**Министерство образования Иркутской области**

Государственное бюджетное профессиональное

образовательное учреждение Иркутской области

«Иркутский авиационный техникум»

(ГБПОУИО «ИАТ»)

КР.09.02.07.22.201.11 ПЗ

ИНФОРМАЦИОННАЯ СИСТЕМА «СКЛАД»

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

Иркутск 2022

Содержание

[ВВЕДЕНИЕ](about:blank) 3

[1. Описание предметной области](about:blank) 5

[2. Анализ инструментальных средств разработки](about:blank) 7

[3.Техническое задание](about:blank) 14

[4. Проектирование ИС](about:blank) 15

[4.1. Структурная схема ИС](about:blank) 15

[4.2. Функциональная схема ИС](about:blank) 19

[4.3. Проектирование базы данных](about:blank) 23

[4.4. Проектирование интерфейса](about:blank) 26

[5. Разработка ИС](about:blank) 29

[5.1. Разработка интерфейса ИС](about:blank) 29

[5.2. Разработка базы данных ИС](about:blank) 39

[5.3. Разработка ИС](about:blank) 39

[6. Документирование программного продукта](about:blank) 41

[6.1. Руководство пользователя ИС](about:blank) 41

[Заключение](about:blank) 50

[СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ](about:blank) 51

[Приложение А – Техническое задание](about:blank) 52

[Приложение Б – Листинг news views.py](about:blank) 57

**ВВЕДЕНИЕ**

Склад – это сложное техническое сооружение, предназначенное для приемки, размещения, накопления, хранения, переработки, отпуска и доставки продукции потребителям.

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 программное обеспечение, с помощью которой можно будет ускорить и облегчить работу на складе.

Целью курсовой работы является создание информационной системы «Склад».

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

* Анализ деятельности современного склада.
* Разработать техническое задание.
* Проанализировать инструментальные средства разработки.
* Создать программное обеспечение, которое будет выполнять следующие функции:
* Создать систему, которая позволила бы автоматизировать контроль над поступающим товаром.
* Сделать так, чтобы поиск товаров осуществлялся с помощью компьютеров.
* Поиск товаров для отгрузки будет проводиться путем поиска соответствующего товара в ПО.
* Создать возможность для просмотра информации о месте хранении товара в ПО.

1. **Описание предметной области**

В качестве предметной области выбран «Склад», который предназначен для осуществления процедуры хранения продукции для сторонних лиц, заинтересованных в этом. Склад должен выполнять следующие функции: прием, учет, хранение и отгрузка готовой продукции, приемка готовой продукции, рассортировка, определение потребности в транспортных средствах, механизированных погрузочных средствах, таре и рабочей силе для отгрузки продукции, согласование планов и условий поставок продукции с основного производства и по договорам со сторонними организациями, организация приемки продукции сторонними организациями, координация деятельности по закупке и продаже продукции с наличием свободных складских площадей, подготовка отчетов об объемах продукции.

Процедура принятия продукции на склад:

* Продукция приходит на склад в сопровождении приходной накладной.
* Контролер на складе, проверяет приходную накладную, и регистрирует ее в книге учета документов.
* Осматривает входящую продукцию, и, если с ней все нормально принимает ее на склад, передавая экспедитору товара выписку (документ) о том, что товар принят на хранение.
* Грузчики отвозят товар в свободное место хранения, и контролер делает запись в книге учета о том, где хранится вновь поступившая продукция.

В ходе работы склада, он нуждается в инвентаризации, которая включает в себя такие стадии как: ответственный работник по переучету продукции, в сопровождении книги переучета, отправляется на склад и в ручную осматривает и переписывает данные о товаре и его количестве; после этого данные сверяются в книге учета товаров, лицами ответственными за документы отчетности на складе и составляется соответствующий отчет, по данным переучета продукции.

Отгрузка товаров со склада проходит следующие стадии:

* Получатель товара подает накладную на отгрузку товара.
* Контролер проверяет эту накладную и регистрирует ее в книге учета входящих документов.
* Далее контролер дает указание работникам склада на поиск нужной продукции и отгрузки ее.
* Затем получатель товара проводит его осмотр, на счет того нужный ли товар отгрузили и в нужном количестве.
* Контролер регистрирует в книге учета факт отгрузки товара.
* Далее контролер выдает получателю груза сопроводительный документ по отгрузке товара.
* Далее происходит непосредственно отгрузка товара техническими средствами.

