текущей и будущей деятельности предприятия. Поэтому аппаратные средства должны быть рассчитаны на длительную эксплуатацию в условиях увеличения объемов обрабатываемой информации, появления новых средств анализа и представления данных. При выборе подходящей конфигурации также следует учитывать ценовую политику на рынке компьютерной техники (стоимость нового оборудования с улучшенными характеристиками в некоторых случаях меньше стоимости морально устаревшего оборудования).

В связи с этим для работы с системой рекомендуются следующие характеристики технического оснащения:

* объем оперативной памяти 4 Гб и выше;
* объем жесткого диска/SSD 256 Гб и более;
* частота процессора 3,6 ГГц и более.

Дополнительные специальные требования к конфигурации ПК:

* принтер формата А4 для печати выходных документов.
  + 1. Подготовка системы к работе.

Данный программный продукт может функционировать в среде Windows 2010 и выше. Система Windows обеспечивает многозадачный графический интерфейс пользователя, который способствует написанию интерактивных программ.

Инсталляция ПО «Склад товаров» осуществляется программой установки. Для работы системы управления базами данных PostgreSQL на компьютере необходимо развернуть сервер.

4.3.4 Описание операций

Для того чтобы запустить программу, необходимо на рабочем столе найти ярлык с названием «Склад» и запустить его. При запуске информационной системы пользователь должен ввести свои учетные данные: логин и пароль.

Пароль хранится в MD5 алгоритме хеширования. После ввода данных необходимо нажать на «Вход» (рисунок 17).

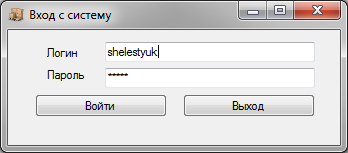


Рисунок 17 – Окно входа в систему

При неверных данных пары логин/пароль система сообщит об этом (рисунок 18). Так как пароль хранится в виде хеша, то это увеличивает безопасность входа в систему. Система производит поиск пользователя по логину, получает его данные и сравнивает с паролем, который ввел пользователь, пароль конвертируется в нужный хеш, с помощью хеш- алгоритма «MD5» и они сверяются.

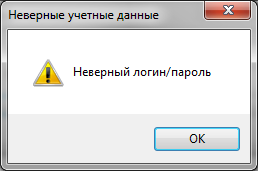


Рисунок 18 – Окно с сообщением об ошибке входа в систему

После удачного входа система отобразит главное окно, с возможностью выбора пунктов из меню (рисунок 19).

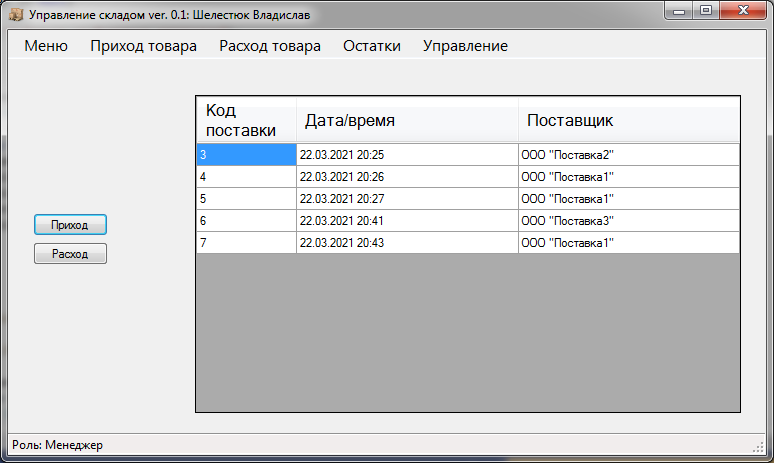


Рисунок 19 – Главное окно «ИС складского учета»

При нажатии на кнопки «Приход» и «Расход» в правой части экрана отобразится информация о последнем приходе или расходе товаров соответственно. Для того чтобы запустить другие окна, необходимо нажать на соответствующий пункт меню. Для оформления процедуры реализации товара выбирается поставщик, номер документа и место размещения товара на стеллаже (рисунок 20).

