

Министерство образования Иркутской области
Государственное бюджетное профессиональное образовательное учреждение
Иркутской области
«Иркутский авиационный техникум»
(ГБПОУИО «ИАТ»)

КП.09.02.07-5.25.221.09 ПЗ

**ИНФОРМАЦИОННАЯ СИСТЕМА
«УПРАВЛЕНИЕ СКЛАДОМ И ЗАПАСАМИ»**

Председатель ВЦК: _____ (М.А. Куряццева)
(подпись, дата)

Руководитель: _____ (М.А. Куряццева)
(подпись, дата)

Студент: _____ (А.А. Горшков)
(подпись, дата)

Иркутск 2025

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	3
1 Описание предметной области ИС.....	5
2 Анализ инструментальных средств для разработки.....	7
3 Техническое задание.....	15
4 Проектирование ИС	17
4.1 Архитектура ИС	17
4.2 Структурные схемы ИС.....	18
4.3 Функциональные схемы ИС.....	22
4.4 Проектирование базы данных.....	24
4.5 Проектирование интерфейса ИС	29
5 Разработка ИС	34
5.1 Разработка кода клиентской части ИС	34
5.2 Разработка кода серверной части ИС.....	40
5.3 Разработка базы данных ИС	44
6 Тестирование ИС.....	48
6.1 Тестовый сценарий	48
6.2 Тестовые наборы данных	52
6.3 Методы тестирования	54
7 Документирование ИС	56
7.1 Руководитель по установке ИС	56
7.2 Руководство пользователя ИС	58
ЗАКЛЮЧЕНИЕ	74
СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ	76
Приложение А – Техническое задание	77
Приложение Б – Листинг «Провайдер YandexServiceProvider».....	87

Изм.	Лист	№ докум.	Подпись	Дата	КП.09.02.07-5.25.221.09.П3		
Разраб.	Горшков А. А.				ИНФОРМАЦИОННАЯ СИСТЕМА «УПРАВЛЕНИЕ СКЛАДОМ И ЗАПАСАМИ» пояснительная записка	Лит.	Лист
Провер.	Кудрявцева М.А.					2	87
Реценз.							
Н. Контр.							
Утверд.	.				ГБПОУИО «ИАТ» ИС-22-1		

ВВЕДЕНИЕ

Современные информационные системы (далее – ИС) управления складом обеспечивают автоматизацию процессов приёма, хранения и выдачи товаров, а также позволяют формировать заявки, контролировать остатки и быстро находить нужные позиции. Это особенно важно для предприятий, где скорость обработки заявок и точность данных напрямую влияют на качество обслуживания клиентов и конкурентоспособность организации.

Актуальность темы курсового проекта обусловлена необходимостью цифровизации и автоматизации складских процессов в условиях современного бизнеса. Рост количества товаров, необходимость оперативного контроля запасов и обработки заявок требуют использования информационных технологий, которые позволяют минимизировать ошибки учёта, ускорить обработку данных и повысить эффективность работы склада.

Целью курсового проекта является проектирование и разработка информационной системы управления складом и запасами.

Для достижения поставленной цели необходимо решить следующие задачи:

- Описать предметную область.
- Провести анализ инструментальных средств разработки.
- Разработать техническое задание: определить функциональные требования, целевую аудиторию и технологии реализации.
- Спроектировать ИС: разработать структуру базы данных, взаимодействия пользователей и интерфейс.
- Реализовать основной функционал системы, включая регистрацию пользователей и сотрудников склада.
- Провести тестирование информационной системы.
- Составить руководство пользователя.

Изм.	Лист	№ докум.	Подпись	Дата
------	------	----------	---------	------

Результатом работы является действующая система, обладающая базовым набором функций для учёта товаров и заявок, а также обеспечения взаимодействия между клиентами, сотрудниками склада и администратором.

Практическая значимость работы заключается в том, что разработанный системы может служить основой для создания полноценного программного продукта, востребованного предприятиями, нуждающимися в эффективной автоматизации складской деятельности.

Иzm.	Лист	№ докум.	Подпись	Дата	KP.09.02.07-5.25.221.09.ПЗ	Лист 4
------	------	----------	---------	------	----------------------------	-----------

1 Описание предметной области ИС

Информационная система «Система управления складом и запасами» представляет собой комплекс взаимосвязанных программных модулей и баз данных, предназначенных для автоматизации процессов учёта товаров, управления складскими операциями, контроля остатков и формирования отчётности.

Основные функции ИС:

- Взаимодействие с пользователем: система предоставляет интуитивно понятный и персонализированный веб-интерфейс для работы пользователей.
- Обработка данных: система принимает, обрабатывает и хранит данные о пользователях, товарах, расположениях на складе и заявках.
- Учёт движения товаров: система обеспечивает ведение журнала товаров, позволяя отслеживать состояние и количество товаров.
- Учёт заявок: пользователи могут формировать заявки на товары, а сотрудники склада видеть активные заявки и обрабатывать их.
- Управление складом: отображение расположения товаров и контроль остатков.
- Анализ и отчётность: система анализирует товары и заявки, а также формирует отчёты в различных форматах для дальнейшего анализа.

Информационная система ориентирована на следующих пользователей:

1 Людей, управляющих складом.

2 Людей, занимающихся предпринимательской деятельностью.

Порядок работы информационной системы:

- 1 Пользователи регистрируются самостоятельно через веб-интерфейс и проходит первоначальную настройку профиля с указанием личных данных (ФИО, телефон, почта). После авторизации они могут просматривать каталог товаров и добавлять интересующие позиции в свои заявки.

Изм.	Лист	№ докум.	Подпись	Дата
------	------	----------	---------	------

2 Сотрудники склада регистрируются исключительно администратором. Они получают доступ к информации о расположении товаров на складе, а также к списку активных заявок для их обработки.

3 Информационная система управление складом и запасами — это универсальная платформа, объединяющая «Клиентов» и «Сотрудников склада» для достижения быстрых результатов. Система предоставляет всем участникам процесса точные данные о товарах, принятия осознанных решений по закупкам. Администратор обеспечивает бесперебойную работу, корректность данных и безопасность системы.

Ключевые сущности предметной области:

- Пользователи (атрибуты: идентификатор, фамилия, имя, отчество, номер телефона, электронная почта, роль, пароль пользователя, идентификатор учетной записи Яндекс).
- Товары (атрибуты: идентификатор, название, идентификатор категории, идентификатор производителя, описание).
- Производители (атрибуты: идентификатор, название, адрес).
- Категории (атрибуты: идентификатор, название, описание).
- Склады (атрибуты: идентификатор, название, адрес).
- Остатки на складе (атрибуты: идентификатор, идентификатор товара, идентификатор склада, количество, последнее обновление).
- Перемещение (атрибуты: идентификатор, идентификатор склада откуда, идентификатор склада куда, идентификатор остатков, количество, статус, тип перемещения, дата создания, идентификатор сотрудника).
- Заявка (атрибуты: идентификатор, идентификатор пользователя, идентификатор остатка, количество, статус, заметки, дата создания, идентификатор склада).

Изм.	Лист	№ докум.	Подпись	Дата
------	------	----------	---------	------

2 Анализ инструментальных средств для разработки

Проектировать структуру ИС удобно через MySQL Workbench, Draw.io, а ее дизайн – через онлайн-сервис Figma. Для реализации ИС подойдут следующие инструменты: Bootstrap, Laravel и PHP со стандартной базой данных MySQL.

MySQL Workbench – инструмент для визуального проектирования баз данных, интегрирующий проектирование, моделирование, создание и эксплуатацию БД в единое бесшовное окружение для системы баз данных MySQL.

Draw.io – это бесплатное онлайн-приложение для создания диаграмм для рабочих процессов, BPM, организационных, сетевых диаграмм, блок-схемм (флоучарты), UML и принципиальных электросхем.

Figma – бесплатный онлайн-сервис для дизайнеров, веб-разработчиков и маркетологов. Он предназначен для создания прототипов сайтов или приложений, иллюстраций и векторной графики. В редакторе можно настроить совместную работу, вносить и обсуждать правки, причём как в браузере, так и через приложение на компьютере.

Laravel – бесплатный веб-фреймворк с открытым кодом, предназначенный для разработки с использованием архитектурной модели MVC. Laravel выпущен под лицензией MIT. Исходный код проекта размещается на GitHub. В результате опроса sitepoint.com в декабре 2013 года о самых популярных PHP-фреймворках Laravel занял место самого многообещающего проекта на 2014 год. Позволяет упростить аутентификацию, маршрутизацию, сессии, кэширование, архитектуру приложения, работу с базой данных.

Bootstrap – бесплатный CSS-фреймворк с открытым исходным кодом для быстрой вёрстки адаптивных интерфейсов сайтов и веб-приложений.

PHP – это скриптовый язык программирования. Имеет открытый исходный код. Изначально создавался для разработки веб-приложений, но в процессе обновлений стал языком общего назначения. Также PHP – интерпретируемый язык. Это означает, что код, который написан на PHP, не компилируется (преобразуется

					КП.09.02.07-5.25.221.09.П3	Лист 7
Изм.	Лист	№ докум.	Подпись	Дата		

в машинный), а выполняется строка за строкой с помощью программы-интерпретатора, которая читает каждую из них и выполняет то, что там указано.

ИС будет содержать в себе информацию – её необходимо хранить, изменять, структурировать и использовать. Это реализуется благодаря базе данных. Были рассмотрены следующие варианты реализации СУБД:

- 1 MySQL.
- 2 PostgreSQL.
- 3 SQLite.

MySQL – свободная реляционная система хранения и управления базами данных. Разработку и поддержку MySQL осуществляет корпорация Oracle, получившая права на торговую марку вместе с поглощённой Sun Microsystems, которая ранее приобрела шведскую компанию MySQL AB. Продукт распространяется как под GNU General Public License, так и под собственной коммерческой лицензией. Помимо этого, разработчики создают функциональность по заказу лицензионных пользователей. Именно благодаря такому заказу почти в самых ранних версиях появился механизм репликации.

PostgreSQL – свободная объектно-реляционная система хранения и управления базами данных. Существует в реализациях для множества UNIX-подобных платформ, включая AIX, различные BSD-системы, HP-UX, IRIX, Linux, macOS, Solaris/OpenSolaris, Tru64, QNX, а также для Microsoft Windows.

SQLite – компактная автономная, высоконадежная и полнофункциональная реляционная система управления базами данных, реализованная в виде библиотеки на языке С. В отличие от клиент-серверных СУБД не является отдельным процессом, к которому подключается приложение. Вместо этого она непосредственно встраивается в конечную программу и становится ее неотъемлемой частью.

Для наглядности сравнения вариантов реализации базы данных была составлена таблица 1.

Изм.	Лист	№ докум.	Подпись	Дата	КП.09.02.07-5.25.221.09.ПЗ	Лист 8

Таблица 1 – Сравнение средств реализации базы данных

Название СУБД	MySQL	PostgreSQL	SQLite
Популярность	+	-	+
Отказоустойчивость	+	+	-
Не требует удаленного сервера	-	-	+
Простота использования	+	-	+

Таким образом, в качестве СУБД для будущего продукта была выбрана MySQL, так как она предоставляет весь необходимый функционал для разработки продукта, и при этом она проста в изучении и овладении, достаточно быстрая, предоставляет возможность гибкой настройки, легко переносимая.

Для взаимосвязи баз данных и северной части продукта необходимо использовать серверный язык. Для реализации этого были рассмотрены два языка программирования – C#, Php и Python.

C# – объектно-ориентированный язык программирования общего назначения.

C# относится к семье языков с С-подобным синтаксисом, из них его синтаксис наиболее близок к C++ и Java. Язык имеет статическую типизацию, поддерживает полиморфизм, перегрузку операторов (в том числе операторов явного и неявного приведения типа), делегаты, атрибуты, события с поддержкой замыканий, LINQ, исключения, комментарии в формате XML.

PHP – скриптовый язык общего назначения, интенсивно применяемый для разработки веб-приложений. В настоящее время поддерживается подавляющим большинством хостинг-провайдеров и является одним из лидеров среди языков, применяющихся для создания динамических веб-сайтов. Действует, как и самостоятельно, так и с фреймворками.

Python – интерпретируемый язык программирования высокого уровня общего назначения, ориентированный на повышение производительности разработчика и читаемости кода. Однако сильно ориентирован на ООП.

Изм.	Лист	№ докум.	Подпись	Дата	КП.09.02.07-5.25.221.09.ПЗ	Лист 9
------	------	----------	---------	------	----------------------------	-----------

Для наглядности сравнения языков программирования была составлена таблица 2.

Таблица 2 – Сравнение языков программирования для разработки ПП

Название языка программирования	C#	PHP	Python
Наличие библиотек	+	+	+
Инструменты для работы с БД	+	+	+
Объектно-ориентированные возможности	-	+	+
Лёгкий понятный синтаксис	-	-	+

Таким образом, PHP будет более лучшим вариантом, так как он имеет большое количество библиотек и имеет инструменты для работы с БД.

Для разработки программного продукта рассмотрены следующие инструментальные средства разработки программных продуктов:

- 1 Visual Studio Code.
- 2 PHPStorm.
- 3 Atom.

Visual Studio Code – текстовый редактор, разработанный Microsoft для Windows, Linux и macOS. Позиционируется как «лёгкий» редактор кода для кроссплатформенной разработки веб- и облачных приложений. Включает в себя отладчик, инструменты для работы с Git, подсветку синтаксиса, IntelliSense и средства для рефакторинга. Имеет широкие возможности для кастомизации: пользовательские темы, сочетания клавиш и файлы конфигурации. Распространяется бесплатно, разрабатывается как программное обеспечение с открытым исходным кодом, но готовые сборки распространяются под проприетарной лицензией.

NetBeans – это свободная интегрированная среда разработки (IDE) под эгидой Apache, поддерживающая разработку на PHP, JavaScript, HTML/CSS и других языках. NetBeans обеспечивает подсветку синтаксиса и автодополнение,

Изм.	Лист	№ докум.	Подпись	Дата

встроенные средства отладки (включая поддержку Xdebug), интеграцию с системами контроля версий Git, удобную навигацию по проекту и поддержку Composer для PHP-проектов. IDE распространяется бесплатно и поддерживается сообществом Apache NetBeans. Atom – текстовый редактор с открытым исходным кодом, разработанный компанией GitHub. Atom позиционировался как гибкий и настраиваемый редактор кода. Он обладал встроенным менеджером пакетов, через который можно было устанавливать расширяющие возможности редактора плагины. Поскольку Atom был продуктом GitHub, его интеграция с системой контроля версий Git была на высшем уровне. Встроенный функционал позволял просматривать изменения, делать коммиты, пушить и управлять ветками прямо из редактора. Однако в июне 2022 года было объявлено об официальном прекращении поддержки Atom.

Сравнение IDE для разработки программного продукта наглядно представлено в таблице 3.

Таблица 3 – Сравнение IDE для разработки программного продукта

Название IDE	NetBeans	Visual Studio Code	Atom
Распространяется бесплатно	+	+	+
Автоматическое сохранение	+	+	+
Возможность расширения функционала библиотеками	+	+	+
Поддержка CSS/HTML/JS	+	+	+

Таким образом, после рассмотрения вариантов средств разработок, было принято решение использовать Visual Studio Code.

Для упрощения и ускорения написания кода, а также обеспечения большей безопасности системы, необходимо использование фреймворков. Для дальнейшей разработки были рассмотрены следующие варианты:

- 1 Laravel.
- 2 Yii.
- 3 CodeIgniter.

Laravel – является одним из популярных фреймворков для PHP. Одна из главных причин популярности заключается в простоте и читаемости кода. Использование Laravel, облегчает решение множества рутинных задач из-за использования собственного движка шаблонов blade, что ведёт к тому, что разработчикам приходится писать, в соответствующих ситуациях, меньше кода. Среди таких задач – маршрутизация, кеширование, авторизация, аутентификация. Благодаря использованию этого фреймворка упрощаются такие задачи, как проверка электронных адресов пользователей, хеширование и сброс паролей. Laravel решает проблему управления структурой базы данных. Структура базы данных описывается на языке PHP, что позволяет хранить ее историю изменений в системе контроля версий.

Yii – это простой, но высокопроизводительный универсальный фреймворк, основанный на компонентной структуре. Он известен благодаря высокой производительности благодаря эффективному использованию механизма lazy loading, но, в первую очередь, он знаменит своей простотой. Yii имеет продуманную архитектуру, которая позволяет легко настраивать и расширять практически любую часть ядра. Также Yii строго следует принципам объектно-ориентированного программирования и стандартам кодирования PSR. Фреймворк имеет в себе генератор кода Gii, который позволяет быстро создавать базовые структуры, на основе которых можно строить собственные решения. Его создатели очень серьёзно подошли к вопросам безопасности. В частности, в него встроены механизмы хеширования паролей, основанные на bcrypt, и средства шифрования.

CodeIgniter – это мощный, легковесный и высокопроизводительный PHP-фреймворк с открытым исходным кодом, созданный для разработчиков, которые нуждаются в простом наборе инструментов для создания полнофункциональных веб-приложений. Философия фреймворка заключается в минимальной

Иzm.	Лист	№ докум.	Подпись	Дата	KP.09.02.07-5.25.221.09.ПЗ	Лист	12

конфигурации, максимальной производительности и простоте освоения. Этот фреймворк не следует строго всем ограничениям архитектуры MVC, в плане архитектуры создаваемых на его основе проектов, он весьма гибок, и не ограничивает разработчиков лишь этой архитектурой. В нём имеется надёжная система защиты контента, он обладает стандартными механизмами для защиты от CSRF- и XSS-атак. Разработчику, присматривающемуся к CodeIgniter, стоит знать о том, что это одна из тех PHP-платформ, которые легче всего освоить. Его, кроме того, очень легко устанавливать. Эти факторы делают его идеальным выбором для новичков.

Сравнение фреймворков для разработки программного продукта наглядно представлено в таблице 4.

Таблица 4 – Сравнение фреймворков для разработки программного продукта

Название фреймворка	Laravel	Yii	CodeIgniter
ORM	ActiveRecord	Doctrine 2, Propel (active record)	ELOQUENT ORM (active record)
Библиотека тестирования	PHPUnit	PHPUnit	PHPUnit
Веб2.0	jQuery HTML5boilerplate		
Система шаблонов	PHP, Simple template parser "{ var_name }"	PHP, Twig	Blade, PHP, Custom

Таким образом было решено использовать фреймворк Laravel.

Для проверки и тестирования ИС, а также обеспечения большей безопасности системы, необходимо использование средств для тестирования. Для дальнейшей разработки были рассмотрены следующие варианты:

- 1 PHPUnit.
- 2 PHPSpec.
- 3 Pest.

PHPUnit – самый популярный фреймворк для модульного тестирования. Он предоставляет большой набор возможностей для написания изолированных тестов,

					КП.09.02.07-5.25.221.09.П3	Лист 13
Изм.	Лист	№ докум.	Подпись	Дата		

проверяющих корректность работы отдельных компонентов системы. Помимо модульных тестов, с его помощью можно создавать интеграционные и функциональные тесты.

PHPSpec – фреймворк, который следует методологии BDD (разработка через поведение). PHPSpec концентрируется на спецификации и описании поведения объектов. Он поощряет разработку «снизу-вверх», заставляя сначала описать желаемое поведение системы на почти естественном языке, а затем реализовать под него код. Это делает его отличным инструментом для проектирования архитектуры приложения на ранних этапах.

Pest – простой и легковесный фреймворк, позиционирующейся как современная альтернатива PHPUnit с целью сделать процесс написания тестов более простым и приятным. Фреймворк построен поверх PHPUnit, что обеспечивает полную совместимость с его возможностями и аннотациями, но предлагает более чистый и лаконичный синтаксис, вдохновленный JavaScript-фреймворком Jest.

Таким образом для тестирования был выбран фреймворк PHPUnit.

