Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерных систем и сетей

Кафедра информатики

*К защите допустить:*

И.о. заведующего кафедрой

информатики

\_\_\_\_\_\_\_\_\_\_\_\_ С.И. Сиротко

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Дипломного проекта

на тему

**МОБИЛЬНОЕ ПРИЛОЖЕНИЕ ДЛЯ КОНТРОЛЯ И УЧЕТА СКЛАДСКИХ ТОВАРОВ**

БГУИР ДП 1-40 04 01 061 ПЗ

Студент М.В. Логвинович

Руководитель В.В. Сапун

Консультанты:

*от кафедры* А.В. Давыдчик

*по технико-экономическому*

*обоснованию* А.Г. Мазайский

Нормоконтролер

Рецензент

Минск 2025

Решением рабочей комиссии

допущен к защите дипломного проекта

Председатель рабочей комиссии

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ( \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ )

Подпись Инициалы и фамилия

«\_\_\_» июня 2025 г

**РЕФЕРАТ**

МОБИЛЬНОЕ ПРИЛОЖЕНИЕ ДЛЯ КОНТРОЛЯ И УЧЕТА СКЛАДСКИХ ТОВАРОВ: дипломный проект /М. В. Логвинович. – Минск: БГУИР, 2025, – п.з. – 80 с., чертежей (плакатов) – 6 л. формата А4.

Результатом настоящего дипломного проекта является разработка программно-аппаратного средства защиты цифрового устройства от клонирования, позволяющее предотвратить многократное клонирование BIT-образа, означающее тиражирование цифровой системы в обход соглашения с производителем.

При разработке программно-аппаратного средства используется язык описания аппаратуры VHDL.

В первом разделе проводится обзор предметной области и подходов к исследуемой теме. Выдвигаются общие требования к созданию программно аппаратного средства.

Во втором разделе проводится описание компонентов и разработка спецификации функциональных требований.

Третий раздел посвящён разработке функциональных компонентов программно-аппаратного средства.

Четвертый раздел посвящен тестированию программно-аппаратного средства и получению экспериментальных результатов.

Пятый раздел посвящен интерпретации полученных значений и их математической оценке.

В шестом разделе приведено технико-экономическое обоснование разработки программно-аппаратного средства.

Заключение содержит краткие выводы по дипломному проекту.

Дипломный проект является завершённым, поставленная задача решена в полной мере. Планируется дальнейшее развитие и совершенствование программно-аппаратного средства. Проект выполнен самостоятельно, проведён анализ оригинальности в системе «Антиплагиат». Процент оригинальности составляет 88,77%. Цитирования обозначены ссылками на публикации, указанные в «Списке использованных источников».

**СОДЕРЖАНИЕ**

[Введение 8](#_Toc193447661)

[1 Анализ аналогов, литературных источников и формирование требований к программному средству 9](#_Toc193447662)

[1.1 Обзор существующих аналогов 9](#_Toc193447663)

[1.1.1 МойСклад 9](#_Toc193447664)

[1.1.2 Zoho Inventory 10](#_Toc193447665)

[1.1.3 Odoo 10](#_Toc193447666)

[1.1.4 ТоргСофт 11](#_Toc193447667)

[1.1.5 Сравнительный анализ аналогов 11](#_Toc193447668)

[1.4. Формирование требований к ПО 12](#_Toc193447669)

[6 ТЕХНИКО-ЭКОНОМИЧЕСКОЕ ОБОСНОВАНИЕ ЭФФЕКТИВНОСТИ разработки и использования мобильного приложения для контроля и учета складских товаров 14](#_Toc193447670)

[6.1. Характеристика программного средства 14](#_Toc193447671)

[6.2. Расчет затрат на разработку и цену программного средства 15](#_Toc193447672)

[6.3. Расчет результата от разработки и использования программного средства 17](#_Toc193447673)

[6.4. Расчет показателей экономической эффективности разработки и использования программного средства 18](#_Toc193447674)

[заключение 19](#_Toc193447675)

[Список использованных источников 20](#_Toc193447676)

**ПЕРЕЧЕНЬ УСЛОВНЫХ ОБОЗНАЧЕНИЙ, СИМВОЛОВ И ТЕРМИНОВ**

# введение

Данный дипломный проект выполнен мной лично, проверен на заимствования, процент оригинальности составляет ХХ% (отчет о проверке на заимствования прилагается).

# 1 Анализ аналогоВ и формирование требований к программному средству

## 1.1 Обзор существующих аналогов

Для разработки эффективного мобильного приложения для контроля и учета складских товаров важно изучить существующие решения на рынке. Анализ аналогов позволяет выявить сильные и слабые стороны конкурентов, определить лучшие практики и сформулировать требования к разрабатываемому программному продукту.

В качестве аналогов будут рассмотрены некоторые общедоступные популярные решения, представленные на рынке.

### **1.1.1**МойСклад

МойСклад – это облачная система для учета товаров, управления продажами и закупками, предназначенная для малого и среднего бизнеса. Сервис позволяет работать с товароучетными операциями как в веб-версии, так и через мобильный клиент [1].

