

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН
Некоммерческое акционерное общество
«АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ
ИМЕНИ ГУМАРБЕКА ДАУКЕЕВА»

Кафедра IT-инжиниринг

«ДОПУЩЕН К ЗАЩИТЕ»

Зав. кафедрой _____
(ученая степень, звание, Ф.И.О.)

_____ « ____ » _____ 20 ____ г.
(подпись)

ДИПЛОМНЫЙ ПРОЕКТ

На тему: «Разработка информационной системы учета продаж супермаркета»
Специальность 5В070300 – Информационные системы

Выполнил Нургалиев А.А.

Группа ИС-16-2

Научный руководитель д.т.н. Ахметов Б.С.

Консультанты:

по экономической части:

(ученая степень, звание,
Ф.И.О.) _____ « ____ » _____ 20 ____ г.
(подпись)

по безопасности жизнедеятельности:

(ученая степень, звание,
Ф.И.О.) _____ « ____ » _____ 20 ____ г.
(подпись)

по применению вычислительной техники:

(ученая степень, звание,
Ф.И.О.) _____ « ____ » _____ 20 ____ г.
(подпись)

(ученая степень, звание,
Ф.И.О.) _____ « ____ » _____ 20 ____ г.
(подпись)

Нормоконтролер: _____

Ф.И.О.) _____ (ученая степень, звание,
_____ «___» _____ 20__ г.
(подпись)

Рецензент: _____
Ф.И.О.) _____ (ученая степень, звание,
_____ «___» _____ 20__ г.
(подпись)

Алматы 2020

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН
Некоммерческое акционерное общество
«АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ
ИМЕНИ ГУМАРБЕКА ДАУКЕЕВА»

Институт систем управления и информационных технологий

Кафедра IT-инжиниринг

Специальность 5В070300 – «Информационные системы»

ЗАДАНИЕ

на выполнение дипломного проекта

Студенту Нургалиеву Алихану Амиржановичу

Тема проекта: Разработка информационной системы учета продаж супермаркета

Утверждена приказом по университету № ___ от «___» _____ 2020 г.

Срок сдачи законченного проекта «___» _____ 2020 г.

Исходные данные к проекту (требуемые параметры результатов исследования (проектирования) и исходные данные объекта): постановленные задачи, отчетная документация (чеки, кассовые книги), этикетки и бирки товаров.

Перечень вопросов, подлежащих разработке в дипломном проекте, или краткое содержание дипломного проекта:

- а) анализ предметной области;
- б) проектирование базы данных;
- в) программная реализация;

- г) экономическое обоснование эффективности;
- д) вопросы безопасности жизнедеятельности и охраны труда.

Перечень графического материала (с точным указанием обязательных чертежей): имеется 26 таблиц и 32 иллюстраций.

Основная рекомендуемая литература:

- 1 Кузнецов С.Д. Базы данных. – М.: Academia, 2012. – 496 с.
- 2 Никсон Р.: Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5. 4-е изд. Издательство – Питер-Трейд, 2016. – 768 с.
- 3 Кузнецов М.В., Симдянов И.В. MySQL 5 – СПб.: БХВ-Петербург, 2010. – 1024 с.
- 4 Горбаченко В.И., Убиенных Г.Ф., Бобрышева Г.В. Проектирование информационных систем с СА ERwin Modeling Suite 7.3: учебное пособие – П.: Издательство ПГУ, 2012. – 154 с.
- 5 Маклафлин Б. PHP и MySQL. Исчерпывающее руководство. 2-е изд. — СПб.: Питер, 2014. — 544 с.: ил. — (Серия «Бестселлеры O’Reilly»). 6 Моргунов Е.П. Язык SQL. Базовый курс: учеб.-практ. пособие – М.: Postgress Professional, 2017. – 256 с.

Консультация по проекту с указанием относящихся к ним разделов проекта

Раздел	Консультант	Сроки	Подпись
Экономическая часть	Габелашвили К.Р	21.04.2020	
Безопасности жизнедеятельности	Приходько Н.Г	30.04.2020	
Программная часть	Майкотов М.Н	14.05.2020	
Нормоконтролер	Абсатарова Б.Р	13.05.2020 – 18.05.2020	

ГРАФИК подготовки дипломной работы (проекта)

Наименования разделов, перечень разрабатываемых вопросов	Сроки представления научному руководителю	Примечания
Анализ и исследование предметной области	17.02.2020 – 26.02.2020	
Проектирование базы данных	27.02.2020 – 23.03.2020	

Программная реализация	24.03.2020 – 30.04.2020	
------------------------	----------------------------	--

Дата выдачи задания « ____ » _____ 2020 г.

Заведующий кафедрой _____ А.А.Досжанова
(подпись)

Научный руководитель проекта _____ Б.С. Ахметов
(подпись)

Задание принял к исполнению студент _____ А.А.Нурғалиев
(подпись)

АҢДАТПА

Дипломдық жоба - бұл супермаркеттердің сатылымын бақылауға мүмкіндік беретін ақпараттық жүйе. Мұндай жүйе кез-келген кәсіпорынға арналған, онда сату, тауарлар мен жұмысшылардың есебін жүргізу қажет. Өзірленген жүйеде әртүрлі перифериялық жабдықтардан ақпарат ала алатын мәліметтер базасы бар, оған бақылау-кассалық машиналар, терминалдар кіреді. Ақпараттық жүйе сізге қағаз есеп беруден бас тартуға және супермаркеттегі тауарлар мен тауарларды бақылау процесін жеделдетуге мүмкіндік береді.

Жүйе компьютерде орнатылған бағдарламалық жасақтамамен жақсы үйлесетін веб-бағдарлама ретінде жасалған.

Жүйенің серверлік жағы MySQL және PHP-де жазылған. Клиент бөлігі HTML және CSS және JavaScript қолдана отырып жазылады.

Өндіріске енгізуден еңбек қауіпсіздігі және өмір қауіпсіздігі, экономикалық тиімділік бойынша есептеулер жүргізілді.

АННОТАЦИЯ

Дипломный проект представляет собой информационную систему, которая позволяет вести учет продаж супермаркета. Такая система предназначена для любого предприятия, в котором надо вести отчетность по продажам, товарам, сотрудникам. Разработанная система имеет базу данных, в которую может поступать информация из различных периферийных аппаратных средств, к которым относятся кассовые аппараты, терминалы. Информационная система позволит отказаться от бумажной отчетности и ускорит процесс мониторинга продаж и товаров в супермаркете.

Система разработана в виде веб-приложения для наилучшей совместимости с установленным ПО на компьютере. Серверная часть системы написана на MySQL и PHP. Клиентская часть написана на языке HTML, с использованием стилей CSS и языка JavaScript.

Проведены расчеты по охране труда и безопасности жизнедеятельности и экономической эффективности от внедрения в производство.

ABSTRACT

The graduation project is an information system that allows you to keep track of supermarket sales. Such a system is intended for any enterprise in which it is necessary to keep records of sales, goods, and employees. The developed system has a database, which can receive information from various peripheral hardware, these include cash registers, terminals. The information system will allow you to abandon paper reporting and accelerate the process of monitoring sales and goods in the supermarket.

The system is designed as a web-application for the best compatibility with installed software on a computer. The server side of the system is written in MySQL and PHP. The client part is written in HTML, using CSS and JavaScript.

The calculations on labor protection and life safety and economic efficiency from implementation in production were carried out.

Содержание

Введение	8
1 Аналитическая часть	10
1.1 Обзор литературы.....	10
1.2 Анализ существующих систем учета продаж	11
1.3 Постановка задачи	13
2 Проектная часть.....	15
2.1 Описание предметной области	15
2.2 Проектирование функциональной структуры	16
2.3 Разработка бизнес-модели ИС	20
2.4 Разработка информационного обеспечения	24
3 Программная реализация	47
3.1 Обоснование выбора среды программирования.....	47
3.2 Структура программного обеспечения	49
3.3 Разработка веб-приложения.....	50
3.4 Описание интерфейса.....	60
4 Экономическое обоснование проекта	65
4.1 Определение объема и трудоемкости разработки ПО	65
4.2 Расчет затрат на разработку ПО	67
4.3 Расчет динамических показателей эффективности капитальных вложений.....	73
5 Безопасность жизнедеятельности.....	76
5.1 Теоретическая часть	76
5.2 Расчетная часть.....	78
Заключение	87
Список литературы	88
Приложение А Техническое задание	90
Приложение Б Листинг программы	91
Приложение В Акт внедрения	121

Введение

В многообразии сфер человеческой деятельности широкое распространение получили технологии, использующие базы данных для систематизации и хранения производственной информации. Структурированная информация лучше анализируется и обрабатывается, а если хранить её в базе, то можно систематически обновлять и дополнять, откуда и следует неизменная актуальность баз данных.

База данных – это некоторое множество данных, организованных по конкретным правилам, включающих принципы описания, хранения и манипулирования данными. Эти данные относятся к конкретной предметной области и организованы так, что могут использоваться для решения большинства задач различными пользователями.

В настоящее время возникло большое количество избыточной информации, в которой часто сложно сориентироваться и найти нужные сведения.

Решением таких проблем являются автоматизированные базы данных, ставшие неотделимой частью практически всех компьютерных систем – от отрасли до отдельного предприятия. За последнее время вырос и растет уровень потребительских качеств систем управления базами данных (СУБД): разнообразие поддерживаемых функций, удобный для пользователя интерфейс, сопряжение с программными продуктами, в частности с другими СУБД, возможности для работы в сети и т.д. СУБД позволяет сводить воедино информацию из самых разных источников (электронные таблицы, другие базы данных) и помогает быстро найти необходимую информацию, донести ее до окружающих с помощью отчетов, графиков или таблиц.