Для создания программного продукта было решено использовать средства:

- 1 Инструментарий MySQL Workbench для проектирования базы данных.
- 2 Сервис Draw.io для функционального и структурного проектирования.
- 3 Сервис Figma для создания прототипов интерфейса.
- 4 Среда разработки MS Visual Studio Code для непосредственной разработки.
- 5 Язык PHP для разработки клиентской ИС.
- 6 СУБД MySQL для работы с данными на серверной части.
- 7 Фреймворк Laravel для разработки серверной части.
- 8 Фреймворк Bootstrap для разработки клиентской части.
- 9 Для тестирования ИС можно использовать встроенные на Laravel unit-тесты PHPUnit.

Изм.	Лист	№ докум.	Подпись	Дата

3 Техническое задание

В начале разработки создавалось техническое задание, в котором указывались основные требования.

Для создания технического задания использовался ГОСТ 34.602 -2020.

Согласно ГОСТ 34.602-2020 техническое задание должно включать следующие разделы:

1. Введение.
- 1.1. Общие сведения.
- 1.2. Цели.
2. Основание для разработки.
 - 2.1. Нормативные документы.
 - 2.2. Проектные документы.
3. Назначение системы.
 - 3.1. Общее описание.
4. Требование к системе.
 - 4.1. Функциональные требования.
5. Технические требования.
 - 5.1. Производительность.
 - 5.2. Надежность.
 - 5.3. Эксплуатационные требования.
 - 5.4. Требования к информационной безопасности.
6. Требование к техническому обеспечению.
 - 6.1. Сервер.
 - 6.2. Хранилище данных.
 - 6.3. Клиентские устройства.
 - 6.3.1. Минимальные требования персонального компьютера (ноутбука).
 - 6.4. Сетевые требования.
7. Требование к программному обеспечению.

Изм.	Лист	№ докум.	Подпись	Дата

- 7.1. Сервер.
- 7.2. Персональный компьютер (ноутбук).
8. Требования к тестированию.
 - 8.1. Общие требования.
 - 8.1.1. Документирование.
 - 8.1.2. Функциональное тестирование.
 9. Организационно-технические требования.
 - 9.1. Этапы разработки.

Техническое задание на разработку программного продукта представлено в Приложении А.

Иzm.	Лист	№ докум.	Подпись	Дата	KP.09.02.07-5.25.221.09.П3	Лист 16
------	------	----------	---------	------	----------------------------	------------

4 Проектирование ИС

4.1 Архитектура ИС

Архитектура программного обеспечения (далее – ПО) информационной системы «Система управления складом и запасами» представляет собой совокупность компонентов и принципов их взаимодействия, обеспечивающих надежность, масштабируемость и безопасность работы приложения.

Для реализации ИС была выбрана клиент-серверная архитектура (рисунок 1), где клиентская часть отвечает за интерфейс взаимодействия с пользователем, а серверная часть – за обработку бизнес-логики и работу с базой данных.

Клиентская часть необходима для отображения пользовательского интерфейса, отправления запросов к системе на сервер и отображения этих данных у пользователя.

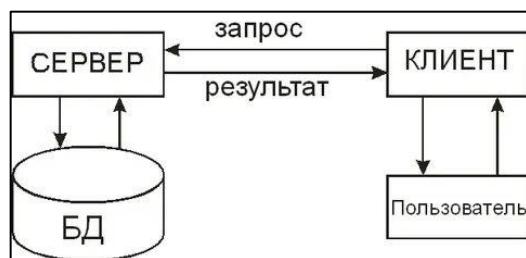


Рисунок 1 – Клиент-серверная архитектура

Основные функции клиента:

- Формирование и отправка запросов к серверу для получения данных о товарах, их наличии, категориях, а также актуальных заявках.
- Обработка ответов сервера и отображение информации пользователю в виде каталога товаров, списка заявок, схемы размещения товаров на складе и отчётов.
- Отправка команд для выполнения операций, таких как добавление товаров в заявку, изменение количества, оформление и отправка заявок, а также управление профилем пользователя.

Изм.	Лист	№ докум.	Подпись	Дата
------	------	----------	---------	------

Серверная часть представляет собой специализированное аппаратное и программное обеспечение, предназначенное для обработки запросов от клиентов, управления системными ресурсами и хранения данных о товарах, заявках и пользователях. Сервер выступает в роли посредника между клиентом и базой данных, обрабатывая запросы и предоставляя клиенту только релевантную информацию.

Основные функции серверной части включают:

- Хранение данных о товарах, их расположении на складе, пользовательских профилях, заявках и истории операций, а также управление доступом к этим данным.
- Обработка входящих запросов от клиентов, включая поиск товаров, формирование и обновление заявок, контроль остатков и генерацию отчётов.
- Формирование и отправка ответов клиенту, включая актуальную информацию о наличии товаров, статусах заявок и уведомлениях.

В архитектуре «клиент-сервер» клиент запрашивает только необходимые данные (например, информацию о конкретном товаре или статусе заявки), а сервер обрабатывает эти запросы и передаёт информацию, снижая нагрузку на клиентскую часть. Такой подход обеспечивает эффективное распределение ресурсов и повышает общую производительность приложения, что особенно важно для оперативного управления складскими процессами и обработки больших объёмов данных.

4.2 Структурные схемы ИС

На этапе проектирования ИС использовался объектно-ориентированный подход с построением структурных схем.

Диаграмма прецедентов (Use Case Diagram) – диаграмма, отражающая отношения между актёрами и прецедентами и являющаяся составной частью модели прецедентов, позволяющей описать систему на концептуальном уровне.

					КП.09.02.07-5.25.221.09.П3	Лист
Изм.	Лист	№ докум.	Подпись	Дата		18

Функциональное проектирование основывается на выделении ключевых сценариев взаимодействия пользователей с системой. Диаграмма прецедентов представлена на рисунке 2. На ней выделено 3 роли актеров и их действия.

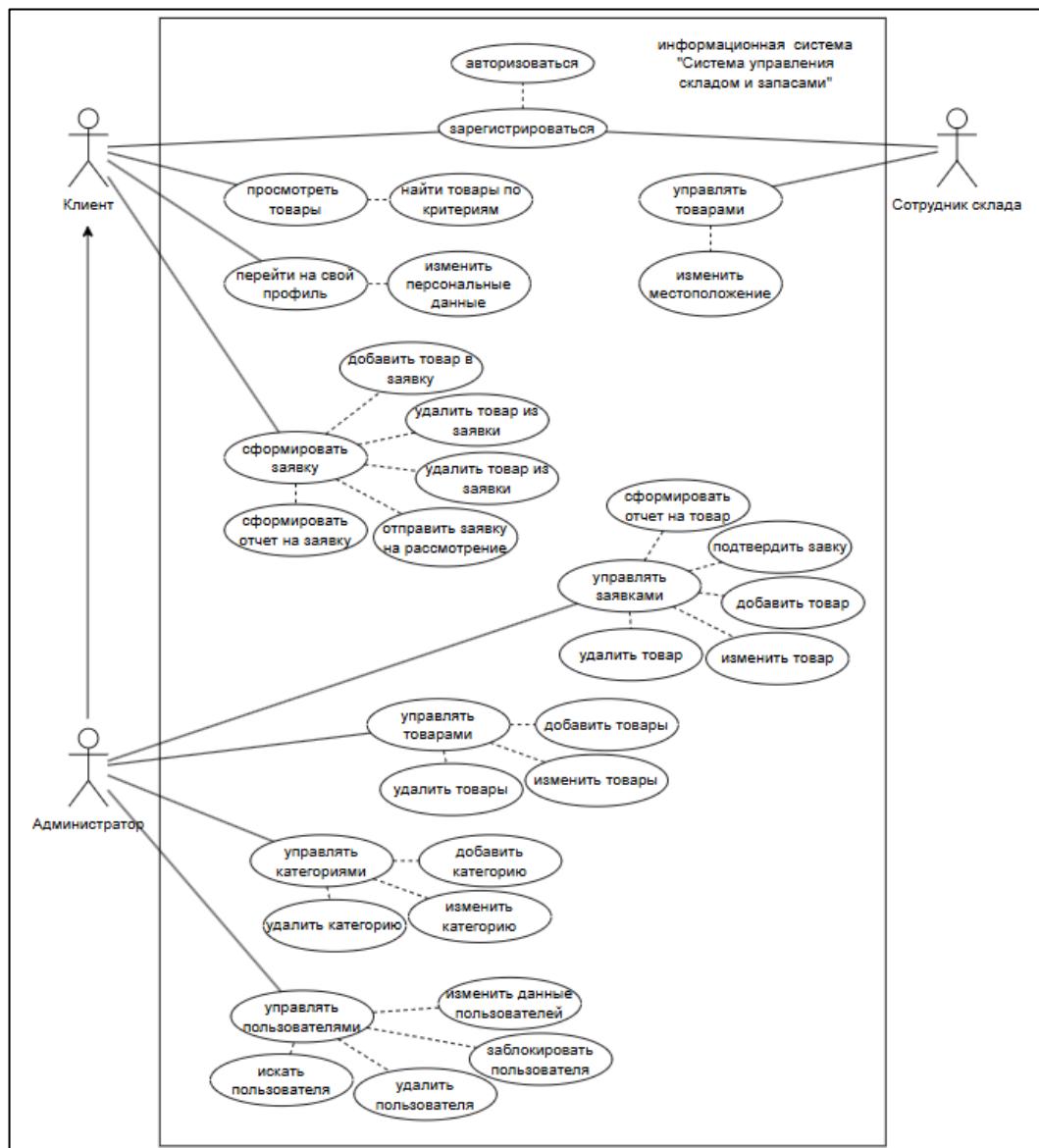


Рисунок 2 – Диаграмма прецедентов

Далее, разрабатывалась диаграмма деятельности. Она разработана в соответствии с методологией UML и отражает последовательность действий участников системы в процессе работы с складскими остатками и заявками. Визуальное разделение на дорожки позволяет четко идентифицировать зоны ответственности каждого участника системы и проследить взаимодействие между

Изм.	Лист	№ докум.	Подпись	Дата	КП.09.02.07-5.25.221.09.ПЗ	Лист 19

различными ролями в процессе работы информационной системы. На рисунке 3 представлена диаграмма деятельности программного продукта

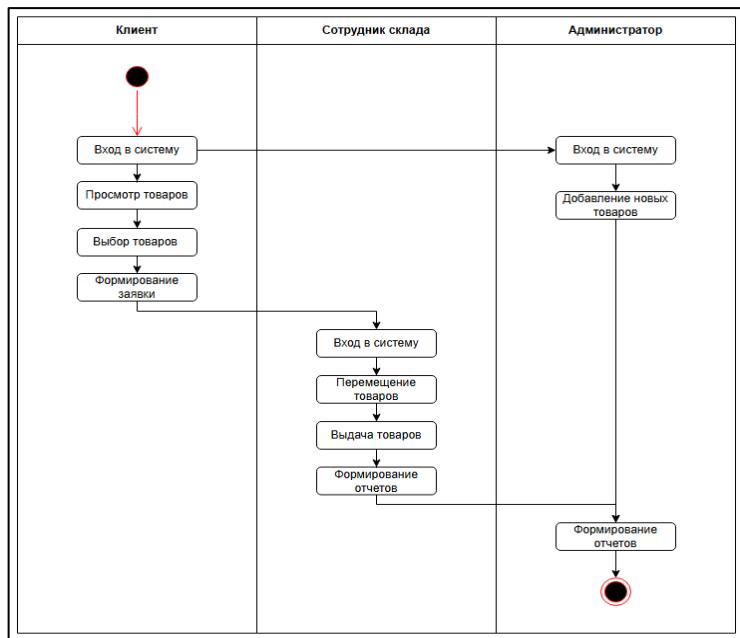


Рисунок 3 – Диаграмма деятельности

Диаграмма взаимодействия системы «Управление складом и остатками» отображает последовательность обмена сообщениями между участниками процесса обработки заявки. Диаграмма выполнена в нотации UML (рисунок 4) и показывает взаимодействие между клиентом, сотрудником склада и приложением в рамках процесса обработки складской заявки.

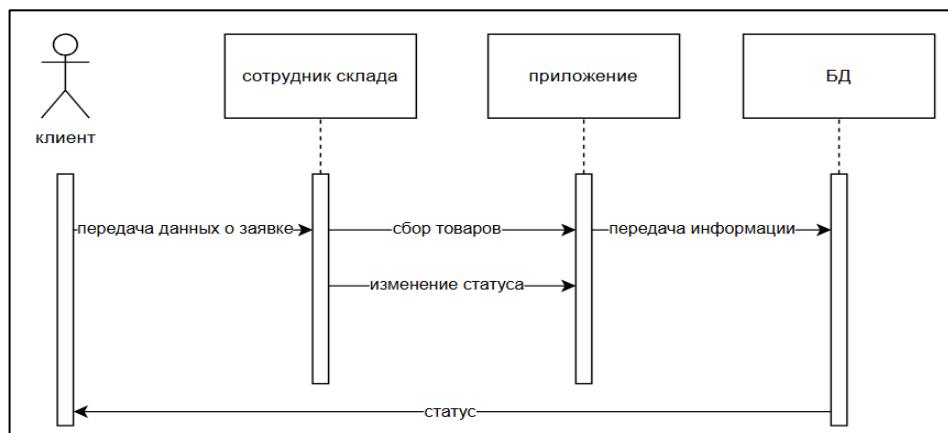


Рисунок 4 – Диаграмма взаимодействия

Изм.	Лист	№ докум.	Подпись	Дата

Для описания внутреннего устройства программного продукта, взаимосвязей между его элементами и распределения функций между ними была построена диаграмма классов. Диаграмма классов, представленная на рисунке 5, разработана в соответствии с нотацией UML. На диаграмме представлены основные классы системы, их атрибуты, методы и взаимосвязи между ними. Модель демонстрирует, какие объекты существуют в системе, какие данные они содержат и как взаимодействуют между собой.

Таким образом, диаграмма классов описывает внутреннюю структуру приложения, показывая, каким образом сущности предметной области преобразованы в программные объекты. Она является концептуальной моделью системы, на основе которой реализуются объекты, контроллеры и сервисы в коде приложения.

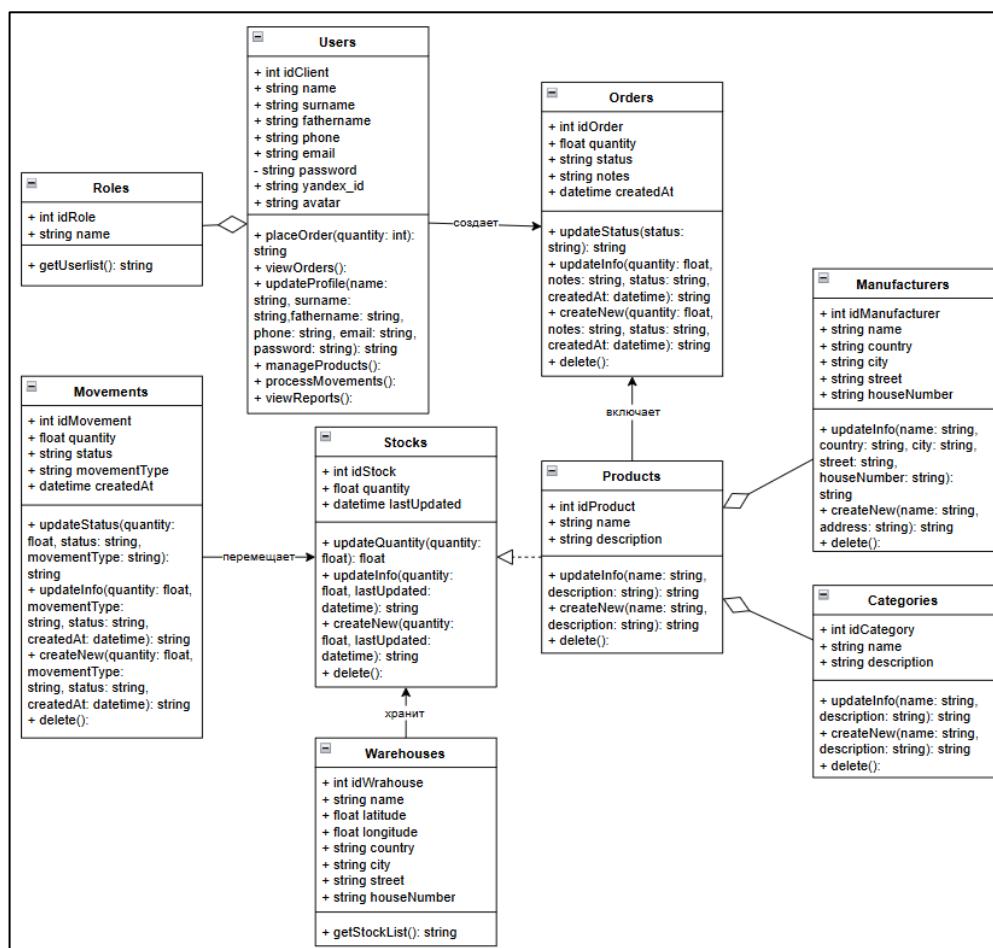


Рисунок 5 – Диаграмма классов

Изм.	Лист	№ докум.	Подпись	Дата

4.3 Функциональные схемы ИС

Функциональные схемы включает контекстную диаграмму, которая представляет информационную систему «Управление складом и остатками» как единый модуль, взаимодействующий с внешними сущностями. Диаграмма выполнена в нотации IDEF0. Контекстная диаграмма отображает систему на верхнем уровне декомпозиции (рисунок 6).

Система взаимодействует с внешними сущностями: клиентами, которые подают заявки и получают уведомления; сотрудниками склада, которые выполняют операции и используют отчеты; поставщиками, которые предоставляют данные о товарах; бухгалтерией, получающей отчетные документы; руководством, использующим аналитические отчеты для принятия решений. Основная цель системы – автоматизация процессов учета товарных остатков, обработки заявок и формирования отчетности для повышения эффективности управления складскими запасами.

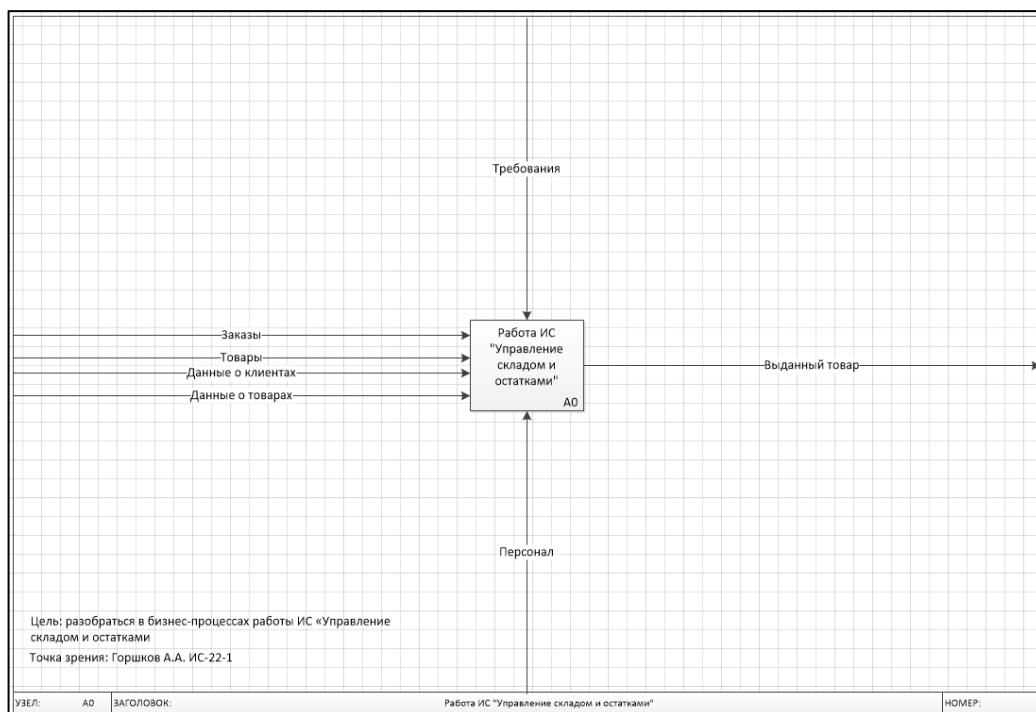


Рисунок 6 – Контекстная диаграмма процессов

Изм.	Лист	№ докум.	Подпись	Дата
------	------	----------	---------	------

КП.09.02.07-5.25.221.09.П3

Лист
22

Далее построена диаграмма декомпозиции, изображенная на рисунке 7, системы «Управление складом и остатками». Она детализирует основные функциональные блоки, выявленные на контекстной диаграмме, и представляет собой декомпозицию основного блока А0 на пять подпроцессов, каждый из которых выполняет специфические функции в рамках общей системы управления складом.