Сервис поддерживает интеграцию с различными сторонними платформами и сервисами, среди которых:

– 1С;

– маркетплейсы (Ozon, Wildberries, Яндекс.Маркет);

– онлайн-кассы (АТОЛ, Эвотор);

– CRM-системы и сервисы доставки.

На рисунке 1.1 представлен графический интерфейс мобильного приложения.

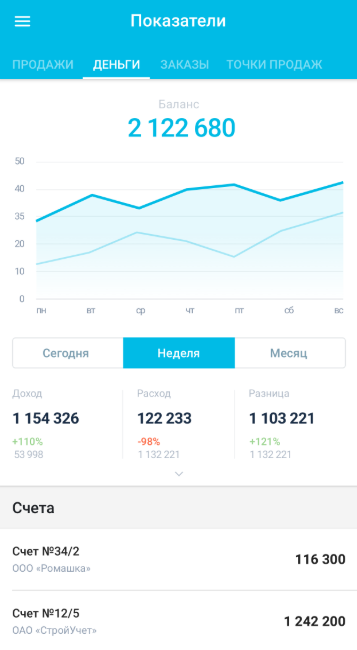


Рисунок 1.1 – МойСклад

Функциональные возможности мобильного приложения включают:

– управление продажами и закупками;

– ведение складского учета;

– автоматизация розничных точек;

– формирование отчетности (финансовой, складской, аналитической);

– контроль финансов;

– работа с клиентской базой.

Приложение обладает интуитивно понятным интерфейсом, что делает его доступным даже для пользователей без специальной подготовки. Оно разработано для платформ iOS и Android и доступно полностью на русском языке. Техническая поддержка доступна через чат, телефон и электронную почту, что облегчает решение возникающих вопросов при работе с системой.

Несмотря на достоинства, мобильный клиент имеет ряд ограничений. В сравнении с веб-версией он предоставляет сокращенный функционал, что может быть неудобно для пользователей, которым требуется полный спектр возможностей сервиса. Кроме того, приложение не поддерживает офлайн-режим, что делает его зависимым от интернет-соединения. Для малого бизнеса сложность начальной настройки и необходимость разбираться в большом количестве функций могут стать дополнительным барьером при внедрении системы. Тем не менее, он удобен для оперативного управления складским учетом и продажами.

### **1.1.2** Zoho Inventory

Zoho Inventory – это облачное решение для управления запасами, заказами и продажами, ориентированное на малый и средний бизнес. Система интегрируется с другими продуктами Zoho, такими как CRM и бухгалтерия, а также с внешними сервисами, что обеспечивает комплексное ведение бизнеса [2].

Помимо полноценной веб-версии, система предлагает мобильное приложение (см. рисунок 1.2), адаптированное для работы на платформах iOS и Android. Приложение Zoho Inventory предоставляет основные инструменты для управления запасами и заказами, что делает его полезным для оперативного контроля складских операций:

1 Управление товарами. Создание новых позиций, добавление SKU и отслеживание товаров по серийным номерам или партиям.

2 Обработка заказов. Формирование и отслеживание заказов, выставление счетов и управление логистикой через интуитивно понятный интерфейс.

3 Уведомления о пополнении запасов. Настройка уровней повторного заказа с получением уведомлений при снижении запасов ниже установленного уровня.

4 Сканирование штрих-кодов. Использование камеры мобильного устройства для сканирования штрих-кодов, что ускоряет процесс добавления и поиска товаров.

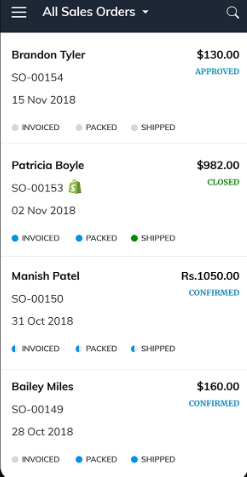


Рисунок 1.2 – Zoho Inventory

Преимущества Zoho Inventory:

– дружелюбный интерфейс и легкость освоения программы новыми сотрудниками;

– возможность синхронизации с другими продуктами Zoho и сторонними сервисами.

Недостатки Zoho Inventory:

– поддержка только английского языка;

– отсутствие офлайн-режима;

– ограниченная функциональность мобильной версии.

### **1.1.3**CloudShop

CloudShop – это программный продукт для автоматизации розничной торговли и управления продажами. Он позволяет вести учет, контролировать складские запасы и обрабатывать заказы как через браузер, так и с мобильных устройств [3].

Основной функционал приложения охватывает ключевые процессы торговли:

1 Складской учет. Контроль остатков, учет закупок и работа с поставщиками.

2 Готовая онлайн-касса для продажи маркированного товара.

3 Продажи и возвраты. Оформление чеков, работа со скидками и дисконтными картами.

4 Печать ценников и этикеток. Генерация штрихкодов и ценников с помощью встроенного конструктора.

5 Работа с клиентами. Ведение базы покупателей, настройка программ лояльности.

На рисунке 1.3 представлен графический интерфейс мобильного приложения CloudShop.