## **Анализ инструментов, используемых в разработке программного продукта**

Грамотно выбранные инструменты для создания программного продукта, заранее определяют его результат.

Проектировать структуру информационной системы удобно через MySQL Workbench и Draw.io, а дизайн – через сервис Figma. Информационная система будет серверной, для работников. Для реализации клиентской части отлично подойдут следующие инструменты: HTML5, CSS3 и JS, включая AJAX.

MySQL Workbench — это унифицированный визуальный инструмент для архитекторов баз данных, разработчиков и администраторов баз данных. MySQL Workbench предоставляет моделирование данных, разработку SQL и комплексные инструменты администрирования для настройки сервера, администрирования пользователей, резервного копирования и многого другого.

Draw.io — это бесплатное кроссплатформенное программное обеспечение для рисования графиков с открытым исходным кодом, разработанное на HTML5 и JavaScript. Его интерфейс можно использовать для создания таких диаграмм, как блок-схемы, каркасы, UML-диаграммы, организационные диаграммы и сетевые диаграммы.

Figma – графический редактор для создания прототипов сайтов и приложений. Над проектом одновременно могут работать несколько человек, так как можно выдать доступ на редактирование или комментирование любому. В Figma обычно создают прототипы сайтов и приложений, иллюстрации, векторную графику, рисуют элементы интерфейса. Ещё здесь создают макеты сайтов для тильды: есть возможность импортировать дизайн.

HTML – язык разметки гипертекста. Язык разметки дает браузеру необходимые инструкции о том, как отображать тексты и другие элементы страницы на мониторе. Язык HTML интерпретируется браузерами и отображается в виде документа, в удобной для человека форме.

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 – это мультипарадигменный язык программирования, который обычно применяется в качестве встраиваемого инструмента для программного доступа к различным объектам приложений. С точки зрения веб-разработки, без знаний этой технологии невозможно заниматься созданием современных интерактивных сайтов. Язык JS – это то, что «оживляет» разметку страниц (HTML) и пользовательский функционал (CMS) сайтов. С помощью этого языка реализуется возможность реакции страницы или отдельных ее элементов на действия посетителя. Сегодня JavaScript является базовым языком программирования для браузеров. Он полностью совместим с операционными системами Windows, Linux, Mac OS, а также всеми популярными мобильными платформами.

AJAX, или, более длинно, Asynchronous Javascript And Xml - технология для взаимодействия с сервером без перезагрузки страниц. За счет этого уменьшается время отклика и веб-приложение по интерактивности больше напоминает десктоп.

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.

Несмотря на название, он может работать не только с MySQL, но и с другими СУБД: Oracle Database, Microsoft SQL Server, PostgreSQL и другими. 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 (БД). 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 БД | MySQL | MySQL Workbench | 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 Workbench, так как она предоставляет весь необходимый функционал для разработки продукта, и при этом она проста в изучении и овладении.

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.
* Бюджетность, экономичность. Найти специалиста не представляется проблемой, стоимость написания программ на php не высока.
* 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:

1. PHPStorm.

2. Netbeans.

3. Atom.

PhpStorm — коммерческая кросс-платформенная [интегрированная среда разработки](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D0%B3%D1%80%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D1%80%D0%B5%D0%B4%D0%B0_%D1%80%D0%B0%D0%B7%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%BA%D0%B8) обеспечивает первоклассную поддержку PHP 5.3–8.1, мгновенно предотвращает ошибки, предоставляет точное автодополнение и безопасные рефакторинги, а также возможность редактирования кода на HTML, CSS и JavaScript.

IDE обеспечивает умное автодополнение, подсветку синтаксиса, расширенные настройки форматирования, проверку ошибок на лету, сворачивание фрагментов кода, поддержку языковых комбинаций и много других возможностей для продуктивной разработки. Автоматические рефакторинги позаботятся о порядке в вашем коде и помогут легко и безопасно выполнять глобальные настройки проекта.

Сотни встроенных инспекций проверяют код прямо в режиме редактирования, а также позволяют проанализировать весь проект на наличие ошибок и проблем в структуре кода. Для большинства инспекций доступны быстрые исправления, которые позволяют откорректировать код мгновенно. Нажмите Alt+Enter, чтобы просмотреть список возможных исправлений для любой инспекции. PhpStorm помогает быстро перемещаться по коду и экономить время при работе над большими проектами. В один клик можно перейти к определению любого метода, функции или переменной, или найти их использования в проекте.

