**Министерство образования Иркутской области**

Государственное бюджетное профессиональное

образовательное учреждение Иркутской области

«Иркутский авиационный техникум»

(ГБПОУИО «ИАТ»)

|  |  |
| --- | --- |
| КП.09.02.07-5.23.201.8 ПЗ |  |

ИНФОРМАЦИОННАЯ СИСТЕМА

«СКЛАДСКОЙ КОМПЛЕКС»

|  |  |  |
| --- | --- | --- |
| Председатель ВЦК: | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись, дата) | (М.А. Кудрявцева) |
| Руководитель: | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись, дата) | (М.П. Морозов) |
| Студент: | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись, дата) | (Д.В. Козлов) |

Иркутск 2023

**СОДЕРЖАНИЕ**

[Введение 3](#_Toc150945581)

[1 Описание предметной области ИС 4](#_Toc150945582)

[2 Анализ инструментов, используемых в разработке программного продукта 5](#_Toc150945583)

[3 Техническое задание 15](#_Toc150945584)

[4 Проектирование ИС 16](#_Toc150945585)

[4.1 Структурная схема ИС 16](#_Toc150945586)

[4.2 Функциональная схема ИС 18](#_Toc150945587)

[4.3 Проектирование базы данных 21](#_Toc150945588)

[5 Разработка ИС 27](#_Toc150945589)

[5.1 Разработка интерфейса ИС 27](#_Toc150945590)

[5.2 Разработка базы данных 34](#_Toc150945591)

[5.3 Разработка ИС 35](#_Toc150945592)

[6. Технологическая документация ИС 38](#_Toc150945598)

[6.1. Руководство пользователя ИС 38](#_Toc150945599)

[Заключение 41](#_Toc150945600)

[Список используемых материалов 42](#_Toc150945601)

[Приложение Б – Листинг User.php 49](#_Toc150945602)

Введение

Складской комплекс – совокупность складских помещений, на доступной площади которых производится хранение товаров и грузов. Помещения могут быть расположены отдельно или иметь общее покрытие с разными системами охраны. Непосредственно на территории складского комплекса производятся складские логистические услуги (размещение, учет, обновление, перемещение). Большая часть складов проектируется для хранения сырья до выполнения операций и готовой продукции до её дистрибьюции. В меньшей степени здесь хранят незавершенное производство, расходуемые материалы и запасные части.

Для более удобного и быстрого поиска товаров, необходимо создать информационную систему, с помощью которой можно будет ускорить и облегчить работу на складе.

Целью курсовой работы является создание информационной системы «Складской комплекс».

Для достижения конечного результата, а именно создание информационной системы «Складской комплекс», необходимо решить следующие задачи:

* разработать техническое задание;
* проанализировать инструментальные средства разработки;
* спроектировать базу данных;
* спроектировать информационную систему;
* спроектировать интерфейс;
* разработать базу данных;
* разработать информационную систему;
* разработать интерфейс;
* продемонстрировать программный продукт;
* создать руководство пользователя по информационной системе.

1. Описание предметной области ИС

Выбранной областью является "Складской комплекс", который предназначен для хранения продукции от сторонних лиц. Складской комплекс включает помещения для хранения товаров, пути для подъезда, зоны погрузки-выгрузки и другие сооружения. Этот комплекс спроектирован для конкретных бизнес-потребностей, обеспечивая эффективное использование пространства, улучшая трудовой процесс и обеспечивая учет продукции. Он также может быть легко пересмотрен или расширен.

На складе проводится инвентаризация, включающая этапы, такие как пересмотр продукции, ответственным сотрудником, который вручную записывает данные о товарах и их количестве, сопровождая книгу переучета. Затем эти данные сверяются с книгой учета товаров сотрудниками, отвечающими за отчетную документацию на складе, и составляется соответствующий отчет по результатам переучета продукции.

Информационная система предназначена для сотрудников и облегчает работу с товарами, позволяя добавлять, изменять и удалять информацию о них.

Таким образом, в функционирование ИС входит:

* авторизация, регистрация;
* разграничение прав доступа;
* манипулирование заказами;
* манипулирование сотрудниками.

1. Анализ инструментов, используемых в разработке программного продукта

Инструменты, правильно выбранные для разработки программы, играют ключевую роль в ее итоговом качестве.

Структурировать информационную систему можно с помощью phpMyAdmin и Draw.io, тогда как для дизайна отлично подходит Figma. Планируемая система будет серверного типа, предназначенная для сотрудников. Для реализации будут использованы технологии HTML5, CSS3, JS, в том числе AJAX.

phpMyAdmin - это веб-приложение на PHP, предоставляющее пользовательский интерфейс для управления базами данных MySQL. С его помощью можно управлять сервером MySQL, выполнять SQL-запросы и просматривать содержание таблиц через веб-браузер. В данном проекте оно применяется для построения ER-диаграмм.

Draw.io - бесплатное приложение для создания диаграмм, основанное на HTML5 и JavaScript. Его можно использовать для разработки различных схем, включая блок-диаграммы, макеты, UML и другие. В этом проекте оно служит для разработки прототипов страниц.

Figma - это инструмент для дизайна и прототипирования веб-сайтов и приложений. Он позволяет нескольким пользователям работать над одним проектом одновременно, предоставляя доступ к редактированию или комментированию. В Figma создаются прототипы, иллюстрации, интерфейсы и даже макеты для платформы Tilda. В этом проекте он служит для визуального оформления.

HTML — это язык разметки, который предоставляет браузеру инструкции по отображению содержимого страницы. Браузеры интерпретируют HTML-код и отображают его в виде веб-страницы, предназначенной для восприятия пользователем.CSS – каскадные таблицы стилей, которые используются для определения стилей (правил) оформления документов — включая дизайн, вёрстку и вариации макета для различных устройств и размеров экрана.

JavaScript представляет собой универсальный язык программирования, который часто используется для внедрения в приложения с целью управления различными их аспектами. В контексте веб-разработки, знание JavaScript является неотъемлемым для создания современных интерактивных веб-сайтов. Этот язык придает жизнь структуре HTML-страниц и обеспечивает функциональность пользовательского интерфейса веб-приложений. JavaScript позволяет страницам и их элементам реагировать на действия пользователей. В настоящее время JavaScript является основным языком программирования для веб-браузеров и полностью совместим с различными операционными системами, включая Windows, Linux, Mac OS, а также мобильными платформами.

Технология AJAX (Asynchronous JavaScript And XML) позволяет взаимодействовать с сервером без необходимости перезагрузки всей веб-страницы. Это сокращает время ожидания и делает веб-приложения более интерактивными, приближая их к функциональности десктопных приложений.Веб-приложение 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. MySQL Workbench;

3. PostgreSQL.

MySQL — одна из наиболее используемых систем управления базами данных. MySQL управляет реляционными базами данных, то есть такими, в которых таблицы связаны между собой. MySQL работает по принципу клиент-сервер. Компьютер пользователя (клиент) отправляет запрос. Сервер баз данных его обрабатывает и предоставляет ответ. Именно поэтому часто можно услышать понятие MySQL-сервер. Это сервер, на котором хранится база данных. Система MySQL написана на языках программирования C и C++. Для работы MySQL используется язык структурированных запросов SQL.

[MySQL Workbench](https://www.mysql.com/products/workbench/) - универсальный инструмент для работы и управления базами данных. С его помощью можно создавать и редактировать таблицы и другие объекты, управлять доступом пользователей и полноценно администрировать БД. Workbench может пригодиться разработчикам, администраторам БД и архитекторам систем.

Одна из главных особенностей Workbench — визуальное моделирование. Оно позволяет увидеть все таблицы и связи между ними в наглядном виде. Это намного упрощает проектирование базы данных.

PostgreSQL — это объектно-реляционная система управления базами данных (ORDBMS), наиболее развитая из открытых [СУБД](https://blog.skillfactory.ru/glossary/subd/) в мире. Имеет открытый исходный код и является альтернативой коммерческим базам данных. СУБД позволяет гибко управлять базами данных. С ее помощью можно создавать, модифицировать или удалять записи, отправлять транзакцию — набор из нескольких последовательных запросов на особом языке запросов SQL.

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

Таблица 1 – Сравнение средств реализации базы данных

|  |  |  |  |
| --- | --- | --- | --- |
| Название СУБД | MySQL | MariaDB | PostgreSQL |
| Большое кол-во типов данных | + | + | + |
| Популярность | + | + | - |
| Отказоустойчивость | - | + | + |
| Не требует удаленного сервера | - | - | - |
| Простота использования | + | - | - |
| Портативность | - | + | - |

Таким образом, в качестве базы данных для будущего продукта была выбрана MySQL, так как она предоставляет весь необходимый функционал для разработки продукта - скорость, возможность использовать в облаке, простота использования, ведь установка не требует особых навыков, а для еще более легкой работы можно использовать дополнительное GUI. Большинство функций для настройки безопасности поддерживаются по умолчанию. MySQL имеет богатый функционал и предлагает бесплатную лицензию для работы с открытым кодом.

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

Python — это активно развивающийся скриптовый язык, который используют для решения большого объема самых разноплановых проблем и задач. Python пригодится в создании компьютерных и мобильных приложений, его применяют в работе с большим объемом информации, при разработке web-сайтов и других разнообразных проектов, используют в машинном обучении. Данный язык программирования используют крупные известные корпорации, такие как Spotify и Амазон (например, для анализа данных и создания алгоритма рекомендаций), YouTube и даже Walt Disney. Таким образом, Python нашел свое место в различных областях — с его помощью можно решить множество задач разной сложности. Синтаксис ядра языка минималистичен, за счёт чего на практике редко возникает необходимость обращаться к документации. Сам же язык известен как интерпретируемый и используется в том числе для написания скриптов. Недостатками языка являются зачастую более низкая скорость работы и более высокое потребление памяти написанных на нём программ по сравнению с аналогичным кодом, написанным на компилируемых языках, таких как C или C++. Недостатками языка являются использование языка в серверной разработке, только благодаря фреймворку, а также зачастую более низкая скорость работы и более высокое потребление памяти написанных на нём программ по сравнению с аналогичным кодом, написанным на компилируемых языках, таких как С или C++. Основные архитектурные черты — динамическая типизация, автоматическое управление памятью, полная интроспекция, механизм обработки исключений, поддержка многопоточных вычислений, высокоуровневые структуры данных. Поддерживается разбиение программ на модули, которые, в свою очередь, могут объединяться в пакеты.

PHP — это распространённый язык программирования общего назначения с открытым исходным кодом. PHP специально сконструирован для веб-разработок и его код может внедряться непосредственно в HTML.

Язык PHP обладает рядом неоспоримых преимуществ:

* Высокая скорость работы и, соответственно, общая производительность ресурсов.
* Простота освоения, простой синтаксис.
* Отличная совместимость и переносимость — php-коды работают одинаково хорошо с разными платформами.
* Набор текста кода и его редактирование можно осуществлять в любом текстовом или html-редакторе.

Для наглядности сравнения языков программирования была составлена таблица 2.

Таблица 2 – Сравнение языков программирования для разработки программного продукта

|  |  |  |
| --- | --- | --- |
| Название языка программирования | Php | Python |
| Наличие библиотек | + | + |
| Инструменты для работы с БД | + | + |
| Объектно-ориентированные возможности | + | + |
| Лёгкий понятный синтаксис | + | - |
| Более активное сообщество | + | - |
| Более лёгкая простая модульность | + | - |

Таким образом, php будет более лучшим вариантом, ведь он более компактный и простой в освоении язык. Он вобрал все лучшие особенности таких популярных языков, как C, Java и Perl, а также его сообщество более активное.