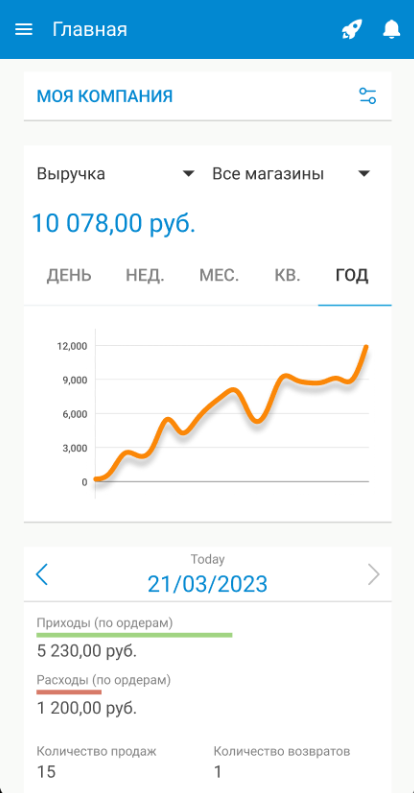


Рисунок 1.3 – CloudShop

CloudShop подходит для магазинов разного формата: от небольших торговых точек до сетевых предприятий. Бесплатный тариф позволяет протестировать основные функции, а мобильное приложение делает управление бизнесом более гибким и доступным.

Среди недостатков можно выделить то, что функционал бесплатной версии рассчитан только на один магазин и одного сотрудника, что может ограничивать возможности бизнеса по мере роста. Кроме того, для работы требуется стабильное интернет-соединение, что может вызывать неудобства при перебоях в сети.

### **1.1.4** Программа «Учет товаров – простой склад»

«Учет товаров – простой склад» — это мобильное приложение для Android, предназначенное для ведения складского учета. Оно ориентировано на малый бизнес, индивидуальных предпринимателей и небольшие торговые точки, которым необходимо вести базовый учет товаров без сложных интеграций и дополнительных модулей [4].

В бесплатной версии доступен следующий функционал:

– создание карточек товара с фото, наименованием, штрихкодом;

– группировка товаров по категориям;

– импорт данных из Excel;

– учет приходных и расходных операций;

– формирование отчетов по обороту товаров, продажам и поступлениям;

– инвентаризация склада.

В платной версии становится доступным указание цен товаров, соответственно предприниматели получают возможность формировать отчеты по прибыли. Появляется возможность устанавливать скидки для покупателей, вести учет по поставщикам и покупателям, а также по нескольким складам или магазинам.

На рисунке 1.4 приведен графический интерфейс мобильного приложения «Учет товаров – простой склад».

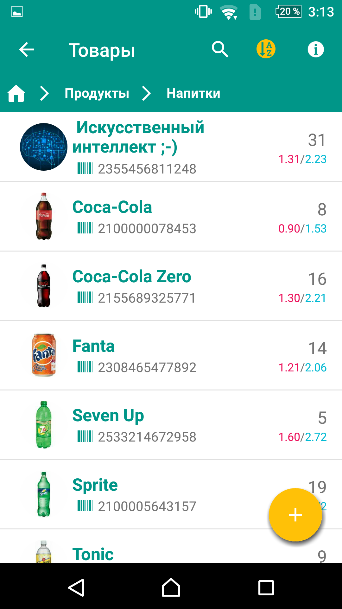


Рисунок 1.4 – Графический интерфейс «Учет товаров – простой склад»

## 1.2. Формирование требований к ПО

Мобильное приложение должно автоматизировать основные складские операции, минимизировать влияние человеческого фактора и повысить прозрачность учета товаров.

Программное средство должно обеспечивать выполнение следующих основных функций:

1 Управление товарами. Создание карточек товаров с наименованием, описанием, фотографией и штрихкодом. Группировка товаров по категориям. Учет остатков товаров.

2 Приемка товара. Регистрация поступлений товаров с указанием поставщика и даты поставки. Автоматическое обновление складских остатков после приемки.

3 Хранение и учет. Просмотр текущего состояния склада. Поддержка поиска и фильтрации товаров. Проведение инвентаризации.

4 Продажа и отгрузка. Регистрация продаж и возвратов. Фиксация бракованного товара.

5 Формирование отчетности. Генерация отчетов по движению товаров, продажам и поступлениям. Анализ остатков на складе. Формирование отчетов по прибыли.

Мобильное приложение разрабатывается для платформы Android, обеспечивая удобный и интуитивно понятный интерфейс, доступный пользователям без специальной подготовки. Высокая производительность позволит быстро выполнять основные операции, такие как добавление, поиск и обновление товаров. Важное внимание уделяется безопасности, включая авторизацию пользователей, разграничение прав доступа, защиту информации от несанкционированного доступа и контроль целостности данных. Надежность системы достигается за счет устойчивости к сбоям и потере данных, а также наличия механизмов восстановления информации после аварийных ситуаций.

Приложение разрабатывается для малого бизнеса и не предполагает поддержку сложных складских систем. В первой версии не предусматривается интеграция с онлайн-кассами и маркетплейсами, но возможна реализация в дальнейшем. Приложение ориентировано на использование одним владельцем или небольшим числом сотрудников.

Таким образом, разработка мобильного приложения для складского учета позволит заказчику автоматизировать складские операции, снизить вероятность ошибок при ведении учета и повысить эффективность бизнес-процессов.