Актуальность дипломного проекта заключается в том, что любая организация в современном мире не может обойтись без информационной системы. Это касается и различного рода супермаркетов. Использование информационной системы позволяет переводить данные, хранящиеся на бумажном носителе в электронный вид, и также способствует оперативному доступу к ним. Это позволяет экономить время и средства на затраты. Кроме того, организация, приобретая исключительно новые качества, становится более конкурентоспособней. Помимо этого, чтобы доступ к этим данным был удобен, разрабатывается веб-приложение. Веб-приложение является программным продуктом, главная часть которого содержится на сервере, а пользовательский интерфейс (UI) отображается в браузере в виде веб-страниц. Первым плюсом, можно назвать то, что приложение независит от того, какая операционная система установлена на компьютере пользователя, является кросс-платформенным. Вторым плюсом веб-приложений - сама концепция такого приложения меняет способ распространения продукта. Разработчики удаляются от традиционных способов распространения программных продуктов путем продажи копий и установки их на каждый компьютер пользователей. Третий плюс для пользователя - пропадает необходимость в

установке и настройке программного обеспечения, так как все уже установлено на серверах и настроено разработчиком. Четвертым плюсом можно назвать то, что требований для работы с приложением не так много, достаточно лишь наличие компьютера и установленного на нем браузера. Следующий положительный момент для самих разработчиков. Так как основная часть веб-приложения располагается на сервере в определенном месте, куда проще стало осуществлять его настройку, поэтому нет надобности содержать много специалистов технической поддержки, занимающихся консультацией пользователей и настройкой приложения на компьютерах.

Созданное веб-приложение с БД можно уже назвать информационной системой. Ведь в основе любой ИС лежит база данных. Информационная система (ИС) – это система, выполняющая функции хранения, поиска и обработки информации, и соответствующие организационные ресурсы (человеческие, технические, финансовые и т. д.), которые обеспечивают и распространяют информацию.

Целью данного дипломного проекта является создание информационной системы для учета продаж супермаркета.

Согласно цели можно выделить следующие задачи:

- проведение анализа предметной области;
- создание логической и физической модели данных;
- программно реализовать базу данных.

1 Аналитическая часть

1.1 Обзор литературы

Для написания дипломного проекта были использованы различные источники информации, к примеру статьи из сети интернет, литература по программированию, ГОСТы. Основной литературой при написании серверной части информационной системы была книга автора Маклафин. Б. «PHP и MySQL», в ней изложены способы управления содержанием сайта, изложены способы взаимодействия базы данных с пользователем и со скриптовым языком PHP.

Исчерпывающим руководством по проектированию информационной системы является книга авторов Горбаченко В.И., Убиенных Г.Ф., Бобрышева Г.В. «Проектирование информационных систем с СА ERwin Modeling Suite 7.3», так как в ней описаны инструкции по необходимому для разработки проекта CASE-средства моделирования баз данных. Также данное пособие знакомит с методологиями моделирования IDEF0, DFD, IDEF3.

Официальный мануал «MySQL::MySQL 8.0 Reference Manual» содержит полную инструкцию по работе с языком SQL и обеспечивает необходимой информацией для проектирования структуры, содержания, взаимодействия данных в базе данных. Уникальность выбранного пособия состоит в том, что она содержит наиболее конкретную информации без излишеств, что в свою очередь позволит сэкономить время.

Для изучения серверного языка программирования была выбрана литература «PHP and MySQL Web Development» ознакамливающая со взаимодействием базы данных с сайтом. Содержит множество примеров и помогает в устранении возможных ошибок в коде.

Хорошим пособием для освоения навыков разработки интерфейса была книга автора Робин Никсон: «Создаем динамический веб-сайт с помощью PHP, MySQL, JavaScript, CSS и HTML5». В первой части книги преподносятся основы языка программирования PHP, включая основы синтаксиса, массивов, функций и объектно-ориентированного программирования. Затем, после усвоения основ PHP, можно переходить к введению в систему управления базами данных MySQL, рассмотрение которой начинается с изучения структуры базы данных MySQL и заканчивается составлением сложных запросов. После этого рассказывается о том, как воспользоваться сочетанием PHP и MySQL, чтобы приступить к созданию собственных динамических веб-страниц путем интегрирования в это сочетание форм и других функциональных возможностей HTML. Затем будут рассмотрены подробности практических аспектов разработки на PHP и MySQL, включая описание различных полезных функций и способов работы с cookies и сессиями, а также способов поддержания высокого уровня безопасности. [15]

В следующих нескольких главах излагаются основы JavaScript, начиная с простых функций и обработки событий и заканчивая доступом к объектной модели документа (DOM), проверкой введенных данных и обработкой ошибок в браузере. [15]

Статьи из сети интернет были преимущественно использованы для поиска решений проблем в ходе проектирования.

1.2 Анализ существующих систем учета продаж

CloudShop [10]

Онлайн система, учет продаж производится в режиме реального времени, что является как преимуществом так и недостатком в некоторой степени если в заведнии интернет периодически пропадает либо вовсе отсутствует. Содержит основные торговые операции, к которым относятся продажа, котроль денежных средств в кассе, добавление товаров.

По отзывам пользователей были сделаны выводы, что система часто выводит ошибки, а центр поддержки отвечает не регулярно. Затратность на пользование системы не соответствует функционалу, который для указанного ценового сегмента является неполноценным.

ПО устанавливается на ПК и мобильные телефоны и имеет определенную совместимость, которая в незначительной степени уменьшает список возможных покупателей.

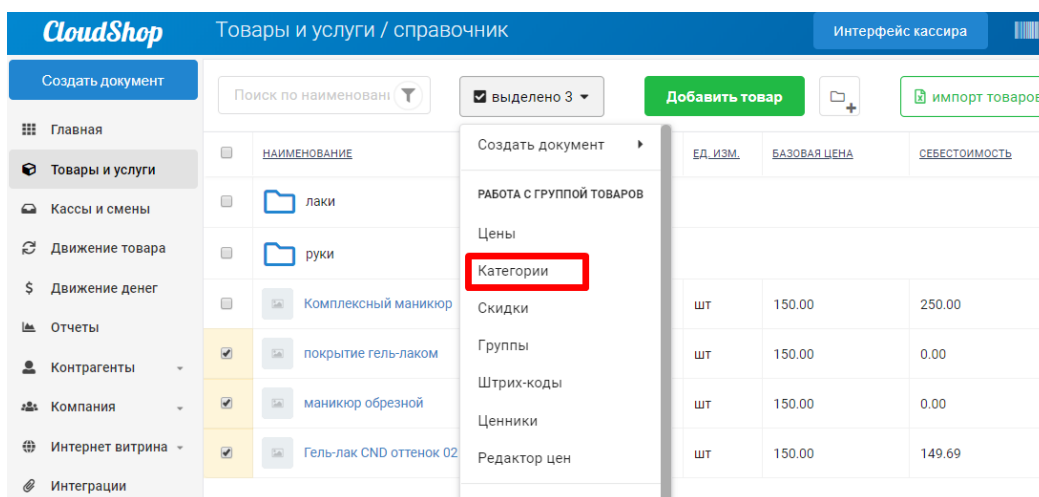


Рисунок 1.1 – Интерфейс

USU - Универсальная система учета [11]

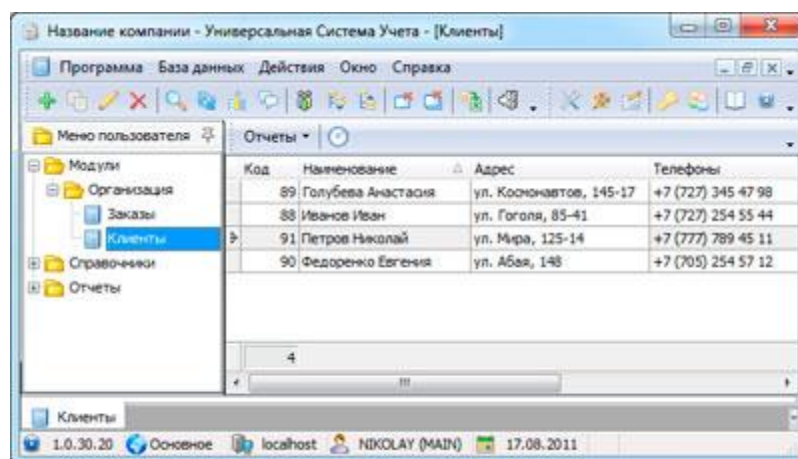


Рисунок 1.2 – Интерфейс 2

Далее представлен краткий перечень возможностей программы Универсальная Система Учета. В зависимости от конфигурации разработанного программного обеспечения список возможностей может меняться.

- большое количество функций для учета товара и продажи.
- обширный функционал, который подойдет любому типу предприятия.
- настраиваемый интерфейс, для удобства пользования каждому сотруднику предприятия.
- более пятидесяти стилей графического оформления программы.
- учет продаж и учет товара.
- окно продажи, в котором имеются такие уникальные функции, как «отложить продажу» или «товар, который спрашивали».
- взаимодействие с ТСД и сканером штрих-кодов.
- проведение инвентаризационных процессов посредством УСУ.
- контроль склада.
- оформление прихода товара и введения первоначальных остатков.
- печать накладных и заявок на закупку продукции.
- прикрепление документации в неограниченных количествах.
- печать документации, с логотипом и реквизитами предприятия.
- ведение клиентской базы.
- СМС - рассылки, голосовые звонки.
- работа с несколькими прайс-листами одновременно.
- ведение базы поставщиков.
- формирования отчетов по продаже, большое количество денежных отчетов.
- графики и диаграммы, которые отображают статистику магазина и товара наглядно [11].