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

* Visual Studio Code
* Netbeans.
* Atom.

Visual Studio C – это продукт от компании Майкрософт. Инструмент, предназначенный для верстальщиков и разработчиков. Один из самых популярных в соответствующей нише.

Представляет собой редактор кода от Microsoft, выступающий «облегченной» интерпретацией VisualStudio. С помощью него можно не только заниматься написанием приложений. Visual Studio Code поддерживает большое количество плагинов, которые позволят «разогнать» редактор до полноценной среды программирования.

Подходит для работы на операционных системах:

* Windows;
* MacOS;
* Linux.

Распространяется на бесплатной основе, благодаря чему набрал огромную популярность. Работает одинаково хорошо как на стареньких устройствах, так и на современных компьютерах.

Visual Studio Code – это редактор, который больше подходит новичкам за счет своей первоначальной «облегченности». Данный продукт может использоваться и продвинутыми разработчиками при создании достаточно сложных кодов. К преимуществам Visual Studio Code относят следующие моменты:

* Простое освоение. Редактор оснащен интуитивно понятным интерфейсом, а также возможностью настройки.
* Небольшой вес.
* Кроссплатформенность.
* Наличие online версии. В случае ее использования для написания программного кода не придется вообще ничего устанавливать. Visual Studio Online работает через браузер.
* Поддержка разных языков. Опция активируется за счет установки плагинов.
* Гибкость.

NetBeans IDE — свободная интегрированная среда разработки приложений (IDE) на языках программирования Java, Python, PHP, JavaScript, C, C++, Ада и ряда других. Проект NetBeans IDE поддерживается и спонсируется компанией Oracle, однако разработка NetBeans ведётся независимым сообществом разработчиков-энтузиастов (NetBeans Community) и компанией NetBeans Org. Последние версии NetBeans IDE поддерживают рефакторинг, профилирование, выделение синтаксических конструкций цветом, автодополнение набираемых конструкций на лету и множество предопределённых шаблонов кода.

Atom (в прошлом Atomicity) — бесплатный текстовый редактор с открытым исходным кодом для Linux, macOS, Windows с поддержкой плагинов, написанных на JavaScript, и встраиваемых под управлением Git. Большинство плагинов имеют статус свободного программного обеспечения, разрабатываются и поддерживаются сообществом.

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

Таблица 3 – Сравнение IDE для разработки программного продукта

|  |  |  |  |
| --- | --- | --- | --- |
| Название IDE | Visual Studio Code | NetBeans | Atom |
| Распространяется бесплатно | +  - | + | + |
| Автоматическое сохранение | + | - | + |
| Подсказки по коду | + | + | + |
| Интеграция с системой контроля версия (GIT) | + | + | - |
| Возможность расширения функционала библиотеками | + | + | + |
| Заточен под PHP-разработку | + | + | + |
| Поддержка CSS/HTML/JS | + | + | + |
| Комфортное использование на слабых ПК | + | - | + |

Таким образом, после рассмотрения вариантов средств разработок, было принято решение использовать Visual Studio Code. Visual Studio Code намного функциональнее, чем приведённые выше Atom и NetBeans. В Visual Studio Code имеется автоматическое сохранение, которое не позволит случайно потерять все наработки, а также он более удобен в работе.

1.2 Обоснование выбора

Для создания программного продукта было решено использовать

средства:

1) Для создания структурных схем, контекстной и диаграмм декомпозиции использовались CASE-средства – Draw.io.

2) Для наглядного составления структуры базы данных использовался инструмент для визуального проектирования баз данных, их редактирования и полного администрирования – MySQL Workbench.

3) Для разработки дизайна web-приложения использовался онлайн-сервис для разработки дизайна и прототипа сайта – Figma.

4) На этапе разработки программного продукта использовались инструменты и среды для разработки: PHP, Laravel, HTML5, CSS3, JavaScript и AJAX, а также редактор кода Visual Studio Code.

Основные преимущества веб-технологий:

1. Распределенность (пользователь может работать с системой из любого места, связанного с WEB-сервером по сети, находясь в любой точке земного шара);
2. Переносимость (Web-клиенты (браузеры) существуют для любых  
   платформ, от настольных компьютеров до сотовых телефонов. Web-сервера используются для большинства платформ, а Web-приложения обычно пишутся на переносимых языках);
3. Привычность интерфейса (почти каждый пользователь компьютера хотя бы раз запускал браузер и работал в нем);
4. Простота установки и обслуживания (новую версию web-приложения не надо устанавливать на все компьютеры – достаточно установить на сервер);
5. Простота интерфейса - Пользователи не любят гигантских окошек с сотнями полей ввода, а программисты и разработчики интерфейсов почему-то любят. Web как раз не поощряет сложный интерфейс, скорее поощряет простой;
6. Простота программирования интерфейса - Создавать HTML из  
   шаблонов куда проще и удобнее, чем создавать графические  
   приложения визуальными редакторами интерфейсов;
7. Простота программирования в целом - 3х-уровневая архитектура:  
   база данных – логика приложения (сервер) – логика представления  
   (клиент, браузер).
8. Техническое задание

Техническое задание, или ТЗ — это документ, в котором фиксируются требования к проекту. Условно ТЗ можно назвать любое поручение исполнителю, главное, чтобы в нем были ясно прописаны характеристики итогового продукта.

Для создания технического задания использовался стандарт ГОСТ 19 201-78.

Согласно ГОСТ 19 201-78 техническое задание должно включать следующие разделы:

Введение.

1. Основание для разработки.
2. Назначение разработки.
3. Требования к программе или программному изделию.
   1. Требования к функциональным характеристикам.
   2. Требования к надежности.
   3. Требования к информационной и программной совместимости.
4. Требования к программной документации.
5. Состав и содержание работ по созданию системы.

ТЗ представлено в Приложении А

1. Проектирование ИС
   1. Структурная схема ИС

Проектирование информационной системы началось с построения диаграммы вариантов использования. На рисунке 1 представлена диаграмма прецедентов Uses CASE. Она содержит 2 актёра, которые могут выполнять суммарно 9 функций, часть из которых может делать только один актёр, а часть – только другой актёр.

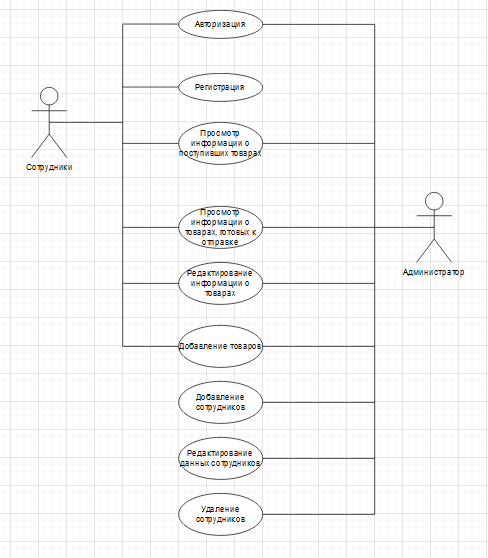


Рисунок 1 - Диаграмма прецедентов

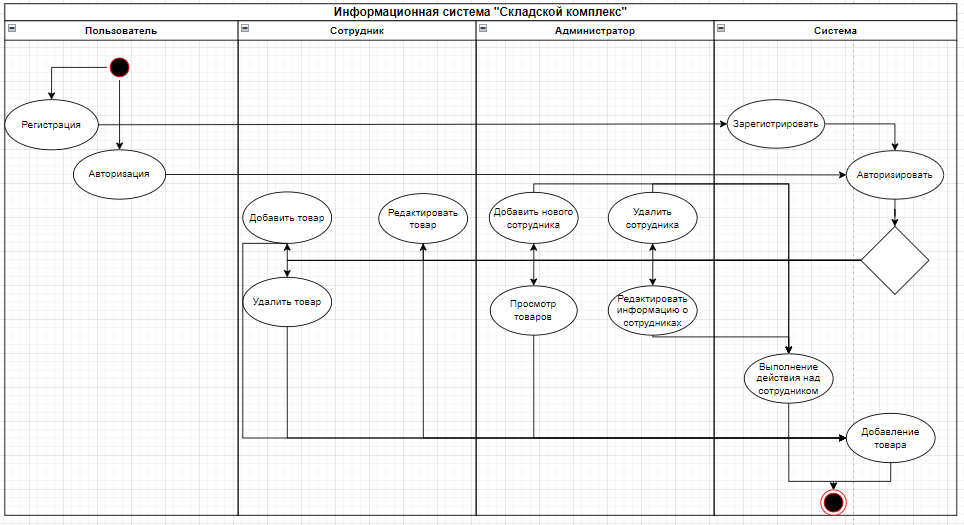


Рисунок 2 – Диаграмма деятельности

На рисунке 2 представлена диаграмма деятельности. Она содержит 4 роли. Процесс начинается с пользователя, который должен зарегистрироваться в системе, чтобы в дальнейшем использовать её или же просто авторизоваться в ней. Права доступа роли «Сотрудник» позволяют добавлять, изменять или же удалять товары, находящиеся на складе. Права доступа роли «Администратор» позволяют добавлять, удалять и редактировать информацию о сотрудниках, которые пользуются данной системой. Также «Администратор» может просматривать товары, находящиеся на складе.

Информационная система обрабатывает данные, а затем сохраняет их в БД.

На рисунке 3 представлена диаграмма развёртывания. Она показывает, что, чтобы пользоваться программным продуктом, необходим web-сервер, на котором размещаются ИС и БД, а также для администратора и сотрудников необходимо каждому по телефону с браузером и выходом в интернет.

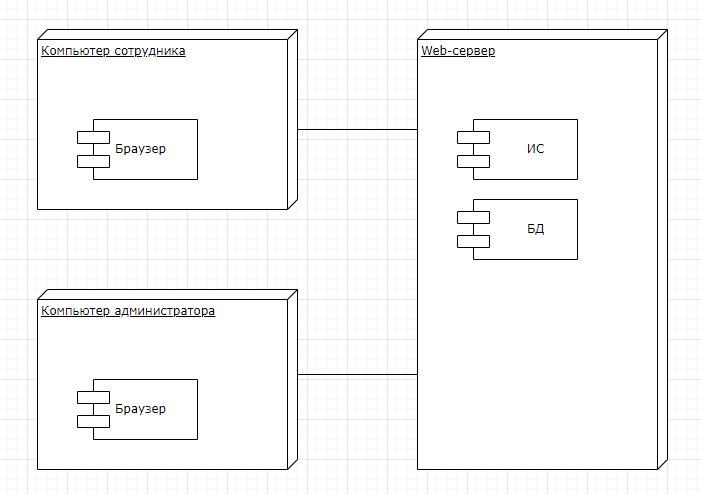


Рисунок 3 – Диаграмма развёртывания

* 1. Функциональная схема ИС

На рисунке 5 представлена контекстная диаграмма, отображающая деятельность ИС.