# 2 Проектирование МОБИЛЬного приложения для КОНТРОЛЯ И учета сКЛАДСКИХ ТОВАРОВ

## 2.1 Диаграмма вариантов использования

В данном разделе представлены пользовательские требования к программному средству, отображенные в виде диаграммы вариантов использования (рисунок 2.1). Диаграмма демонстрирует ключевые сценарии работы с системой, а также взаимодействие пользователей с основными функциями приложения.

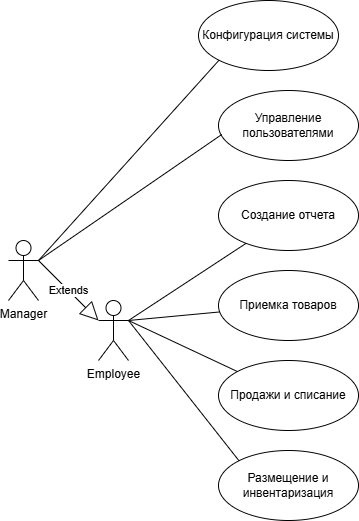


Рисунок 2.1 – Диаграмма вариантов использования программного средства

Мобильное приложение для контроля и учета складских товаров поддерживает два типа пользователей: администратор (Admin) и сотрудник, работник склада (Manager). Администратор обладает расширенными возможностями и может управлять учетными записями сотрудников, изменять настройки приложения, просматривать аналитические отчеты по продажам и прибыли. Он также имеет полный доступ ко всем функциям, доступным сотруднику.

Сотрудник выполняет операции, связанные с управлением товарными позициями, включая добавление новых товаров, редактирование их характеристик и учет изменений на складе. Он может фиксировать поступления товаров, оформлять списания и возвраты, а также выполнять инвентаризацию. В рамках своих полномочий сотрудник имеет доступ к формированию отчетов о движении товаров, но не может просматривать финансовую аналитику и управлять настройками учетных записей.

Таким образом, пользователю с ролью Employee доступны следующие варианты использования:

– создание отчета;

– приемка товаров;

– добавление продаж и списаний;

– выполнение размещения товаров и инвентаризации.

Пользователь с ролью Admin, в свою очередь, связан с Employee отношением обобщения, которое показывает доступность всех вариантов использования сотрудника.

## 2.2 Пользовательский интерфейс и удобство использования

Разработка мобильного приложения направлена на создание удобного, интуитивно понятного и эффективного пользовательского интерфейса, который минимизирует вероятность ошибок и ускоряет выполнение складских операций.

Разработка интерфейса осуществляется с учетом принципов удобства использования, доступности и визуальной гармонии. Приложение должно быть адаптировано для работы на различных мобильных устройствах, обеспечивая корректное отображение элементов интерфейса и удобство взаимодействия с сенсорными экранами.

Основной экран включает в себя ключевые функции складского учета, представленные в виде удобных кнопок и быстрых действий, позволяющих минимизировать количество нажатий для выполнения базовых операций. Использование крупных интерактивных элементов, четких контрастных шрифтов и интуитивных пиктограмм способствует быстрому восприятию информации.

Система навигации организована по принципу минимального количества уровней вложенности, что позволяет пользователю легко находить необходимую информацию и совершать действия без избыточных переходов. Важные функции, такие как добавление товаров, оформление поступлений и инвентаризация, доступны на главном экране или в быстром доступе через панель навигации.

Для сотрудников, работающих на складе, предусмотрена поддержка работы с камерой устройства для сканирования штрихкодов, что ускоряет процесс идентификации товаров. Приложение реализует возможность работы в темном и светлом режимах для снижения нагрузки на зрение при длительном использовании.

Особое внимание уделяется адаптивности и доступности интерфейса: поддержка масштабирования текста, цветовые схемы с учетом потребностей людей с нарушениями зрения, а также минимизация когнитивной нагрузки за счет использования привычных элементов управления.

Таким образом, графический интерфейс приложения позволит упростить складские процессы, снизить количество ошибок при работе с данными и повышение общей эффективности использования мобильного приложения.

На рисунке 2.2 приведен прототип главного экрана мобильного приложения.