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. Для разработки программ в среде NetBeans и для успешной инсталляции и работы самой среды NetBeans должен быть предварительно установлен Sun JDK или J2EE SDK подходящей версии. Среда разработки NetBeans по-умолчанию поддерживала разработку для платформ J2SE и J2EE. Начиная с версии 6.0 NetBeans поддерживает разработку для мобильных платформ J2ME, C++ (только g++) и PHP без установки дополнительных компонентов.

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.

Atom основан на Electron (ранее известный как Atom Shell) — фреймворке кроссплатформенной разработки с использованием Chromium и io.js. Редактор написан на CoffeeScript и LESS. Версия 1.0 была выпущена 25 июня 2015 г.

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

Таблица 3 – Сравнение IDE для разработки программного продукта

| Название IDE | PHPStorm | 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 PHPStorm. PHPStorm намного функциональнее, чем приведённые выше Atom и NetBeans. В PHPStorm имеется автоматическое сохранение, которое не позволит случайно потерять все наработки, а также он более удобен в работе.

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 PHPStorm.

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);  
   привычность интерфейса (почти каждый пользователь компьютера хотя бы раз запускал браузер и работал в нем);  
   простота установки и обслуживания (новую версию web-приложения не надо устанавливать на все компьютеры – достаточно установить на сервер).
3. 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.
4. Простота программирования интерфейса - Создавать HTML из  
   шаблонов куда проще и удобнее, чем создавать графические  
   приложения визуальными редакторами интерфейсов.
5. Простота программирования в целом - 3х-уровневая архитектура:  
   база данных – логика приложения (сервер) – логика представления  
   (клиент, браузер)
6. **Техническое задание**

**ВСТАВИТЬ ПУНКТ ПО ТЕХНИЧЕСКОМУ ЗАДАНИЮ**

1. **Проектирование ИС**

**4.1 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.

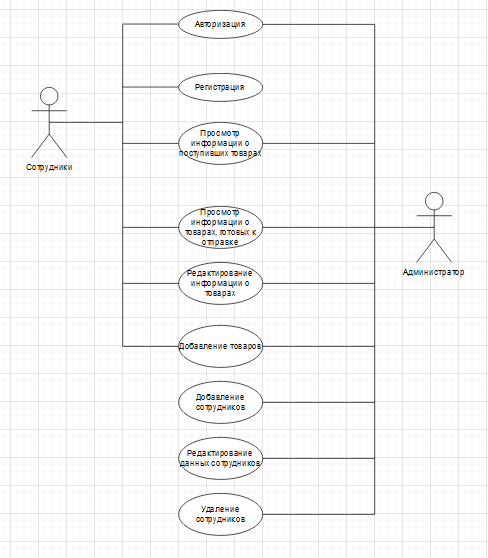
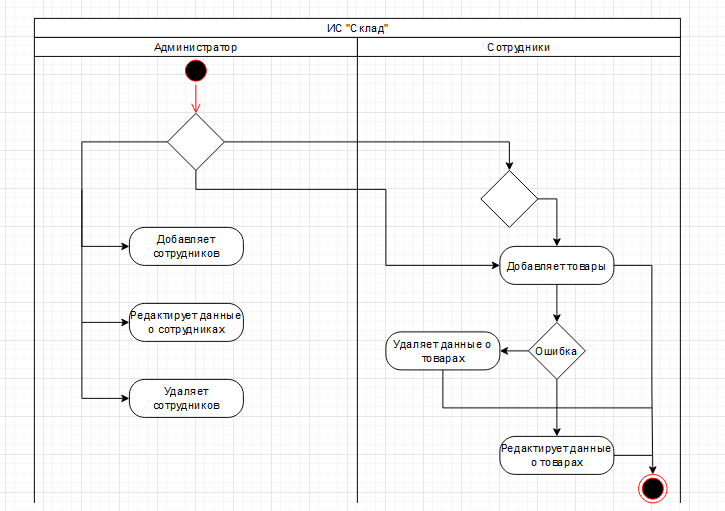


Рисунок 1 - Диаграмма прецедентов

На рисунке 2 представлена диаграмма деятельности. Она содержит 2 роли. Процесс начинается с администратора, который добавляет сотрудников, редактирует или удаляет их. Далее сотрудник или администратор добавляет товар, если есть какая-то ошибка, то изменяет или же удаляет товар.