Проанализировав данную систему можно отметить, что главным преимуществом этого продукта является многофункциональность, недостатком можно отметить высокую стоимость

1.3 Постановка задачи

Исходя из проведенного ранее анализа были определены вопросы к разработанной информационной системе:

- учет продажи товаров;
- хранение и вывод информации о должностях, количестве штатных единиц, стаже сотрудников;
- добавление/удаление сотрудников;
- хранение и вывод информации о сотрудниках - ФИО, ИИН, дата рождения, адрес, контактный номер, банковский счет;
- ведение номенклатуры;
- добавление номенклатуры;
- ведение отчетности по продажам сотрудников;
- ведение отчетности по чекам за день/месяц;
- ведение отчетности по заказам;
- предоставить информацию о наличии товаров;
- добавление/удаление товара;
- предоставить информацию о заработной плате сотрудников;
- хранение и вывод информации о покупке - номер чека, тип оплаты, валюта оплаты, информация о возврате, дата покупки;
- функционал, разчитанный для текущей предметной области;
- во избежания ошибок с ПО решено разработать интерфейс в виде веб-приложения.

Требования к информационному обеспечению:

- необходимо исследовать тему учета продаж в торговой отрасли и структуру супермаркета и на их основе построить логическую модель базы данных с сущностями, которая должна отвечать следующим критериям:
 - сущность и атрибут именуется в именительном падеже;
 - сущность определяется первичным ключем;
 - между собой сущности должны иметь связь: один-к-одному, один-ко-многим, многие-к-одному, многие-ко-многим;
 - СУБД должна обеспечивать безопасность, целостность данных, их сохранность;
 - столбцам таблиц указать тип данных, размер.

На основе логической строится физическая модель базы данных:

- наименования, атрибуты, столбцы именуется на английском языке;
- исключается связь многие-ко-многим.

Требования к программному обеспечению:

- Windows (32-bit или 64-bit): Windows 8 / Windows 7 / Windows Server 2008 / Windows Vista / Windows XP SP3;
- Минимальные аппаратные требования: 500 МБ свободной RAM и 4 ГБ свободного места на HDD;
- наличие Microsoft Visual C++ 2005-2008-2010-2012-2013-2015-2019 Re-distributable Package;
- наличие браузера;
- наличие OpenServer;

2 Проектная часть

2.1 Описание предметной области

Предметной областью, описанной в данном дипломном проекте, является супермаркет. Как было сказано выше, для данного супермаркета необходимо разработать информационную систему для автоматизации учета продаж.

Супермаркет занимается продажей одежды. Данная компания имеет свои отделения в различных точках города. Рабочий процесс начинается в 10:00 и заканчивается в 22:00. Раз в месяц проходит инвентаризация. В супермаркете работают сотрудники: два администратора, два кассира и 8 продавцов-консультантов.

Каждый сотрудник имеет свой идентификационный номер. Наряду с клиентами они являются основными объектами в осуществлении бизнес-процесса супермаркета.

В обязанности администратора супермаркета входят:

- организация бесперебойной работы торгового зала;
- руководство персоналом;
- контроль состояния торгового зала, в том числе чистоты и порядка, соблюдения санитарно-гигиенических требований;
- консультирование покупателей;
- помощь в проведении инвентаризации;
- предотвращение и разрешение конфликтных ситуаций;
- обучение продавцов, организация и контроль их работы в торговом зале;

- заказ и приемка товаров;

Обязанности кассира:

- осуществление денежных операций;
- прием и выдача денег клиенту;
- определение подлинности купюр;
- работа с пластиковыми банковскими картами;
- учет и контроль денежной наличности;
- ведение кассовой отчетности, оформление отчетных документов;
- инкассация денег и передача их инкассатору.

В должностные обязанности продавца-консультанта входит:

- консультирование покупателей;
- приём и выкладка товара;
- продажа товара;
- ведение отчетности;

- поддержание чистоты в супермаркета;
- участие в инвентаризации.

Покупателям помогают продавцы-консультанты в выборе одежды, после чего консультант отправляет клиента на кассу, где производится регистрация покупок и прием денег, а также регистрируется сам консультант по идентификационному номеру. Консультант помогает кассиру в упаковке товара. Каждый товар имеет свой уникальный код, который однозначно его идентифицирует - штрихкод. Кроме того, товар имеет артикул. Артикул товара - это сочетание букв, цифр, символов, которое обозначает данную модель товара.

По завершении рабочего дня кассир производит закрытие смены на всех РМ типа «Касса» и осуществляет снятие на них сменных отчетов. Также снимает Z-отчеты с банковских терминалов. Кассир также заполняет кассовую книгу, формирует отчет о продажах и далее отправляет по электронной почте все имеющиеся отчеты в офис руководству. Директор получает из системы, либо по почте аналитические отчеты, на основе которых производится принятие управленческих решений.

На основе данных о текущих продажах и об остатках товара в супермаркете, используя базовые аналитические отчеты, кассир или администратор формирует заказы.

БД после создания должна отвечать на следующие основные запросы:

- а) в каком количестве был продан тот или иной товар за определенное время;
- б) на какую сумму была продажа за определенное время;
- в) список сотрудников супермаркета;
- г) какое количество фискальных/банковских чеков было выписано;
- д) в каком количестве имеется товар;
- е) сумма продаж определенного сотрудника за определенный период;
- ж) информация о держателях бонусных карт и их бонусах;
- з) список сотрудников определенной должности;
- и) общая сумма продаж по терминалу определенного банка;
- к) информация о инвентаре супермаркета.

2.2 Проектирование функциональной структуры

Организационная структура супермаркета показана на рисунке 2.1.

Исходя из организационной структуры супермаркета в данной ИС можно выделить следующие подсистемы:

- 1) Подсистема «Должность». Основные функции: хранение и вывод информации о должностях, количестве штатных единиц, стаже, подсистема связана с подсистемой «Сотрудник»

2) Подсистема «Сотрудник». Основные функции: хранение и вывод информации о сотрудниках - ФИО, ИИН, дата рождения, адрес, контактный номер, банковский счет.

3) Подсистема «Товар». Основная функция данной подсистемы - это хранение, изменение и вывод информации о товаре - наименовании, артикул, цена.

4) Подсистема «Номенклатура». Основные функции: хранение и вывод информации о товаре - штрихкод, количество единиц.

5) Подсистема «Фискальный чек». Основные функции: хранение и вывод информации о покупке - номер чека, тип оплаты, валюта оплаты, информация о возврате, дата покупки.

Функциональная система изображена на рисунке 2.2.