В правой части окна отображается таблица с выбором нужного товара из списка с указанием его количества (если в системе отсутствует нужное количество, то система сообщит об этом), далее нажимается кнопка «Оформить», данные вводятся в систему и сохраняются. Для просмотра остатка на складе необходимо в главном окне выбрать «Остатки».

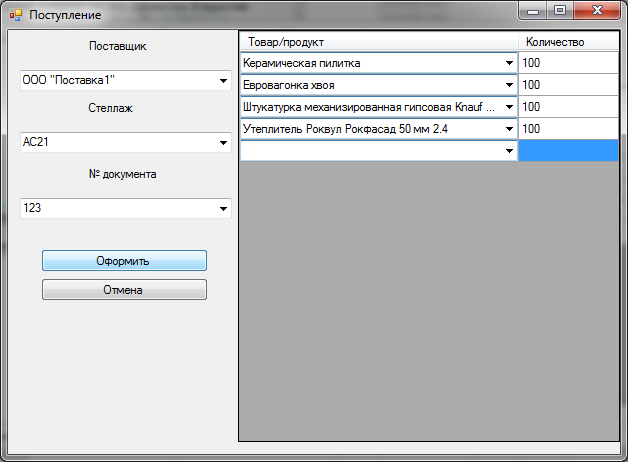


Рисунок 20 – Форма «Поступление товара»

С помощью поиска всегда удобно можно найти и отсортировать необходимые товары и продукты на складе (рисунок 21) при этом на форме отобразится его фактический остаток с названием и изображением товара.

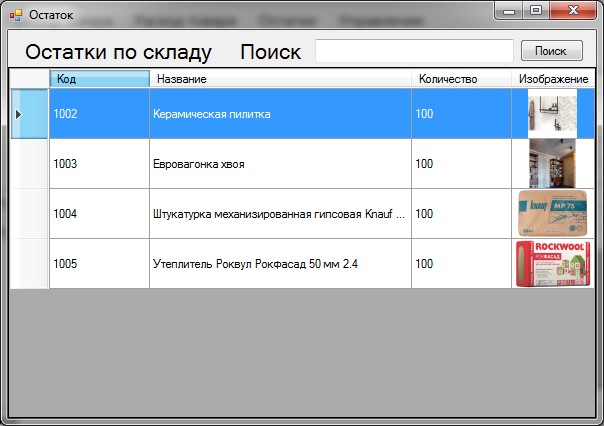


Рисунок 21 – Окно «Остаток»

Если же необходимо отгрузить товары, то нужно выбрать в главном окне «Расход товара». На этой форме отображается код товара, название товара, а также количество и изображение, необходимое для визуального восприятия. В «Выдаче товара» необходимо также выбрать товары и указать их количество (рисунок 22).

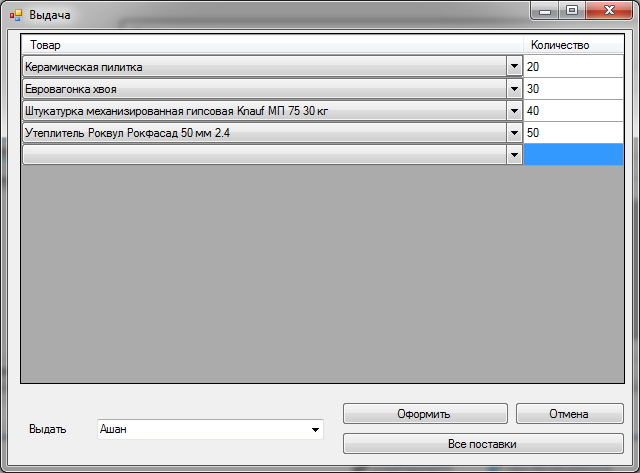


Рисунок 22 – Форма «Выдача товара»

Все товары хранятся в системе, для того, чтобы не создавать каждый раз им новые характеристики. Если же их нет, то приходится создавать их вручную, с помощью формы управления товарами, в которой товару назначается название, стоимость от поставщика, единица измерения, а также фотография товара (рисунок 23). На форме можно редактировать эти параметры в любой момент.