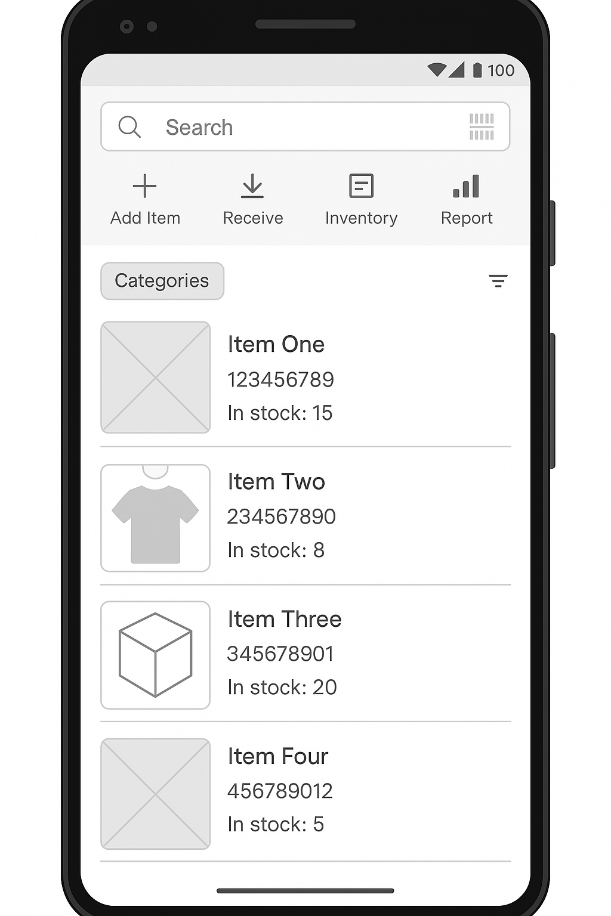


Рисунок 2.2 –Прототип главного экрана приложения

# 3 Проектирование программного средства

## 3.1 Разработка архитектуры ПС

Разрабатываемое мобильное приложение для контроля и учета складских товаров, построенное по принципам чистой архитектуры (Clean Architecture) с использованием шаблона проектирования MVI (Model-View-Intent). Архитектура приложения разделена на четко определённые слои, каждый из которых отвечает за конкретные задачи и обеспечивает высокий уровень модульности, тестируемости и масштабируемости.

Для обеспечения бизнес-логики, обработки данных и взаимодействия с внешними источниками информации применяются следующие архитектурные уровни:

1 Приложение (app) — отвечает за запуск приложения, навигацию и связывание слоёв.

2 Презентационный слой (presentation) — реализует MVI-подход, включает состояния экрана, намерения пользователя и управление состоянием через ViewModel.

3 Слой домена (domain) — содержит бизнес-логику, интерфейсы репозиториев и use case-ов.

4 Слой данных (data) — реализует источники данных (локальные и удалённые), включая работу с Room, Retrofit и репозитории.

5 Инфраструктура (например: network, database) — содержит конкретные реализации взаимодействия с API и локальной БД.

Серверная часть построена с использованием Kotlin + Spring Boot, предоставляет REST API для обмена данными между клиентом и сервером. В архитектуре бэкенда выделяются следующие компоненты:

– Контроллеры (REST Controllers)

– Сервисы (Services)

– Репозитории (Repositories)

– Сущности и DTO

– Конфигурации безопасности (JWT, OAuth2, CORS и т.д.)

Архитектура системы обеспечивает раздельную ответственность компонентов, что упрощает сопровождение, масштабирование и повторное использование кода. Взаимодействие между клиентом и сервером осуществляется через HTTP-запросы, передающие данные в формате JSON.

## 3.2 Разработка логической и физической моделей базы данных

Логическая модель базы данных представляет собой основу для хранения и обработки информации в мобильном приложении. Она представляет собой детализированную версию концептуальной модели, в которой схематически отражены ограничения, наименования сущностей и их взаимосвязи, не зависящие от конкретной платформы.

Набор основных сущностей, который необходим для разработки приложения, в себя включает:

– товар (Product) – хранит информацию о наименовании, количестве, цене и штрихкоде товаров.

– категория товаров (Category) – объединяет товары по видам и характеристикам.

– приемка товара (StockReceipt) – фиксирует поступление товаров на склад от поставщиков.

– позиции приемки (StockReceiptItem) – содержит сведения о товарах, входящих в конкретную поставку.

– продажа (Sale) – регистрирует факт реализации товара клиенту.

– позиции продажи (SaleItem) – описывает состав продажи, включая количество и стоимость товаров.

– поставщик (Supplier) – содержит информацию о контрагентах, от которых поступает товар.

– сотрудник (Employee) – представляет пользователей системы с различными уровнями доступа.

– инвентаризация (InventoryCheck) – фиксирует результаты сверки остатков товаров.

– позиции инвентаризации (InventoryCheckItem) – содержит информацию о расхождениях между ожидаемым и фактическим количеством товара.

Данная модель обеспечивает структурированное хранение информации, необходимое для эффективного управления складскими процессами. На рисунке 2.3 представлена схема логической модели базы данных.