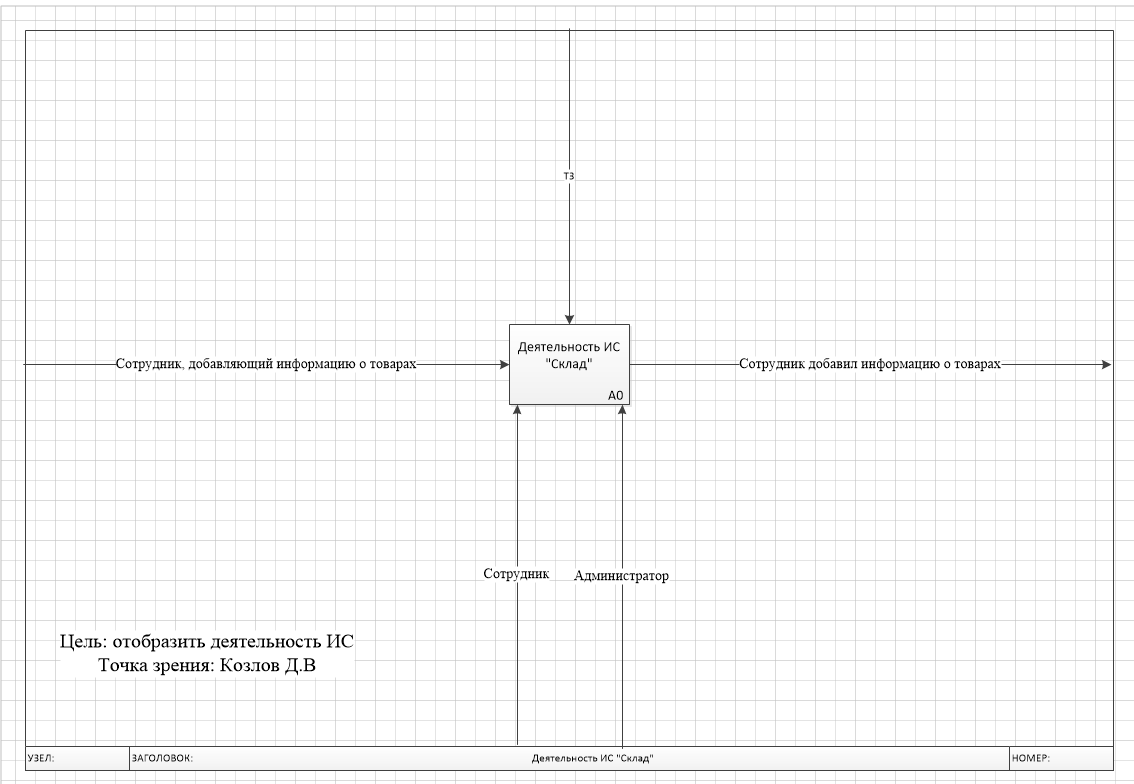


Рисунок 5 – Контекстная диаграмма IDEF0

На рисунке 6 представлена диаграмма декомпозиций (А1), отображающая деятельность ИС более подробно предыдущей.

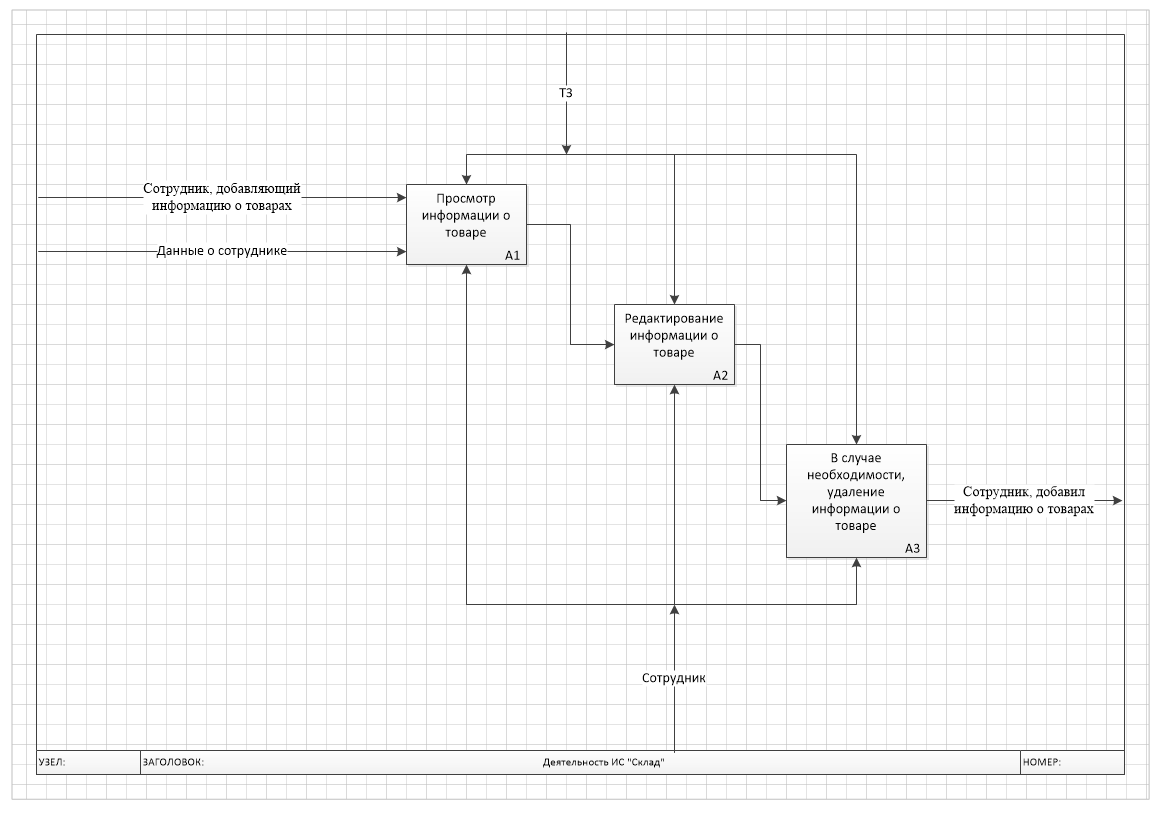


Рисунок 6 – Диаграмма декомпозиций А1

На рисунке 7 представлена диаграмма классов. Она содержит 15 классов, среди них база данных, html-страница, информационная система «Складской комплекс», пользователь, сотрудник, администратор, прибывший товар, товар готовый к отгрузке, товар.

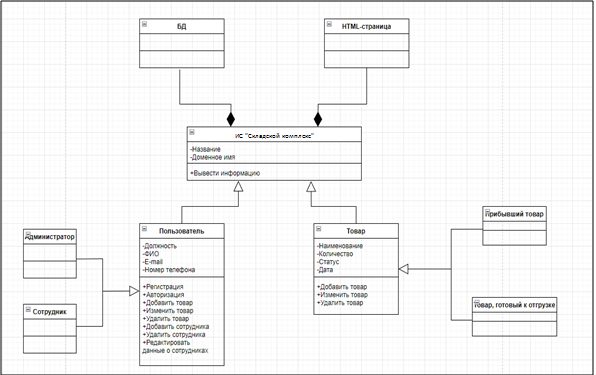


Рисунок 7 – Диаграмма классов

На рисунке 8 представлена диаграмма потоков данных. В центре всего сотрудник, который может взаимодействовать с товарами, для соответствующих действий в ИС предусмотрены соответствующие БД.

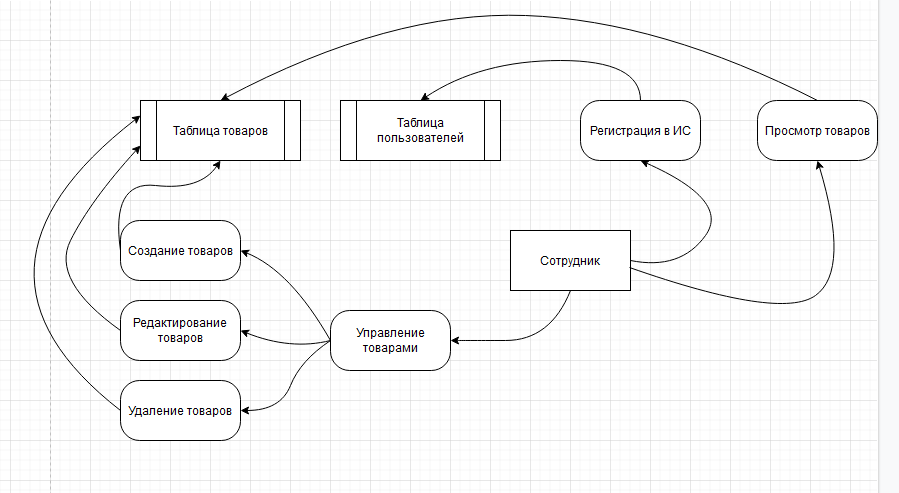


Рисунок 8 - Диаграмма потоков данных

* 1. Проектирование базы данных

Проектирование базы данных начинается с концептуального проектирование базы данных.

Концептуальное проектирование – построение семантической модели предметной области, то есть информационной модели наиболее высокого уровня абстракции. Такая модель создаётся без ориентации на какую-либо конкретную СУБД и модель данных.

На рисунке 9 представлена инфологическая модель базы данных.

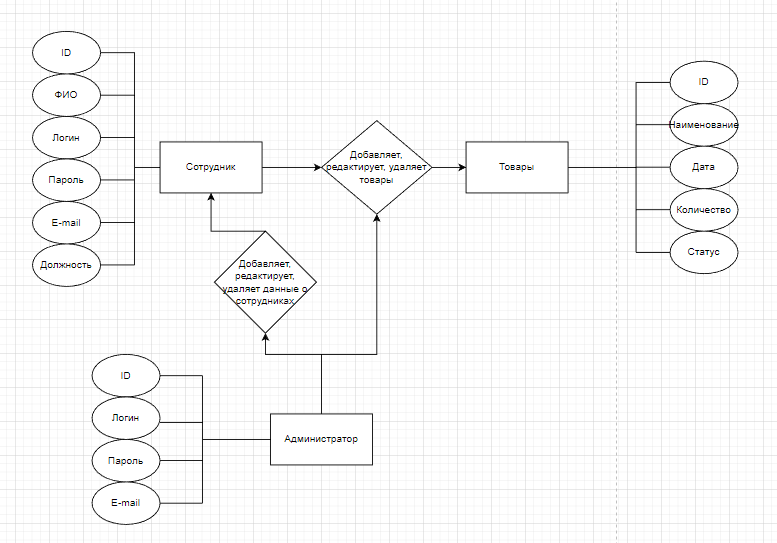


Рисунок 9 – Инфологическая модель базы данных

На инфологической модели базы данных выше схематично отображены сущности системы, их атрибуты и связи между ними. Так, в прямоугольник отображены сущности, такие как: Сотрудник, Товары, Администратор. В овалах отображены атрибуты сущностей, например: ФИО, E-mail, ID и другие. Ромбами изображены связи между сущностями, например сотрудник добавляет товары.

Далее происходит преобразование концептуальной модели в логическую модель, по формальным правилам. Таким образом, логическое (даталогическое) проектирование — создание схемы базы данных на основе конкретной модели данных, например, реляционной модели данных.

На даталогической модели базы данных (рисунок 10) отображены сущности приложения, а также первичные и внешние ключи, связывающие сущности между собой.