Для удаления ненужного товара необходимо выбрать товар в списке товаров и нажать на кнопку «Удалить». Отредактировать информацию о товаре можно нажав на кнопку «Изменить». Если при добавлении товара нет нужной единицы измерения, то рядом с полем со списком единиц измерения можно открыть сразу форму и добавить нужное измерение.

Параметр – это некая величина, значение которой служит для различения элементов определенного большинства между собой. В своем роде величина, постоянная в пределах данного явления или задачи, но при переходе к другому явлению или задаче может изменить своё значение.

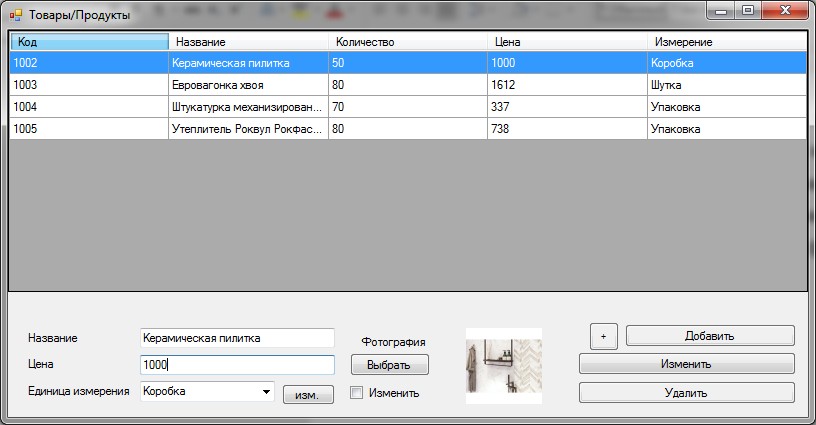


Рисунок 23– Форма «Управление товарами»

Управление поставщиками осуществляется с помощью соответствующей формы (рисунок 24), на которой вводится следующая информация: код, имя, адрес, телефон, в зависимости от различных ситуаций данные могут меняться. Информация отображается в виде табличной сетки данных.

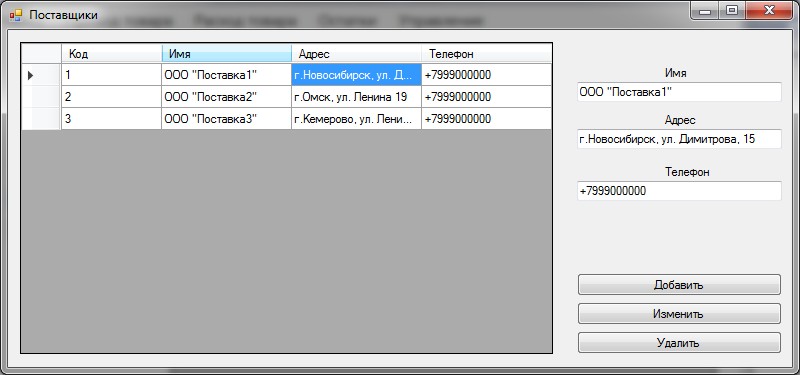


Рисунок 24 – Форма «Управление поставщиками»

4.3.5 Аварийные ситуации

Риск возникновения аварийной ситуации в данной программе очень мал, основными причинами данных ситуаций возможны только несоблюдении условий выполнения технологического процесса, а также при потере части данных в базе данных.

В случае отсутствия подключения к базе данных выводится диалоговое окно «Ошибка подключения к базе данных»

.

# Заключение

Учёт товаров на складе предоставляет важную особенность в деятельности любого склада, а использование информационных технологий на складе предоставляет сокращение времени процесса учета в разы.

В ходе выполнения курсовой работы были рассмотрены цели и задачи информационных систем для поддержки складского учета, изучены технологии, применяемые на складах организаций, организационная структура склада, выявлены должностные роли. Проведенный анализ существующих бизнес-процессов организации показал, что во многих бизнес-процессах есть возможность сократить время на выполнение процесса. В связи с этим и было принято решение о разработке программного обеспечения складского учета.