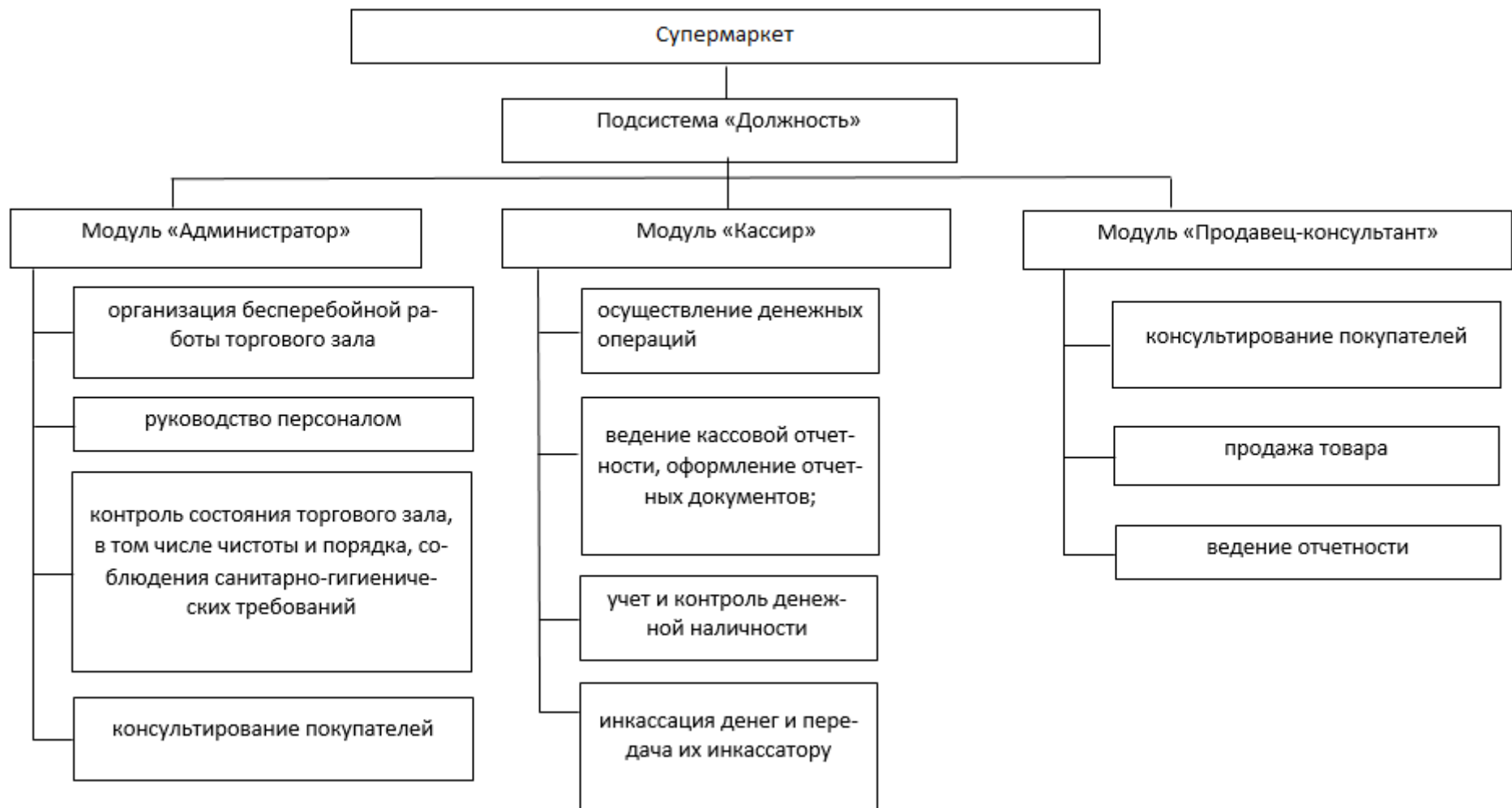


Рисунок 2.1 - Организационная структура

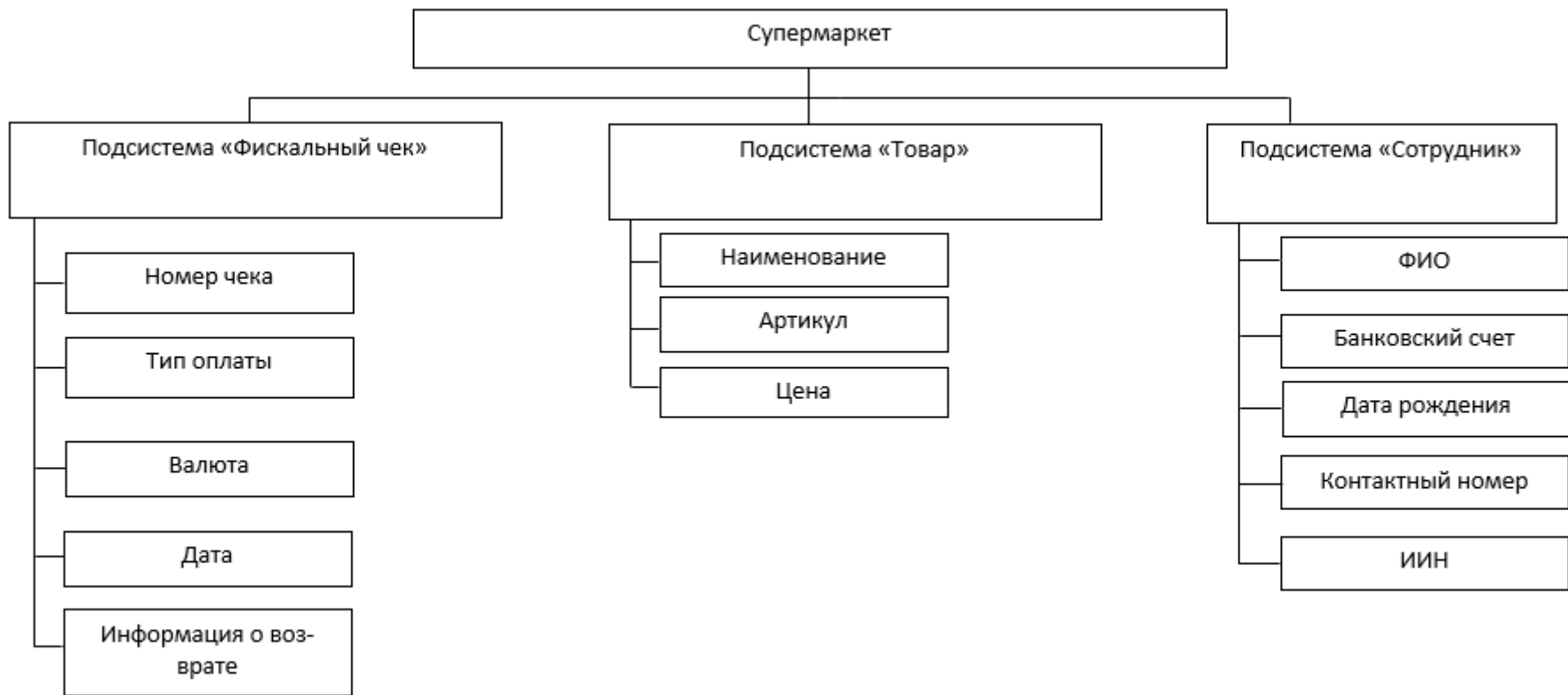


Рисунок 2.2 - Функциональная структура

2.2 Разработка бизнес модели ИС

2.2.1 Предметная область

Информационная система учета продаж супермаркета представляет собой ряд таблиц в которых хранятся такие данные как фискальный чек, номенклатура, товар, сотрудники и т.д

2.2.2 Предпроектное обследование

2.2.2.1 Функциональная и информационная модели

2.2.2.1.1 Функциональная модель

Главное назначение информационной системы супермаркета является обеспечение доступа к информации о товарах в супермаркете, покупках, сотрудниках.

Сотруднику предоставлен доступ к данным к которым ему можно получить доступ через веб-приложение, которая через sql-запросы выводит нужную информацию.

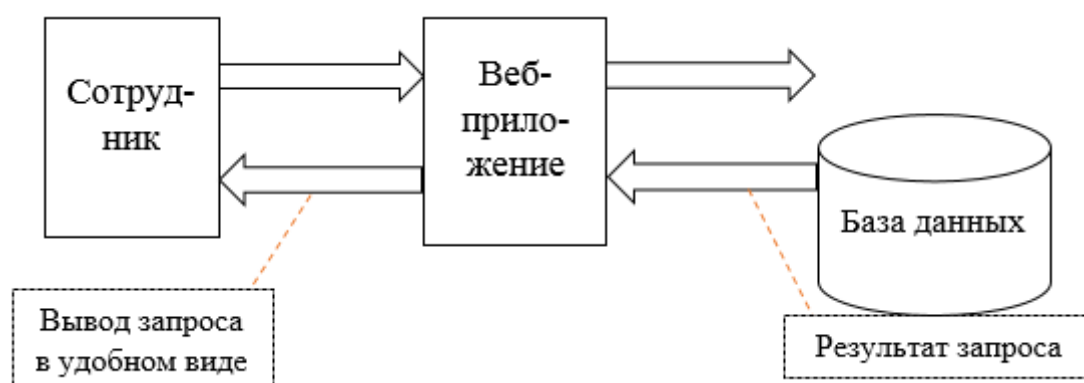


Рисунок 2.3 - Функциональная модель

2.2.2.1.2 Информационная модель

Информационная модель

Информационная модель подсистемы «Товары» для сотрудника



Рисунок 2.4 - Информационная модель подсистемы «Товары» для сотрудника

2.2.3 Выделение бизнес-процессов

Необходимо выделить бизнес-процессы, отобразить их в виде схем.

В работе выделяются следующие бизнес-процессы: «Предоставление информации о товарах», «Предоставление информации о продажах», «Просмотр проданных товаров отдельными сотрудниками». Просмотр информации о товарах (рисунок 2.5).

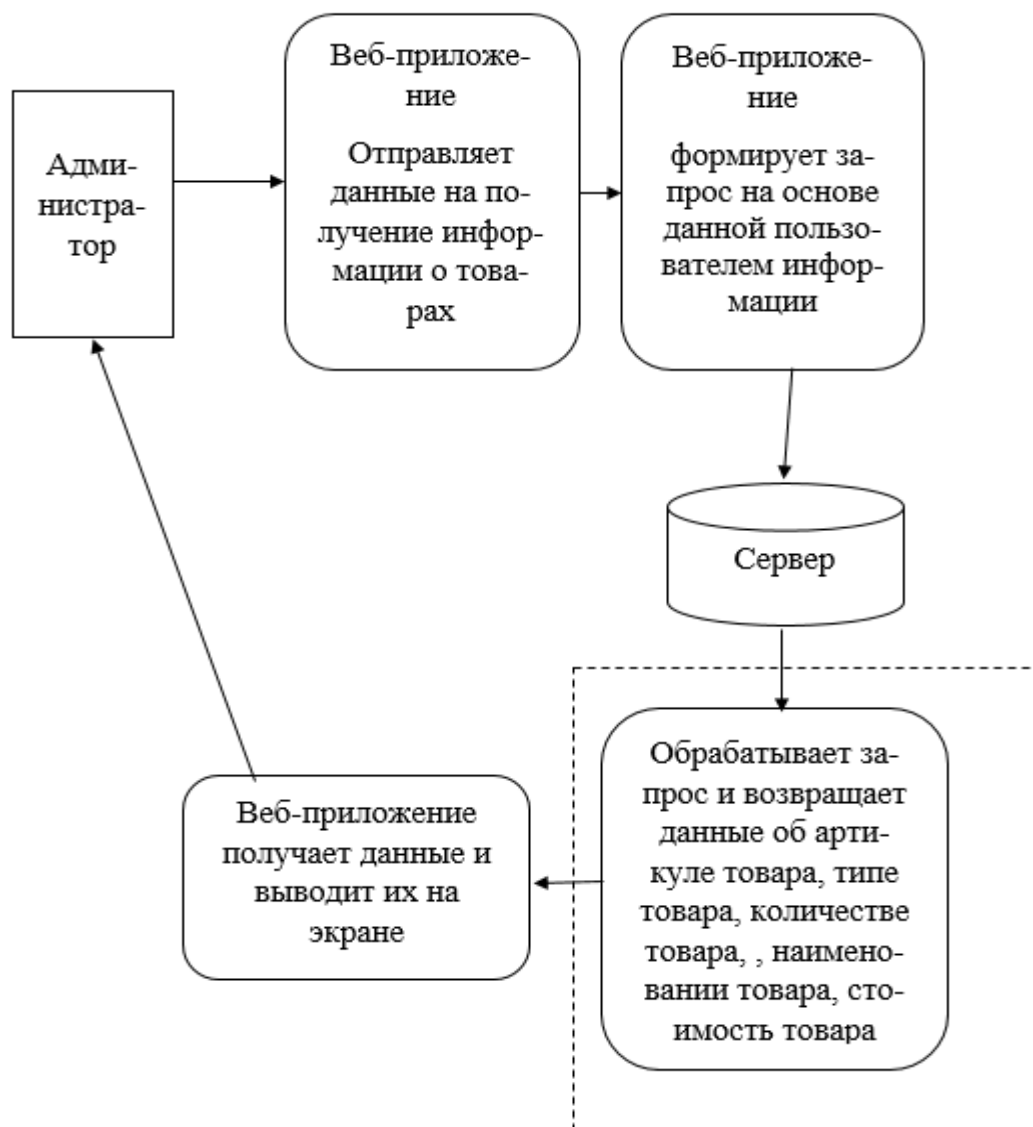


Рисунок 2.5 - Бизнес-процесс «Просмотр информации о товарах»

2.2.4 Анализ и оптимизация бизнес-процессов

Необходимо провести анализ выявленных бизнес-процессов, провести оптимизацию бизнес-процессов и составить их схемы.