Рисунок 2 – Диаграмма деятельности

На рисунке 3 представлена диаграмма компонентов. В общей сложности 4 элементов, которые служат для просмотра, редактирования и удаления информации о прибывшем товаре, а также товаре готовом к отгрузке.

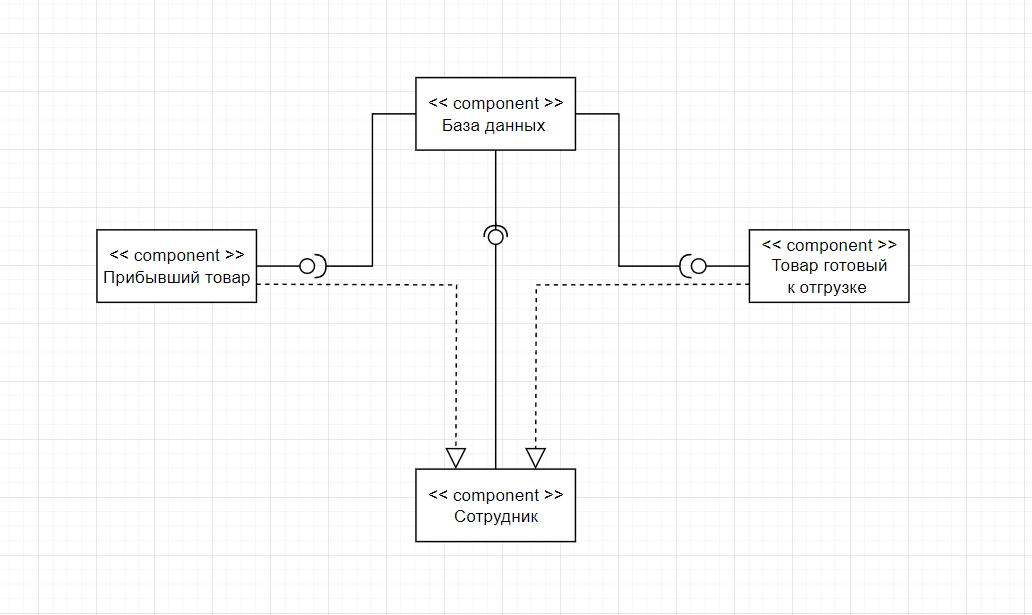


Рисунок 3 – Диаграмма компонентов

На рисунке 4 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.

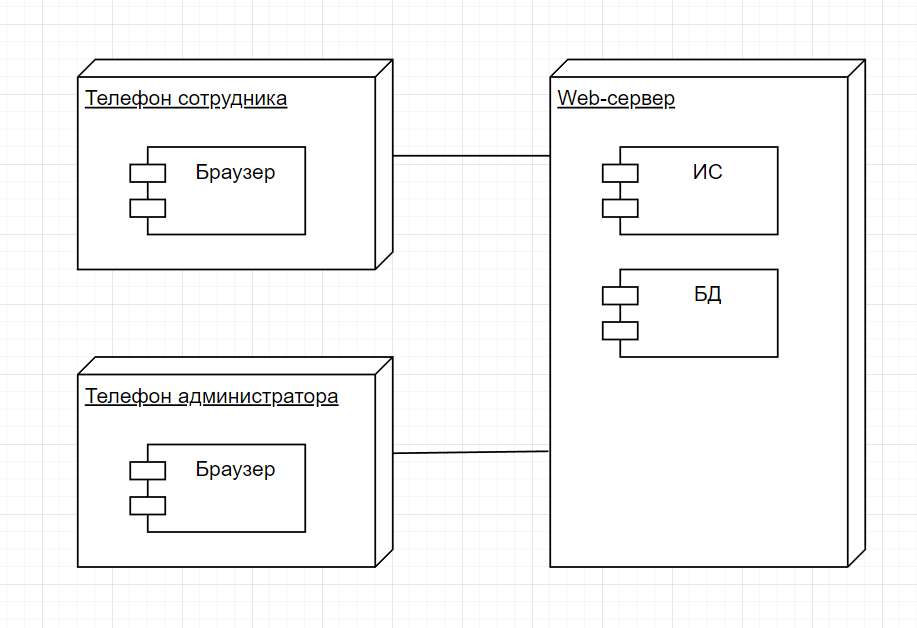


Рисунок 4 – Диаграмма развёртывания

* 1. **Функциональная схема ИС**

На рисунке 5 представлена контекстная диаграмма, отображающая деятельность ИС.