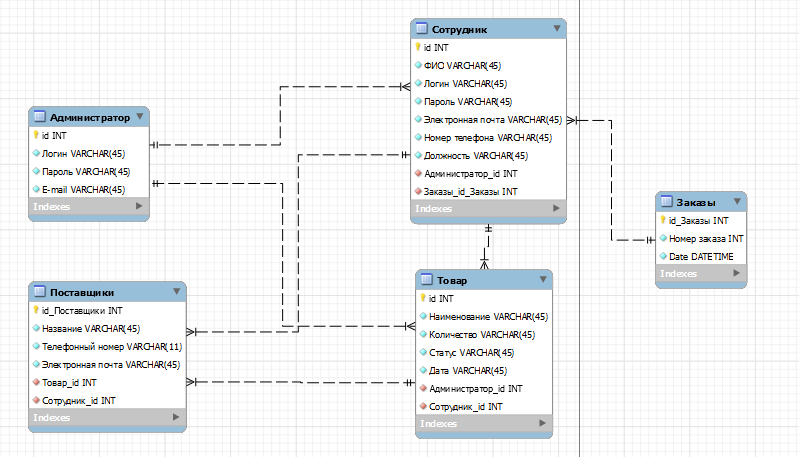


Рисунок 10 – Даталогическая модель базы данных

На рисунке 11 представлена ER-диаграмма базы данных. Она содержит 6 таблиц для полного функционирования и качественной сортировки информации.

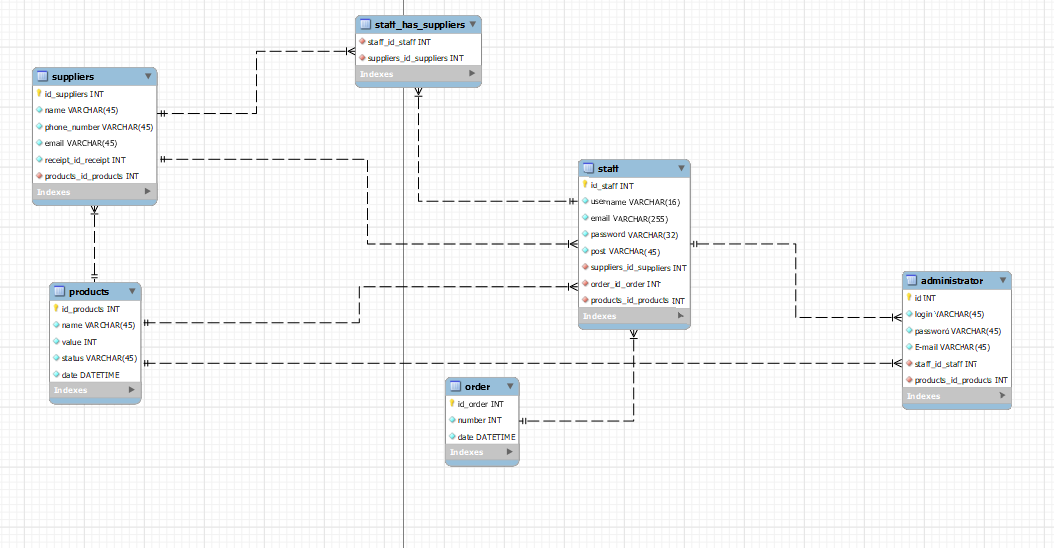


Рисунок 11 – ER-модель базы данных

Таблица 1 – Таблица staff

|  |  |  |
| --- | --- | --- |
| Таблица | Тип данных | Описание |
| Id | INT | ID |
| Login | VARCHAR(16) | Логин |
| Email | VARCHAR(255) | Электронная почта |
| Password | VARCHAR(32) | Пароль |
| Post | VARCHAR(45) | Должность |
| suppliers\_id\_suppliers | INT | Вторичный ключ |
| order\_id\_order | INT | Вторичный ключ |
| products\_id\_products | INT | Вторичный ключ |

Таблица 2 – Таблица administrator

|  |  |  |
| --- | --- | --- |
| Таблица | Тип данных | Описание |
| Id | INT | ID |
| Login | VARCHAR(45) | Логин |
| Email | VARCHAR(255) | Электронная почта |
| Password | VARCHAR(45) | Пароль |
| staff\_id\_staff | INT | Вторичный ключ |
| products\_id\_products | INT | Вторичный ключ |

Таблица 3 – Таблица order

|  |  |  |
| --- | --- | --- |
| Таблица | Тип данных | Описание |
| Id | INT | ID |
| Number | INT | Номер заказа |
| Date | DATETIME | Дата |
| staff\_id\_staff | INT | Вторичный ключ |
| products\_id\_products | INT | Вторичный ключ |

Таблица 4 – Таблица products

|  |  |  |
| --- | --- | --- |
| Таблица | Тип данных | Описание |
| Id | INT | ID |
| Name | VARCHAR(45) | Наименование |
| Value | INT | Количество |
| Status | VARCHAR(45) | Статус товара |
| Date | DATETIME | Дата |

Таблица 5 – Таблица suppliers

|  |  |  |
| --- | --- | --- |
| Таблица | Тип данных | Описание |
| Id | INT | ID |
| Name | VARCHAR(45) | Наименование |
| phone\_number | INT | Телефонный номер |
| Email | VARCHAR(45) | Электронная почта |
| receipt\_id\_receipt | INT | Вторичный ключ |
| products\_id\_products | INT | Вторичный ключ |

Таблица 6 – Таблица staff\_has\_suppliers

|  |  |  |
| --- | --- | --- |
| Таблица | Тип данных | Описание |
| staff\_id\_staff | INT | Вторичный ключ |
| suppliers\_id\_suppliers | INT | Вторичный ключ |

**4.4. Проектирование интерфейса**

Для разработки пользовательского интерфейса был выбран инструмент Figma — онлайн-сервис для разработки интерфейсов и прототипирования с возможностью организации совместной работы в режиме реального времени.

В результате проектирование интерфейса будущей информационной системы были спроектированы прототипы трёх страниц: главная страница (рисунок 12), страница авторизации (рисунок 13), страница просмотра списка товаров (рисунок 14). Благодаря созданию прототипов, разработка непосредственно программного продукта будет значительно облегчена, за счёт наглядных примеров будущих страниц ИС.

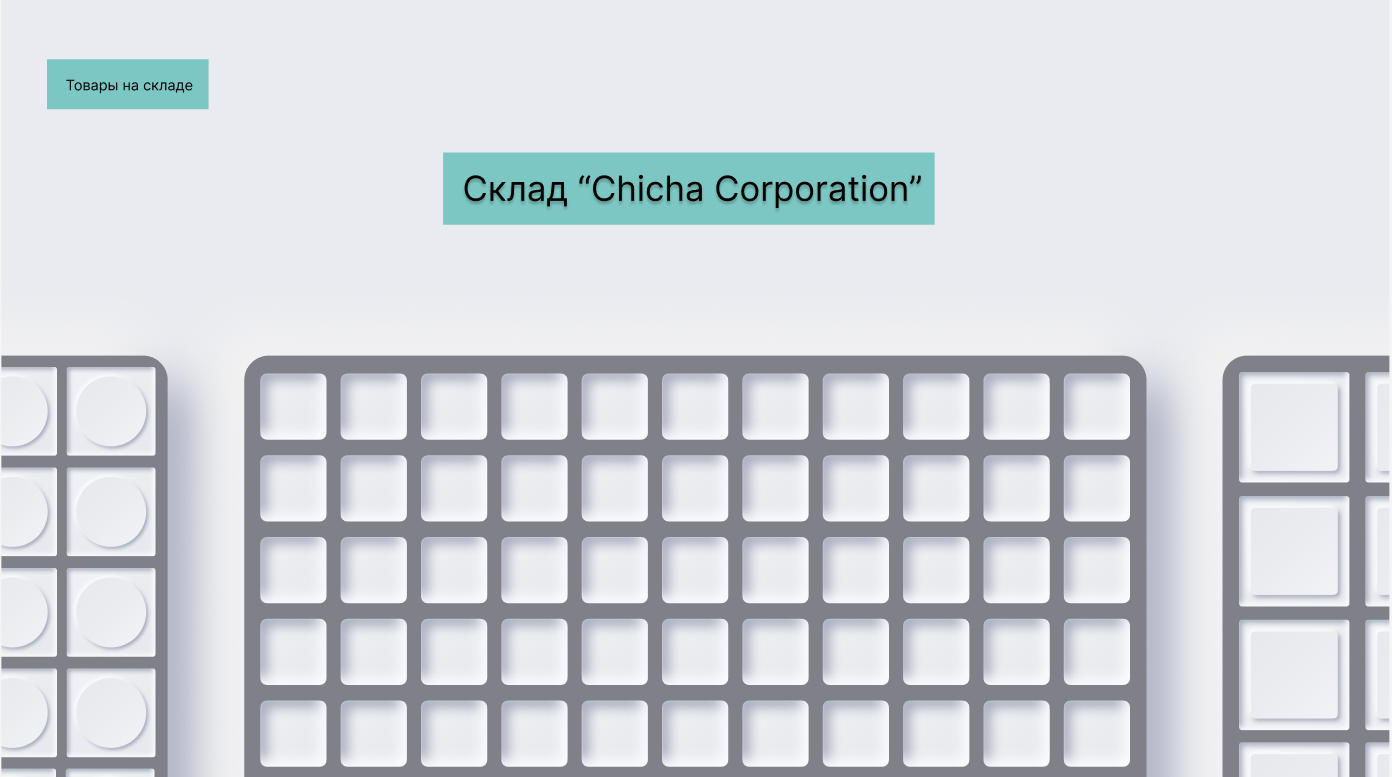


Рисунок 12 – Главная страница



Рисунок 13 – Страница авторизации

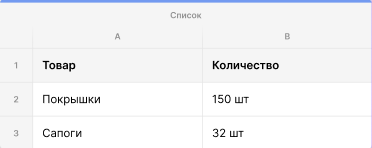
****

Рисунок 14 – Страница со списком товаров

1. Разработка ИС
   1. Разработка интерфейса ИС

В информационной системе «Склад» были поставлены и выполнены следующие задачи:

для сотрудников:

* авторизация;
* добавление товаров;
* удаление товаров;
* редактирование товаров.

для администратора:

* создание сотрудников;
* удаление сотрудников;
* редактирование сотрудников;
* добавление новых товаров;
* добавление товаров;
* удаление товаров;
* редактирование товаров.

Для серверной операционной системы, на которой будет располагаться программный код ИС, выдвигаются следующие минимальные требования:

* процессор: Intel Pentium 4 или более поздней версии;
* ОЗУ 512 Мб DDR3;
* 15 ГБ на HDD;
* поддержка БД MySQL;
* операционная система Windows;
* доступ к сети Интернет.

Требования к клиентской части ИС, необходимые для пользования информационной системой:

* процессор Intel Pentium 4 или более поздней версии;
* ОЗУ 2 ГБ.
* доступ к сети Интернет;
* HDD 1ГБ;
* Браузер;
* встроенный видеоадаптер;
* клавиатура, мышь;
* операционная система: Windows.

Общий размер дисковой памяти, занимаемой информационной системой, составляет 2,60 МБ (рисунок 15).



Рисунок 16 – Объем дисковой памяти ИС

Объем потребляемой ОЗУ составляет 30 668К (что примерно равно 30Мб) на одну вкладку в браузере Яндекс (рисунок 15).



Рисунок 16 – Потребляемая ОЗУ в Яндекс браузере

На рисунках 17, 18 показан html код для основной страницы, которую видят все пользователи не зависимо от того, зарегистрированы ли они или нет, а также прошли ли авторизацию и результат кода.