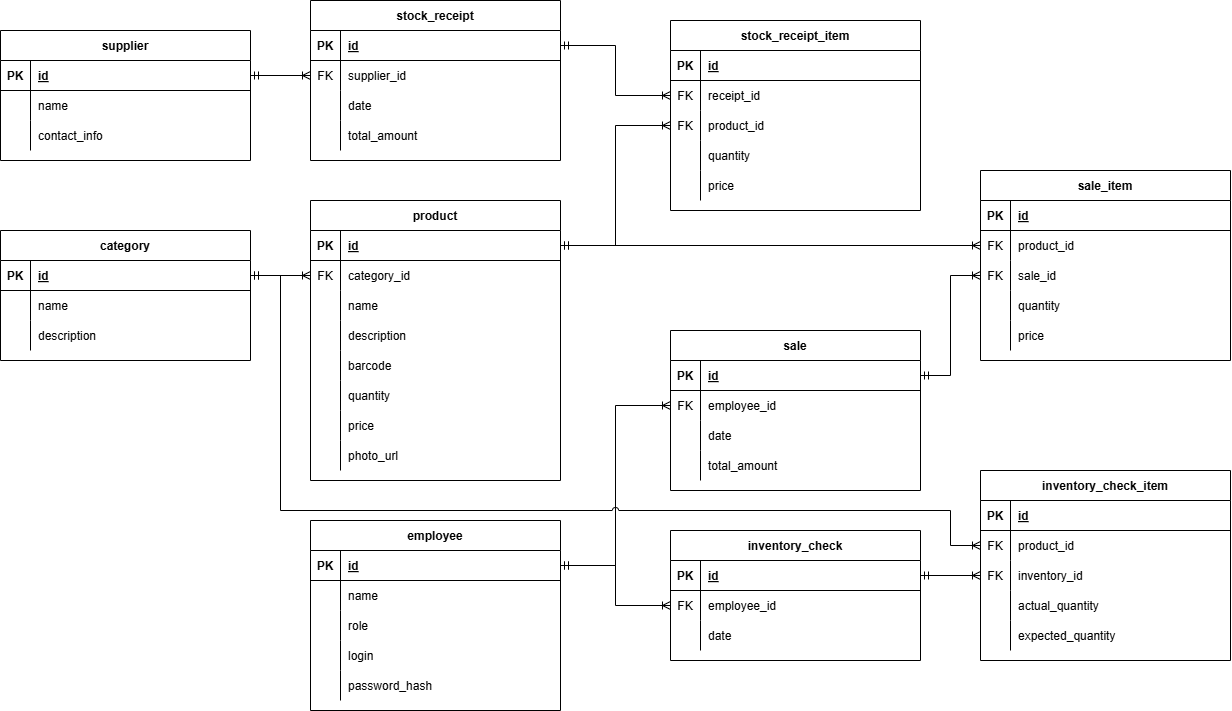


Рисунок 2.3 – Логическая модель базы данных

Сущность «product» включает следующие атрибуты:

– id – уникальный идентификатор;

– name – наименование товара;

– description – описание;

– barcode – штрихкод;

– category\_id – ссылка на категорию;

– quantity – текущее количество на складе;

– price – цена за единицу;

– photo\_url – ссылка на изображение.

Сущность «category» включает следующие атрибуты:

– id – уникальный идентификатор;

– name – наименование категории;

– description – описание.

Сущность «stock\_receipt» содержит следующие атрибуты:

– id – уникальный идентификатор;

– supplier\_id – ссылка на поставщика;

– date – дата поступления;

– total\_amount – общая сумма приемки.

Сущность «stock\_receipt\_item» включает следующие атрибуты:

– id – уникальный идентификатор;

– receipt\_id – ссылка на приемку;

– product\_id – ссылка на товар;

– quantity – количество;

– price – цена за единицу.

Сущность «sale» содержит следующие атрибуты:

– id – уникальный идентификатор;

– date – дата продажи;

– total\_amount – общая сумма;

– employee\_id – ссылка на продавца.

Сущность «sale\_item» включает следующие атрибуты:

– id – уникальный идентификатор;

– sale\_id – ссылка на продажу;

– product\_id – ссылка на товар;

– quantity – количество;

– price – цена за единицу.

Сущность «supplier» включает следующие атрибуты:

– id – уникальный идентификатор;

– name – название компании;

– contact\_info – контактные данные.

Сущность «employee» включает следующие атрибуты:

– id – уникальный идентификатор;

– name – ФИО;

– role – должность;

– login – учетная запись;

– password\_hash – защищенный пароль.

Сущность «inventory\_check» содержит следующие атрибуты:

– id – уникальный идентификатор;

– date – дата проведения;

– employee\_id – сотрудник, проводивший инвентаризацию.

Сущность «inventory\_check\_item» включает следующие атрибуты:

– id – уникальный идентификатор;

– inventory\_id – ссылка на инвентаризацию;

– product\_id – ссылка на товар;

– actual\_quantity – фактическое количество;

– expected\_quantity – ожидаемое количество.

Между сущностями обеспечиваются следующие связи:

1 Один товар может относиться к одной категории, но категория может содержать много товаров.

2 Один товар может участвовать во многих приемках и продажах.

3 Один поставщик может быть связан с несколькими приемками.

4 Один сотрудник может проводить несколько инвентаризаций или оформлять продажи.

5 Один чек продажи может включать несколько позиций товаров.

Разработанная логическая модель базы данных обеспечивает целостное хранение информации о товарах, складских операциях и движении запасов. Она формирует основу для надежного и масштабируемого хранения данных в мобильном приложении.

Физическая модель данных дополняет логическую, уточняя её с учетом особенностей выбранной технологии базы данных. Оба уровня моделирования определяют структуру, организацию и принципы обработки данных, обеспечивая их эффективное хранение, извлечение и управление.

На рисунке 2.4 представлена физическая модель базы данных.