Диаграмма демонстрирует полный цикл обработки товаров от поступления данных до физического перемещения и координации персонала, обеспечивая целостное представление о функционировании складской информационной системы на уровне детализированных бизнес-процессов.

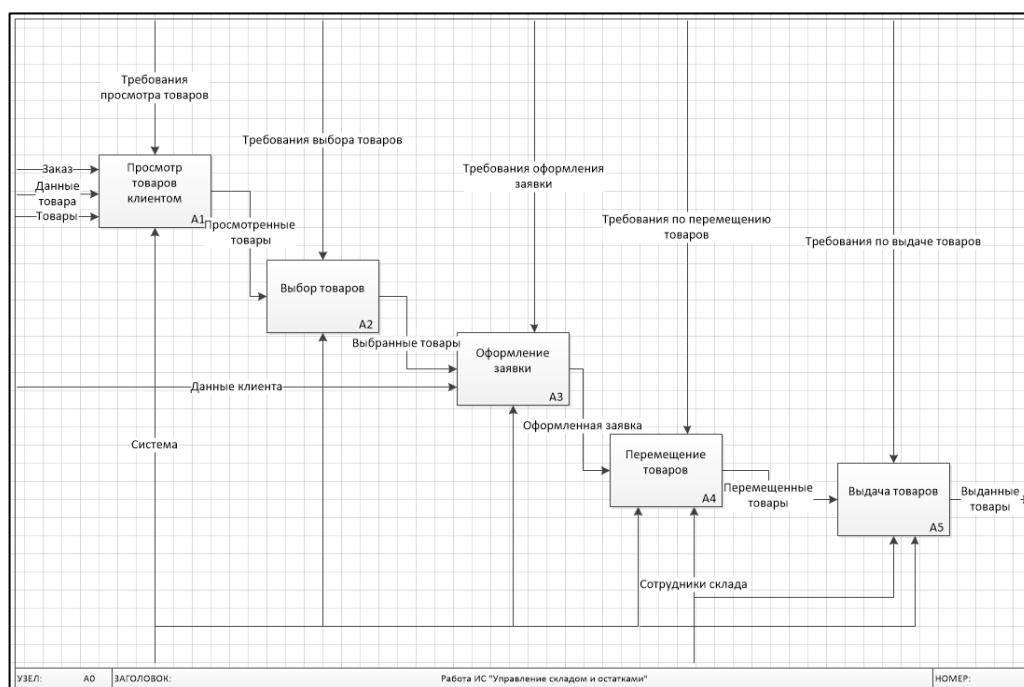


Рисунок 7 – Диаграмма декомпозиции процесса

Для более детального понимания логики функционирования системы и анализа процессов обработки информации была разработана диаграмма потоков данных (DFD). Диаграмма потоков данных DFD, находящаяся на рисунке 8, графически представляет потоки данных в разрабатываемой информационной системе. С её помощью описываются входные и выходные потоки данных, и

Изм.	Лист	№ докум.	Подпись	Дата

хранилища этих данных. DFD позволяет определить, какие данные обрабатываются, какие процессы их преобразуют, и каким образом информация перемещается между пользователями, системой и базой данных.

DFD-диаграмма отображает процесс создания заявки, его логике.

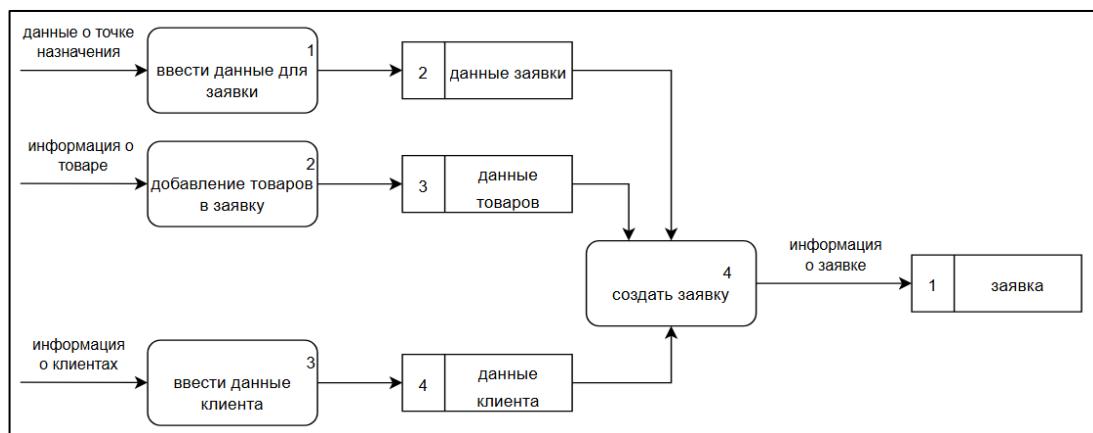


Рисунок 8 – Диаграмма потоков данных

4.4 Проектирование базы данных

В начале проектирования базы данных создается инфологическая модель на рисунке 9 отражает полную логическую структуру базы данных и взаимосвязи между всеми объектами предметной области. Её разработка позволяет обеспечить надёжное хранение и обработку информации, снизить избыточность данных и реализовать все ключевые функции системы: учёт товаров, обработку заявок, отслеживание перемещений и ведение отчётности.

На основании инфологической модели будет построена физическая модель базы данных в MySQL, включающая таблицы, индексы, ограничения целостности и связи между ними.

						КП.09.02.07-5.25.221.09.ПЗ	Лист 24
Изм.	Лист	№ докум.	Подпись	Дата			

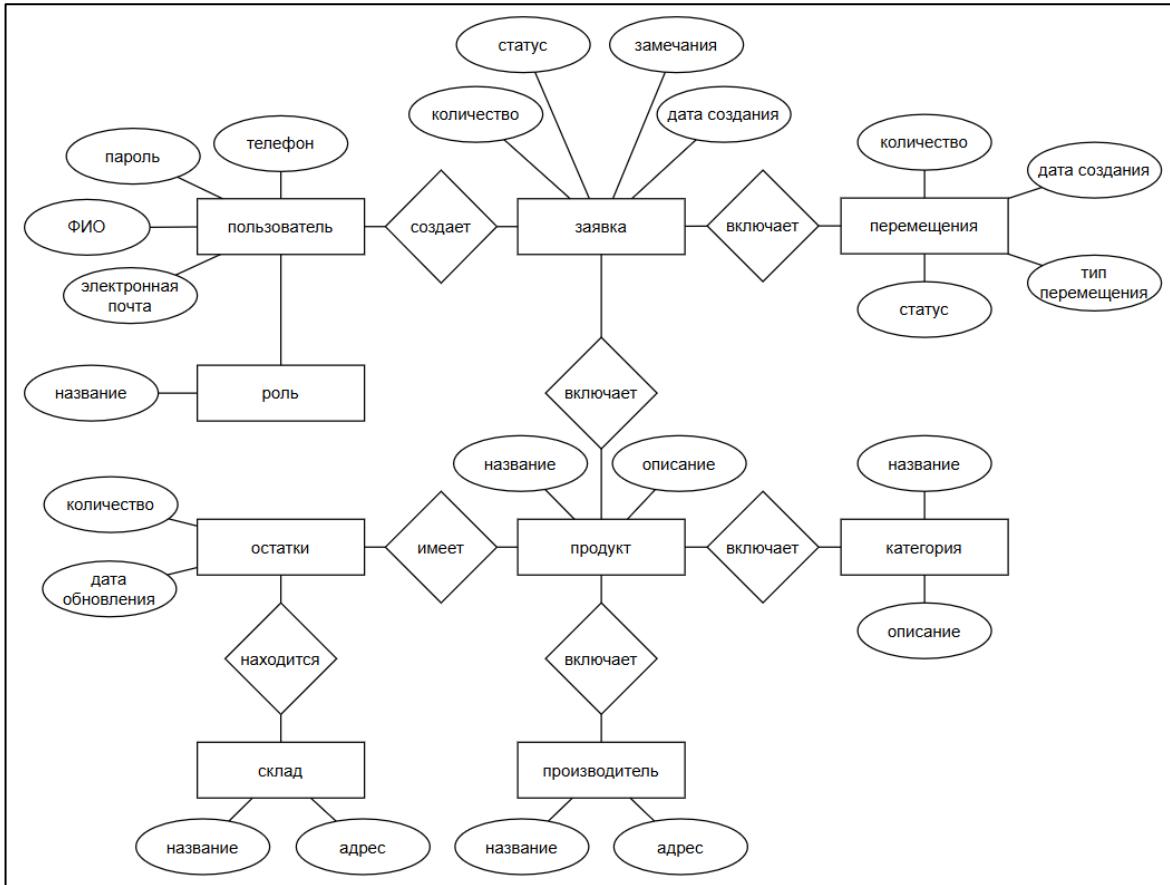


Рисунок 9 – Инфологическая модель базы данных

Для обеспечения целостности, структурированности и удобства хранения данных в информационной системе была спроектирована ER-модель базы данных.

ER-модель базы данных, изображенная на рисунке 10, разработана для описания логической структуры информационной системы «Управление складом и запасами» и отражает взаимосвязи между основными сущностями, входящими в предметную область. Модель служит основой для построения реляционной базы данных, обеспечивающей хранение и обработку данных о пользователях, товарах, складах, остатках, заявках и операциях перемещения.

Изм.	Лист	№ докум.	Подпись	Дата
------	------	----------	---------	------

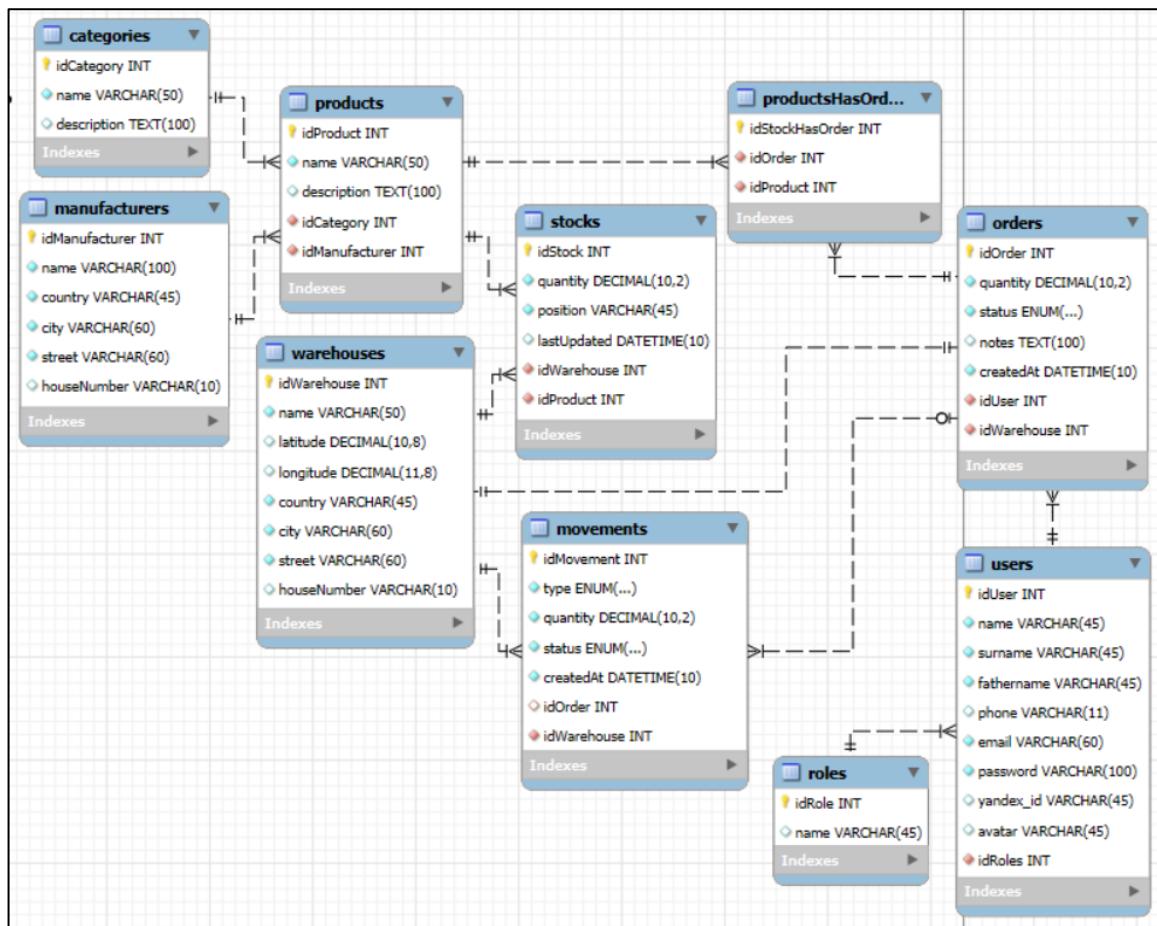


Рисунок 10 – ER-модель базы данных

База данных имеет 9 таблиц

Таблица 5 – Таблица «Users»

Атрибут	Тип данных	Описание
idUser	INT	Идентификатор пользователя (PK)
name	VARCHAR(45)	Имя пользователя
surname	VARCHAR(45)	Фамилия пользователя
fathername	VARCHAR(45)	Отчество пользователя
phone	VARCHAR(11)	Номер телефона пользователя
email	VARCHAR(60)	Адрес электронной почты пользователя
password	VARCHAR(100)	Пароль пользователя в зашифрованном виде
yandex_id	VARCHAR(255)	Идентификатор учетной записи Яндекс
avatar	VARCHAR(255)	Изображение пользователя
idRole	INT	Внешний ключ на таблицу «Roles»

Таблица 6 – Таблица «Roles»

Атрибут	Тип данных	Описание
idRole	INT	Идентификатор роли (PK)
name	VARCHAR(45)	Название роли пользователя в системе

Таблица 7 – Таблица «Categories»

Атрибут	Тип данных	Описание
idCategory	INT	Идентификатор категории (PK)
name	VARCHAR(50)	Название категории товаров
description	TEXT(100)	Описание категории

Таблица 8 – Таблица «Manufacturers»

Атрибут	Тип данных	Описание
idManufacturer	INT	Идентификатор производителя (PK)
name	VARCHAR(100)	Название производителя
country	VARCHAR(45)	Страна адреса производителя
city	VARCHAR(60)	Город адреса производителя
street	VARCHAR(60)	Улица адреса производителя
houseNumber	VARCHAR(10)	Номер дома производителя

Таблица 9 – Таблица «Products»

Атрибут	Тип данных	Описание
idProduct	INT	Идентификатор товара (PK)
name	VARCHAR(50)	Наименование товара
description	TEXT(100)	Подробное описание товара
idCategory	INT	Внешний ключ на таблицу «Categories»
idManufacturer	INT	Внешний ключ на таблицу «Manufacturers»

Таблица 10 – Таблица «Warehouses»

Атрибут	Тип данных	Описание
idWarehouse	INT	Идентификатор склада (PK)
name	VARCHAR(50)	Наименование склада
latitude	DECIMAL(10,8)	Ширина расположения склада
longitude	DECIMAL(11,8)	Долгота расположения склада
country	VARCHAR(45)	Страна адреса склада
city	VARCHAR(60)	Город адреса склада
street	VARCHAR(60)	Улица адреса склада
houseNumber	VARCHAR(10)	Номер дома склада

Таблица 11 – Таблица «Stocks»

Атрибут	Тип данных	Описание
idStock	INT	Идентификатор записи об остатках (PK)
idProduct	INT	Внешний ключ на таблицу «Products»
idWarehouse	INT	Внешний ключ на таблицу «Warehouses»
quantity	DECIMAL(10,2)	Количество товара на складе
position	VARCHAR(45)	Позиция товара на складе
lastUpdate	DATETIME	Дата и время последнего обновления информации

Таблица 12 – Таблица «Orders»

Атрибут	Тип данных	Описание
idOrder	INT	Идентификатор заявки (PK)
idUser	INT	Внешний ключ на таблицу «Пользователи»
idWarehouse	INT	Склад, с которого выполняется заявка
notes	TEXT	Заметки для заявки
quantity	DECIMAL(10,2)	Количество заказанного товара
status	ENUM('pending', 'confirmed', 'rejected', 'completed')	Статус выполнения заявки (ожидает, подтверждено, отклонено, завершено)
createdAt	DATETIME	Дата и время создания заявки

Таблица 13 – Таблица «ProductsHasOrders»

Атрибут	Тип данных	Описание
idProductHasOrder	INT	Идентификатор таблицы ProductsHasOrders (PK)
idProduct	INT	Внешний ключ на таблицу «Stocks»
idOrder	INT	Внешний ключ таблицу «Orders»

Таблица 14 – Таблица «Movements»

Атрибут	Тип данных	Описание
idMovement	INT	Идентификатор перемещения (PK)
idWarehouse	INT	Склад, куда перемещается товар
quantity	DECIMAL(10,2)	Количество перемещённого товара
status	ENUM('in Progress', 'complete')	Статус перемещения («в процессе», «выполнен»)
type	ENUM('between warehouses', 'for order')	Тип перемещения («между складами», «для заявки»)
createdAt	DATETIME	Дата и время операции

4.5 Проектирование интерфейса ИС

Пользовательский интерфейс обеспечивает упрощенное взаимодействие пользователя с программным продуктом. Интерфейс позволяет путем использования различных элементов управления сделать взаимодействие продуктом максимально удобным и понятным для конченого пользователя.

Прототип пользовательского интерфейса для информационной системы разрабатывался в программе Figma. На рисунке 11 изображена главная страница, с которой сталкивается не авторизированный пользователь при переходе по ссылке. На главной странице представлена основная информация о системе, её основные функции и расположение главного склада. Пользователь, нажав на кнопку «Войти», может авторизоваться в приложении.

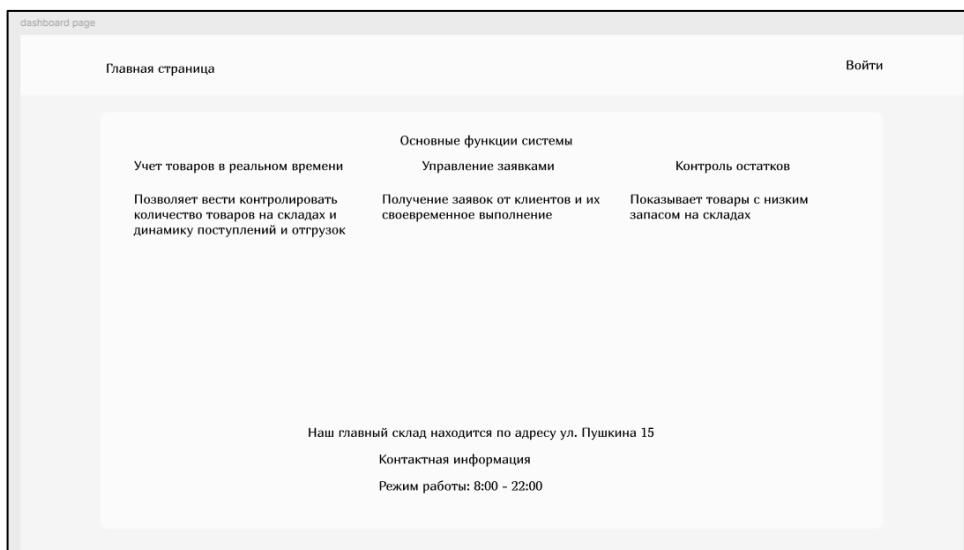


Рисунок 11 – Окно главной страницы

На рисунке 12 представлен интерфейс окон пользователей для авторизации в приложении. Любой пользователь имеет возможность создать новую учётную запись.