На основе анализа бизнес-процессов были сформированы требования на разработку программного обеспечения, выполнено прототипирование эскизных интерфейсов системы и определены сущности и атрибуты программного продукта, а также установлены связи между ними.

В соответствии с требованиями и эскизами прототипов было разработано программное обеспечение поддержки складской деятельности организации. После разработки было проведено описание функций в системе.

Разработанная информационная система складского учета позволяет:

* управлять различными данными в базе данных;
* вести учет по приему товаров, а также отгрузки.

Информационная система прошла стадию тестирования в мае 2023 года. В дальнейшем систему можно доработать под необходимые требования от организации.

Все поставленные задачи для достижения цели курсовой работы были выполнены.

# Список использованных источников и литературы

1. ГОСТ Р 58412-2019. Национальный стандарт Российской Федерации. Защита информации. Разработка безопасного программного обеспечения. Угрозы безопасности информации при разработке программного обеспечения (утвежден и введен в действие Приказом Росстандарта от 21.05.2019 N 204-ст).
2. ГОСТ Р ИСО/МЭК 12207-2010. Информационная технология (ИТ). Системная и программная инженерия. Процессы жизненного цикла программных средств (утвержден и введен в действие Приказом Федерального агентства по техническому регулированию и метрологии от 30.11.2010 г. № 631-ст).
3. Автоматизация склада: как автоматизировать работу и бизнес-процессы склада. Первый Бит: сайт. – URL: <https://barnaul.1cbit.ru/blog/avtomatizatsiya-sklada-kak-avtomatizirovat-rabotu-i-biznes-protsessy-sklada/> (дата обращения: 20.04.2023).
4. Зверева В. П. Сопровождение и обслуживание программного обеспечения компьютерных систем: учеб. для студ. учреждений сред. проф. образования / В. П. Зверева, А. В. Назаров. - М.: Издательский центр "Академия", 2018. – 256 с.
5. Многоуровневые системы клиент-сервер [Электронный ресурс] – Режим доступа: https://[www.osp.ru/nets/1997/06/142618/](http://www.osp.ru/nets/1997/06/142618/) (дата обращения: 20.03.2021).
6. Руководство по PostgreSQL: сайт. – URL: <https://metanit.com/sql/postgresql/> (дата обращения: 03.05.2023).
7. Федорова Г. Н. Разработка, администрирование и защита баз данных: учебник для студ. учреждений сред проф. образования – 5-е изд. М.: Издательский центр "Академия", 2021. – 288 с.
8. Федорова Г.Н. Разработка программных модулей программного обеспечения для компьютерных систем: учебник. Среднее профессиональное образование, профессиональная подготовка / Г.Н Федорова. – М.: Академия, 2016. - 384 с.
9. Федорова Н.Г. Осуществление интеграции программных модулей: учебник для студ.учреждений сред.проф.образования. – М.: Издательский центр «Академия», 2021. – 272 с.
10. Язык программирования C#: сайт. – URL: <https://metanit.com/sharp/> (дата обращения: 03.05.2023).

# Приложение А

# Листинги кода PostgreSQL по созданию таблиц

1. *Таблица «goods»*

CREATE TABLE IF NOT EXISTS public.goods

(

"КодТовар" integer NOT NULL GENERATED ALWAYS AS IDENTITY ( INCREMENT 1 START 1 MINVALUE 1 MAXVALUE 2147483647 CACHE 1 ),

"Наименование товара" character varying(250) COLLATE pg\_catalog."default",

"Единица измерения" integer,

CONSTRAINT product\_pkey PRIMARY KEY ("КодТовар"),

CONSTRAINT "Товар" FOREIGN KEY ("Единица измерения")

REFERENCES public.unit ("КодЕдиницы") MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION

NOT VALID

)