Таблица 1 – анализ бизнес процессов

Произведем оценку критических факторов успеха работы подсистем. Бизнес-процессы	Эффективное использование на компьютере	Автономный режим функционирования	Удобство отображения	Совместимость (поддержка ОС)
---	---	-----------------------------------	----------------------	------------------------------

Продолжение таблицы 1

Предоставление информации кассиру	5	4	4	4
Предоставление информации администратору	5	5	4	5

Просмотр информации о товаре на рисунке 2.6.

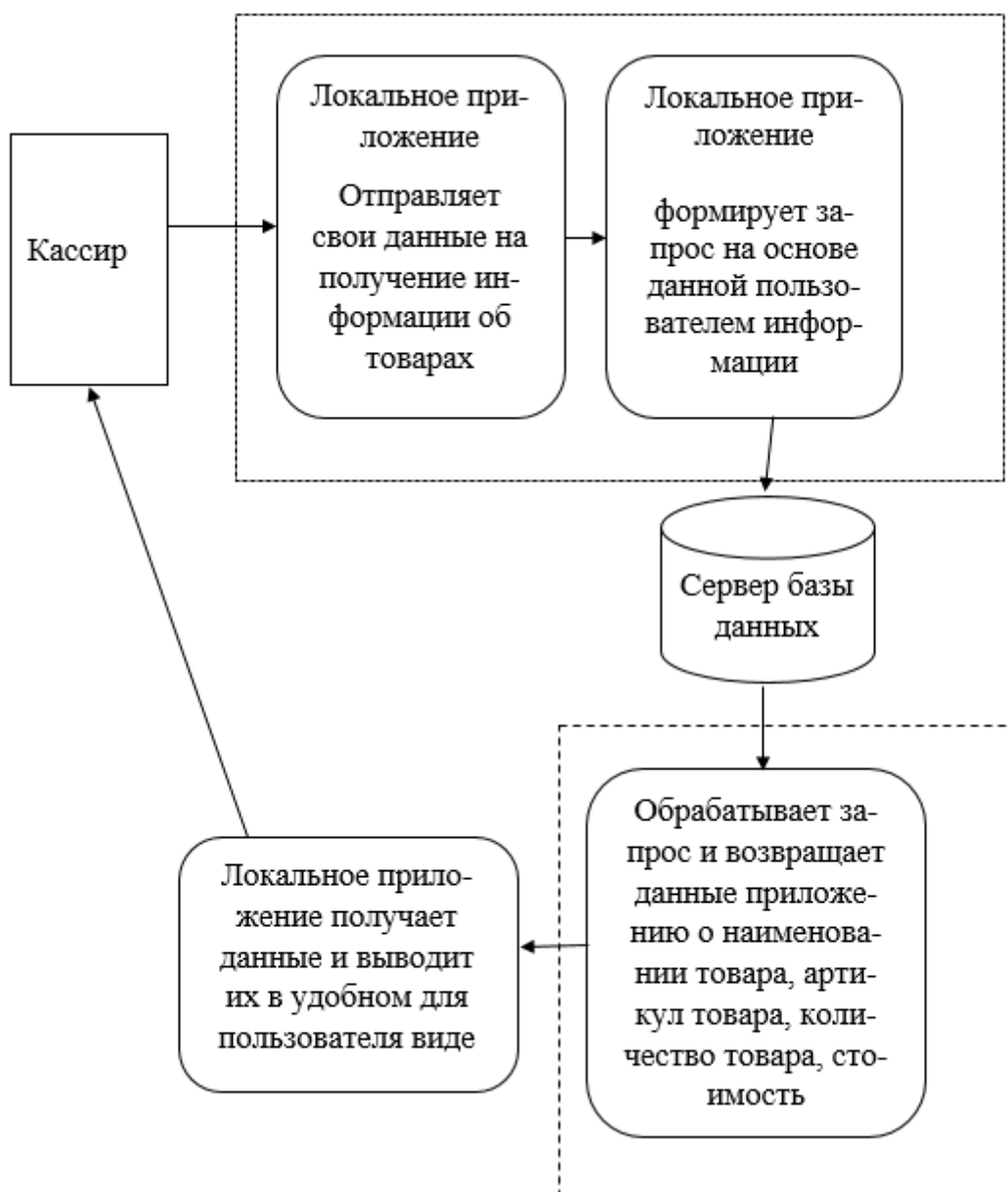


Рисунок 2.6 - Бизнес-процесс «Просмотр информации о товарах»

2.3 Разработка информационного обеспечения

2.3.1 Диаграмма UML

Диаграмма UML - это диаграмма, основанная на UML (унифицированном языке моделирования), целью которого является визуальное представление системы, визуализация происходит вместе с основными участниками, действиями, ролями, классами и артефактами. Данные диаграммы помогают лучше понимать, поддерживать, изменять и документировать информацию о системе. Основными диаграммами, которые описывают систему, являются диаграммы вариантов использования, диаграммы последовательной и кооперативные диаграммы. Говоря элементарно, UML - это модернизированный подход к программному обеспечению для моделирования и документирования. Де-факто, это одна из самых популярных методик по моделированию бизнес-процессов [1].

Существуют две наиболее широкие категории - это диаграмма поведенческого UML и диаграмма структурного UML. Они охватывают все другие типы. Исходя из названия, некоторые диаграммы UML пытаются проанализировать и изобразить структуру системы или процесса, в то время как другие описывают поведение системы, ее участников (актеров) и компонентов ее построения. Различные типы разбиты следующим образом [2]:

- диаграмма классов;
- диаграмма компонентов;
- диаграмма композитной/составной структуры;
- диаграмма развёртывания;
- диаграмма объектов;
- диаграмма пакетов;
- диаграмма деятельности;
- диаграмма автомата;
- диаграмма вариантов использования;
- диаграммы коммуникации и последовательности;
- диаграмма обзора взаимодействия;
- диаграмма синхронизации.

2.3.2 Построение диаграммы вариантов использования

Краеугольным камнем системы являются функциональные требования, которые система выполняет. Диаграммы прецедентов используются для анализа требований к системе высокого уровня. Эти требования выражены в разных вариантах использования. Компоненты этой UML-диаграммы [1]:

- функциональные требования - представлены как варианты использования; глагол, описывающий действие;

- акторы, взаимодействующие с системой; актор может быть человеком, организацией или внутренним или внешним приложением;
- отношения между акторами и вариантами использования представлены прямыми стрелками.

В приведенных ниже рисунках показаны диаграммы UML вариантов использования для системы, описанной в техническом задании проекта. В данном случае у нас есть кассир и администратор. На рисунке 2.7 изображена UML-диаграмма вариантов использования для актора «Кассир», на рисунке 2.8 - для актора «Администратор».

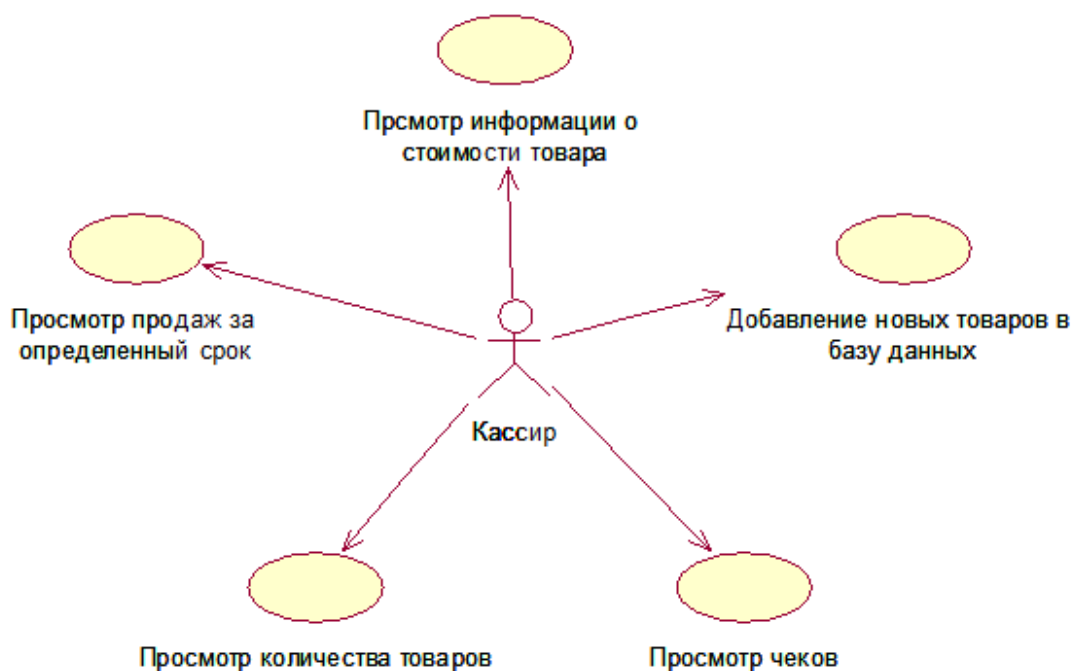


Рисунок 2.7 - UML-диаграмма вариантов использования для актора «Кассир»



Рисунок 2.8 - UML-диаграмма вариантов использования для актора «Администратор»

В круглых контейнерах выражаются действия, которые выполняют актеры. Такими действиями являются: Как можно заметить, UML-диаграммы сценариев использования хороши для демонстрации динамического поведения между участниками внутри системы, упрощая представление системы и не отражая детали реализации.