Рисунок 17 – Часть HTML кода основной страницы

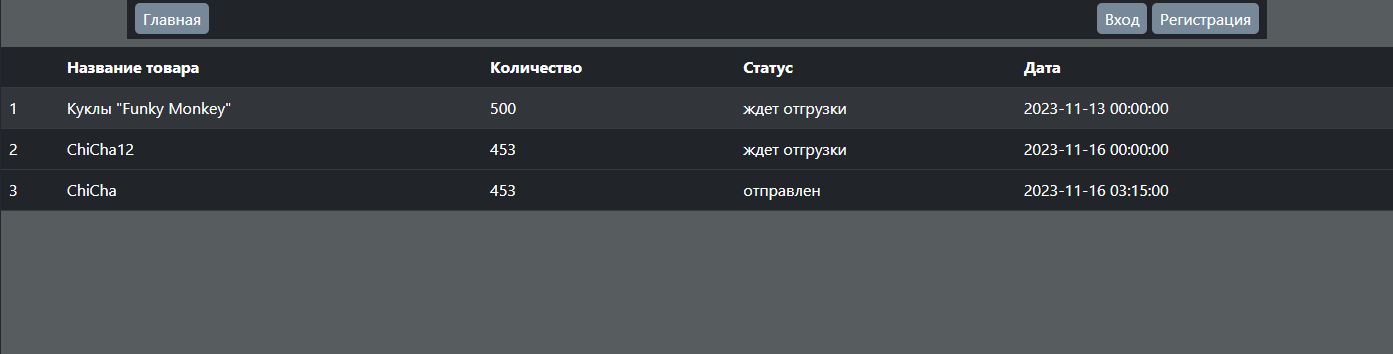


Рисунок 18 – Результат HTML кода основной страницы

На рисунках 19, 20 показан html код для регистрации новых сотрудников, для последующего вноса в базу данных и дальнейшего прохождения авторизации, зарегистрированы ли они или нет, а также прошли ли авторизацию и результат кода.



Рисунок 19 – Часть HTML кода страницы регистрации

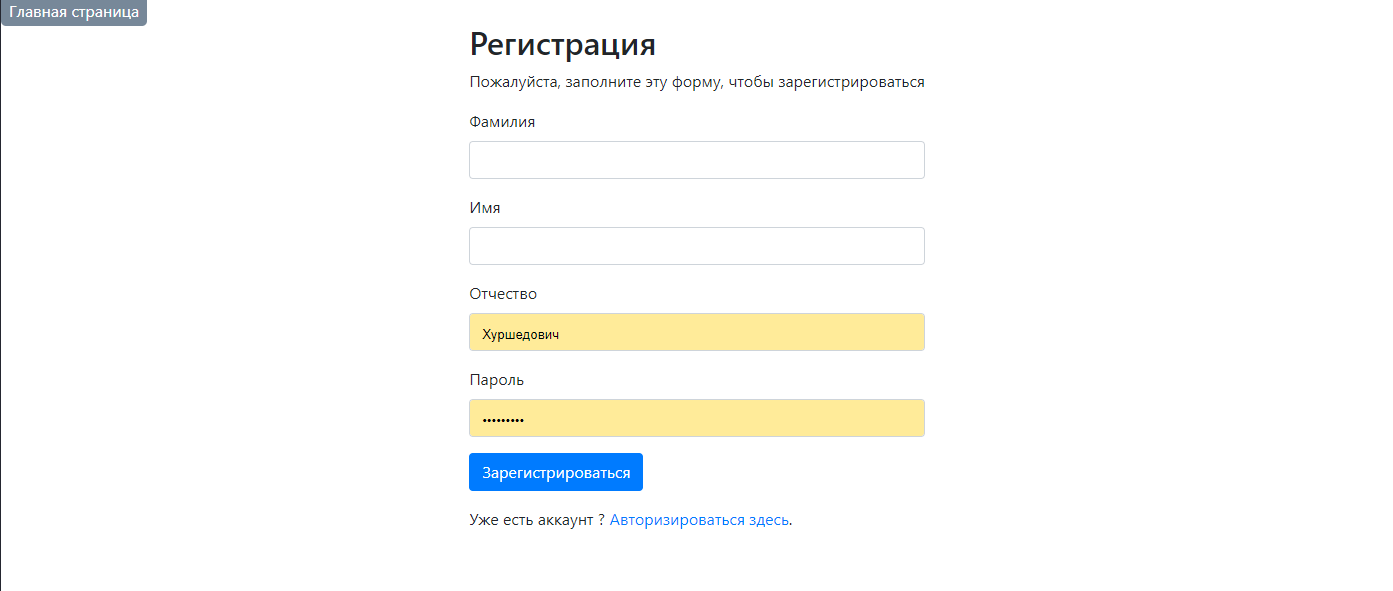


Рисунок 20 – Результат html кода страницы регистрации

На рисунках 21, 22 изображен html код страницы авторизации пользователя, для последующего использования информационной системы, и результат кода.



Рисунок 21 – Часть HTML кода страницы авторизации

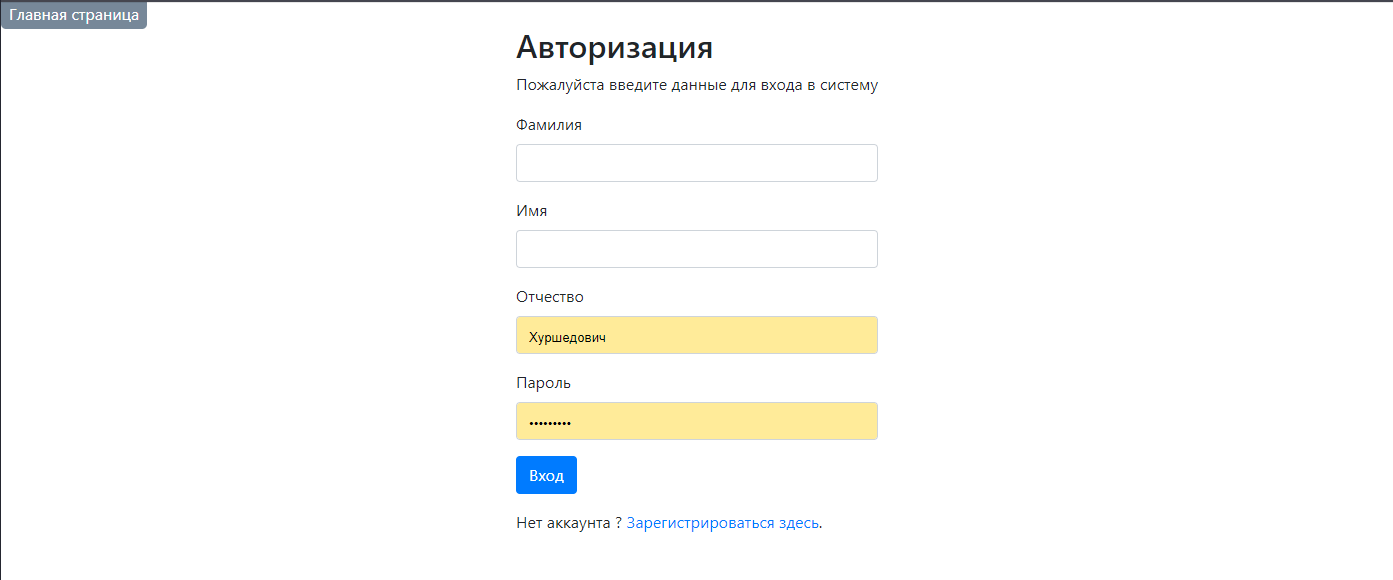


Рисунок 22 – Результат html кода страницы авторизации

Далее было создано 2 личных кабинета общего вида: для администратора и сотрудника. На рисунках 23, 24 изображен html код личного кабинета администратора, с помощью которого администратор может выполнять все действия в ИС.



Рисунок 23 – Часть html код личного кабинета администратора

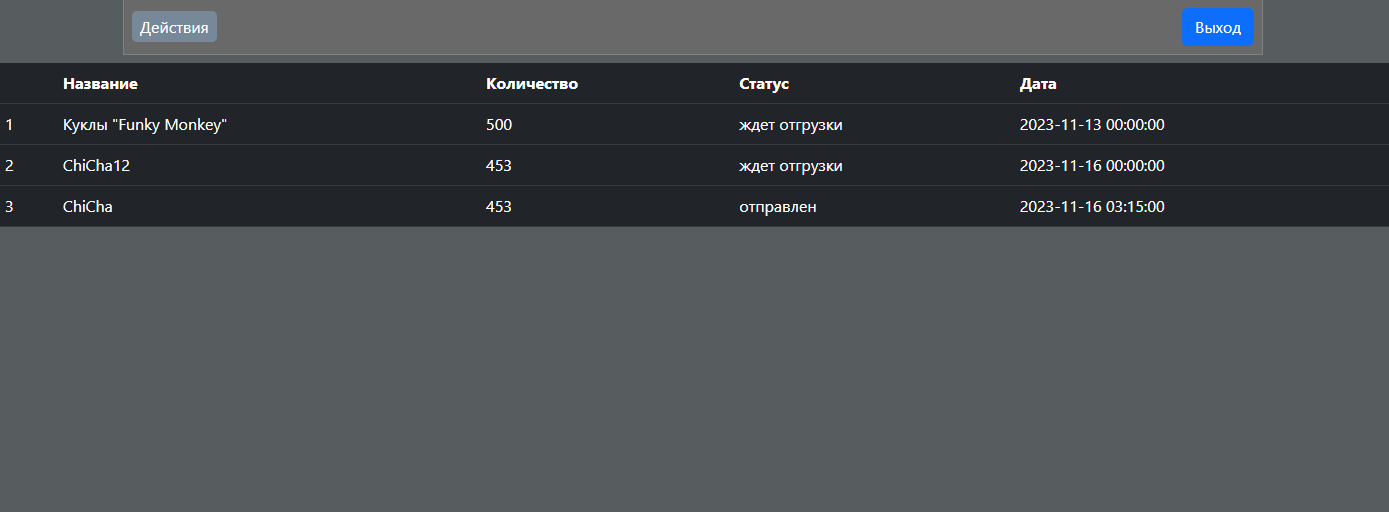


Рисунок 24 – Итоговой вид личного кабинета администратора

На рисунках 25, 26 изображен html код личного кабинета сотрудника, с помощью которого сотрудник может выполнять действия над товарами, прибывшими на склад.