Изм.	Лист	№ докум.	Подпись	Дата	КП.09.02.07-5.25.221.09.ПЗ	Лист 29
------	------	----------	---------	------	----------------------------	------------

The screenshot shows a light gray login form titled "login page". It contains two input fields: "Эл. почта" (Email) and "Пароль" (Password). Below these is a checkbox labeled "Запомнить меня" (Remember me). At the bottom left is a link "Войти при помощи Telegram" (Log in via Telegram), and at the bottom right is a dark gray button labeled "Войти" (Log in).

Рисунок 12 – Окно авторизации

На рисунке 13 изображено окно для регистрации пользователя в системе. Для этого потребуется его основная информация, а именно: фамилия, имя и отчество, электронная почта, номер телефона и пароль, который потребуется ввести еще раз для подтверждения.

The screenshot shows a light gray registration form titled "signup page". It contains seven input fields: "Имя" (Name), "Фамилия" (Surname), "Отчество" (Middle name), "Эл. почта" (Email), "Телефон" (Phone), "Пароль" (Password), and "Подтвердите пароль" (Confirm password). At the bottom left is a link "Уже зарегистрированы?" (Already registered?), and at the bottom right is a dark gray button labeled "Зарегистрироваться" (Register).

Рисунок 13 – Окно регистрации

Изм.	Лист	№ докум.	Подпись	Дата	КП.09.02.07-5.25.221.09.П3	Лист 30
------	------	----------	---------	------	----------------------------	------------

На рисунке 14 представлено окно товаров для администратора. Все последующие окна представлены для роли «Администратор». На странице присутствует шапка сайта, на которой находится навигация по приложению и возможность перейти в личный кабинет пользователя, ниже представлен основной блок с информацией о имеющихся товарах, отсортированные по категориям. Имеется возможность добавления товаров и изменения товаров, просмотра категорий и производителей

The screenshot shows a web-based administration interface for managing products. At the top, there is a navigation bar with links: Главная страница, Товары (selected), Остатки, Заявки, Перемещения, Админ-панель, and Имя пользователя. Below the navigation bar, there are three tabs: Добавить товар, Категории, and Производители, with 'Добавить товар' being the active tab. A search bar labeled 'Поиск' is also present. The main content area is divided into two sections: 'Категория: Принтеры и МФУ' and 'Категория: Бумажная продукция'. Each section contains three product cards, each with fields for Название (Name), Производитель (Manufacturer), Наличие (Stock), and Описание (Description). Each card also has an 'Изменить' (Edit) button.

Рисунок 14 – Окно товаров

На рисунке 15 изображено окно добавления товара, для этого потребуется заполнить требуемые данные о товаре: категория, название, производителя, количество товара, склад, на котором будет находиться товар и описание. Остальные окна добавления остатков, заказов, перемещений выполнены в таком же стиле

Изм.	Лист	№ докум.	Подпись	Дата	КП.09.02.07-5.25.221.09.ПЗ	Лист 31
------	------	----------	---------	------	----------------------------	------------

add product admin page

Главная страница	Товары	Остатки	Заявки	Перемещения	Админ-панель	Имя пользователя										
Добавить категорию Добавить производителя																
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 15%;">Категория:</td> <td>Принтеры и МФУ</td> </tr> <tr> <td>Название:</td> <td>МФУ HP LaserJet Pro m428fdw</td> </tr> <tr> <td>Производитель:</td> <td>HP Inc</td> </tr> <tr> <td>Наличие:</td> <td>7 шт</td> </tr> <tr> <td>Описание:</td> <td>Лазерное МФУ с Wi-Fi, автоматической двусторонней печатью</td> </tr> </table>							Категория:	Принтеры и МФУ	Название:	МФУ HP LaserJet Pro m428fdw	Производитель:	HP Inc	Наличие:	7 шт	Описание:	Лазерное МФУ с Wi-Fi, автоматической двусторонней печатью
Категория:	Принтеры и МФУ															
Название:	МФУ HP LaserJet Pro m428fdw															
Производитель:	HP Inc															
Наличие:	7 шт															
Описание:	Лазерное МФУ с Wi-Fi, автоматической двусторонней печатью															
Добавить товар																

Рисунок 15 – Окно добавления товара

На рисунок 16 представлено окно остатков, на котором отображается информация по остаткам товаров, отсортированных по складам.

stocks admin page

Главная страница	Товары	Остатки	Заявки	Админ-панель	Имя пользователя
Склад № 1					
id	Категория:	Название:	Производитель:	Позиция	Количество
1	Принтеры и МФУ	МФУ HP LaserJet Pro m428fdw	HP Inc	A2	7 шт
2	текст	текст	текст	текст	текст
3	текст	текст	текст	текст	текст
4	текст	текст	текст	текст	текст
Склад № 2					
id	Категория:	Название:	Производитель:	Позиция	Количество
1	Принтеры и МФУ	МФУ HP LaserJet Pro m428fdw	HP Inc	A2	7 шт
2	текст	текст	текст	текст	текст

Рисунок 16 – Окно остатков

На рисунке 17 изображены заявки, созданные пользователями. Администратор имеет возможность создавать и изменять уже созданные заявки, а также при необходимости их удалять.

Изм.	Лист	№ докум.	Подпись	Дата	КП.09.02.07-5.25.221.09.П3 Лист 32
------	------	----------	---------	------	--

orders admin page						
Главная страница	Товары	Остатки	Заявки	Перемещения	Админ-панель	Имя пользователя
Добавить заявку						Заявка № 1
id	Категория:	Название:	Производитель:	Склад	Количество	
1	Принтеры и МФУ	МФУ HP LaserJet Pro m428fdw	HP Inc	№3	2 шт	
2	текст	текст	текст	текст	текст	
3	текст	текст	текст	текст	текст	
4	текст	текст	текст	текст	текст	
						Изменить

Рисунок 17 – Окно заявок

У пользователя с ролью «Администратор» имеется доступ к админ-панели, представленной на рисунке 18, на которой отображаются все пользователи, отсортированные по ролям. Пользователь имеет право добавлять пользователей и изменять данные пользователей.

admin-panel page						
Главная страница	Товары	Остатки	Заявки	Перемещения	Админ-панель	Имя пользователя
Добавить пользователя						Администраторы
id	ФИО	Телефон	Эл. почта	Роль	Статус	
1	Васильев А.А.	89021125051	vasilki@gmail.com	Администратор	Активен	Изменить
2	текст	текст	текст	текст	текст	Изменить
3	текст	текст	текст	текст	текст	Изменить
4	текст	текст	текст	текст	текст	Изменить

Рисунок 18 – Окно админ-панели

При проектировании интерфейса были разработаны окна для авторизации, регистрации, главной страницы, товаров, добавления товаров, остатков, заявок и админ-панели.

5 Разработка ИС

5.1 Разработка кода клиентской части ИС

Для взаимодействия пользователя с продуктом необходимо реализовать пользовательский интерфейс, соответствующий всем указанным требованиям.

Далее на рисунках 19, 20 и 21 будут представлены вид страницы для отображения местоположения склада на карте и часть кода этой страницы, отвечающий за вывод информации пользователю.

Место для вывода информации о расположении склада на карте решено было разместить в отдельной вкладке со всей информацией о складе.

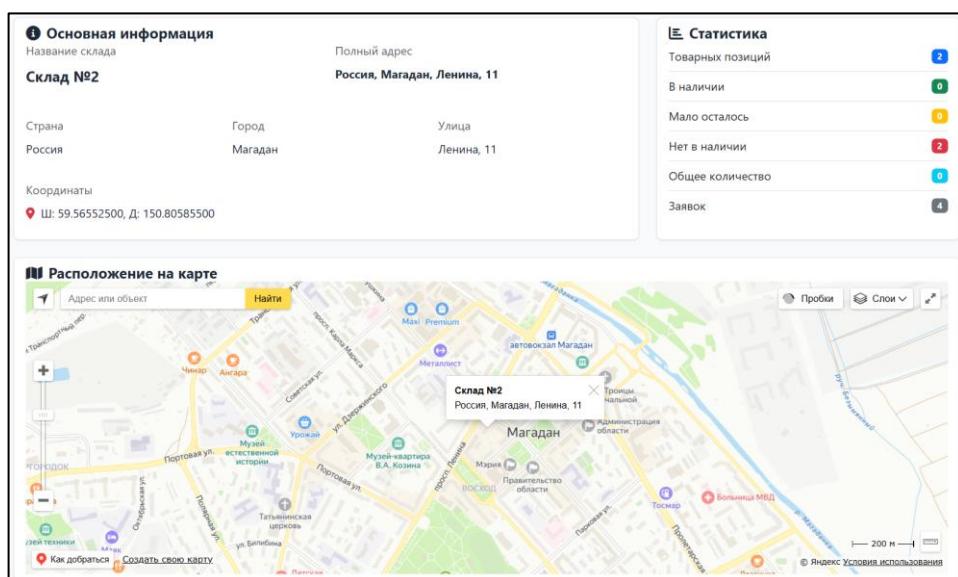


Рисунок 19 – Вид страницы отображения местоположения склада

Допускается что у склада может не быть координат для отображения склада на карте. Поэтому при инициализации блока с интеграцией с картами проверяются координаты у склада. При отсутствии координат блок с картами не будет отображаться у пользователя.

Изм.	Лист	№ докум.	Подпись	Дата

```

<!-- Карта -->
@if($warehouse->hasCoordinates())


<div class="col-12">
        <div class="card-custom px-3 py-2">
            <div class="card-header bg-transparent border-bottom-0">
                <h5 class="fw-bold mb-0">
                    |   <i class="fas fa-map me-2"></i>Расположение на карте
                </h5>
            </div>
            <div class="card-body p-0">
                <div id="map" style="height: 400px; border-radius: 0 0 8px 8px;"></div>
            </div>
        </div>
    </div>
@endif


```

Рисунок 20 – Часть кода для вывода информации

```

@if($warehouse->hasCoordinates())
@push('scripts')
<script src="https://api-maps.yandex.ru/2.1/?apikey={{ env('YANDEX_MAPS_API_KEY') }}&lang=ru_RU" type="text/javascript"></script>
<script>
ymaps.ready(function() {
    var map = new ymaps.Map('map', {
        center: [{!! $warehouse->latitude !!}, {!! $warehouse->longitude !!}],
        zoom: 15
    });

    var placemark = new ymaps.Placemark([{!! $warehouse->latitude !!}, {!! $warehouse->longitude !!}], {
        balloonContent: '<strong>{{ $warehouse->name }}</strong><br>{{ $warehouse->full_address }}'
    }, {
        preset: 'islands#redIcon'
    });

    map.geoObjects.add(placemark);
    placemark.balloon.open();
});
</script>
@endpush
@endif

```

Рисунок 21 – Часть кода, отвечающая за определение координат

На рисунке 22 представлен вид страницы создания заявки от лица пользователя с ролью администратор. В заявке у пользователя может быть несколько товаров, поскольку товары, указанные в заявке, могут храниться на разных складах, после создания заявки система автоматически создаст перемещения товаров из склада хранения в склад получения клиентом товара.

Изм.	Лист	№ докум.	Подпись	Дата	КП.09.02.07-5.25.221.09.ПЗ	Лист 35

Новая заявка

Клиент *

Выберите клиента

Склад получения *

Выберите склад

Товары в заявке

+ Добавить товар

Товар *	Количество *	Доступно
Выберите товар	0.00	- ×

Примечание

Дополнительная информация...

Всего товаров Общее количество Уникальных позиций

1 0.00 0

X Отмена Создать заявку

Рисунок 22 – Вид страницы создания заявки

На рисунке 23 и 24 представлен код для добавления товаров в заявку.

```
<!-- товары -->


<div class="d-flex justify-content-between align-items-center mb-3">
        <h6 class="fw-bold mb-0">Товары в заявке</h6>
        <button type="button" class="btn btn-sm btn-outline-gray" id="addProduct">
            <i class="fas fa-plus me-1"></i>Добавить товар
        </button>
    </div>

    <div id="products-container">
        <!-- товары добавляются динамически -->
    </div>


```

```
@error('products')
    <div class="text-danger small mt-2">{{ $message }}</div>
@enderror
</div>
```

Рисунок 23 – Часть кода для отображения товаров в заявке

					КП.09.02.07-5.25.221.09.ПЗ	Лист 36
Изм.	Лист	№ докум.	Подпись	Дата		

```

<!-- Шаблон для товара -->
<template id="product-template">
    <div class="card-custom mb-3 product-item">
        <div class="card-body px-2 py-2">
            <div class="row align-items-center">
                <div class="col-md-5">
                    <label class="form-label">Товар *</label>
                    <select class="form-select product-select" name="products[][id]" required>
                        <option value="">Выберите товар</option>
                        @foreach($products as $product)
                            <option value="{{ $product->id }}"
                                data-quantity="{{ $product->total_quantity }}"
                                data-category="{{ $product->category->name }}"
                                data-manufacturer="{{ $product->manufacturer->name }}>
                                {{ $product->name }} (Доступно: {{ $product->total_quantity }})
                            </option>
                        @endforeach
                    </select>
                </div>
                <div class="col-md-4">
                    <label class="form-label">Количество *</label>
                    <input type="number" step="0.01" min="0.01" class="form-control quantity-input"
                        name="products[][quantity]" required placeholder="0.00"
                        oninput="validateQuantity(this)">
                </div>
                <div class="col-md-2">
                    <label class="form-label">Доступно</label>
                    <div class="form-control-plaintext available-quantity text-center fw-bold">--</div>
                </div>
                <div class="col-md-1">
                    <label class="form-label">&ampnbsp</label>
                    <button type="button" class="btn btn-outline-danger w-100 remove-product">
                        <i class="fas fa-trash"></i>
                    </button>
                </div>
            </div>
            <div class="product-info mt-2 small text-muted" style="display: none;">
                <span class="category"></span> • <span class="manufacturer"></span>
            </div>
        </div>
    </div>
</template>
@endsection

```

Рисунок 24 – Часть кода для выбора товаров в заявке

На рисунке 25 представлен вид страницы с отчетами по остаткам, перемещениям, выполненным заявкам. Для создания отчета по выполненным заявкам необходимо указать начальную и конечную дату, а также указать склад.

						КП.09.02.07-5.25.221.09.ПЗ	Лист 37
Изм.	Лист	№ докум.	Подпись	Дата			

The screenshot shows a dashboard titled 'Отчеты' (Reports) with three separate report forms:

- Отчет по заявкам**: Reports completed requests for the selected period. It includes fields for 'Начальная дата *' (09.11.2025), 'Конечная дата *' (09.12.2025), and 'Склад (опционально)' (All warehouses). A 'Скачать' (Download) button is present.
- Отчет по перемещениям**: Reports item movements for the selected period. It includes fields for 'Начальная дата *' (09.11.2025), 'Конечная дата *' (09.12.2025), and 'Статус' (All statuses). A 'Скачать' (Download) button is present.
- Отчет по остаткам**: Reports current item inventories at warehouses. It includes fields for 'Склад' (All warehouses), 'Статус остатков' (All statuses), and a 'Скачать' (Download) button.

Рисунок 25 – Вид страницы с отчетами

```
<!-- Отчет по заявкам -->
<div class="col-lg-4 mb-4">
    <div class="card-custom h-100 px-3 py-2">
        <div class="card-header bg-transparent border-bottom-0">
            <h5 class="fw-bold mb-0">
                | <i class="fas "></i>Отчет по заявкам
            </h5>
        </div>
        <div class="card-body">
            <p class="text-muted mb-4">Отчет по выполненным заявкам за выбранный период</p>

            <form action="{{ route('reports.orders') }}" method="GET">
                <div class="mb-3">
                    <label for="orders_start_date" class="form-label">Начальная дата *</label>
                    <input type="date" class="form-control" id="orders_start_date" name="start_date" value="{{ now()->subMonth()->format('Y-m-d') }}" required>
                </div>
                <div class="mb-3">
                    <label for="orders_end_date" class="form-label">Конечная дата *</label>
                    <input type="date" class="form-control" id="orders_end_date" name="end_date" value="{{ now()->format('Y-m-d') }}" required>
                </div>
                <div class="mb-3">
                    <label for="orders_warehouse" class="form-label">Склад (опционально)</label>
                    <select class="form-select" id="orders_warehouse" name="warehouse_id">
                        <option value="">Все склады</option>
                        @foreach(\App\Models\Warehouse::all() as $warehouse)
                            <option value="{{ $warehouse->id }}>{{ $warehouse->name }}</option>
                        @endforeach
                    </select>
                </div>

                <button type="submit" name="download" value="1" class="btn btn-outline-gray w-100">
                    | <i></i>Скачать
                </button>
            </form>
        </div>
    </div>
</div>
```

Рисунок 26 – Код для окна отчета по заявкам

Далее после скачивания документа с отчетом пользователю будет доступна таблица с данными относящимся к заявке, а именно: пользователь, создавший

Изм.	Лист	№ докум.	Подпись	Дата	КП.09.02.07-5.25.221.09.ПЗ	Лист 38

заявку, склад получения, количество видов товаров, суммы всех товаров в заявке, дату создания и дату выполнения.