2.3.3 Построение диаграммы последовательностей и кооперативной диаграммы

Диаграммы последовательности, вероятно, являются наиболее важными диаграммами UML не только в компьютерном мире, но и в качестве моделей уровня разработки для разработки бизнес-приложений. В последнее время они стали популярными в описании бизнес-процессов из-за своей наглядности.

Исходя из названия, диаграммы последовательности описывают последовательность сообщений и взаимодействий, которые происходят между субъектами и объектами.

Поскольку основными компонентами являются сообщения, которыми обмениваются объекты, диаграммы связи строятся точно также как и диаграммы последовательности. Единственное различие между двумя вышеназванными диаграммами состоит в том, что объекты на диаграммах связи показаны с помощью ассоциативных соединений [1].

Основными и наиболее важными прецедентами для актора «Администратор» являются «Просмотр штатных единиц». Построим

диаграмму последовательностей. На рисунке 2.9 и 2.10 продемонстрирована диаграмма последовательности и кооперативная диаграмма соответственно для прецедента «Просмотр штатных единиц».

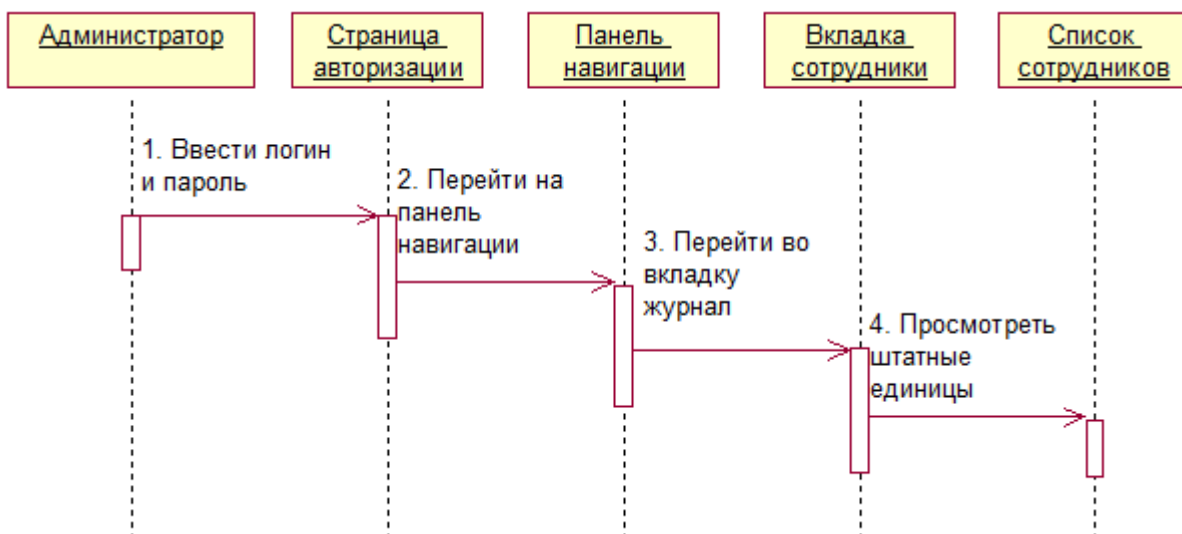


Рисунок 2.9 - UML-диаграмма последовательности для прецедента «Просмотр штатных единиц»



Рисунок 2.10 - Кооперативная диаграмма для прецедента «Просмотр штатных единиц»

На рисунке 2.11 и 2.12 продемонстрирована диаграмма последовательности и кооперативная диаграмма соответственно для прецедента «Просмотр артикула товара».

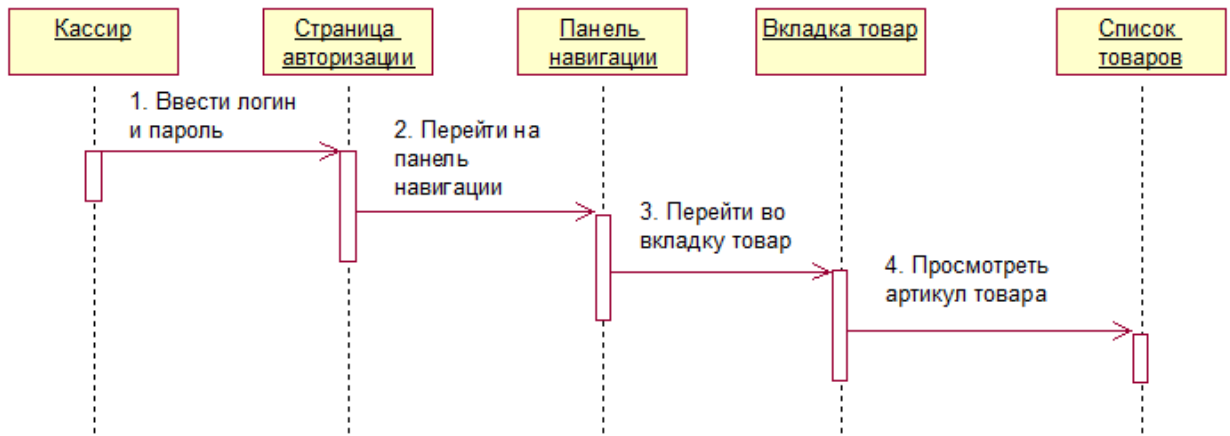


Рисунок 2.11 - UML-диаграмма последовательности для прецедента «Просмотр артикула товара»

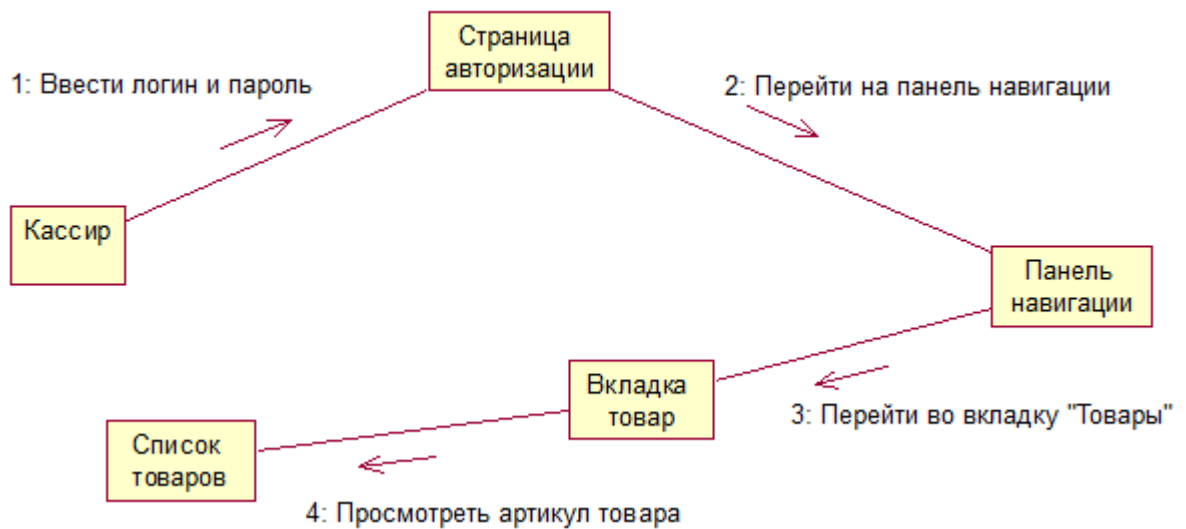


Рисунок 2.12 - Кооперативная диаграмма для прецедента «Просмотр артикула товара»

2.3.4 Обоснование выбора СУБД

Проанализируем следующие наиболее популярные СУБД на данный момент времени и на основе данного анализа выберем подходящую СУБД для нашей информационной системы:

- Oracle;
- PostgreSQL;
- Microsoft SQL Server;
- MySQL.

Изучив литературу по СУБД Oracle [3], можно выделить следующие плюсы данной системы:

- инновации для повседневной работы. С Oracle 12c в качестве гибридного облачного программного обеспечения ежедневно появляются инновационные технологии облачных вычислений;

- сильная техническая поддержка и документация. Oracle обеспечивает достойную поддержку клиентов и предоставляет исчерпывающую техническую документацию по нескольким ресурсам. Таким образом, благодаря этому, можно найти решения любых возникающих проблем;

- большая емкость. Благодаря недавно выпущенной функции мультиарендности, архитектура базы данных теперь упрощает упаковку многих баз данных и плавное управление ими;

Однако, основополагаясь на вышеназванные достоинства СУБД Oracle, стоит отметить, что для информационной системы мобильного приложения нет необходимости использовать облачные вычисления. Большая емкость - это весомое достоинство, однако для приложения не нужно обрабатывать большие корпоративные данные. Кроме того, у данной СУБД имеются существенные недостатки [3]:

- высокая стоимость. Стандартная версия, которая не включает в себя все доступные функции, стоит 17 500 долларов за единицу. Enterprise Edition стоит более 47000 долларов за единицу;

- ресурсоемкая. Для установки требуется не только много места на диске, но также необходимо учитывать постоянные обновления оборудования, если происходит его развертывание на месте;

- кривая тяжелого обучения. База данных Oracle – это не система, которую можно сразу начать использовать. Для этого лучше иметь сертифицированных инженеров Oracle DB.