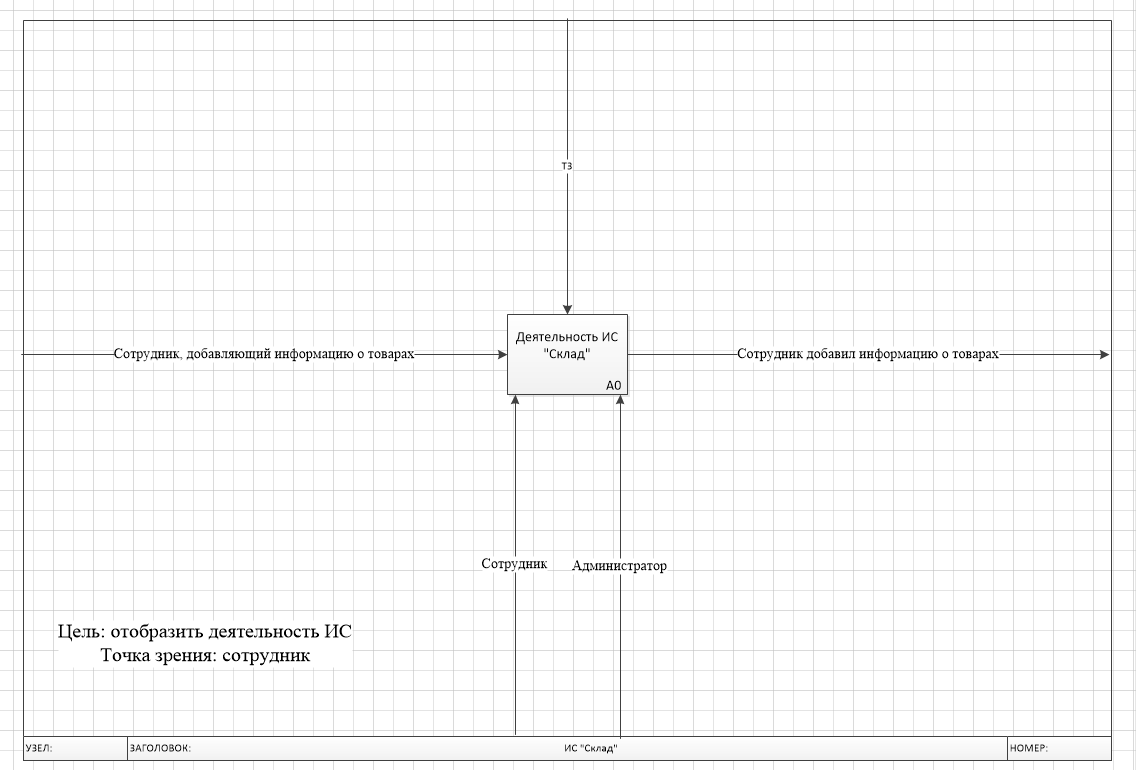


Рисунок 5 – Контекстная диаграмма IDEF0

На рисунке 18 представлена диаграмма декомпозиций (А1), отображающая деятельность ИС более подробно предыдущей.