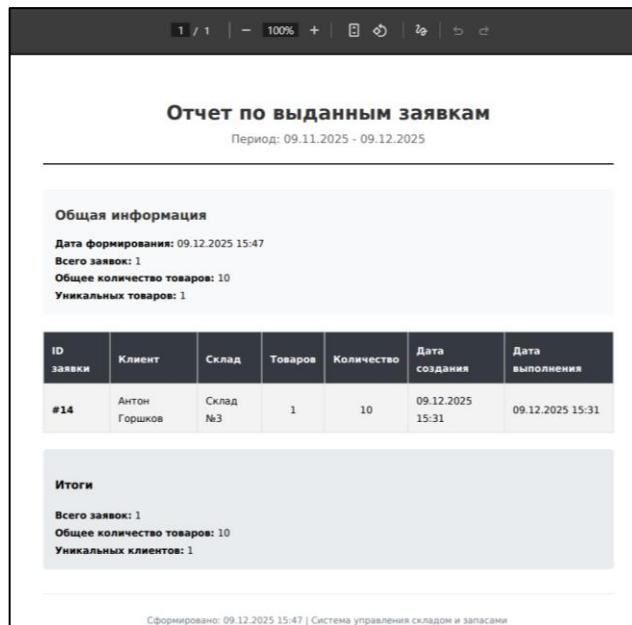


Рисунок 27 – Содержание сформированного отчета по выданным заявкам

```

@if($orders->count() > 0)


| ID заявки                              | Клиент                                                | Склад                                                 | Товаров                                                 | Количество                                                            | Дата создания                                                         | Дата выполнения                                                       |
|----------------------------------------|-------------------------------------------------------|-------------------------------------------------------|---------------------------------------------------------|-----------------------------------------------------------------------|-----------------------------------------------------------------------|-----------------------------------------------------------------------|
| <strong>#{{ \$order-&gt;id }}</strong> | <strong>{{ \$order-&gt;user-&gt;full_name }}</strong> | <strong>{{ \$order-&gt;warehouse-&gt;name }}</strong> | <strong>{{ \$order-&gt;products-&gt;count() }}</strong> | <strong>{{ \$order-&gt;products-&gt;sum('pivot.quantity') }}</strong> | <strong>{{ \$order-&gt;created_at-&gt;format('d.m.Y H:i') }}</strong> | <strong>{{ \$order-&gt;updated_at-&gt;format('d.m.Y H:i') }}</strong> |



### Итоги:



Всего заявок: {{ $orders->count() }}



Общее количество товаров: {{ $orders->sum(function($order) { return $order->products->sum('pivot.quantity'); }) }}



Уникальных клиентов: {{ $orders->unique('user_id')->count() }}


```

Рисунок 28 – Код для вывода данных о выданной заявки

Изм.	Лист	№ докум.	Подпись	Дата	Лист
					КП.09.02.07-5.25.221.09.П3 39

5.2 Разработка кода серверной части ИС

Без серверной части система не сможет выполнять свои основные функции. Поскольку разработка ведется с использованием фреймворка Laravel в серверной части проекта прописываются модели, контроллеры и маршруты. Далее будут описаны ключевые фрагменты серверной части разрабатываемой системы для страницы с пользователями. Для интеграции с авторизацией через сервис Яндекса в таблицу к пользователю был добавлен атрибут yandex_id.

```
use HasFactory, Notifiable;

protected $fillable = [
    'name',
    'surname',
    'fathername',
    'phone',
    'email',
    'password',
    'role_id',
    'yandex_id',
    'avatar',
];

protected $hidden = [
    'password',
    'remember_token',
];

protected function casts(): array
{
    return [
        'email_verified_at' => 'datetime',
        'password' => 'hashed',
    ];
}

public function role()
{
    return $this->belongsTo(Role::class);
}
```

Рисунок 29 – Представление модели пользователей

Изм.	Лист	№ докум.	Подпись	Дата

```

public function store(Request $request)
{
    $validated = $request->validate([
        'surname' => 'required|string|max:45',
        'name' => 'required|string|max:45',
        'fathername' => 'nullable|string|max:45',
        'phone' => 'required|string|unique:users',
        'email' => 'required|string|email|max:60|unique:users',
        'password' => ['required', 'confirmed', Rules\Password::defaults()],
        'role_id' => 'required|exists:roles,id',
    ]);

    $user = User::create([
        'surname' => $validated['surname'],
        'name' => $validated['name'],
        'fathername' => $validated['fathername'],
        'phone' => $validated['phone'],
        'email' => $validated['email'],
        'password' => Hash::make($validated['password']),
        'role_id' => $validated['role_id'],
    ]);

    return redirect()->route('users.show', $user)
        ->with('success', 'Пользователь успешно создан.');
}

public function show(User $user)
{
    $user->load(['role', 'orders.warehouse', 'orders.products']);

    return view('users.show', compact('user'));
}

public function edit(User $user)
{
    $roles = Role::orderBy('name')->get();

    return view('users.edit', compact('user', 'roles'));
}

```

Рисунок 30 – Часть кода контроллера пользователей

```

// Пользователи (только для админа)
Route::middleware(['auth', 'role:admin'])->group(function () {
    Route::get('/users', [UserController::class, 'index'])->name('users.index');
    Route::get('/users/create', [UserController::class, 'create'])->name('users.create');
    Route::post('/users', [UserController::class, 'store'])->name('users.store');
    Route::get('/users/{user}/show', [UserController::class, 'show'])->name('users.show');
    Route::get('/users/{user}/edit', [UserController::class, 'edit'])->name('users.edit');
    Route::put('/users/{user}', [UserController::class, 'update'])->name('users.update');
    Route::delete('/users/{user}', [UserController::class, 'destroy'])->name('users.destroy');
});

```

Рисунок 31 – Маршруты для страниц относящимся к пользователям

Для работы с сервисом был установлен пакет Socialite. Laravel Socialite – пакет для упрощения интеграции социальной аутентификации в приложения на основе Laravel. Он позволяет пользователям входить в систему с помощью учётных

Изм.	Лист	№ докум.	Подпись	Дата

данных социальных сетей. Пакет использует протокол OAuth для предоставления пользователям разрешения на доступ к их информации из социальных сетей. Он обрабатывает весь процесс аутентификации, включая перенаправление пользователей на соответствующую платформу, получение и хранение токенов аутентификации, а также извлечение данных пользователей. Код провайдера для работы сервиса Яндекс ID будет представлен в приложении Б – Листинг.

```
app > Http > Controllers > Auth > YandexAuthController.php > YandexAuthController
15  class YandexAuthController extends Controller
29      public function callback(Request $request)
62          $code = $request->get('code');
63          $client = new \GuzzleHttp\Client();
64
65          \Log::info('Starting direct API call to Yandex...');
66
67          // 1. Получаем access token
68          $tokenResponse = $client->post('https://oauth.yandex.ru/token', [
69              'form_params' => [
70                  'grant_type' => 'authorization_code',
71                  'code' => $code,
72                  'client_id' => env('YANDEX_CLIENT_ID'),
73                  'client_secret' => env('YANDEX_CLIENT_SECRET'),
74              ],
75              'headers' => [
76                  'Content-Type' => 'application/x-www-form-urlencoded',
77              ],
78          ]);
79
80          $tokenData = json_decode($tokenResponse->getBody(), true);
81          \Log::info('Token response from Yandex', $tokenData);
82
83          if (!isset($tokenData['access_token'])) {
84              throw new \Exception('Не получен access_token от Яндекс. Ответ: ' . json_encode($tokenData));
85          }
86
87          $accessToken = $tokenData['access_token'];
88
89          // 2. Получаем данные пользователя
90          $userResponse = $client->get('https://login.yandex.ru/info', [
91              'headers' => [
92                  'Authorization' => 'OAuth ' . $accessToken,
93                  'Accept' => 'application/json',
94              ],
95          ]);
96
97
98
99
100
101
102
103
104
```

Рисунок 32 – Часть кода контроллера для авторизации пользователя через Яндекс

```
"yandex" => [
    'client_id' => env('YANDEX_CLIENT_ID'),
    'client_secret' => env('YANDEX_CLIENT_SECRET'),
    'redirect' => env('YANDEX_REDIRECT_URI'),
],
```

Рисунок 33 – Настройка файла с сервисами для работы с Яндекс

Изм.	Лист	№ докум.	Подпись	Дата

```

class YandexServiceProvider extends ServiceProvider
{
    public function register()
    {
        //
    }

    public function boot()
    {
        $socialite = $this->app->make(Factory::class);

        $socialite->extend('yandex', function () use ($socialite) {
            $config = config('services.yandex');

            return $socialite->buildProvider(\SocialiteProviders\Yandex\Provider::class, $config);
        });
    }
}

```

Рисунок 34 – Код провайдера для авторизации через Яндекс

Для поиска склада на картах, необходимо в модели и миграции для склада указать широту и долготу. При создании склада при правильно введенном адресе можно автоматически определить широту и долготу.

```

class Warehouse extends Model
{
    use HasFactory;

    protected $fillable = [
        'name',
        'latitude',
        'longitude',
        'country',
        'city',
        'street',
        'house_number'
    ];

    public function stocks()
    {
        return $this->hasMany(Stock::class);
    }

    public function orders()
    {
        return $this->hasMany(Order::class);
    }
}

```

Рисунок 35 – Представление модели для склада

Изм.	Лист	№ докум.	Подпись	Дата
------	------	----------	---------	------

```

public function map()
{
    $warehouses = Warehouse::withCount(['stocks', 'orders'])
        ->orderBy('name')
        ->get();

    return view('warehouses.map', compact('warehouses'));
}

public function mapData()
{
    $warehouses = Warehouse::withCount(['stocks', 'orders'])
        ->whereNotNull('latitude')
        ->whereNotNull('longitude')
        ->get()
        ->map(function ($warehouse) {
            return [
                'id' => $warehouse->id,
                'name' => $warehouse->name,
                'latitude' => $warehouse->latitude,
                'longitude' => $warehouse->longitude,
                'address' => $warehouse->getFullAddressAttribute(),
                'stocks_count' => $warehouse->stocks_count,
                'orders_count' => $warehouse->orders_count,
                'url' => route('warehouses.show', $warehouse),
            ];
        });

    return response()->json($warehouses);
}

```

Рисунок 36 – Часть кода контроллера склада для вывода на карту точек со складом

5.3 Разработка базы данных ИС

Для функционирования ИС необходимо реализовать базу данных. База данных представляет собой организованный набор структурированных данных, хранящихся в цифровом формате.

Разработка базы данных для информационной системы «Управление складом и запасами» осуществляется с помощью создания миграций фреймворка Laravel, а с помощью инструмента PhpMyAdmin ведется управление базой данных. База данных состоит из 18 таблиц. Из которых Laravel создает автоматически 7 таблиц. На рисунке 37 представлена структура базы данных.

Изм.	Лист	№ докум.	Подпись	Дата
------	------	----------	---------	------

Таблица	Действие	Строки	Тип	Сравнение	Размер
cache	Обзор Структура Поиск Вставить Очистить Удалить	0	InnoDB	utf8mb4_unicode_ci	16,0 КИБ
cache_locks	Обзор Структура Поиск Вставить Очистить Удалить	0	InnoDB	utf8mb4_unicode_ci	16,0 КИБ
categories	Обзор Структура Поиск Вставить Очистить Удалить	3	InnoDB	utf8mb4_unicode_ci	16,0 КИБ
failed_jobs	Обзор Структура Поиск Вставить Очистить Удалить	0	InnoDB	utf8mb4_unicode_ci	16,0 КИБ
jobs	Обзор Структура Поиск Вставить Очистить Удалить	0	InnoDB	utf8mb4_unicode_ci	16,0 КИБ
job_batches	Обзор Структура Поиск Вставить Очистить Удалить	0	InnoDB	utf8mb4_unicode_ci	16,0 КИБ
manufacturers	Обзор Структура Поиск Вставить Очистить Удалить	1	InnoDB	utf8mb4_unicode_ci	16,0 КИБ
migrations	Обзор Структура Поиск Вставить Очистить Удалить	13	InnoDB	utf8mb4_unicode_ci	16,0 КИБ
movements	Обзор Структура Поиск Вставить Очистить Удалить	6	InnoDB	utf8mb4_unicode_ci	80,0 КИБ
orders	Обзор Структура Поиск Вставить Очистить Удалить	7	InnoDB	utf8mb4_unicode_ci	48,0 КИБ
order_products	Обзор Структура Поиск Вставить Очистить Удалить	9	InnoDB	utf8mb4_unicode_ci	48,0 КИБ
password_reset_tokens	Обзор Структура Поиск Вставить Очистить Удалить	0	InnoDB	utf8mb4_unicode_ci	16,0 КИБ
products	Обзор Структура Поиск Вставить Очистить Удалить	2	InnoDB	utf8mb4_unicode_ci	48,0 КИБ
roles	Обзор Структура Поиск Вставить Очистить Удалить	3	InnoDB	utf8mb4_unicode_ci	32,0 КИБ
sessions	Обзор Структура Поиск Вставить Очистить Удалить	1	InnoDB	utf8mb4_unicode_ci	48,0 КИБ
stocks	Обзор Структура Поиск Вставить Очистить Удалить	5	InnoDB	utf8mb4_unicode_ci	48,0 КИБ
users	Обзор Структура Поиск Вставить Очистить Удалить	4	InnoDB	utf8mb4_unicode_ci	64,0 КИБ
warehouses	Обзор Структура Поиск Вставить Очистить Удалить	6	InnoDB	utf8mb4_unicode_ci	16,0 КИБ
18 таблиц		Всего	60 InnoDB utf8mb4_0900_ai_ci		
576,0 КИБ					

Рисунок 37 – Структура базы данных

Структура таблиц соответствует схеме базы данных из пункта 4.3. на рисунке 38 изображена структура таблицы «users», имеющей необходимые атрибуты для работы пользователей, регистрирующихся в системе, так и авторизованных при помощи Yandex OAuth.

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно
1	id	bigint		UNSIGNED	Нет	<i>Нет</i>		AUTO_INCREMENT
2	name	varchar(45)	utf8mb4_unicode_ci		Нет	<i>Нет</i>		
3	surname	varchar(45)	utf8mb4_unicode_ci		Нет	<i>Нет</i>		
4	fathername	varchar(45)	utf8mb4_unicode_ci		Да	<i>NULL</i>		
5	phone	varchar(255)	utf8mb4_unicode_ci		Да	<i>NULL</i>		
6	email	varchar(60)	utf8mb4_unicode_ci		Нет	<i>Нет</i>		
7	password	varchar(255)	utf8mb4_unicode_ci		Да	<i>NULL</i>		
8	role_id	bigint		UNSIGNED	Нет	<i>Нет</i>		
9	remember_token	varchar(100)	utf8mb4_unicode_ci		Да	<i>NULL</i>		
10	created_at	timestamp			Да	<i>NULL</i>		
11	updated_at	timestamp			Да	<i>NULL</i>		
12	yandex_id	varchar(255)	utf8mb4_unicode_ci		Да	<i>NULL</i>		
13	avatar	varchar(255)	utf8mb4_unicode_ci		Да	<i>NULL</i>		

Рисунок 38 – Структура таблицы «users»

Изм.	Лист	№ докум.	Подпись	Дата	Лист 45
					КП.09.02.07-5.25.221.09.П3

```

public function up(): void
{
    Schema::create('users', function (Blueprint $table) {
        $table->id('id');
        $table->string('name', 45);
        $table->string('surname', 45);
        $table->string('fathername', 45)->nullable();
        $table->string('phone', 11);
        $table->string('email', 60)->unique();
        $table->string('password', 100);
        $table->foreignId('role_id')->constrained('roles')->default(3);
        $table->rememberToken();
        $table->timestamps();
    });

    Schema::create('password_reset_tokens', function (Blueprint $table) {
        $table->string('email')->primary();
        $table->string('token');
        $table->timestamp('created_at')->nullable();
    });

    Schema::create('sessions', function (Blueprint $table) {
        $table->string('id')->primary();
        $table->foreignId('user_id')->nullable()->index();
        $table->string('ip_address', 45)->nullable();
        $table->text('user_agent')->nullable();
        $table->longText('payload');
        $table->integer('last_activity')->index();
    });
}

```

Рисунок 39 – Миграция для создания таблицы «users»

Также в программном продукте присутствует интеграция с сервисом Яндекс.Карты. Для корректной работы с сервисом в структуру таблицы «warehouses» были внесены дополнительные атрибуты, отвечающие за широту и долготу расположения склада.

Изм.	Лист	№ докум.	Подпись	Дата
------	------	----------	---------	------

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно
1	id 	bigint		UNSIGNED	Нет	<i>Нет</i>		AUTO_INCREMENT
2	name	varchar(50)	utf8mb4_unicode_ci		Нет	<i>Нет</i>		
3	latitude	decimal(10,8)			Да	<i>NULL</i>		
4	longitude	decimal(11,8)			Да	<i>NULL</i>		
5	country	varchar(45)	utf8mb4_unicode_ci		Нет	<i>Нет</i>		
6	city	varchar(60)	utf8mb4_unicode_ci		Нет	<i>Нет</i>		
7	street	varchar(60)	utf8mb4_unicode_ci		Нет	<i>Нет</i>		
8	house_number	varchar(10)	utf8mb4_unicode_ci		Нет	<i>Нет</i>		
9	created_at	timestamp			Да	<i>NULL</i>		
10	updated_at	timestamp			Да	<i>NULL</i>		

Рисунок 40 – Структура таблицы «warehouses»

```
Schema::create('warehouses', function (Blueprint $table) {
    $table->id();
    $table->string('name', 50);
    $table->decimal('latitude', 10, 8)->nullable();
    $table->decimal('longitude', 11, 8)->nullable();
    $table->string('country', 45);
    $table->string('city', 60);
    $table->string('street', 60);
    $table->string('house_number', 10);
    $table->timestamps();
});
```

Рисунок 41 – Код миграции для создания таблицы «warehouses»

```
class Warehouse extends Model
{
    use HasFactory;

    protected $fillable = [
        'name',
        'latitude',
        'longitude',
        'country',
        'city',
        'street',
        'house_number'
    ];

    public function stocks()
    {
        return $this->hasMany(Stock::class);
    }

    public function orders()
    {
        return $this->hasMany(Order::class);
    }
}
```

Рисунок 42 – Модель таблицы «warehouses»

Изм.	Лист	№ докум.	Подпись	Дата	КП.09.02.07-5.25.221.09.П3	Лист 47
------	------	----------	---------	------	----------------------------	------------

6 Тестирование ИС

6.1 Тестовый сценарий

Для тестирования разработанной системы применялось функциональное тестирование, направленное на проверку соответствия программного продукта его функциональным требованиям. Для проведения тестирования использовался фреймворк PHPUnit, встроенный в Laravel.

Таблица 15 – Сценарий создания заявки клиентом

Параметр	Описание
ID теста	FUNC-POS-01
Название теста	Создание заявки клиентом
Предусловия	Пользователь с ролью «Клиент» авторизован в системе, есть записи о товарах
Шаги	<ol style="list-style-type: none">1) Переход на страницу создания заявки2) Выбор склада получения3) Добавление товаров с указанием количества4) Нажатие кнопки "Создать заявку"
Тестовые данные	Склад «Главный», товар «Арматура», количество: 2
Ожидаемый результат	Заявка успешно создана, автоматически сформированы перемещения

Таблица 16 – Сценарий изменения статуса перемещения сотрудником

Параметр	Описание
ID теста	FUNC-POS-02
Название теста	Изменение статуса перемещения с «В прогресс» на «Выполнен»
Предусловия	Пользователь с ролью «Сотрудник» авторизован, есть активные перемещения
Шаги	1) Переход на страницу перемещений 2) Нажатие кнопки "Выполнить" у выбранного перемещения
Тестовые данные	Перемещение ID: 1, товар: «Арматура», количество: 5
Ожидаемый результат	Перемещение отмечено как выполненное, остатки на складах обновлены

Таблица 17 – Сценарий создания товара администратором

Параметр	Описание
ID теста	FUNC-POS-03
Название теста	Создание продукта пользователем администратором
Предусловия	Пользователь с ролью «Администратор» авторизован в системе, имеются записи о производителях и категориях, зарегистрированы клиенты
Шаги	1) Переход на страницу создания заявки 2) Выбор склада получения 3) Добавление товаров с указанием количества 4) Нажатие кнопки "Создать заявку"
Тестовые данные	Склад «Основной», товар «Монитор», количество: 2, ID клиента: 2
Ожидаемый результат	Заявка успешно создана, автоматически сформированы перемещения

Таблица 18 – Сценарий валидации данных при создании заявки

Параметр	Описание
ID теста	FUNC-NEG-01
Название теста	Валидация данных при создании заявки
Предусловия	Пользователь с ролью «Клиент» авторизован в системе
Шаги	1) Переход на страницу создания заявки 2) Попытка создания заявки с пустыми полями и отрицательными значениями
Тестовые данные	Пустое название, отрицательная цена, нулевое количество
Ожидаемый результат	Система выводит сообщения об ошибках валидации

Таблица 20 – Сценарий проверки разграничения прав

Параметр	Описание
ID теста	FUNC-NEG-02
Название теста	Доступ к админ-панели обычным пользователем
Предусловия	Пользователь с ролью «Клиент» авторизован в системе
Шаги	1) Попытка перехода на страницу управления пользователями
Тестовые данные	URL: «/users/index.php»
Ожидаемый результат	Доступ запрещен. Редирект на страницу с началом работы в системе

Таблица 21 – Сценарий создания заявки с неверным количеством товара

Параметр	Описание
ID теста	FUNC-NEG-03
Название теста	Попытка создания заявки с недостаточным количеством товара
Предусловия	Пользователь с ролью «Клиент» авторизован в системе, имеются записи о товарах с низким запасом
Шаги	1) Попытка создания заявки с количеством товара больше доступного
Тестовые данные	Склад «Главный», товар «Арматура», количество: 5, доступно: 1
Ожидаемый результат	Система выводит сообщение о недостатке товара

Результаты проведенного тестирования представлены на рисунках 43, 44 и 45. Тестовый сценарий «RoleAccessTest» (рисунок 43) проверяет доступ различных ролей к функциям приложения.

```
PASS Tests\Feature\RoleAccessTest
✓ dashboard access for all roles
✓ user management access
✓ order management access
✓ movement management access
✓ product management access
✓ warehouse management access
✓ crud operations for different roles
✓ order edit access levels
✓ yandex oauth access for all

Tests: 9 passed (48 assertions)
Duration: 0.70s
```

Рисунок 43 – Результат выполнения тестов по разграничению пользователей по ролям

Тестовый сценарий «OrderCreationTest» (рисунок 44) разработан для проверки функционала по созданию заявок для различных ролей пользователей. А

тестовый сценарий «MovementCompletionTest» (рисунок 45) служит для проверки доступа пользователей к изменению статуса перемещения.

```
PASS Tests\Feature\OrderCreationTest
✓ client can create order
✓ admin can create order for client
✓ employee cannot create order
✓ guest cannot create order
✓ order creation validation
✓ cannot create order with insufficient stock
✓ movements created automatically for order

Tests: 7 passed (24 assertions)
Duration: 0.52s
```

Рисунок 44 – Результат выполнения тестов по созданию заявок

```
PASS Tests\Feature\MovementCompletionTest
✓ employee can complete movement
✓ admin can complete movement
✓ client cannot complete movement
✓ cannot complete already completed movement

Tests: 4 passed (12 assertions)
Duration: 0.45s
```

Рисунок 45 – Результат выполненных тестов по изменению статуса перемещения

6.2 Тестовые наборы данных

В качестве тестовых данных выступают фабрики фреймворка Laravel, позволяющие создавать реалистичные тестовые данные. Фабрика (Factory) используется для фабрик моделей – классов, которые определяют, как генерировать тестовые данные для моделей Eloquent. Вместо ручного создания экземпляров моделей фабрики позволяют определять атрибуты по умолчанию. Ниже приведены основные фабрики для данных.