Рассмотрим следующую в списке СУБД PostgreSQL. Прочитав и изучив статьи, посвященные данной системе управления базами данных, можно выделить следующие плюсы [3]:

- масштабируемость. Вертикальная масштабируемость является отличительной чертой PostgreSQL, в отличие от СУБД MySQL;

- поддержка пользовательских типов данных. PostgreSQL по умолчанию поддерживает большое количество типов данных, таких как JSON, XML, H-Store и другие. Кроме того, он позволяет пользователям детерминировать свои собственные типы данных;

- открытый исходный код, управляемый сообществом. Postgres является полностью открытым и поддерживается сообществом, что укрепляет его как целостную экосистему.

Так как в проектируемой базе данных приложения будут использоваться только встроенные типы данных, нет необходимости в создании пользовательских типов данных. Также у данной СУБД есть существенный минус – наличие противоречивой документации. Поскольку сообщество PostgreSQL довольно распространено, документация не соответствует одинаковым стандартам для всех функций Postgre.

Проанализировав источники по СУБД MSSQL, необходимо выделить следующие достоинства [3]:

- разнообразие версий;
- комплексное решение для бизнес-данных. MSSQL, ориентированный в основном на коммерческие решения, предоставляет множество дополнительных возможностей для бизнеса;
- богатая документация и помощь сообщества;
- поддержка облачных баз данных.

После изучения рекомендуемой литературы [3-4] по MySQL были отмечены следующие достоинства:

- бесплатная установка. Комьюнити версия MySQL бесплатна для скачивания. MySQL Community Edition – базовый набор инструментов для индивидуального использования;
- простой синтаксис и легкая сложность. Структура и стиль MySQL очень просты. Разработчики даже считают MySQL базой данных с человеческим языком;
- облако-совместимый.

Сравнение вышеперечисленных СУБД наглядно представлено в таблице 2.

Таблица 2 - Сравнение СУБД

	Структура данных	Лицензия	Документация	Масштабируемость	Другие структуры данных	К о
MySQL	SQL	GNU Generally Public License	✓ ✓	Vertical, complex	*	
Oracle	Multi-model, SQL	Proprietary	✓ ✓ ✓	Vertical	SQL	
PostgreSQL	Object-relational, SQL	Open-source	✓ ✓	Vertical	NoSQL	
MSSQL Server	T-SQL	Proprietary	✓ ✓ ✓	Vertical, complex	SQL, NoSQL	

Изучив все аспекты вышеназванных СУБД и принимая во внимание все их достоинства и недостатки, в качестве системы управления базами данных для своей информационной системы было решено выбрать СУБД MySQL. Основной причиной выбора является тот факт, что данная СУБД бесплатная, имеет довольно несложную установку и легко настраиваемая. Так как выбор основывался в основном между двумя продуктами Oracle - MySQL и СУБД Oracle, первая отличительная особенность MySQL сыграла свою нишу. Для того, чтобы развернуть свою базу данных в Oracle необходимо заплатить

немалые деньги. Кроме того, серверная часть приложения будет написана на языке программирования PHP. При помощи связки PHP + MySQL получаются быстрые приложения и сайты, работающие в режиме реального времени. Другим аспектом выбора СУБД MySQL является то, что ее синтаксис очень прост и похож на человеческий язык и для манипулирования данными используется язык SQL. Несмотря на то, что в MySQL нет многих функций, которые имеются в Oracle, разработчики постоянно занимаются усовершенствованием.

Таким образом, MySQL - это быстрая, надежная система баз данных с открытым исходным кодом, которая также предлагает большое количество функций и отлично подходит для средних баз данных.

Выбор, кроме вышеперечисленных аргументов, учитывался также из того факта, что заказчик использует СУБД MySQL.

2.3.5 Логическая и физическая модели

Для создания модели базы данных использовано CASE-средство ERwin Data Modeler. При использовании любого CASE-средства вначале строится логическая модель БД в виде диаграммы с указанием сущностей и связей между ними. Логической моделью называется абстрактный взгляд на данные, на ней данные представлены так, как выглядят в реальном мире, и могут называться так, как они называются в реальном мире, например «Супермаркет», «Сотрудник», «Товар» и т.д. Представляемые на логическом уровне объекты модели, называются сущностями и атрибутами. Логическая модель является универсальной и не связана с конкретной реализацией СУБД. На рисунке 2.12 представлена логическая модель БД «Супермаркет».

На основании полученной логической модели переходят к физической модели данных. Физическая модель данных, в отличие от логической, зависит от конкретного СУБД. Физическая модель содержит всю информацию о всех объектах БД. Физическая модель зависит от конкретного СУБД. Учитывая этот факт, одна и та же логическая модель может соответствовать нескольким различным физическим моделям. Если в логической модели не имеет значения, какой конкретно тип данных имеет атрибут, то в физической модели важно описать всю информацию о конкретных физических объектах - таблицах, колонках, индексах и т.д. Разделение модели данных на логические и физические позволяет решить несколько важных задач. Для переключения между логической и физической моделью данных служит список выбора в левой части панели инструментов. На рисунке 2.14 изображена физическая модель базы данных. На основании физической модели можно сгенерировать либо саму БД или DDL-скрипт, который, в свою очередь, может быть использован для генерации БД. [17]

Связь между сущностями характеризуется следующим набором параметров:

а) *именем* – указывается в виде глагола и определяет семантику связи;

б) *кратностью (кардинальность, мощность)*: один-к-одному (1:1), один-ко-многим и многие-ко-многим. Кратность показывает, какое количество экземпляров одной сущности определяется экземпляром другой.

в) *типом*: идентифицирующая (атрибуты одной сущности, называемые внешним ключом, входят в состав дочерней и служат для идентификации ее экземпляров, т.е. входят в ее первичный ключ) и неидентифицирующая (внешний ключ имеется в дочерней сущности, но не входит в состав первичного ключа);

г) *обязательностью*: обязательная (при вводе нового экземпляра в дочернюю сущность заполнение атрибутов внешнего ключа обязательно и для введенных значений должен существовать экземпляр в родительской сущности) и необязательная (заполнение атрибутов внешнего ключа в экземпляре дочерней сущности необязательно).