Рисунок 25 – Часть html кода личного кабинета сотрудника

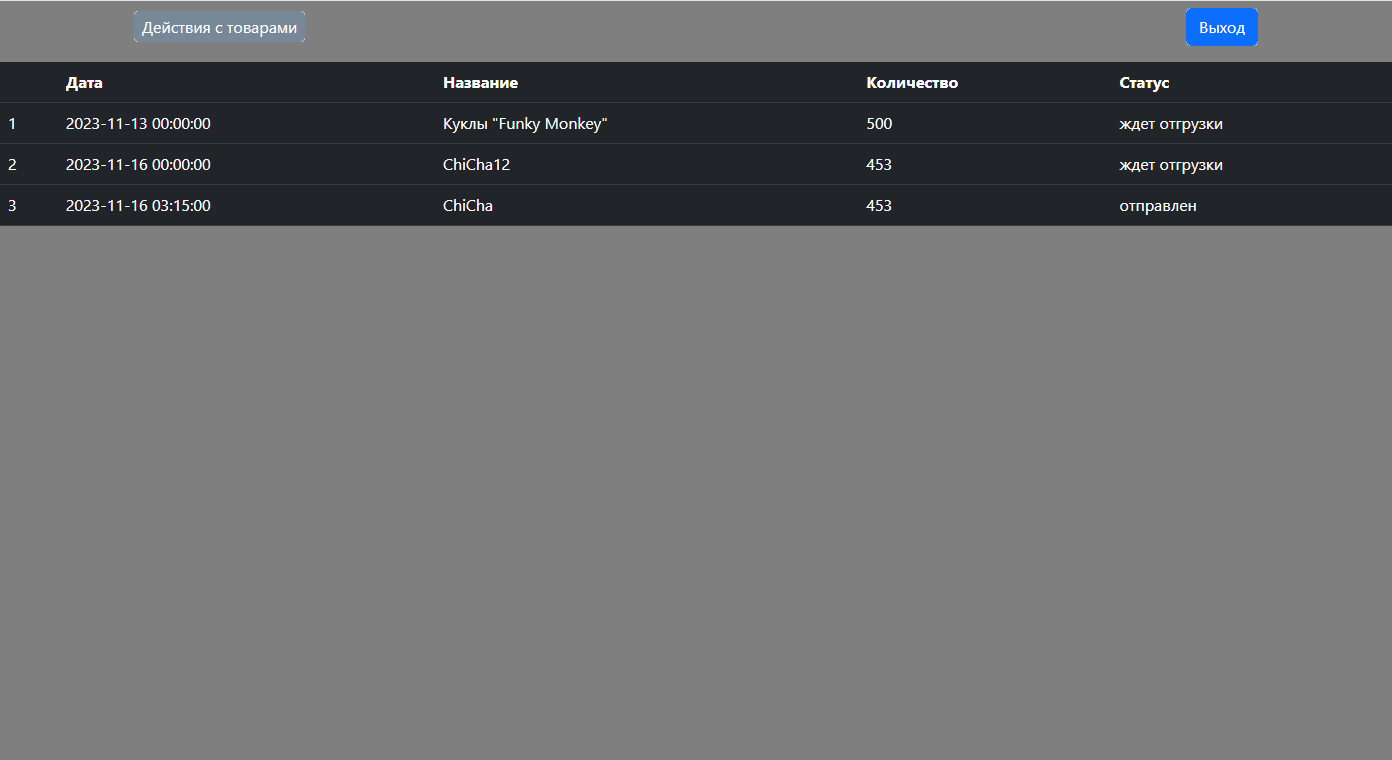


Рисунок 26 – Результат html кода личного кабинета сотрудника

* 1. Разработка базы данных

После того, как разработан интерфейс, необходимо где-то хранить данные, которые будут заноситься пользователями ИС. Для этого необходимо разработать базу данных (далее – БД).

Разрабатывалась БД информационной системы «Склад» во встроенном менеджере баз данных PHPStorm.

БД состоит из 6 таблиц: users, suppliers, staff, products, order, administrator

Структура таблиц соответствует ER-модели из рисунка 11 и таблицам 4-8.

Таблица users на рисунке 27.

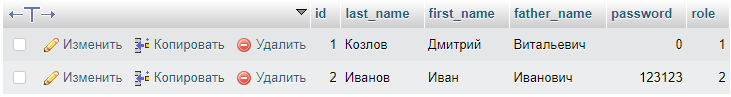


Рисунок 27 – Таблица users

Таблица staff на рисунке 28.



Рисунок 28 – Таблица staff

Таблица products на рисунке 29.

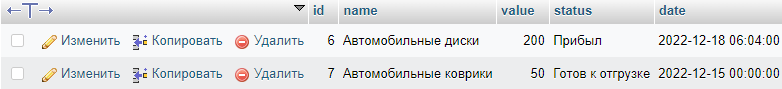


Рисунок 29 – Таблица products

Таблица order на рисунке 30.



Рисунок 30 – Таблица order

Таблица suppliers на рисунке 31.



Рисунок 31 – Таблица suppliers

Таблица administrator на рисунке 32.



Рисунок 32 – Таблица administrator

* 1. Разработка ИС

Для того, чтобы ИС работала, был создан файл db.php (рисунок 34), в котором осуществляется подключение к БД через PDO (PHP Data Object). В подключении указывается адрес хостинга, а также название БД, кодировка текста, имя пользователя и пароль (который по умолчанию является пустым).



Рисунок 34 – Подключение к базе данных

Затем были реализованы функции авторизации, регистрации и выхода из учётной записи с помощью моделей auth.php, registration.php и logout.php (рисунки 35, 36, 37). Также были созданы представления авторизации (auth.php) и регистрации (registration.php). (рисунок 38, 39) Полная версия программного кода контроллера находится в приложении Б.



Рисунок 35 – Модель auth.php



Рисунок 36 – Модель registration.php

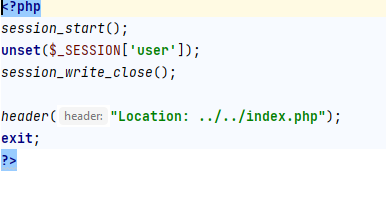


Рисунок 37 – Модель logout.php

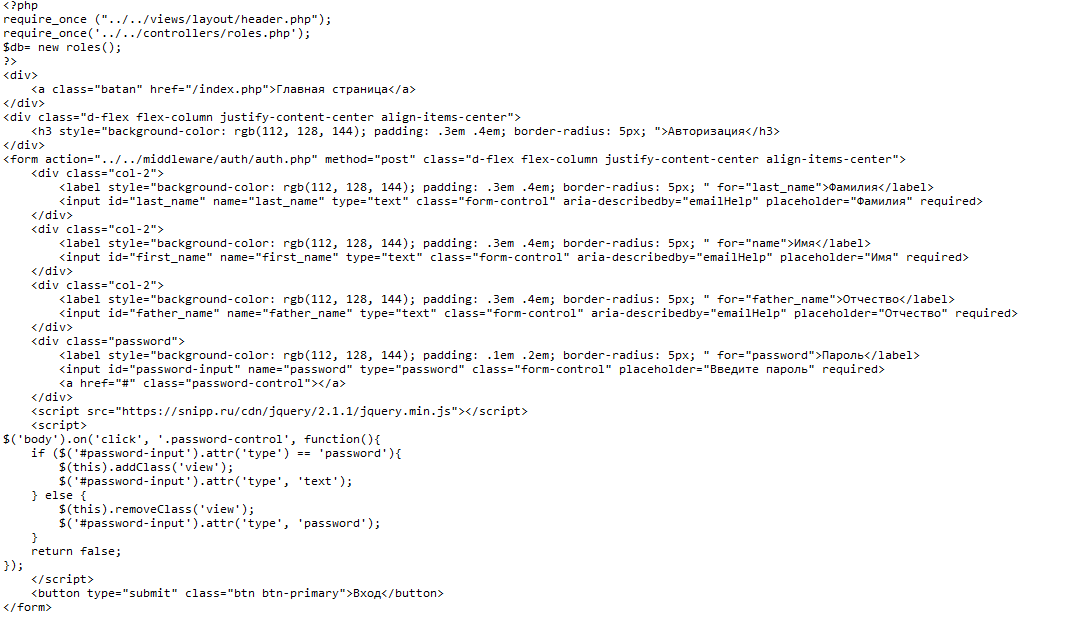


Рисунок 38 – Представление auth.php

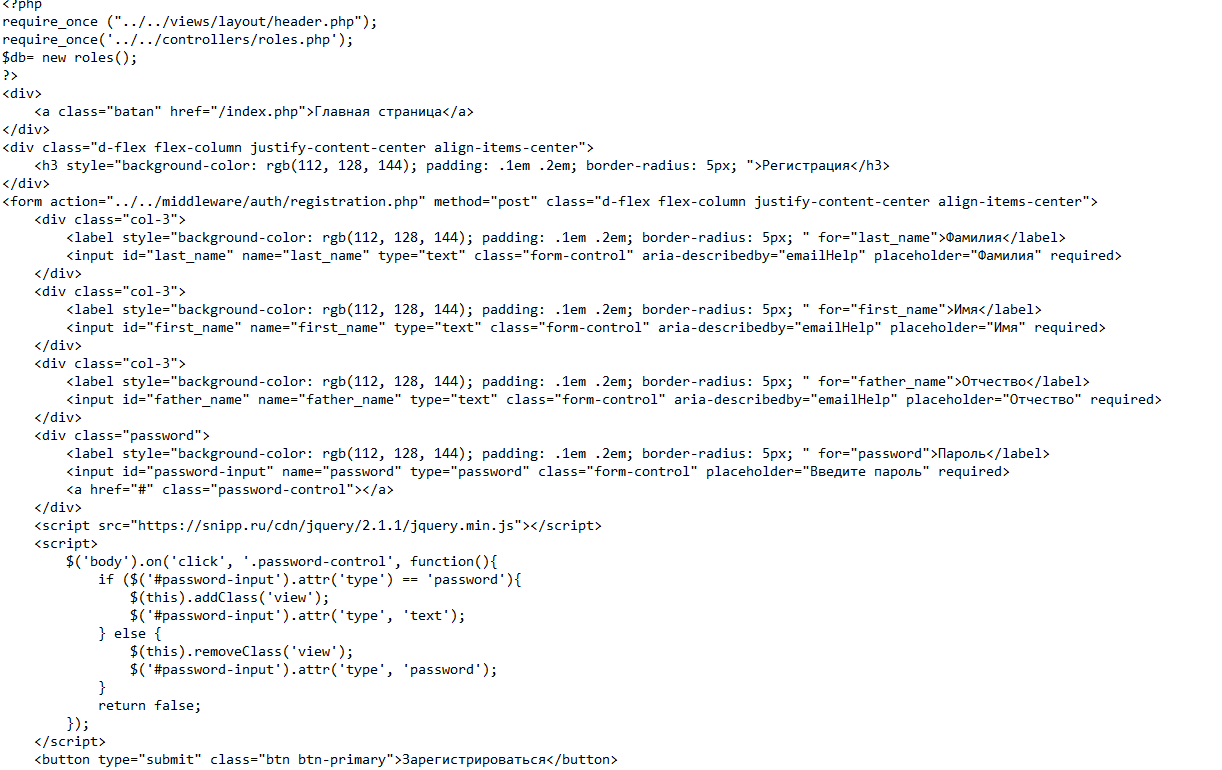


Рисунок 39 – Представление registration.php

* 1. Тестирование ИС

Тестирование и отладка являются двумя важными процессами в разработке программного продукта.

Тестирование — это процесс проверки программного продукта на соответствие требованиям и ожиданиям пользователя. Оно включает в себя различные виды тестов, такие как модульные тесты, интеграционные тесты, системные тесты и пользовательские тесты. Цель тестирования заключается в выявлении ошибок, дефектов и несоответствий, а также в проверке функциональности, производительности и безопасности продукта.

Отладка — это процесс идентификации и исправления ошибок в программном продукте. Она включает в себя анализ кода, выполнение шагов программы для идентификации проблемного участка, использование инструментов для отслеживания ошибок и исправления дефектов. Основная цель отладки состоит в обнаружении и исправлении ошибок, которые могут привести к неправильной работе программы или сбоям.

Тестирование и отладка являются неотъемлемой частью разработки программного продукта и имеют важное значение. Они позволяют обнаруживать и устранять ошибки и дефекты еще на ранних этапах разработки, что ведет к повышению качества и надежности программного продукта. Эти процессы также помогают улучшить производительность и безопасность программы, а также повысить удовлетворенность пользователей. Без должного тестирования и отладки программа может содержать ошибки, которые могут привести к непредсказуемым результатам, нарушению функциональности и негативному впечатлению пользователей.