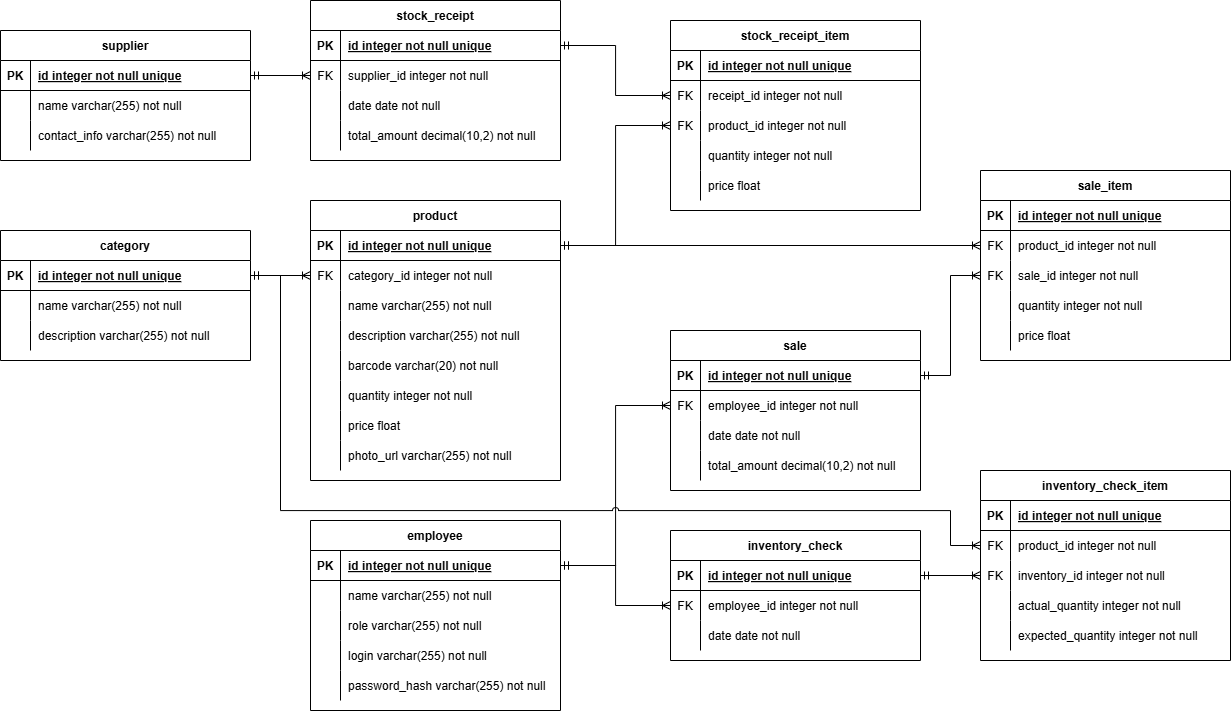


Рисунок 2.4 – Физическая модель базы данных

## 3.3 Разработка алгоритма ПС и алгоритмов отдельных модулей

# 6 ТЕХНИКО-ЭКОНОМИЧЕСКОЕ ОБОСНОВАНИЕ ЭФФЕКТИВНОСТИ разработки и использования мобильного приложения для контроля и учета складских товаров

## 6.1. Характеристика программного средства

В условиях современного рынка, где конкуренция и требования к операционной эффективности постоянно растут, многие предприятия сталкиваются с проблемами ручного учета, что приводит к ошибкам, потерям времени и ресурсов. Разработанное в рамках дипломного проекта программное средство представляет собой мобильное приложение для контроля и учета складских товаров, которое позволяет устранить эти недостатки, обеспечивая автоматизацию и повышение точности учета. Это решение особенно актуально для компаний, стремящихся оптимизировать свои бизнес-процессы и минимизировать издержки, связанные с человеческим фактором.

Цель разработки – автоматизация процессов складского учета, повышение точности и скорости операций, снижение затрат на управление складом.

Область применения – складские комплексы, дистрибьюторские центры, производственные предприятия, торговые сети и логистические компании.

Функциональные возможности приложения:

– ведение базы складских товаров (идентификатор, наименование, количество, срок годности и пр.);

– автоматизированный учет поступления и расхода товаров;

– контроль остатков и анализ товарооборота;

– формирование отчетов и аналитики.

Актуальность разработки обусловлена необходимостью устранения ошибок, связанных с ручным учетом, оптимизацией логистики и складских процессов, а также повышением скорости и точности инвентаризации. Приложение позволяет снизить затраты на персонал за счет автоматизации рутинных операций и удовлетворяет потребность в современных инструментах для эффективного управления складскими операциями, что особенно важно в условиях растущих требований к точности и скорости обработки данных на складах.

Ожидаемый экономический эффект от внедрения программного средства:

– снижение потерь от недостач и излишков;

– повышение операционной эффективности склада;

– экономия времени на проведение инвентаризации;

– улучшение контроля за товарными запасами;

– сокращение времени на поиск и обработку товаров;

– повышение удовлетворенности клиентов за счет оперативного выполнения заказов;

– снижение затрат на хранение излишков и минимизация простоев из-за отсутствия товаров.

## 6.2. Расчет затрат на разработку и цену программного средства

Для разработки программного средства необходимы следующие категории исполнителей:

− инженер-программист, отвечающий за разработку серверной части;

− инженер-программист, отвечающий за разработку клиентской части;

− тестировщик;

− дизайнер;

Расчет затрат на основную заработную плату специалистов осуществляется в таблице 6.1. Принимая среднее количество рабочих дней в месяце равным 21, получим, что количество рабочих часов каждого сотрудника равно 168 часов в месяц.