Изм.	Лист	№ докум.	Подпись	Дата	КП.09.02.07-5.25.221.09.ПЗ	Лист 52
------	------	----------	---------	------	----------------------------	------------

```

public function definition(): array
{
    return [
        'name' => $this->faker->randomElement(['admin', 'employee', 'client']),
        'created_at' => now(),
        'updated_at' => now(),
    ];
}

public function admin(): static
{
    return $this->state(fn (array $attributes) => [
        'name' => 'admin',
    ]);
}

public function employee(): static
{
    return $this->state(fn (array $attributes) => [
        'name' => 'employee',
    ]);
}

```

Рисунок 46 – Фабрика для ролей пользователей

```

public function definition(): array
{
    return [
        'from_warehouse_id' => Warehouse::factory(),
        'to_warehouse_id' => Warehouse::factory(),
        'product_id' => Product::factory(),
        'quantity' => $this->faker->randomFloat(2, 1, 100),
        'status' => 'in_progress',
        'type' => 'between_warehouses',
        'order_id' => null,
        'created_at' => now(),
        'updated_at' => now(),
    ];
}

```

Рисунок 47 – Фабрика для перемещений

```

public function definition(): array
{
    return [
        'user_id' => User::factory(),
        'warehouse_id' => Warehouse::factory(),
        'notes' => $this->faker->optional()->sentence(),
        'status' => 'pending',
        'created_at' => now(),
        'updated_at' => now(),
    ];
}

public function forUser(User $user): static
{
    return $this->state(fn (array $attributes) => [
        'user_id' => $user->id,
    ]);
}

public function forWarehouse(Warehouse $warehouse): static
{
    return $this->state(fn (array $attributes) => [
        'warehouse_id' => $warehouse->id,
    ]);
}

public function pending(): static
{
    return $this->state(fn (array $attributes) => [
        'status' => 'pending',
    ]);
}

```

Рисунок 48 – Фабрика для заявок

Изм.	Лист	№ докум.	Подпись	Дата	KPI.09.02.07-5.25.221.09.ПЗ	Лист 53

```

public function definition(): array
{
    return [
        'name' => $this->faker->words(3, true),
        'description' => $this->faker->optional()->paragraph(),
        'category_id' => Category::factory(),
        'manufacturer_id' => Manufacturer::factory(),
        'created_at' => now(),
        'updated_at' => now(),
    ];
}

```

Рисунок 49 – Фабрика для продуктов

```

public function definition(): array
{
    return [
        'product_id' => Product::factory(),
        'warehouse_id' => Warehouse::factory(),
        'quantity' => $this->faker->randomFloat(2, 0, 1000),
        'position' => $this->faker->optional()->bothify('Стеллаж ##-Полка #'),
        'last_update' => now(),
        'created_at' => now(),
        'updated_at' => now(),
    ];
}

```

Рисунок 50 – Фабрика для остатков

6.3 Методы тестирования

OrderCreationTest – тестирование создания и управления заявками.

- Проверка создания заявок клиентами.
- Проверка создания заявок администратором для клиентов.
- Тестирование валидации входных данных.
- Проверка автоматического создания перемещений.

RoleAccessTest – тестирование доступа на основе ролей.

- Проверка доступа к главной странице для разных ролей.
- Тестирование управления пользователями.
- Проверка доступа к управлению заявками и перемещениями.
- Тестирование CRUD операций для разных ролей.

MovementCompletionTest – тестирование выполнения перемещений.

- Проверка выполнения перемещений сотрудниками.

Изм.	Лист	№ докум.	Подпись	Дата

- Тестирование ограничений доступа для клиентов.
- Проверка обработки ошибок при недостатке товара.

Иzm.	Лист	№ докум.	Подпись	Дата	КП.09.02.07-5.25.221.09.П3	Лист
						55

7 Документирование ИС

7.1 Руководитель по установке ИС

Информационную систему «Управление складом и запасами» можно установить и развернуть на персональном компьютере для ознакомления с функционалом. Для начала работы следует установить программное обеспечение необходимое для запуска и эксплуатирования программного продукта, указанные в анализе инструментальных средств, а именно:

- Visual Studio Code – для подключения к новой базе данных и миграции таблиц в базу данных.
- OpenServer 6.0.0 – виртуальный сервер для работы системы.
- PhpMyAdmin – для управления базой данных продукта
- PHP 8.3 – язык, на котором работает продукт.
- Любой браузер, указанный в техническом задании для взаимодействия с системой.

Первым шагом следует перенести папку с проектом в папку «OSPanel/home», для возможности запустить проект на виртуальном сервере (рисунок 51).

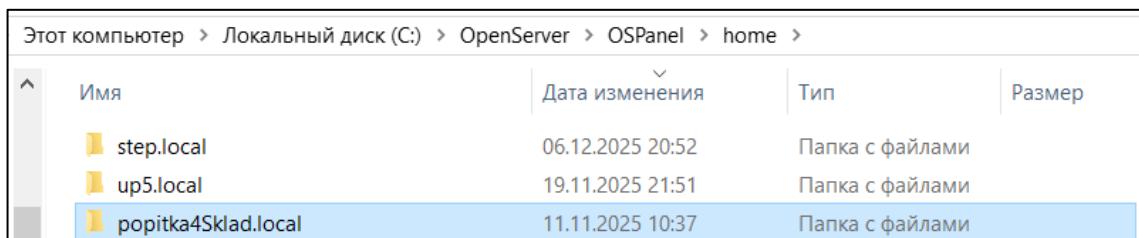


Рисунок 51 – Расположение папки с проектом на OpenServer

Далее необходимо открыть папку проекта в Visual Studio Code и изменить строки подключения к базе данных в файле .env (рисунок 52). Следует вводить данные соответствующие вашим параметрам подключения.

Изм.	Лист	№ докум.	Подпись	Дата	КП.09.02.07-5.25.221.09.ПЗ	Лист 56
------	------	----------	---------	------	----------------------------	------------

```
DB_CONNECTION=mysql  
DB_HOST=mysql-8.0  
DB_PORT=3306  
DB_DATABASE=popitka3Sklad  
DB_USERNAME=root  
DB_PASSWORD=
```

Рисунок 52 – Подключение к базе данных

Выполнить команды после ввести следующие команды в терминал редактора кода для создания таблиц в базе данных. Код представлен на рисунке 53.

```
php artisan migrate:fresh
```

Рисунок 53 – Код для переноса таблиц в базу данных

Также для корректной работы фреймворка Bootstrap следует ввести команду, представленную на рисунке 54. И для работы с сервисами Яндекс следует ввести команду, изображенную на рисунке 55.

```
npm run dev
```

Рисунок 54 – Код для работы фреймворка

```
php artisan serve --host=127.0.0.1 --port=8000
```

Рисунок 55 – Код для запуска проекта на конкретном адресе и порте

Чтобы начать работу с продуктом необходимо в поисковую строку браузера ввести запрос <http://127.0.0.1:8000>.

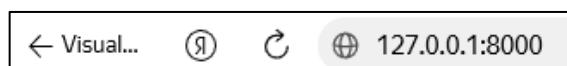


Рисунок 55 – Адрес ИС

На этом установка и настройка окружения для работы информационной системы заканчивается.

Изм.	Лист	№ докум.	Подпись	Дата

7.2 Руководство пользователя ИС

1 Введение

Настоящее руководство пользователя распространяется на информационную систему (далее – ИС) «Управление складом и запасами» версии 1.0. Руководство содержит технические характеристики, описание программного обеспечения, принципы взаимодействия пользователя с ИС и указания, необходимые для правильной эксплуатации.

2 Назначение

Информационная система представляет собой программный продукт, состоящий из клиентской и серверной части. Система предназначается для управления товарами на складе и автоматизации выполнения заявок от клиентов.

3 Подготовка к работе

Для начала работы с ИС «Управление складом и запасами» на автоматизированном рабочем месте (далее - АРМ) администратора системы должен быть предустановлен один из предложенных WEB – браузеров: Microsoft Edge, Mozilla Firefox, Opera, с установленными обновлениями не позднее 2020 года. Также должна быть установлена среда Open Server Panel (далее OSP) с готовыми модулями: MySQL-8.0 и PHP-8.3.

4 Работа с информационной системой

4.1 Запуск системы и авторизация

Инструкция по запуску приложения на персональном компьютере находится в пункте 7.1 Руководство по установке ИС. После выполнения всех

Изм.	Лист	№ докум.	Подпись	Дата	КП.09.02.07-5.25.221.09.ПЗ	Лист 58
------	------	----------	---------	------	----------------------------	------------

шагов из пункта 7.1 пользователь попадает на страницу, представленную на рисунке 56, с общей информацией о системе.

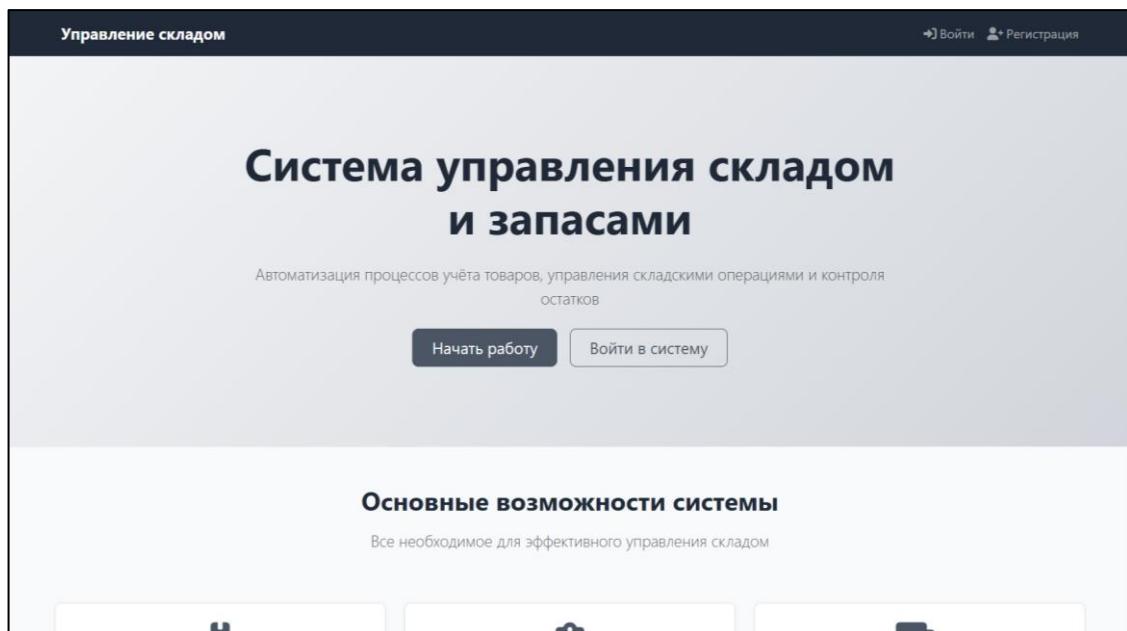


Рисунок 56 – Страница с информацией о системе

The registration form is titled 'Регистрация' (Registration) and has the subtext 'Создайте новый аккаунт' (Create a new account). It features a red button 'Я Зарегистрироваться через Яндекс' (I register through Yandex). Below this, there are fields for 'Фамилия *' (Surname *), 'Имя *' (Name *), and 'Отчество' (Middle name). There is also a field for 'Телефон *' (Phone *) containing the number '8 (XXX) XXX-XX-XX'. An 'Email *' field is present, along with 'Пароль *' (Password *) and 'Подтверждение *' (Confirmation *). A large dark blue button at the bottom is labeled 'Зарегистрироваться' (Register). At the bottom right, there is a link 'Уже есть аккаунт? Войти' (Already have an account? Log in).

Рисунок 57 – Регистрация пользователя

						КП.09.02.07-5.25.221.09.П3	Лист 59
Изм.	Лист	№ докум.	Подпись	Дата			

Для получения доступа к данным необходимо пройти процедуру регистрации (рисунок 57): указать фамилию, имя, отчество, номер телефона электронную почту и пароль, а затем нажать кнопку «Зарегистрироваться». Также предусмотрен вариант авторизации в системе при помощи учетной записи Яндекс. После выбора входа при помощи Яндекс, пользователя переносит на окно, изображенное на рисунке 58.

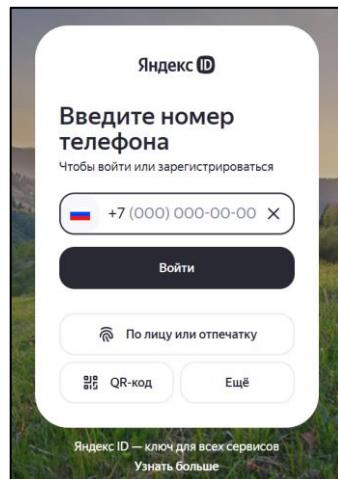


Рисунок 58 – Авторизация при помощи учетной записи Яндекс

После успешной авторизации при переходе в профиль учетной записи в ИС будет выведена следующая информация, представленная на рисунке 59.

Редактирование профиля

Основная информация

Имя * Горшков

Фамилия * Антон

Отчество

Роль client

Контактная информация

Email * anton2006gor@yandex.ru

Телефон * 7000000000

Я Управление через Яндекс

Ваш аккаунт привязан к Яндекс ID. Для изменения пароля и email используйте Яндекс аккаунт.

Я Управление Яндекс аккаунтом

Дата регистрации: 16.11.2025 12:37

Сохранить изменения

Рисунок 59 – Данные учетной записи Яндекс

Изм.	Лист	№ докум.	Подпись	Дата

4.2 Администратор

4.2.1 Главная страница

После успешной регистрации вы попадаете на главную страницу приложения (рисунок 60).

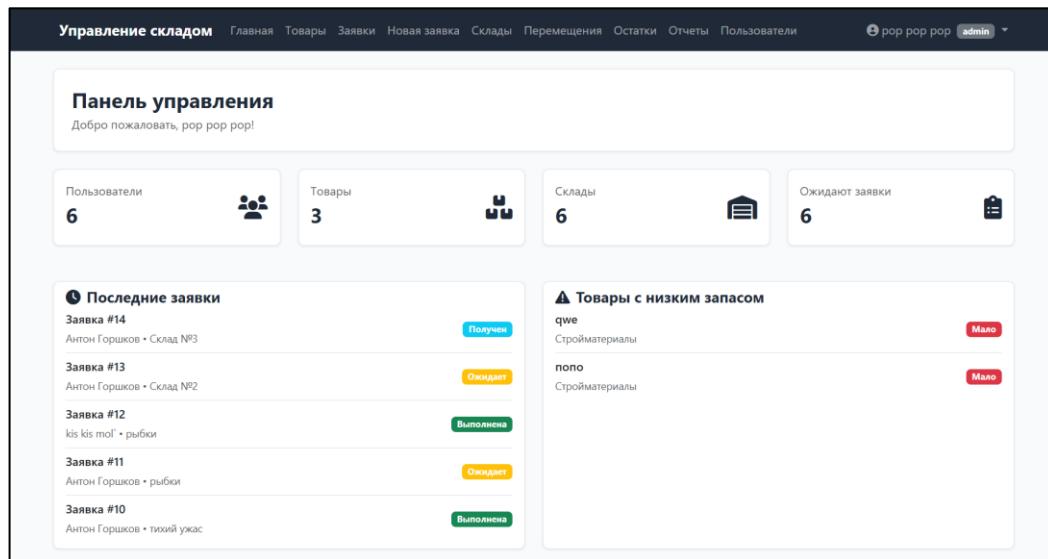


Рисунок 60 – Главная страница

На главной странице ИС представлены список последних заявок, предупреждения о товарах с низким запасом на складах, а также возможности подсчет общей информации системы.

Навигация сайта находится в верхушке сайта, с нее можно перейти на остальные страницы.

Выход из системы осуществляется через кнопку «Выйти» в правом верхнем углу при нажатии на часть с информацией о пользователе.

4.2.2 Товары

После перехода на страницу «Товары» откроется окно со списком товаров, зарегистрированных в системе (рисунок 61).

Изм.	Лист	№ докум.	Подпись	Дата	КП.09.02.07-5.25.221.09.П3	Лист 61
------	------	----------	---------	------	----------------------------	---------

Товары						
Управление товарами и остатками на складах						
Поиск по названию...		Все категории		Все производители		Найти
Название	Категория	Производитель	Общее количество	Склады	Дата создания	Действия
Арматура	Металлоконструкции	ЗАО "Бешеки"	105	1 скл.	10.12.2025	
Пеноплекс	Стройматериалы	ООО "Тсос"	10	3 скл.	11.11.2025	
Пеноблок	Стройматериалы	ООО "Тсос"	41	4 скл.	11.11.2025	

Рисунок 61 – подробная информация о заказе

4.2.3 Создание товара, категории и производителя

Пользователь с ролью имеет право создавать записи о товарах, категорий и производителях. При выборе функции «Добавить товар» открывается соответствующая страница, на которой есть возможность внести данные для нового товара (рисунок 62).

+ Добавить товар	
Название товара *	Категория *
<input type="text"/>	Выберите категорию
Производитель *	Склад размещения *
Выберите производителя	Выберите склад
Количество *	Позиция на складе
<input type="text" value="0"/>	Например: Стеллаж А, Полка 2
Описание	
<input type="text" value="Подробное описание товара..."/>	
<input type="button" value="Отмена"/>	<input type="button" value="Сохранить товар"/>

Рисунок 62 – Создание нового товара

Пользователь может перейти на страницу с информацией о категориях, представленную на рисунке 63.

Категории товаров				
Управление категориями товарной номенклатуры				
Поиск по названию...		Найти		
Название	Описание	Товаров	Статус	Действия
Металлоконструкции	—	1	Активно	

Рисунок 63 – Категории товаров

Изм.	Лист	№ докум.	Подпись	Дата

Для добавления новой категории следует пройти те же этапы, как и с добавлением товара в базу данных. Процесс добавления не отличается для других данных используемых для работы приложения, следует на странице, относящейся к необходимой вкладке, перейти по кнопке «Добавить (название)» и ввести в доступные поля данные. Необходимые поля для создания записи выделены знаком «*», остальные поля могут оставаться пустыми. При заполненных данных следует нажать на кнопку «Сохранить» или «Создать» после выполнения всех шагов пользователя перенаправит на страницу с информацией о добавленном товаре. Схожим принципом работает редактирование записей.