Таблица 7 – Сценарий тестирования для сотрудника 1

|  |  |  |  |
| --- | --- | --- | --- |
| **Поле** | | **Описание** | |
| Дата(ы) теста | | 5 октября 2023 | |
| Приоритет  тестирования  (Низкий/Средний/  Высокий) | | Высокий | |
| Заголовок/название теста | | Тестовый сценарий 1: переход во вкладку со всеми товарами | |
| Этапы теста | | 1. Сотрудник входит в систему, используя свои учетные данные 2. Сотрудник попадает в свой личный кабинет, где переходит в интерфейс просмотра всех товаров 3. Сотрудник просматривает список всех товаров, которые отображаются из базы данных | |
| Тестовые данные | | Данные о товарах (ID, наименование товара, время прибытия, статус) | |
| Ожидаемый  результат | Сотрудник успешно просматривает список всех товаров, отображаемых в интерфейсе, включая информацию о наименовании товара, его количестве, времени прибытия, а также статусах товаров | |
| Фактический  результат | Сотрудник без проблем просматривает список всех товаров, которые выводятся напрямую из базы данных. | |

Таблица 8 – Сценарий тестирования для сотрудника 2

|  |  |
| --- | --- |
| **Поле** | **Описание** |
| Дата(ы) теста | 5 октября 2023 |
| Приоритет  тестирования  (Низкий/Средний/  Высокий) | Высокий |
| Заголовок/название теста | Тестовый сценарий 2: добавление новых товаров, используя внутренний интерфейс информационной системы |
| Этапы теста | 1. Сотрудник входит в систему, используя свои учетные данные 2. Сотрудник попадает в свой личный кабинет, где переходит в интерфейс просмотра всех товаров 3. Сотрудник нажимает на кнопку добавления нового товара 4. Сотрудник переходит в интерфейс добавления нового товара, где вводит необходимые данные в поля 5. Сотрудник добавляет новый товар |
| Тестовые данные | Данные о товарах (ID, наименование товара, время прибытия, статус) |
| Ожидаемый  результат | Сотрудник успешно просматривает список всех товаров, отображаемых в интерфейсе, включая информацию о наименовании товара, его количестве, времени прибытия, а также статусах товаров |
| Фактический  результат | Сотрудник без проблем просматривает список всех товаров, которые выводятся напрямую из базы данных. |

Таблица 9 – Сценарий тестирования для администратора 1

|  |  |
| --- | --- |
| **Поле** | **Описание** |
| Дата(ы) теста | 6 октября 2023 |
| Приоритет  тестирования  (Низкий/Средний/  Высокий) | Высокий |
| Заголовок/название теста | Тестовый сценарий 1: Регистрация нового пользователя администратором |
| Этапы теста | 1. Администратор входит в систему, используя свои учетные данные. 2. Администратор переходит в окно добавления сотрудника. 3. Администратор нажимает кнопку "Добавить нового сотрудника". 4. Администратор вводит данные нового пользователя: имя, фамилия, логин, пароль 5. Администратор подтверждает создание пользователя. 6. Система сохраняет нового пользователя в базе данных. |
| Тестовые данные | Имя: Руслан  Фамилия: Шаинов  Логин: RuslanCHES  Пароль:112233 |
| Ожидаемый  результат | Система успешно создает нового пользователя, пользователь отображается в списке существующих сотрудников. |
| Фактический  результат | Новый пользователь добавлен и отображается в системе |

Таблица 10 – Сценарий тестирования для администратора 2

|  |  |
| --- | --- |
| **Поле** | **Описание** |
| Дата(ы) теста | 6 октября 2023 |
| Приоритет  тестирования  (Низкий/Средний/  Высокий) | Средний |
| Заголовок/название теста | Тестовый сценарий 2: Просмотр всех товаров администратором |
| Этапы теста | 1. Администратор входит в систему, используя свои учетные данные. 2. Администратор переходит в интерфейс просмотра всех товаров. 3. Администратор просматривает список всех товаров. |
| Тестовые данные | Данные о товарах (ID, наименование товара, время прибытия, статус) |
| Ожидаемый  результат | Администратор успешно просматривает список всех товаров, включая всю информацию, указанную выше |
| Фактический  результат | Администратор успешно просматривает список всех товаров. |

Был создан чек-лист, представленный в таблице 11.

Таблица 11- Чек лист тестирования

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Тест | Вводные данные | Ожидаемый результат | Фактический результат | Результат тестирования | Комментарий |
| Авторизация | Данные пользователя: логин, пароль | Пользователь вошел в систему | Пользователь авторизирован | Неуспешно | - |
| Регистрация | Данные пользователя: логин, пароль | Данные добавляются в систему | Данные добавлены | Неуспешно | - |
| Добавление товаров | Данные товаров: id, статус, наименование, количество, время | Данные о новых товарах добавляются в базу данных | Данные сохранены | Неуспешно | - |
| Удаление товаров | Данные товаров: id, статус, наименование, количество, время | Данные о товарах удаляются из системы | Данные удалены | Неуспешно | - |
| Редактирование товаров | Данные товаров: id, статус, наименование, количество, время | Данные товаров успешно изменяются | Изменения в данных товарах успешно внеслись | Неуспешно | - |

1. Технологическая документация ИС
   1. Руководство пользователя ИС

Lля того, чтобы войти в веб-сервер нужно запустить файл start.bat, скопировать ссылку http://localhost:8080, и вставить её в поисковую строку браузера. После этого откроется главная страница системы.

В данном руководстве пользователя рассмотрен пример личного кабинета администратора. На рисунке 27 можно видеть список товаров, поступивших на склад, а также 2 кнопки: «Выход» и «Действия с товарами» для взаимодействия с товарами.

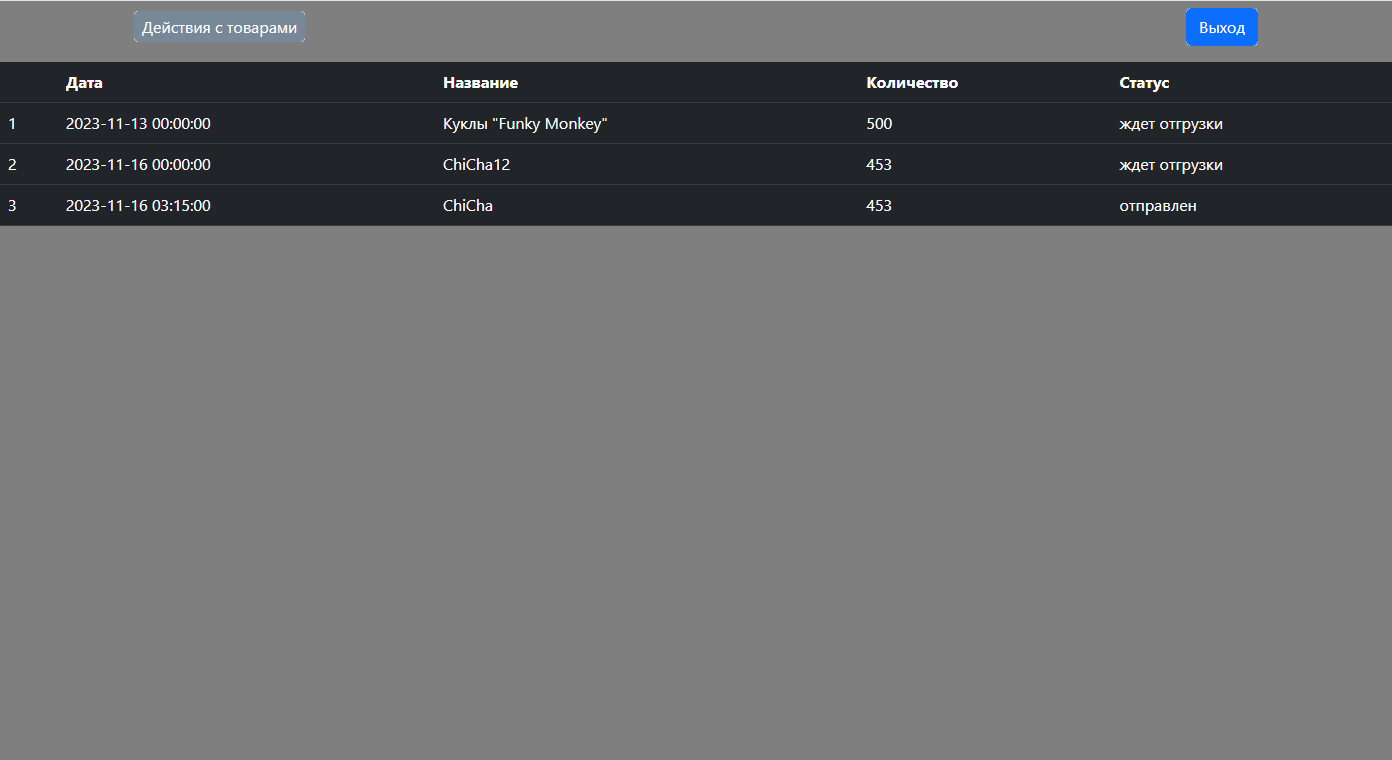


Рисунок 27 – Личный кабинет пользователя

При нажатии на кнопку «Действия с товарами» открывается страница, на которой можно выбрать либо «Удаление товара», либо «Добавление товара», при нажатии на «Добавление товара» открывается соответствующее меню. Это можно увидеть на рисунке 28.

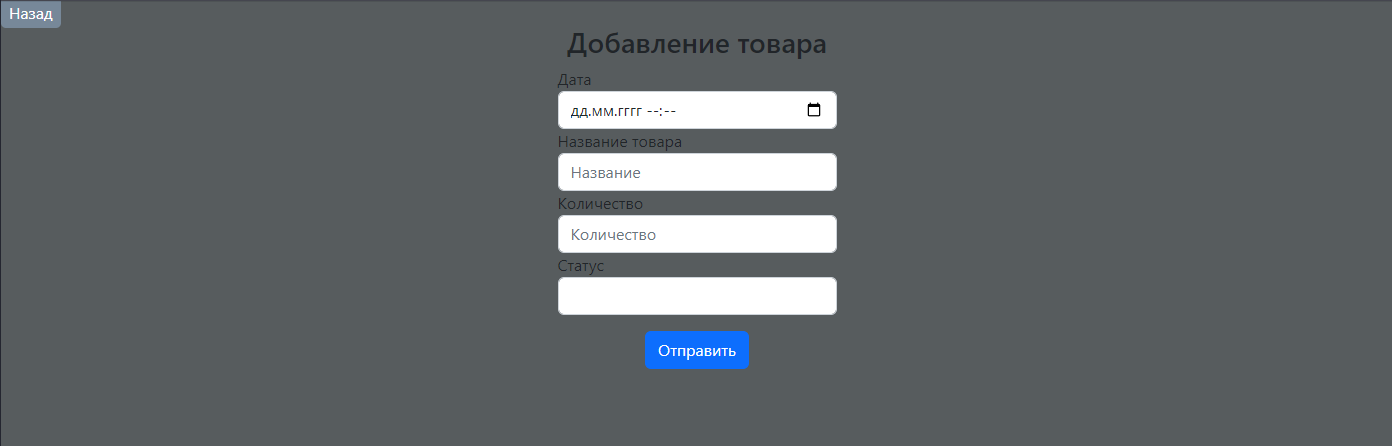


Рисунок 28 – Создание товара

При нажатии на кнопку «Удаление товара», открывается соответствующая вкладка. Представлено на рисунке 29.