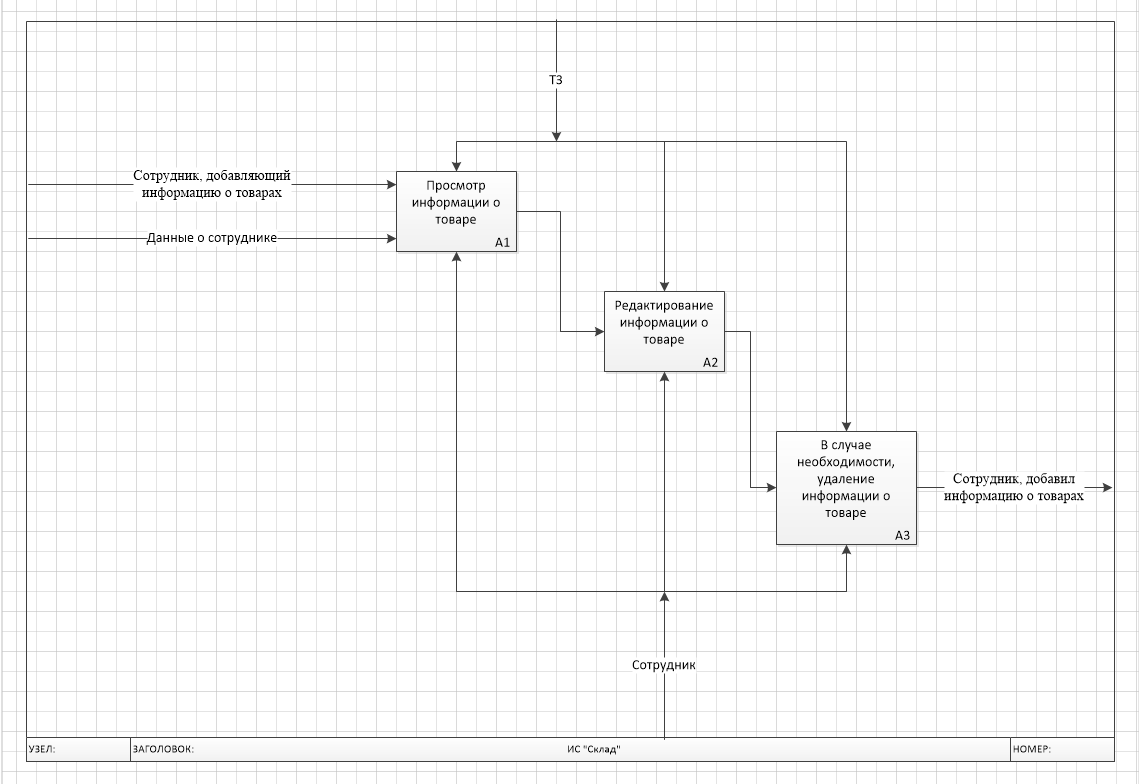


Рисунок 6 – Диаграмма декомпозиций А1

На рисунке 7 представлена диаграмма классов. Она содержит 15 классов, среди них база данных, html-страница, информационная сеть «Ремонт и Дизайн», пользователь, просто пользователь, автор публикаций, администратор, суперадмин, публикация, публикации про ремонт, дизайн, лайфхаки, уведомления, обращения в публикацию и отчёты.