Производители				
Управление производителями товаров				
Производитель	Адрес	Товаров	Статус	Действия
ЗАО "Бещеки" ID: 2	Россия, Магадан, Ленина, 12	1	✓ Активен	<input type="button" value=""/> <input type="button" value=""/> <input type="button" value=""/>
ООО "Тсос" ID: 1	Россия, Магадан, Поп, 5	2	✓ Активен	<input type="button" value=""/> <input type="button" value=""/> <input type="button" value=""/>

Рисунок 64 – Страница со списком Производителей

4.2.4 Заявки и создание заявки

Все записи о заявках доступны к просмотру на странице «Заявки». В открывшемся окне будут выведены все данные заявок (рисунок 65). Есть возможность создания новых заявок (рисунок 66). При создании заявки от роли администратора следует указать: клиента, склад получения, товары и их количество и примечание при необходимости. После создания заявки происходит проверка на наличие товаров в заявке и имеющихся товаров на складе, если товаров недостаточно, то создаются перемещения с данными товарами из других складов для выполнения заявки.

Изм.	Лист	№ докум.	Подпись	Дата		КП.09.02.07-5.25.221.09.П3	Лист 63

Заявки						
Все заявки системы						
ID заявки	Статус	Склад	Клиент	Дата с	Дата по	
#14	Антон Горшков anton2006gor@yandex.ru	Склад №3	1 шт.	Получен	09.12.2025 15:31	
#13	Антон Горшков anton2006gor@yandex.ru	Склад №2	1 шт.	Ожидает	09.12.2025 13:36	

Рисунок 65 – Список заявок

Новая заявка

Клиент *

Выберите клиента

Склад получения *

Выберите склад

Товары в заявке

Товар *	Количество *	Доступно
Выберите товар	0.00	

Примечание

Дополнительная информация...

+ Добавить товар
× Отмена
Создать заявку

Рисунок 66 – Создание заявки

4.2.5 Склады и отображение склада на карте, создание склада

Информация о складах располагается на странице «Склады» (рисунок 67). О каждом складе можно увидеть подробную информацию, представленную на рисунке 68, о самом складе и его местоположение на карте, если перейти по кнопке с изображением глаза. Информацию о складах могут просматривать все роли пользователей.

Склады					
Управление складскими помещениями					
Название	Адрес	Товаров	Заявок	Координаты	Действия
главный склад Создан: 11.11.2025	Магадан Портовая, 45 Россия	3	2	Есть	

Рисунок 67 – Информация о складах

Изм.	Лист	№ докум.	Подпись	Дата	КП.09.02.07-5.25.221.09.П3	Лист 64

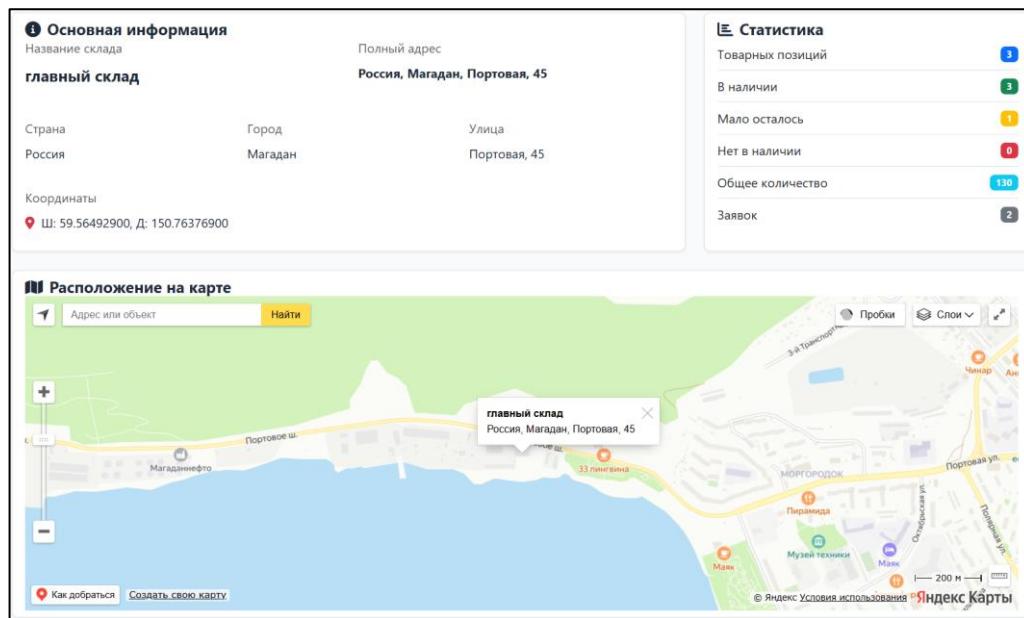


Рисунок 68 – Подробная информация о складе

Для создания новой записи склада, необходимо указать адрес и нажать на кнопку «Определить координаты», после поля с данными широты и долготы автоматически заполняются (рисунок 69).

+ Добавить склад

Основная информация

Название склада *

Введите название склада

Адрес

Страна * Город * Улица *

Россия Например: Москва Например: Ленина

Номер дома *

Например: 15 Определить координаты

Широта 55.755826 Долгота 37.617300

Расположение на карте

Адрес или объект Найти Яндекс Карты

Рисунок 69 – Добавление Склада

Изм.	Лист	№ докум.	Подпись	Дата

4.2.6 Пользователи

Администратор имеет право просматривать список всех пользователей системы, а также имеет право на изменение статуса пользователей и создание нового пользователя (рисунок 70). Создание пользователя ничем не отличается от шаблона по добавлению записей.

Пользователи						
Управление пользователями системы						
Все роли		Все типы		Поиск по имени или email...		+ Добавить пользователя
Пользователь	Контактная информация	Роль	Заявки	Дата регистрации	Действия	
Выборгина Варвара Валерьевна	varavina2@yandex.ru +7 (956) 132-15-53	employee	0	11.12.2025		
Олегов Олег Олегович	oleg@oleg +7 (950) 222-33-44	client	0	10.12.2025		
kis kis mol'	kis@kis +7 (950) 145-25-44	client	1	07.12.2025		
Антон Горшков	anton2006gor@yandex.ru 70000000000	client	6	16.11.2025		

Рисунок 70 – Пользователи

4.2.7 Перемещения

Перемещения в основном формируются для переноса товаров между складами (рисунок 71), поэтому большую часть из них создает система для автоматизации. Однако перемещения могут происходить между складами, поэтому администратор имеет право создавать перемещения (рисунок 72).

Перемещения									
Все перемещения системы									
Статус	Тип	Склад	Товар	Дата с	Дата по	Действия	+ Новое перемещение		
Все статусы	Все типы	Все склады	Все товары	дд.мм.гггг	дд.мм.гггг				
#20	Пеноблок Стройматериалы	тихий ужас	Склад №3	10.00	Для заявки #14	Выполнено	09.12.2025 15:31		
#19	Пеноблок Стройматериалы	главный склад	тихий ужас	2.00	Меж складами	Выполнено	09.12.2025 15:18		
#18	Пеноблок Стройматериалы	тихий ужас	Какиш	10.00	Для заявки #13	Выполнено	09.12.2025 13:36		
#17	Пеноплекс Стройматериалы	главный склад	тихий ужас	2.00	Меж складами	Выполнено	08.12.2025 14:43		

Рисунок 71 – Список перемещений

Изм.	Лист	№ докум.	Подпись	Дата	КП.09.02.07-5.25.221.09.П3	Лист 66
------	------	----------	---------	------	----------------------------	------------

+ Новое перемещение

Тип перемещения *

Между складами

Товар *

Арматура (Общее количество: 105)

Склад-источник *

главный склад - 105.00 шт.

Будут показаны только склады, на которых есть выбранный товар

Склад-назначения *

Склад №3 (Магадан)

Количество *

10

Доступно на выбранном складе: 105.00

Отмена Создать перемещение

Рисунок 72 – Создание нового перемещения

4.2.8 Остатки

Система должна обеспечивать управление остатками на складах. На странице «Остатки» выводится список количества товаров на складах (рисунок 73).

Остатки на складах					
Управление товарными остатками на складах					
Поиск	Склад	Товар	Категория	Статус	
<input type="text" value="Название товара"/>	<input type="text" value="Все склады"/>	<input type="text" value="Все товары"/>	<input type="text" value="Все категории"/>	<input type="text" value="Все статусы"/>	Добавить остаток
Товар	Склад	Количество	Позиция	Последнее обновление	Действия
Пеноблок Стройматериалы • ООО "Тсос"	главный склад Магадан	17.00	A2	09.12.2025 15:18	
Пеноплекс Стройматериалы • ООО "Тсос"	главный склад Магадан	8.00 Мало осталось	A3	08.12.2025 15:02	
Арматура Металлоконструкции • ЗАО "Бещеки"	главный склад Магадан	105.00	A6	10.12.2025 08:51	

Рисунок 73 – Остатки на складе

Изм.	Лист	№ докум.	Подпись	Дата	КП.09.02.07-5.25.221.09.ПЗ	Лист 67
------	------	----------	---------	------	----------------------------	---------

4.2.9 Отчеты

Система поддерживает формирование отчетов по выданным заявкам, перемещениям и по текущим остаткам на складах. Вид страницы представлен на рисунке 74. Для создания отчета по выданным заявкам, необходимо указать период времени и при необходимости склад, с которого забирали заявки. Для отчета по перемещениям необходимо указать период, статус перемещений, и также склад, куда перемещали товар. Для формирования отчета по остаткам можно указать склад и статус остатков.

The screenshot shows a user interface for generating reports in PDF format. It includes three main sections:

- Отчет по заявкам**: Reports issued requests for the selected period. It has fields for 'Начальная дата' (14.11.2025) and 'Конечная дата' (14.12.2025), a dropdown for 'Склад (опционально)' (All warehouses), and a 'Скачать' (Download) button.
- Отчет по перемещениям**: Reports movements of goods for the selected period. It has fields for 'Начальная дата' (14.11.2025) and 'Конечная дата' (14.12.2025), a dropdown for 'Статус' (All statuses), a dropdown for 'Склад (опционально)' (All warehouses), and a 'Скачать' (Download) button.
- Отчет по остаткам**: Reports current inventory levels on warehouses. It has a dropdown for 'Склад' (All warehouses), a dropdown for 'Статус остатков' (All status), and a 'Скачать' (Download) button.

Рисунок 74 – Страница для создания отчетов

The screenshot shows a detailed report for movements of goods. At the top, it displays the title 'Отчет по перемещениям товаров' and the period '14.11.2025 - 14.12.2025'. Below this is a section titled 'Общая информация' containing the following data:

- Дата формирования: 14.12.2025 16:30
- Всего перемещений: 10
- Общее количество: 60
- Выполнено: 5

Below this is a table with columns: ID, Товар, Из склада, В склад, Количество, Тип, Статус, and Дата создания. One row is shown:

ID	Товар	Из склада	В склад	Количество	Тип	Статус	Дата создания
#7	Пеноблок	главный склад	Склад №2	5.00	для заявки	В процессе	14.11.2025 04:38

Рисунок 75 – Отчет по перемещениям товаров

					КП.09.02.07-5.25.221.09.П3	Лист 68
Изм.	Лист	№ докум.	Подпись	Дата		

4.3 Сотрудник склада

4.3.1 Главная страница

После авторизации за пользователя с ролью «Сотрудник склада» пользователь попадает на главную страницу (рисунок 76). Данной роли доступен функционал только для просмотра заявок, остатков и изменения статуса перемещения.

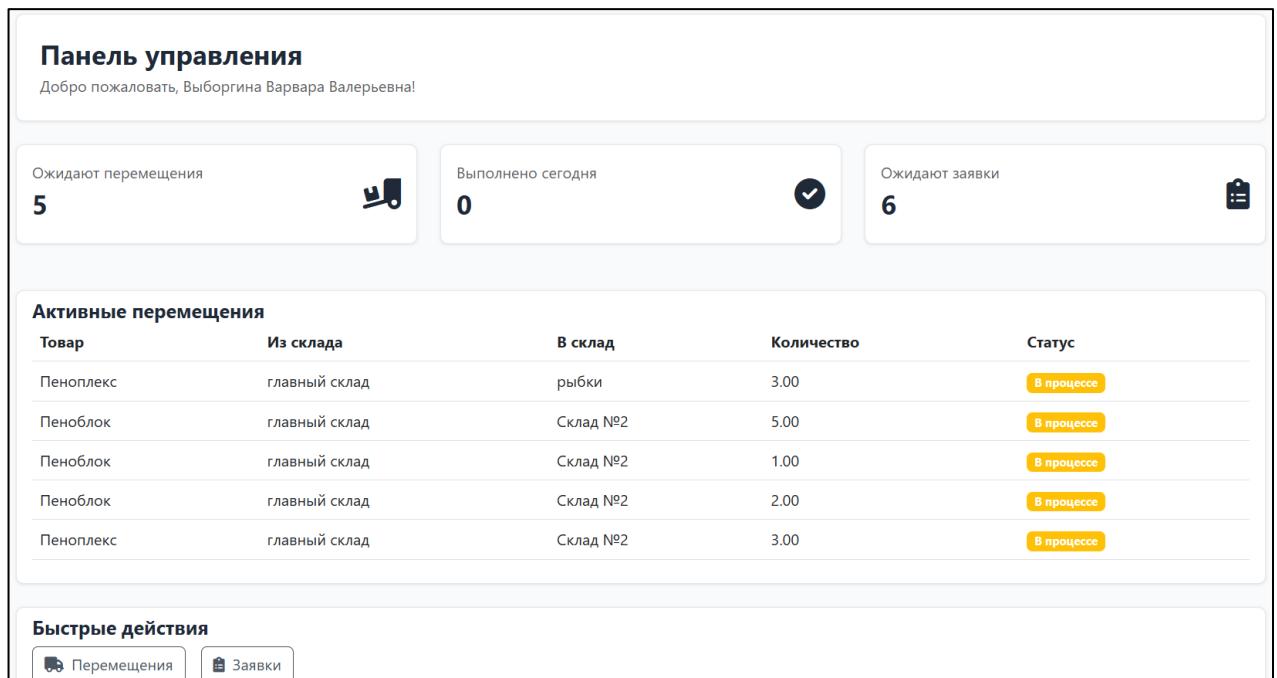


Рисунок 76 – Главная страница сотрудника

4.3.2 Перемещения

Сотрудник склада имеет доступ к просмотру всех невыполненных перемещений для заявок (рисунок 77). Он может открыть перемещение в отдельном окне и просмотреть детальную информацию и изменить статус на «Выполнен» (рисунок 78).

Изм.	Лист	№ докум.	Подпись	Дата	КП.09.02.07-5.25.221.09.П3	Лист 69
------	------	----------	---------	------	----------------------------	---------

Перемещения

Активные перемещения для выполнения

Статус	Тип	Склад	Товар	Дата с	Дата по			
Все статусы	Все типы	Все склады	Все товары	дд.мм.гггг	дд.мм.гггг			
ID	Товар	Из склада	В склад	Количество	Тип	Статус	Дата создания	Действия
#16	Пеноплекс Стройматериалы	главный склад	рыбки	3.00	Для заявки #8	В процессе	08.12.2025 14:14	
#7	Пеноблок Стройматериалы	главный склад	Склад №2	5.00	Для заявки #7	В процессе	14.11.2025 04:38	
#4	Пеноблок Стройматериалы	главный склад	Склад №2	1.00	Для заявки #5	В процессе	11.11.2025 05:00	
#1	Пеноблок Стройматериалы	главный склад	Склад №2	2.00	Для заявки #3	В процессе	11.11.2025 04:33	
#2	Пеноплекс Стройматериалы	главный склад	Склад №2	3.00	Для заявки #3	В процессе	11.11.2025 04:33	

Рисунок 77 – Страница перемещений

Перемещения / #21

Перемещение #21

Детальная информация о перемещении

Основная информация

Номер перемещения
#21

Статус
В процессе

Тип перемещения
Для заявки

Количество
5.00

Связанная заявка
Заявка #15
Клиент: Антон Горшков
Склад: рыбки

Дата создания
12.12.2025 23:32

Последнее обновление
12.12.2025 23:32

Информация

Товар	Пеноплекс
Категория	Стройматериалы
Производитель	ООО "Tsoc"

Рисунок 78 – Страница просмотра конкретного перемещения

Изм.	Лист	№ докум.	Подпись	Дата	КП.09.02.07-5.25.221.09.П3	Лист 70
------	------	----------	---------	------	----------------------------	------------

4.3.3 Остатки

Поскольку пользователь имеет роль «Сотрудник склада», значит имеет возможность просматривать (рисунок 79) и изменять данные об остатках товара на складах (рисунок 80), а именно его позицию на складе и хранимое количество.

Остатки на складах						+ Добавить остаток
Управление товарными остатками на складах						
Поиск	Склад	Товар	Категория	Статус		
<input type="text" value="Название товара"/>	<input type="text" value="Все склады"/>	<input type="text" value="Все товары"/>	<input type="text" value="Все категории"/>	<input type="text" value="Все статусы"/>		
Товар	Склад	Количество	Позиция	Последнее обновление	Действия	
Пеноблок Стройматериалы • ООО "Тсос"	главный склад Магадан	17.00	A2	09.12.2025 15:18		
Пеноплекс Стройматериалы • ООО "Тсос"	главный склад Магадан	3.00 Мало осталось	A3	12.12.2025 23:36		
Арматура Металлоконструкции • ЗАО "Бещеки"	главный склад Магадан	105.00	A6	10.12.2025 08:51		
Пеноблок Стройматериалы • ООО "Тсос"	Склад №2 Магадан	0.00	A1	12.11.2025 13:07		

Рисунок 79 – Список остатков на складах

Редактировать остаток Назад

Товар Пеноблок Стройматериалы • ООО "Тсос"	Склад главный склад Магадан, Портовая
Количество *	Позиция на складе
<input type="text" value="17,00"/>	<input type="text" value="A2"/>
Информация Создан: 11.11.2025 03:58 Обновлен: 09.12.2025 15:18	Статус В наличии
Просмотр	Отмена Обновить остаток

Рисунок 80 – Редактирование остатка

					Лист 71	КП.09.02.07-5.25.221.09.ПЗ
Изм.	Лист	№ докум.	Подпись	Дата		

4.4 Клиент

4.4.1 Главная страница

После регистрации в приложении пользователь может просмотреть функции (рисунок 81), которые система предлагает для использования: просмотр товаров, просмотр заявок и создание новой заявки, а также просмотр информации о складах и хранимых на них товарах.

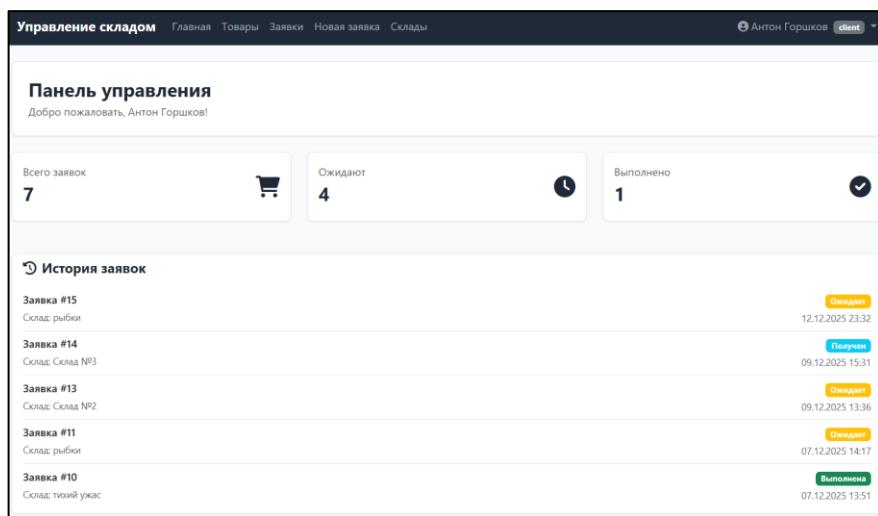


Рисунок 81 – Главная страница у клиента

Просмотрев интересующие товары, пользователь может перейти по кнопке «Новая заявка» на страницу создания заявки. На рисунке 82 представлен интерфейс страницы. Пользователю для создания заявки необходимо выбрать склад, из которого клиент сможет забрать товары из своей заявки, после следует выбрать товары. В заявке может быть указано несколько товаров, также система учитывает введенное количество товара и имеющееся на складах количество, в случае превышения система выдаст предупреждение о несоответствии. При желании можно указать дополнительную информацию для заявки. Данные заявки можно изменить только пока она имеет статус «Ожидание», дальнейшие действия с заявкой не допускаются.