1. *Таблица «consumption\_goods»*

CREATE TABLE IF NOT EXISTS public.consumption\_goods

(

"КодРасхода" integer NOT NULL GENERATED ALWAYS AS IDENTITY ( INCREMENT 1 START 1 MINVALUE 1 MAXVALUE 2147483647 CACHE 1 ),

"КодПоставщики" integer,

"КодМенеджер" integer,

"Код документа" integer NOT NULL,

"Дата расхода" timestamp with time zone,

"Товары" character varying COLLATE pg\_catalog."default",

CONSTRAINT consumption\_goods\_pkey PRIMARY KEY ("КодРасхода", "Код документа"),

CONSTRAINT "consumption\_goods\_КодМенеджер\_fkey" FOREIGN KEY ("КодМенеджер")

REFERENCES public.users ("КодПользователи") MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION

NOT VALID,

CONSTRAINT "Расход" FOREIGN KEY ("КодПоставщики")

REFERENCES public.suppliers ("КодПоставщики") MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION

NOT VALID

)

1. *Таблица «delivery\_goods»*

CREATE TABLE IF NOT EXISTS public.delivery\_goods

(

"КодПоставка" integer NOT NULL GENERATED ALWAYS AS IDENTITY ( INCREMENT 1 START 1 MINVALUE 1 MAXVALUE 2147483647 CACHE 1 ),

"КодПоставщики" integer,

"КодМенеджер" integer,

"КодДокумент" integer NOT NULL,

"Дата поставки" timestamp with time zone,

"Товары" character varying COLLATE pg\_catalog."default",

CONSTRAINT delivery\_goods\_pkey PRIMARY KEY ("КодПоставка", "КодДокумент"),

CONSTRAINT "delivery\_goods\_КодМенеджер\_fkey" FOREIGN KEY ("КодМенеджер")

REFERENCES public.users ("КодПользователи") MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION

NOT VALID,

CONSTRAINT "Поставка" FOREIGN KEY ("КодПоставщики")

REFERENCES public.suppliers ("КодПоставщики") MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION

NOT VALID

)

1. *Таблица «equipment»*

CREATE TABLE IF NOT EXISTS public.equipment

(

"КодКомплектация" integer NOT NULL GENERATED ALWAYS AS IDENTITY ( INCREMENT 1 START 1 MINVALUE 1 MAXVALUE 2147483647 CACHE 1 ),

"Стеллаж" integer,

"Ячейка" character varying COLLATE pg\_catalog."default",

"Время" timestamp with time zone,

CONSTRAINT equipment\_pkey PRIMARY KEY ("КодКомплектация"),

CONSTRAINT "Комплектация" FOREIGN KEY ("Стеллаж")

REFERENCES public.rack ("КодСтеллажи") MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION

NOT VALID

)

1. *Таблица «inventory»*

CREATE TABLE IF NOT EXISTS public.inventory

(

"КодИнветаризация" integer NOT NULL GENERATED ALWAYS AS IDENTITY ( INCREMENT 1 START 1 MINVALUE 1 MAXVALUE 2147483647 CACHE 1 ),

"Время" timestamp with time zone,

"Описание" text COLLATE pg\_catalog."default",

"КодПользователи" integer,

CONSTRAINT inventory\_pkey PRIMARY KEY ("КодИнветаризация"),

CONSTRAINT "Инвентаризация" FOREIGN KEY ("КодПользователи")

REFERENCES public.users ("КодПользователи") MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION

NOT VALID

)

1. *Таблица «rack»*

CREATE TABLE IF NOT EXISTS public.rack

(

"КодСтеллажи" integer NOT NULL, GENERATED ALWAYS AS IDENTITY ( INCREMENT 1 START 1 MINVALUE 1 MAXVALUE 2147483647 CACHE 1 ),

"Название" character varying COLLATE pg\_catalog."default",

"Вместимость" numeric,

"Категория" integer,

"Ячейки" integer,

CONSTRAINT shelving\_pkey PRIMARY KEY ("КодСтеллажи"),

CONSTRAINT "Категория" FOREIGN KEY ("Категория")

REFERENCES public.rack\_category ("КодКатегория") MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION

NOT VALID

)