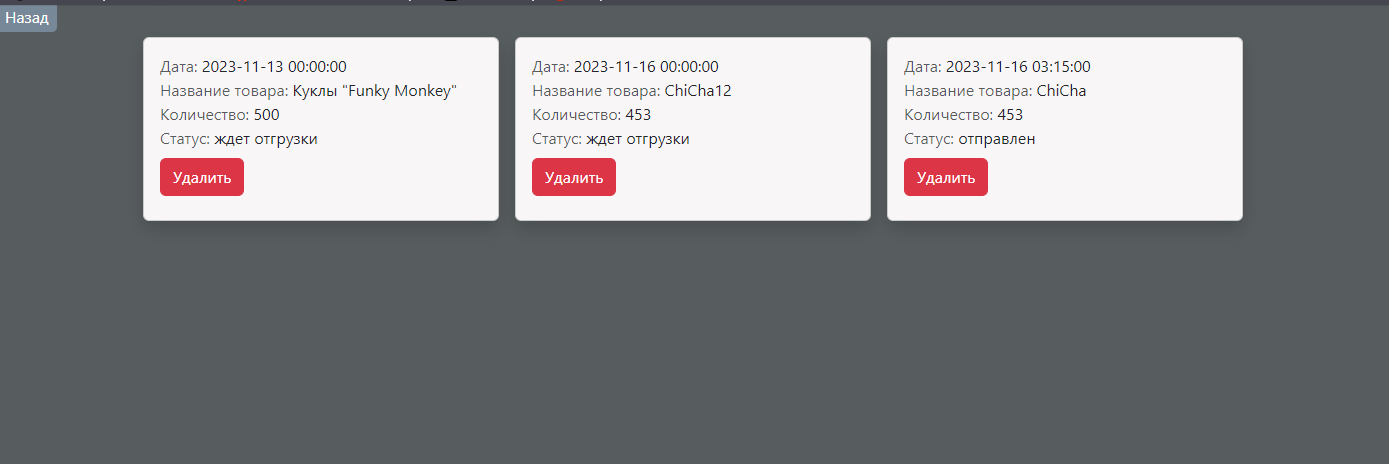


Рисунок 29 – Удаление товара

Заключение

В ходе выполнения, данной курсовой работы, была разработана информационная система «Складской комплекс» – система, в которой сотрудники могут легко взаимодействовать с товарами, что заметно облегчает их работу. Был определен и реализован следующий функционал информационной системы:

для сотрудников:

* авторизация;
* добавление товаров;
* удаление товаров;
* редактирование товаров.

для администратора:

* создание сотрудников;
* удаление сотрудников;
* редактирование сотрудников;
* добавление новых товаров;
* удаление товаров;
* редактирование товаров.

Были рассмотрены возможные реализации ИС с использованием разных технологий и языков программирования, но в результате анализа инструментальных средств разработки выбор остановился на языке программирования PHP.

Был разработан браузерный программный продукт, с широким функционалом, а также понятным и лёгким интерфейсом.

В дальнейшем ИС может развиваться путём расширения функционала и совершенствования интерфейса, тем самым набирая большее сообщество пользователей, а также и актуальность программного продукта.

Все поставленные цели и задачи курсового проекта были успешно выполнены.

Список используемых материалов

1. Colorscheme – Цвета HTML. Таблица из 147 имён цветов для HTML и CSS – URL: https://colorscheme.ru/html-colors.html (дата обращения: 15.10.2022). – Текст: электронный.
2. Htmlbook – Самоучитель HTML4 – URL: http://htmlbook.ru/samhtml (дата обращения: 15.10.2023). – Текст: электронный.
3. HTML5 BOOK – Основы CSS – URL: https://html5book.ru/osnovy-css/ (дата обращения: 20.10.2023). – Текст: электронный.
4. Itnan – MariaDB в сравнении с MySQL в 2022 году – URL: https://itnan.ru/post.php?c=1&p=662870 (дата обращения: 24.10.2023). – Текст: электронный.
5. JetBrains – Возможности PhpStorm – URL: https://www.jetbrains.com/ru-ru/phpstorm/features/ (дата обращения: 25.10.2023). – Текст: электронный.
6. MDN – Основы HTML – URL: https://developer.mozilla.org/ru/docs/Learn/Getting\_started\_with\_the\_web/HTML\_basic (дата обращения: 22.09.2023). – Текст: электронный.
7. Oracleplsql – URL: https://oracleplsql.ru/mariadb-manual.html (дата обращения: 24.10.2023). – Текст: электронный.
8. Php.net – Что такое PHP? – URL: https://www.php.net/manual/ru/intro-whatis.php (дата обращения: 24.09.2022). – Текст: электронный.
9. StudFiles – Инфологическое моделирование. Er-модель. – URL: https://studfile.net/preview/6862142/page:11/ (дата обращения: 28.09.2022). – Текст: электронный.
10. zaLinux – Изучение MySQL/ MariaDB – URL: https://zalinux.ru/?p=760 (дата обращения: 24.09.2023). – Текст: электронный.

**Приложение А – Техническое задание  
Министерство образования Иркутской области**

Государственное бюджетное профессиональное

образовательное учреждение Иркутской области

«Иркутский авиационный техникум»

(ГБПОУИО «ИАТ»)

|  |  |
| --- | --- |
| КП.09.02.07-5.23.201.10 ПЗ |  |

ТЕХНИЧЕСКОЕ ЗАДАНИЕ

ИНФОРМАЦИОННАЯ СИСТЕМА

«СКЛАДСКОЙ КОМПЛЕКС»

|  |  |  |
| --- | --- | --- |
| Руководитель: | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись, дата) | (М.П. Морозов) |
| Студент: | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись, дата) | (Д.В. Козлов) |

Иркутск 2023

**1 Общие сведения**

Наименование работы: информационная система «Складской комплекс».

Исполнитель: студент иркутского авиационного техникума, группы ИС 20-1, Козлов Д.В

Разработка информационной системы проходит в рамках курсовой работы.

Сроки разработки информационной системы с 21.09.2023 по 22.11.2023 года.

**2 Цели и назначение создания автоматизированной системы**

Курсовая работа нацелена на создание информационной системы "Складской комплекс", которая предоставляет функциональность для добавления новых заказов, редактирования существующих и отслеживания состояния товаров.

В рамках этой информационной системы будут реализованы следующие возможности:

* Аутентификация пользователей.
* Регистрация новых пользователей.
* Создание записей о заказах.
* Редактирование существующих записей.
* Удаление записей.

**3 Характеристика объекта автоматизации**

Данная информационная система разрабатывается с целью упростить процессы работы сотрудников склада, облегчая создание и отслеживание заказов и их статусов.

**4 Требования к системе в целом**

**4.1 Требования к структуре и функционированию приложения**

Функции веб-приложения:

1. Окно «Авторизации»:
   1. авторизация пользователей.
2. Окно «Регистрация»:
   1. регистрация пользователей.
3. Окно «Главная»:
   1. Информация о заказах;
      1. добавление наименований;
      2. удаление наименований;
      3. редактирование данных.
   2. Окно «Сотрудники»:
      1. добавление сотрудников;
      2. удаление сотрудников;
      3. редактирование данных.

**4.2 Требования к надежности**

Для гарантировано надежной работы, необходимо осуществлять проверку корректности входных данных и обеспечивать валидность полей. Входные данные получаются от пользователя через клавиатуру и отображаются в отдельных ячейках таблицы.

**4.3 Требования к безопасности**

Для обеспечения безопасности в информационной системе, необходимо реализовать разграничение прав доступа.

**4.4 Требования к эксплуатации, техническому обслуживанию, ремонту и хранению компонентов системы**

Минимальные системные требования для сервера:

1. Процессор: Intel Pentium 4 2.0Ghz / AMD XP 2200+;
2. Оперативная память: 512 Мб;
3. Жёсткий диск: 150мб;
4. Операционная система: Windows 7/8/10.
5. Версия MySQL 5.0 и выше;

Минимальные системные требования для рабочей станции:

1. Процессор: Intel Pentium 4 2.0Ghz / AMD XP 2200+;
2. Оперативная память: 512 Мб;
3. Жёсткий диск: 150мб;
4. Операционная система: Windows 7/8/10.

**5 Требования к документированию**

Основным документом, регламентирующими использование информационной системы является руководство пользователя.

Основным документом, регламентирующими разработку информационной системы является техническое задание.

**6 Состав и содержание работ по созданию системы**

В таблице 1 представлены плановые сроки начала и окончания работы по созданию веб-приложения.

Таблица 1 – Этапы выполнения разработки

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование этапов курсового проекта | Срок | | Объем |
| Начало | Окончание |
| Предпроектное исследование предметной области (выбор темы, постановка цели, задач, описание области применения, исследование предметной области) | 15.09.23 | 21.09.23 | 10% |
| Разработка технического задания (выбор архитектуры программного обеспечения, выбор типа пользовательского интерфейса, выбор языка и среды программирования) | 22.09.23 | 30.09.23 | 20% |
| Проектирование программного обеспечения.  (разработка структурной и функциональной схемы ПО, проектирование базы данных (инфологическое, ER-модель, физическая модель) | 02.10.23 | 12.10.23 | 50% |
| Разработка (программирование) программного продукта | 13.10.23 | 06.11.23 | 80% |
| Тестирование и отладка программного продукта (функциональное тестирование, составление тест-планов) | 07.11.23 | 14.11.23 | 90% |
| Составление программной документации (оформление ПЗ, написание руководства пользователя, составление презентации и речи) | 15.11.23 | 22.11.23 | 100% |
| Защита курсового проекта | 23.11-30.11 | |

Приложение Б – Листинг User.php

<?php

require ('db.php');

class Staff extends DB

{

public function get(){

return $this->DBAll('SELECT \* from staff');

}

public function deleteStaff($request){

$req=json\_decode($request);

return $this->transaction(

'DELETE from staff where id='.$req->id,

'Сотрудник удален');

}

public function createStaff($request){

$req = json\_decode($request);

$last\_name = $req->last\_name;

$first\_name = $req->first\_name;

$father\_name = $req->father\_name;

$post = $req->post;

$connect = $this->connect();

try{

$connect->beginTransaction();

$connect->exec("INSERT INTO staff (last\_name,first\_name,father\_name,post) values ('{$last\_name}','{$first\_name}','{$father\_name}','{$post}')");

$connect->commit();

return json\_encode([

'message'=>'Сотрудник добавлен'

]);

}catch (PDOException $e){

$connect->rollBack();

return json\_encode([

'message'=>$e->getMessage()

]);

}

}

public function updateStaff($request){

$req = json\_decode($request);

$id = $req->id;

$last\_name = $req->last\_name;

$first\_name = $req->first\_name;

$father\_name = $req->father\_name;

$post = $req->post;

$connect = $this->connect();

try{

$connect->beginTransaction();

$connect->exec("UPDATE staff SET last\_name='{$last\_name}', first\_name='{$first\_name}', father\_name='{$father\_name}', post='{$post}'

WHERE id={$id} ");

$connect->commit();

return json\_encode([

'message'=>'Сотрудник обновлён'

]);

}catch (PDOException $e){

$connect->rollBack();

return json\_encode([

'message'=>$e->getMessage()

]);

}

}

}