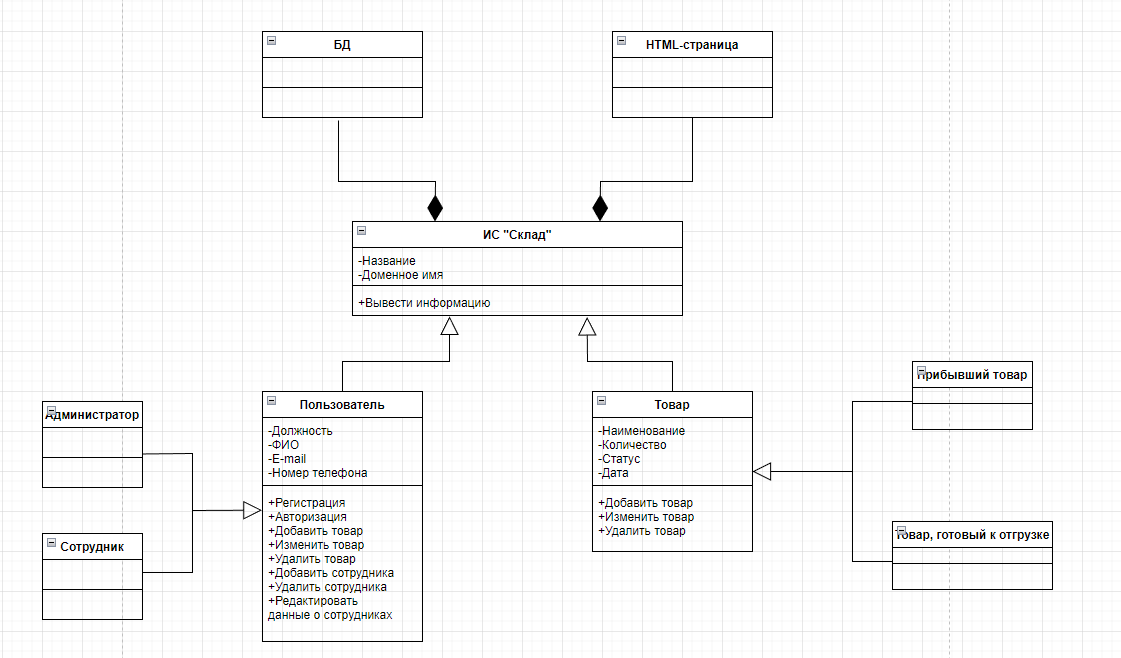


Рисунок 7 – Диаграмма классов

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

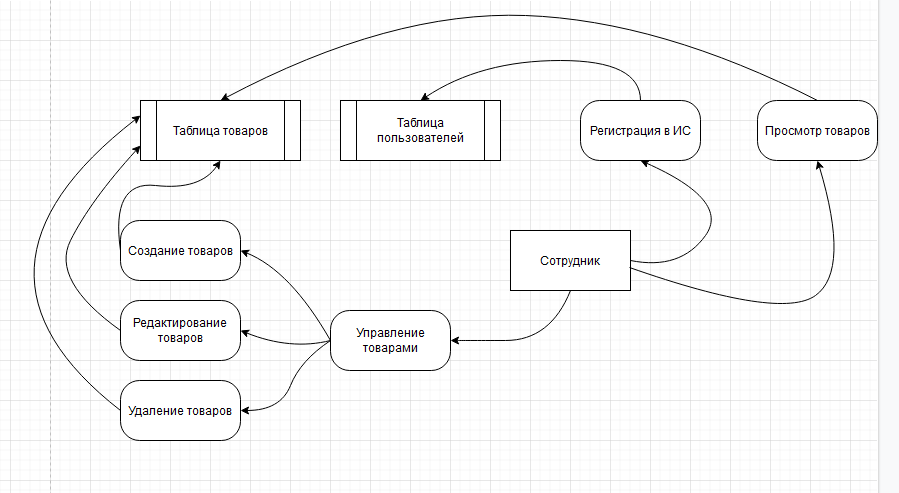


Рисунок 8 - Диаграмма потоков данных

**4.3 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.

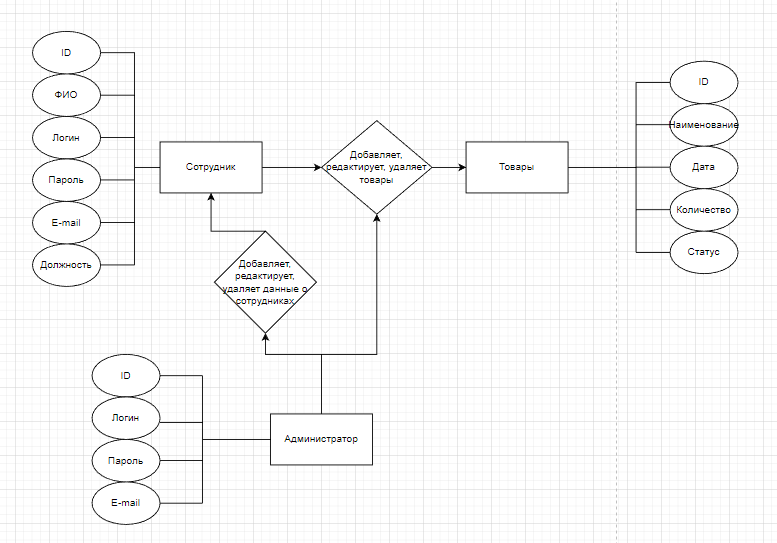


Рисунок 9 – Инфологическая модель базы данных

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.

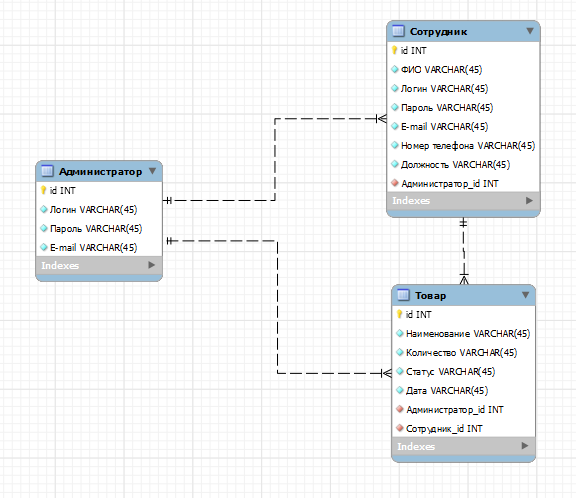


Рисунок 10 – Даталогическая модель базы данных

На рисунке 1 представлена ER-диаграмма базы данных. Она содержит 6 таблиц для полного функционирования и качественной сортировки информации.