1. *Таблица «rack\_category»*

CREATE TABLE IF NOT EXISTS public.rack\_category

(

"КодКатегория" integer NOT NULL GENERATED ALWAYS AS IDENTITY ( INCREMENT 1 START 1 MINVALUE 1 MAXVALUE 2147483647 CACHE 1 ),

"Названия" character varying(100) COLLATE pg\_catalog."default",

CONSTRAINT "Shelving\_category\_pkey" PRIMARY KEY ("КодКатегория")

)

1. *Таблица «role»*

CREATE TABLE IF NOT EXISTS public.role

(

"КодРоли" integer NOT NULL GENERATED ALWAYS AS IDENTITY ( INCREMENT 1 START 1 MINVALUE 1 MAXVALUE 2147483647 CACHE 1 ),

"Название" character varying(50) COLLATE pg\_catalog."default" NOT NULL,

CONSTRAINT " role\_pkey" PRIMARY KEY ("КодРоли")

)

1. *Таблица «suppliers»*

CREATE TABLE IF NOT EXISTS public.suppliers

(

"КодПоставщики" integer NOT NULL GENERATED ALWAYS AS IDENTITY ( INCREMENT 1 START 1 MINVALUE 1 MAXVALUE 2147483647 CACHE 1 ),

"Название" character varying(250) COLLATE pg\_catalog."default",

"Адрес" character varying(250) COLLATE pg\_catalog."default",

"Телефон" character varying(12) COLLATE pg\_catalog."default",

CONSTRAINT "Suppliers\_pkey" PRIMARY KEY ("КодПоставщики")

)

1. *Таблица «unit»*

CREATE TABLE IF NOT EXISTS public.unit

(

"КодЕдиницы" integer NOT NULL GENERATED ALWAYS AS IDENTITY ( INCREMENT 1 START 1 MINVALUE 1 MAXVALUE 2147483647 CACHE 1 ),

"Название" character varying(50) COLLATE pg\_catalog."default",

CONSTRAINT unit\_pkey PRIMARY KEY ("КодЕдиницы")

)

1. *Таблица «users»*

CREATE TABLE IF NOT EXISTS public.users

(

"КодПользователи" integer NOT NULL GENERATED ALWAYS AS IDENTITY ( INCREMENT 1 START 1 MINVALUE 1 MAXVALUE 2147483647 CACHE 1 ),

"ФИО" character varying(100) COLLATE pg\_catalog."default",

"Логин" character varying(25) COLLATE pg\_catalog."default",

"Пароль" character varying(25) COLLATE pg\_catalog."default",

"Роль" integer,

CONSTRAINT users\_pkey PRIMARY KEY ("КодПользователи"),

CONSTRAINT "Роль" FOREIGN KEY ("Роль")

REFERENCES public.role ("КодРоли") MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION

NOT VALID

)

# Приложение Б

# Листинги кода на языке C# по созданию пользовательского интерфейса

# Приложение В

# Тест-кейсы

|  |  |
| --- | --- |
| Элементы тест-кейса | Значения |
| Название проекта | Программное обеспечение «Склад товаров» |
| Номер версии | v 1.0.0.1 |
| Имя тестера | Ефимов А.С. |
| Даты тестирования | 06.05.2023 – 17.05.2023 |
| Test Сase # | TC\_UI\_1 |
| Приоритет теста | Высокий |
| Название теста | Проверка авторизации с правильным логином и паролем |
| Резюме испытаний | Проверка авторизации с правильным логином и паролем и переходы на форму |
| Шаги тестирования | ввод пары логин-пароль для учетной записи Менеджер;  ввод пары логин-пароль для учетной записи Бухгалтер;  ввод пары логин-пароль для учетной записи Администратор;  проверка перехода на формы личных кабинетов |
| Данные тестирования | Для правильной пары логин – пароль: СТР 170-171 |
| Ожидаемый результат |  |
| Фактический результат |  |
| Предпосылки |  |
| Постусловия |  |
| Статус |  |
| Комментарии |  |

# Приложение Г

# Таблицы базы данных