Принимаем размер премии и иных стимулирующих выплат равным 50% от основной заработной платы.

Таблица 6.1 – Расчет затрат на основную заработную плату исполнителей, необходимых для разработки программного средства

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Категория исполнителя | Месячный оклад, р. | Часовой оклад, р. | Трудоемкость работ, ч | Итого, р. |
| Инженер-программист (серверная часть) | 2500 | 14,88 | 300 | 4464 |
| Инженер-программист (клиентская часть) | 2300 | 13,69 | 280 | 3833,2 |
| Тестировщик | 1800 | 10,71 | 200 | 2142 |
| Дизайнер | 2000 | 11,90 | 180 | 2142 |
| Итого | | | | 12581,2 |
| Премия и иные стимулирующие выплаты | | | | 6290,6 |
| Всего затрат на основную заработную плату разработчиков | | | | 18871,8 |

Расчет дополнительной заработной платы разработчиков осуществляется по формуле:

|  |  |
| --- | --- |
|  | (6.1) |

где – норматив дополнительной заработной платы, принятый в размере 10%.

Расчет отчислений на социальные нужды выполняется по формуле:

|  |  |
| --- | --- |
|  | (6.2) |

где – норматив отчислений в ФЗСН и Белгосстрах, принятый в размере 34,6%.

Расчет отчислений на прочие расходы производится по формуле:

|  |  |
| --- | --- |
|  | (6.3) |

где – норматив прочих расходов, принятый в размере 30%.

Общая сумма затрат на разработку высчитывается по формуле:

|  |  |
| --- | --- |
|  | (6.4) |

Плановая прибыль, включаемая в цену программного средства, рассчитывается по формуле:

|  |  |
| --- | --- |
|  | (6.5) |

где – рентабельность затрат на разработку программного средства в размере 30%.

Отпускная цена программного средства осуществляется по формуле:

|  |  |
| --- | --- |
|  | (6.6) |

В таблице 6.2 указаны итоговые расчетные данные по формированию цены программного средства на основе затрат.

Таблица 6.2 – Формирование цены программного средства на основе затрат

|  |  |  |
| --- | --- | --- |
| Наименование статьи затрат | Формула/таблица для расчета | Значение, р. |
| 1. Основная заработная плата разработчиков | Таблица 6.1 | 18871,8 |
| 2. Дополнительная заработная плата разработчиков | Формула (6.1) | 1887,18 |
| 3. Отчисления на социальные нужды | Формула (6.2) | 7182,61 |
| 4. Прочие расходы | Формула (6.3) | 5661,54 |
| 5. Общая сумма затрат на разработку | Формула (6.4) | 33603,13 |
| 6. Плановая прибыль, включаемая в цену программного средства | Формула (6.5) | 10080,94 |
| 7. Отпускная цена программного средства | Формула (6.6) | 43684,07 |

## 6.3. Расчет результата от разработки и реализации программного средства

Предполагается, что программное средство будет реализовано организацией-разработчиком по отпускной цене. В таком случае, экономический эффект для организации-разработчика будет определяться по формуле:

|  |  |
| --- | --- |
|  | (6.7) |

где ‒ ставка налога на прибыль.

Таким образом, экономический эффект от разработки для организации-разработчика составит 8064,75р.

## 6.4. Расчет показателей экономической эффективности разработки и реализации программного средства

Для организации-разработчика программного средства оценка эффективности осуществляется с помощью расчета простой нормы прибыли по формуле:

|  |  |
| --- | --- |
|  | (6.8) |

По итогам оценки эффективности инвестиций (затрат) можно сделать вывод, что разработка программного средства является экономически целесообразной как для организации-разработчика, так и для организации-заказчика. Экономический эффект от разработки для организации-разработчика составил 8064,75 рублей, что подтверждает рентабельность проекта. Простая норма прибыли, рассчитанная для организации-разработчика, составила 23,99%, что свидетельствует о высокой эффективности вложенных средств.

Для организации-заказчика разработка программного средства силами сторонней компании выгоднее, чем самостоятельная реализация или приобретение аналогичного решения у других разработчиков. Это обусловлено отсутствием необходимых кадровых ресурсов, высокой стоимостью самостоятельной разработки, а также временными затратами на создание и внедрение подобного решения. Кроме того, организация-разработчик обладает необходимым опытом и специализированными знаниями, что позволяет минимизировать риски и обеспечить высокое качество продукта.

Таким образом, проект является экономически обоснованным и привлекательным для обеих сторон, обеспечивая значительный экономический эффект и повышение операционной эффективности для организации-заказчика.

# заключение

# Список использованных источников

[1] МойСклад [Электронный ресурс]. – Электронные данные. – Режим доступа: https://www.moysklad.ru/by/. – Дата доступа: 02.04.2025.

[2] Online Inventory Software – Zoho Inventory [Электронный ресурс]. – Электронные данные. – Режим доступа: https://www.zoho.com/inventory/. – Дата доступа: 02.04.2025.

[3] CloudShop – бесплатная программа для магазина. Учет товаров в магазине [Электронный ресурс]. – Электронные данные. – Режим доступа: https://cloudshop.ru/. – Дата доступа: 02.04.2025.