Изм.	Лист	№ докум.	Подпись	Дата
------	------	----------	---------	------

Новая заявка

Склад получения *

Товары в заявке

+ Добавить товар

Товар *	Количество *	Доступно
Пеноблок (Доступно: 41)	100	41
Стройматериалы • ООО "Тсос"		
Арматура (Доступно: 105)	27	105
Металлоконструкции • ЗАО "Бещеки"		

Примечание

Дополнительная информация...

Всего товаров 2 Общее количество 127.00 Уникальных позиций 2

Рисунок 82 – Создание заявки и предупреждение и превышении доступного количества

После создания заявки на странице с подробной информацией о заявке (рисунок 83) можно отследить статусы перемещений на указанный в заявке склад и сам статус заявки. После получения товаров из заявки клиентом статус заявки меняется с «Выполнен» на «Получено».

Заявка #16

← Назад | Редактировать

Детальная информация о заявке

Основная информация

Номер заявки #16
Статус Ожидает

Клиент
Антон Горшков
anton2006gor@yandex.ru

Склад получения
Склад №2
Магадан, Ленина, 11

Дата создания
14.12.2025 15:40
Последнее обновление
14.12.2025 15:40

Статистика

Товаров в заявке 1
Общее количество 10
Перемещения 1
Выполнено перемещений 0

Товары в заявке

Товар	Категория	Производитель	Количество	Доступно на складах
Пеноблок	Стройматериалы	ООО "Тсос"	10.00	41

Связанные перемещения

Товар	Из склада	В склад	Количество	Статус	Дата создания
Пеноблок	главный склад	Склад №2	10.00	В процессе	14.12.2025 15:40

Рисунок 83 – Подробная информация о заявке

Иzm.	Лист	№ докум.	Подпись	Дата		КП.09.02.07-5.25.221.09.П3	Лист 73

ЗАКЛЮЧЕНИЕ

В ходе выполнения курсового проекта была успешно разработана информационная система «Управление складом и запасами», полностью соответствующая поставленным целям и задачам. Основной целью являлось создание современного, удобного и функционального программного продукта, автоматизирующего ключевые процессы складского учёта, управления запасами, обработки заявок и формирования отчётности. Цель достигнута в полном объёме.

В рамках проекта решены следующие задачи:

- Проведено детальное исследование предметной области, определены ключевые сущности и процессы, подлежащие автоматизации.
- Выполнен сравнительный анализ инструментальных средств разработки, на основе которого были выбраны наиболее подходящие технологии: PHP, Laravel, MySQL, Bootstrap и др.
- Разработано техническое задание в соответствии с ГОСТ 34.602-2020, определяющее функциональные, технические и эксплуатационные требования к системе.
- Осуществлено комплексное проектирование ИС, включая разработку архитектуры, функциональных и структурных схем (диаграммы прецедентов, деятельности, классов, IDEF0, DFD), проектирование базы данных (ER-модель) и пользовательского интерфейса.
- Проведена непосредственная разработка системы, включающая создание клиентской и серверной частей, реализацию базы данных, интеграцию с внешними сервисами (Яндекс.Карты, Яндекс OAuth) и обеспечение безопасности.
- Выполнено функциональное тестирование системы с использованием PHPUnit, подтвердившее корректность работы всех основных модулей и сценариев использования.
- Составлена полная программная документация, включая руководство пользователя и руководство по установке.

Изм.	Лист	№ докум.	Подпись	Дата
------	------	----------	---------	------

Был реализован следующий функционал программного продукта:

- Многопользовательская система с ролевым доступом (Гость, Клиент, Сотрудник склада, Администратор).
- Регистрация и авторизация (в том числе через OAuth Яндекс).
- Полный цикл управления товарами, категориями, производителями и складами.
- Создание и отслеживание заявок с автоматическим формированием перемещений.
- Визуализация складов на карте с использованием Яндекс.Карт.
- Формирование отчётов по выполненным операциям.
- Интуитивно понятный адаптивный веб-интерфейс, разработанный с использованием Bootstrap.

Информационная система «Управление складом и запасами» представляет собой практичный и масштабируемый программный продукт, способный повысить эффективность работы склада за счёт автоматизации рутинных операций, минимизации ошибок учёта и предоставления актуальной аналитической информации.

Изм.	Лист	№ докум.	Подпись	Дата	КП.09.02.07-5.25.221.09.ПЗ	Лист 75
------	------	----------	---------	------	----------------------------	------------

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

- 1 [github.com](https://github.com/barryvdh/laravel-dompdf) - DOMPDF Wrapper for Laravel: – URL: <https://github.com/barryvdh/laravel-dompdf> (дата обращения: 27.10.2025). – Текст: электронный.
- 2 [laravel.com](https://laravel.com/docs/12.x/socialite) – Laravel Socialite: – URL: <https://laravel.com/docs/12.x/socialite> (дата обращения 24.11.2025). – Текст: электронный.
- 3 [laravel.com](https://laravel.su/docs/12.x/documentation) – Документация по работе с Laravel 12.x: – URL: <https://laravel.su/docs/12.x/documentation> (дата обращения: 30.10.2025). – Текст: электронный.
- 4 ospanel.io – Документация по работе с Open Panel Server – URL: <https://ospanel.io/> (дата обращения: 10.10.2025). – Текст: электронный.
- 5 [php.net](https://www.php.net/manual/ru/index.php) – Документация по работе с php – URL: <https://www.php.net/manual/ru/index.php> (дата обращения: 25.10.2025). – Текст: электронный.
- 6 [timeweb.com](https://timeweb.com/ru/docs/virtualnyj-hosting/bazy-dannyh/osnovy-raboty-v-phpmyadmin/) – Документация по работе с phpMyAdmin – URL: <https://timeweb.com/ru/docs/virtualnyj-hosting/bazy-dannyh/osnovy-raboty-v-phpmyadmin/> (дата обращения: 30.10.2025). – Текст: электронный.
- 7 [yandex.ru](https://yandex.ru/maps-api/docs/geocoder-api/index.html) – API Геокодера: – URL: <https://yandex.ru/maps-api/docs/geocoder-api/index.html> (дата обращения: 30.11.2025). – Текст: электронный.
- 8 [yandex.ru](https://yandex.ru/maps-api/docs/js-api/common/quickstart.html) – JavaScript API: – URL: <https://yandex.ru/maps-api/docs/js-api/common/quickstart.html> (дата обращения: 24.11.2025). – Текст: электронный.
- 9 [yandex.ru](https://yandex.ru/dev/id/doc/ru/) – Яндекс ID: – URL: <https://yandex.ru/dev/id/doc/ru/> (дата обращения 30.11.2025). – Текст: электронный.
- 10 Как создать руководство пользователя: – URL: <https://weeek.net/ru/blog/user-manuals> (дата обращения: 12.12.2025). – Текст: электронный.

Изм.	Лист	№ докум.	Подпись	Дата

Приложение А – Техническое задание

Министерство образования Иркутской области

Государственное бюджетное профессиональное образовательное учреждение

Иркутской области

«Иркутский авиационный техникум»

(ГБПОУИО «ИАТ»)

Техническое задание

**ВЕБ-ПРИЛОЖЕНИЕ «УПРАВЛЕНИЕ СКЛАДОМ И
ЗАПАСАМИ»**

Руководитель:

(М.А. Кудрявцева)

(подпись, дата)

Студент:

(А.А. Горшков)

(подпись, дата)

Иркутск 2025

Изм.	Лист	№ докум.	Подпись	Дата
------	------	----------	---------	------

КП.09.02.07-5.25.221.09.П3

Лист
77

1 Введение

1.1 Общие сведения

Документ представляет собой техническое задание на создание информационной системы (далее – ИС) «Система управления складом и запасами», предназначенной для автоматизации

Вид программного продукта: информационная система.

1.2 Цели и задачи

Целью создания ИС «Система управления складом и запасами» является автоматизация процессов учёта товаров, управления складскими операциями, контроля остатков и формирования отчётности.

Задачи ИС включают:

- Автоматизацию процессов приёма, хранения и выдачи товаров.
- Формирование заявок, контроль остатков и поиск позиций.
- Учёт движения товаров, обработку заявок и анализ данных.

2 Основания для разработки

2.1 Нормативные документы

Документ основывается на следующих нормативных документах:

- ГОСТ 34.602-2020 "Информационная технология. Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированной системы".
- Методические указания по КП МДК 05.02 2024 ИС_4 курс.

2.2 Проектные документы

Проектные документы включают:

					КП.09.02.07-5.25.221.09.П3	Лист 78
Изм.	Лист	№ докум.	Подпись	Дата		

- Задание на дипломное проектирование.
- Методические указания по выполнению дипломного проектирования.
- Пояснительную записку.
- Руководство пользователя.

3 Назначение приложения

3.1 Общее описание

ИС «Система управления складом и запасами» предназначена для использования сотрудниками складов, логистами, менеджерами по закупкам и администраторами. Приложение будет поддерживать процессы учёта товаров, управления остатками, формирования заказов, проведения инвентаризаций и генерации отчётов. Для администраторов предусмотрены функции управления пользователями, настройки справочников и мониторинга действий в системе.

4 Требования к приложению

4.1 Функциональные требования

1. Функции роли «Гость»;
 - Просмотр главной страницы, раздела «О системе»;
 - Просмотр описания функционала;
 - Регистрация и авторизация:
 - Регистрация в системе;
 - Авторизация в системе;
 - Авторизация в системе с помощью сервиса Яндекс;
2. Функции роли «Сотрудник склада»;

Все функции роли «Гость»

 - Поиск товаров:
 - Поиск по названию;

Изм.	Лист	№ докум.	Подпись	Дата	КП.09.02.07-5.25.221.09.ПЗ	Лист 79

- Поиск по местоположению;

- Управление товарами:

- Изменение позиций товара на складе;

3. Функции роли «Клиент»:

Все функции роли «Гость»;

- Управление личными данными в личном кабинете:

– Изменение регистрационных данных (ФИО, адрес электронной почты, номер телефона);

- Смена пароля;

- Выход из аккаунта;

- Просмотр доступных товаров:

- Поиск по названию;

- Фильтрация товаров по категориям;

- Отображение по доступности (в наличии или нет в наличии);

- Управление заявками:

- Добавление товаров в заявку;

- Изменение количества добавленных товаров;

- Удаление добавленных товаров;

- Отправка заявки на рассмотрение;

4. Функции роли «Администратор»:

Все функции роли «Клиент»;

- Управление заявками;

- Подтверждение или отклонение заявки;

- Формирование отчета по заявке;

- Управление товарами:

- Изменение данных товаров;

- Добавление новых товаров;

- Формирование отчета о товарах;

Изм.	Лист	№ докум.	Подпись	Дата
------	------	----------	---------	------

- Удаление товаров
- Управление категориями:
- Добавление категорий;
- Изменение категорий;
- Удаление категорий;
- Просмотр списка пользователей;
- Возможность блокировки пользователя;
- Возможность смены роли пользователя;
- Фильтрация списка пользователей по имени, электронной почте, номеру телефона, дате регистрации аккаунта;

4.2 Технические требования

4.2.1 Производительность

- Количество одновременных пользователей: до 1000 активных сессий;
- Количество запросов в секунду (RPS/QPS): 100 RPS;
- Среднее время отклика: не более 3000 мс.

4.2.2 Надежность

- Валидация пользовательского ввода на клиенте - все формы, поля, файлы и параметры должны проверяться перед отправкой запроса на сервер.
- Валидация пользовательского ввода на сервере - все входные данные, независимо от наличия клиентской проверки, проходят повторную строгую валидацию.
- Защита от SQL-инъекций за счёт применения параметризованных запросов или ORM с встроенной защитой.

4.3 Эксплуатационные требования

						КП.09.02.07-5.25.221.09.П3	Лист 81
Изм.	Лист	№ докум.	Подпись	Дата			

4.3.1.1 Среда функционирования:

- Поддержка современных браузеров: Яндекс браузер, Edge.

4.3.1.2 Установка и развёртывание:

- Разработка инструкции по установке и запуску.

4.3.1.3 Интеграция:

- С сервисом Яндекс.карты.
- Регистрация и авторизация при помощи Яндекс.

4.4 Требования к информационной безопасности

ИС должна обеспечивать защиту данных и предотвращение несанкционированного доступа в соответствии со следующими требованиями:

1. Аутентификация и авторизация:

– Система должна обеспечивать аутентификацию пользователей перед предоставлением доступа к функциональным модулям. Механизм аутентификации: по электронной почте и паролю или при помощи учетной записи в мессенджере Telegram.

– Пароли пользователей должны храниться в зашифрованном виде с использованием стойкого криптографического алгоритма Всгурт.

– Для каждой пользовательской сессии должен формироваться уникальный токен. Токен (JWT, session ID) должен иметь ограниченное время жизни (30 минут) и быть защищён от подделки. При смене пароля токен должен обновляться.

– Поддержка двухфакторной аутентификации (2FA, TOTP).

2. Защита от распространённых уязвимостей:

– Защита от несанкционированного доступа к данным - проверка прав доступа к каждому ресурсу на серверной стороне.

5 Требования к техническому обеспечению

Изм.	Лист	№ докум.	Подпись	Дата	КП.09.02.07-5.25.221.09.ПЗ	Лист 82

5.1 Сервер

- Количество процессоров: 1 шт.
- Количество ядер одного процессора: 8 ядер.
- Тактовая частота одного процессора: 2.5 ГГц.
- Оперативная память: 32 ГБ.
- Пропускная способность сети: 1 Гбит/с.
- Клиентские рабочие станции: рекомендуется ПК с ОС Windows 10/11, современный встроенный GPU с поддержкой WebGL 2.0 или дискретная видеокарта с 4 ГБ видеопамяти, не менее 8 ГБ ОЗУ, не менее 10 ГБ свободного места на диске.

5.2 Хранилище данных

- Тип накопителей: SSD.
- Общий объем хранилища: 1 ТБ.
- IOPS на чтение: 5000.
- IOPS на запись: 3000.

5.3 Клиентские устройства

5.3.1 Минимальные требования для персонального компьютера (ноутбука)

- Количество ядер процессора: 4 ядра.
- Тактовая частота процессора: 2.0 ГГц.
- Оперативная память: 8 ГБ.
- Тип диска: HDD или SSD.
- Свободного места на диске: не менее 10 ГБ.
- Разрешение дисплея: Full HD (1920×1080).

					КП.09.02.07-5.25.221.09.П3	Лист 83
Изм.	Лист	№ докум.	Подпись	Дата		

- Пропускная способность сети: 100 Мбит/с.

5.4 Сетевые требования

- Доступ к сети Интернет со скоростью не менее 1 Гбит/с.
- Поддержка следующих сетевых протоколов: HTTP/1.1, HTTP/2, WebSocket, DNS, TCP/IP.
- Поддержка IPv4 (обязательно), IPv6 (при наличии).
- Возможность работы через защищённые порты 443 (HTTPS) и 80 (HTTP).

6 Требования к программному обеспечению

6.1 Сервер

- · Операционная система Ubuntu Server 24.04 LTS (возможные аналоги: Windows Server 2022 Standard/Datacenter).
 - · Сервер базы данных: MySQL 8.4 LTS (возможные варианты: PostgreSQL 16, MariaDB 11.3).
 - · Веб-сервер: Nginx 1.24 (или Apache HTTP Server 2.4.58).
 - · Интерпретатор: PHP 8.3+, Node.js 22+.
 - · Менеджер пакетов: Composer 2.7 (для PHP), npm 10.8 или yarn 4.1 (для Node.js).
 - · Система контроля версий: Git 2.45.

6.2 Персональный компьютер (ноутбук)

1. Операционная система Windows 10 / 11 (64-bit), Linux Ubuntu 22.04 / 24.04 LTS, macOS 12 Monterey и выше.
2. Веб-браузеры (актуальные версии, не старше двух релизов): Яндекс.Браузер 25.4+.

Изм.	Лист	№ докум.	Подпись	Дата	КП.09.02.07-5.25.221.09.П3	Лист 84

3. Необходимые компоненты браузера:
 - Поддержка HTML5, CSS3, ECMAScript 2015+ (ES6+).
 - Поддержка Web Storage API, IndexedDB, Fetch API.
 - Поддержка WebSocket, Service Workers, PWA.
 - Включена поддержка JavaScript и cookies.

7 Требования к тестированию

7.1 Общие требования

Тестирование ИС должно охватывать все ключевые аспекты функциональности, производительности и безопасности. Этап тестирования (п. 7.1, этап 4) должен включать:

- Документирование.
- Функциональное тестирование (минимум 3 функции).

7.1.1 Документирование

Тестовые сценарии (test cases) – охватывающие ключевые функциональные и нефункциональные требования к системе для проверки работы:

- в типовых и исключительных ситуациях;
 - при вводе корректных и некорректных данных;
- при различных ролях пользователей (администратор, клиент, сотрудник склада).

Тестовая документация должна включать:

- план тестирования; · перечень тестовых сценариев;
- отчёт о результатах тестирования и выявленных дефектах.

7.1.2 Функциональное тестирование

Изм.	Лист	№ докум.	Подпись	Дата	КП.09.02.07-5.25.221.09.ПЗ	Лист 85
------	------	----------	---------	------	----------------------------	------------

- Проверка корректности работы ИС в соответствии с функциональными требованиями (п. 4.1).
- Проверка работы пользовательских сценариев (регистрация, аутентификация, навигация, операции CRUD, отчёты и т.д.).
- Проверка корректности взаимодействия с внешними системами (Yandex Maps, Yandex OAuth).
- Проверка обработки ошибок и сообщений пользователю при некорректных действиях.

8 Организационно-технические требования

8.1 Этапы разработки

В таблице 1 представлены сроки и этапы разработки ИС.

Таблица 1 – Сроки и этапы разработки ИС

Номер	Наименование этапов	Начало
1	Предпроектное исследование предметной области	18.09.25
2	Разработка технического задания (выбор архитектуры программного обеспечения, выбор типа пользовательского интерфейса, выбор языка и среды программирования)	23.09.25
3	Проектирование программного обеспечения. (разработка структурной и функциональной схемы ПО, проектирование базы данных (ER-модель))	25.10.25
4	Разработка (программирование) программного продукта	30.12.25
5	Тестирование и отладка программного продукта (функциональное тестирование, составление тест-планов)	14.12.25
6	Составление программной документации (написание руководства пользователя)	16.12.25

Приложение Б – Листинг «Провайдер YandexServiceProvider»

```
<?php
// app/Providers/YandexServiceProvider.php

namespace App\Providers;

use Illuminate\Support\ServiceProvider;
use Laravel\Socialite\Contracts\Factory;

class YandexServiceProvider extends ServiceProvider
{
    public function register()
    {
        //
    }

    public function boot()
    {
        $socialite = $this->app->make(Factory::class);

        $socialite->extend('yandex', function () use ($socialite) {
            $config = config('services.yandex');

            return $socialite->buildProvider(\SocialiteProviders\Yandex\Provider::class,
                $config);
        });
    }
}
```

Изм.	Лист	№ докум.	Подпись	Дата