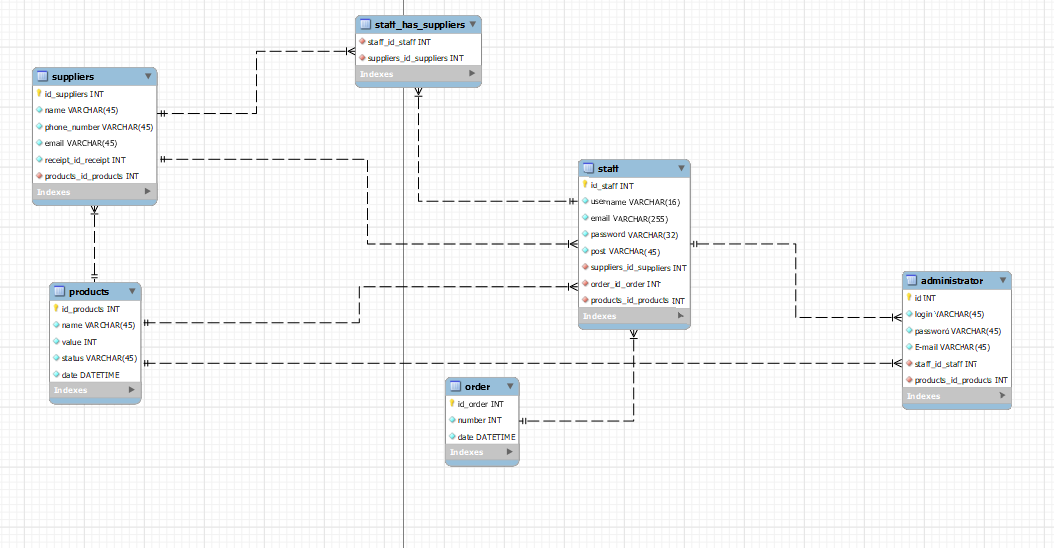


Рисунок 10 – ER-модель базы данных

Таблица 4 – 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 5 – 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 6 – Таблица order

| Таблица | Тип данных | Описание |
| --- | --- | --- |
| id | INT | ID |
| number | INT | Номер заказа |
| date | DATETIME | Дата |
| password | VARCHAR(45) | Пароль |
| staff\_id\_staff | INT | Вторичный ключ |
| products\_id\_products | INT | Вторичный ключ |

Таблица 7 – 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 8 – 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 9 – Таблица staff\_has\_suppliers

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

**Приложение А – Техническое задание**

Министерство образования Иркутской области

Государственное бюджетное профессиональное образовательное учреждение

Иркутской области

«Иркутский авиационный техникум»

(ГБПОУИО «ИАТ»)

|  |  |
| --- | --- |

ТЕХНИЧЕСКОЕ ЗАДАНИЕ

ИНФОРМАЦИОННАЯ СИСТЕМА

«СКЛАД»

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

Иркутск 2022

**1 Общие сведения**

Наименование работы: информационная система «Склад».

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 21.09.2022 по 06.12.2022 года.

**2 Цели и назначение создания автоматизированной системы**

Целью курсовой работы является создание информационной системы «Склад», которое включает в себя возможности внесение в списки новых заказов и редактирования старых, отслеживания товаров.

В данной информационной системе реализуется следующий функционал:

− авторизация;

− регистрация;

− создание записей;

− редактирование записей;

− удаление записей.

**3 Характеристика объекта автоматизации**

Данная информационная система разрабатывается для облегчения работы сотрудникам склада. При помощи системы будет проще создавать новые заказы, отслеживать заказы склада, статус заказов.

**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 обеспечения надежности необходимо проверять корректность получаемых данных и реализовать валидность полей. Входные данные поступают в виде значений c клавиатуры. Эти значения отображаются в отдельных полях 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 разработки | Срок | |
| --- | --- | --- |
| Начало | Окончание |
| Предпроектное исследование предметной области (выбор темы, постановка цели, задач, описание области применения, исследование предметной области) | 15.09.22 | 21.09.22 |
| Разработка технического задания (выбор архитектуры программного обеспечения, выбор типа пользовательского интерфейса, выбор языка и среды программирования) | 21.09.22 | 26.09.22 |
| Проектирование программного обеспечения.  (разработка структурной и функциональной схемы ПО, проектирование базы данных (инфологическое, ER-модель, физическая модель) | 26.09.22 | 25.10.22 |
| Разработка (программирование) и отладка программного продукта | 25.10.22 | 28.11.22 |
| Составление программной документации (написание руководства пользователя) | 28.11.22 | 06.12.22 |