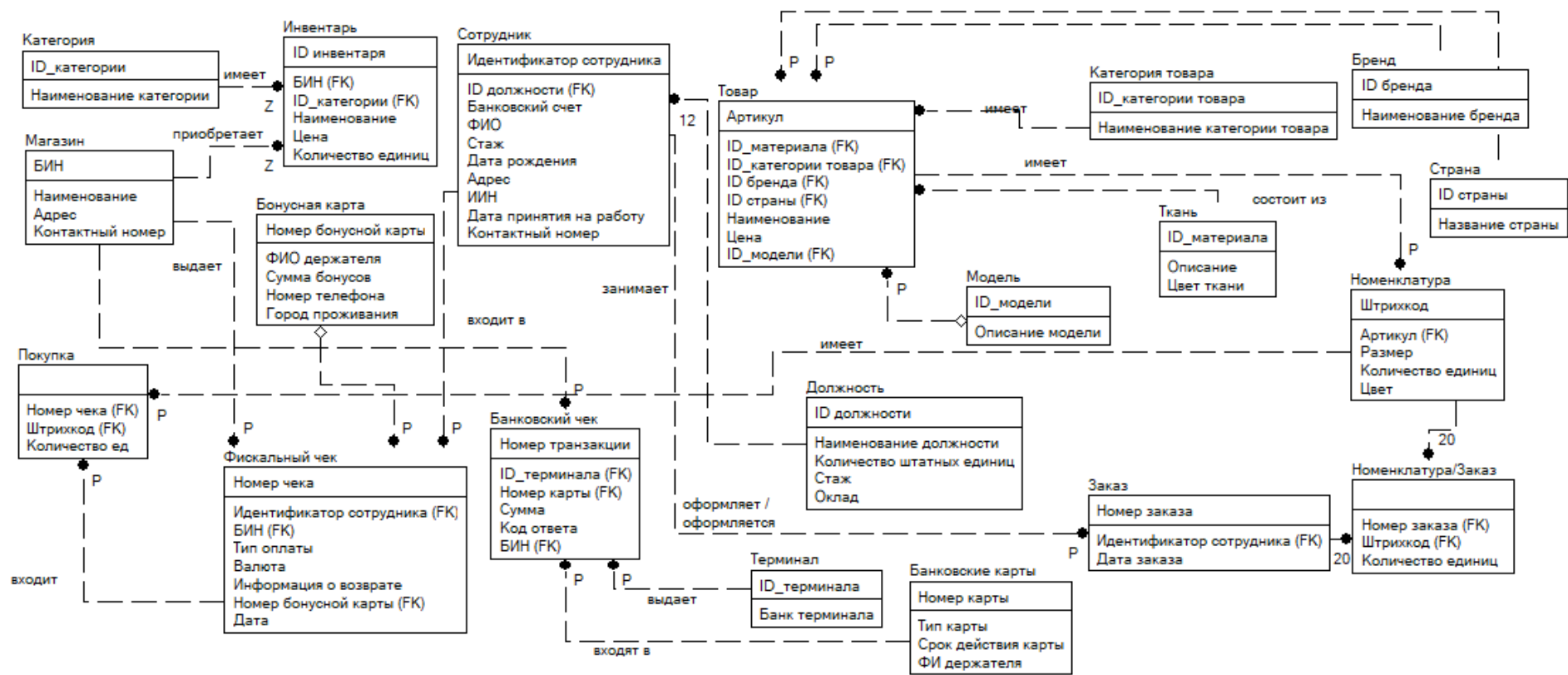


Рисунок 2.13 - Логическая модель БД в ErWin

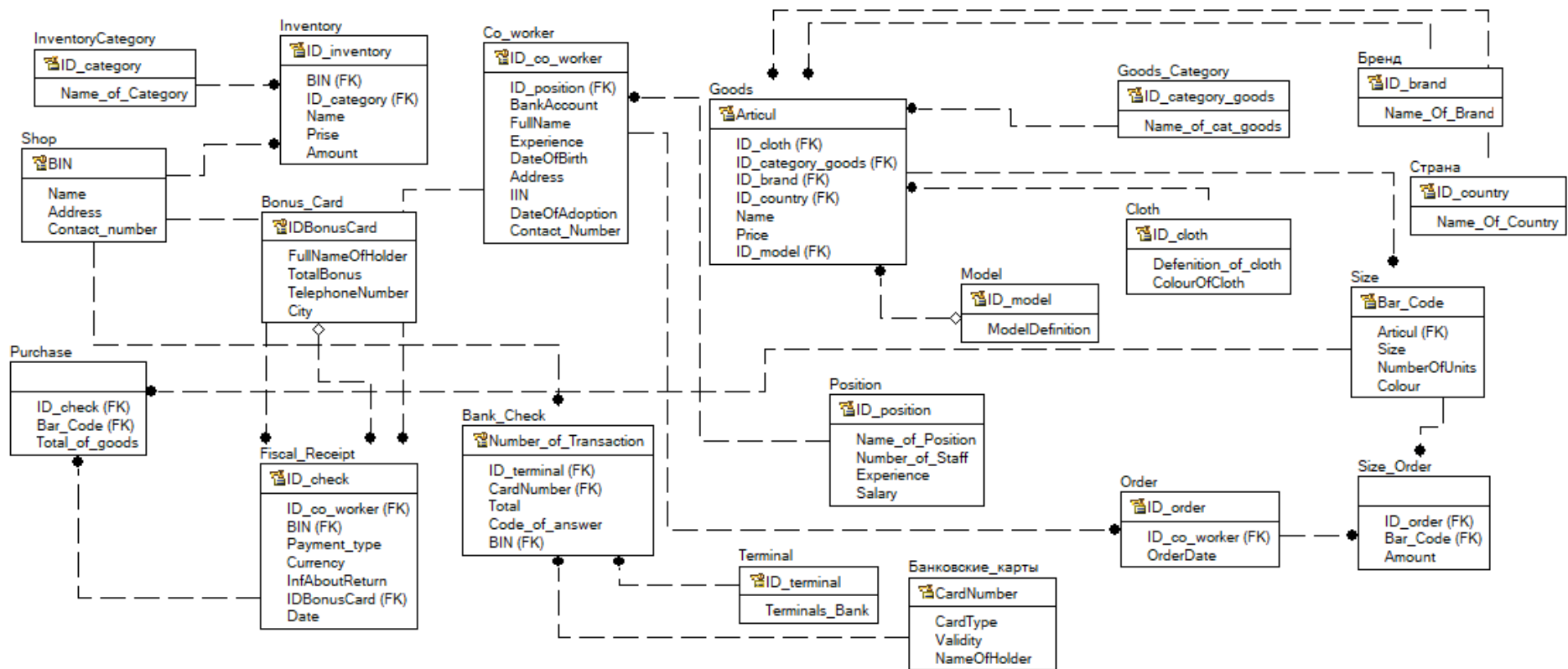


Рисунок 2.14 - Физическая модель БД в ErWin

2.3.4 Структура таблиц БД

База данных ИС состоит из 20 таблиц: POSITION (ДОЛЖНОСТЬ), INVENTORY (ИНВЕНТАРЬ), INVENTORY_CATEGORY (КАТЕГОРИЯ ИНВЕНТАРЯ), SUPERMARKET (СУПЕРМАРКЕТ), BONUSCARD (БОНУСНАЯ КАРТА), CLOTH (МАТЕРИАЛ), TERMINAL (ТЕРМИНАЛ), MODEL (МОДЕЛЬ), GOODS (ТОВАР), GOODS_CATEGORY (КАТЕГОРИЯ ТОВАРА), COUNTRY (СТРАНА), BRAND (БРЕНД), CO_WORKER (СОТРУДНИК), PURCHASE (ПОКУПКА), FISCAL_RECEIPT (ФИСКАЛЬНЫЙ ЧЕК), BANK_CHECK (БАНКОВСКИЙ ЧЕК), BANKCARD (БАНКОВСКАЯ КАРТА), ORDER (ЗАКАЗ), SIZE (НОМЕНКЛАТУРА), SIZE/ORDER (НОМЕНКЛАТУРА/ЗАКАЗ)

Рассмотрим более подробно каждую таблицу, их значение и функции.

Таблица «POSITION» («ДОЛЖНОСТЬ») предназначена для хранения данных о должностях, их наименование, количество штатных единиц, стаж, оклад. Ее структура приведена в таблице 3.

Таблица 3 - Структура таблицы «POSITION» («ДОЛЖНОСТЬ»)

№	Наименование поля	Тип поля	Размер	NULL	Назначение поля
1	ID_Position	INTEGER		NULL	Первичный ключ, хранит идентификатор должности
2	Name_of_Position	VARCHAR	20	NULL	Хранит наименование должности
3	Number_of_Staff	VARCHAR	20	NULL	Хранит количество штатных единиц
4	Experience	CHAR	18	NULL	Хранит количество опыта
5	Salary	INTEGER		NULL	Хранит данные об окладе

Таблица «INVENTORY» («ИНВЕНТАРЬ») предназначена для хранения данных об инвентаре супермаркета, наименовании, категории инвентаря, стоимость, количества единиц.

Таблица 4 - Структура таблицы «INVENTORY» («ИНВЕНТАРЬ»)

№	Наименование поля	Тип поля	Размер	NULL	Назначение поля
1	ID_inventory	INTEGER		NULL	Первичный ключ, хранит идентификатор инвентаря

Продолжение таблицы 4

2	BIN_FK	NUMERIC	12	NOT NULL	Внешний ключ из таблицы «SUPERMARKET». Хранит БИН супермаркета
3	ID_category_FK	INTEGER		NOT NULL	Внешний ключ из таблицы «InventoryCategory». Хранит идентификатор категории
4	Name	VARCHAR	20	NULL	Хранит наименование инвентаря
5	Prise	INTEGER		NULL	Хранит стоимость инвентаря
6	Amount	INTEGER		NULL	Хранит количество инвентаря

Таблица «INVENTORY_CATEGORY» («КАТЕГОРИЯ ИНВЕНТАРЯ») предназначена для хранения данных о наименовании категории инвентаря.

Таблица 5 - Структура таблицы «INVENTORY_CATEGORY» («КАТЕГОРИЯ ИНВЕНТАРЯ»)

№	Наименование поля	Тип поля	Размер	NULL	Назначение поля
1	ID_category	INTEGER		NULL	Является первичным ключом, хранит идентификатор категории

2	Name_of_Category	VAR-CHAR	20	NULL	Хранит наименование категории
---	------------------	----------	----	------	-------------------------------

Таблица «SUPERMARKET» («СУПЕРМАРКЕТ») предназначена для хранения данных о БИН, наименовании, адресе, контактного номера супермаркета.

Таблица 6 - Структура таблицы «SUPERMARKET» («СУПЕРМАРКЕТ»)

№	Наименование поля	Тип поля	Размер	NULL	Назначение поля
1	BIN	NUMERIC	12	NULL	Первичный ключ, хранит БИН супермаркета.

Продолжение таблицы 6

2	Name	VARCHAR	40	NULL	Хранит наименование супермаркета
3	Address	VARCHAR	40	NULL	Хранит адрес супермаркета
4	Contact_number	NUMERIC	11	NULL	Хранит контактный номер супермаркета

Таблица «BONUSCARD» («БОНУСНАЯ КАРТА») предназначена для хранения данных о носителях бонусных карт, их ФИО, количество бонусов, телефонный номер, город проживания. Структура таблицы приведена в таблице ниже.

Таблица 7 - Структура таблицы «BONUSCARD» («БОНУСНАЯ КАРТА»)

№	Наименование поля	Тип поля	Размер	NULL	Назначение поля
1	IDBonusCard	INTEGER		NOT NULL	Первичный ключ. Хранит идентификатор

					бонусной карты
2	FullNameOfHolder	VARCHAR	20	NULL	Хранит ФИО владельца карты
3	TotalBonus	INTEGER		NULL	Хранит данные о количестве бонусов
4	TelephoneNumber	INTEGER		NULL	Хранит данные о номере телефона владельца
5	City	VARCHAR	20	NULL	Хранит данные о городе проживания

Таблица «CLOTH» («МАТЕРИАЛ») предназначена для хранения сведений о номере аудитории, количестве компьютеров, телевизоров, посадочных мест, а также типе аудитории.

Таблица 8 - Структура таблицы «CLOTH» («МАТЕРИАЛ»)

№	Наименование поля	Тип поля	Размер	NULL	Назначение поля
1	ID_cloth	INTEGER		NULL	Первичный ключ, хранит идентификатора материала

Продолжение таблицы 8

2	Definition_of_cloth	VARCHAR	20	NULL	Хранит информацию об описании материала
3	ColourOfCloth	VARCHAR	20	NOT NULL	Хранит информацию цвете материала

Таблица «TERMINAL» («ТЕРМИНАЛ») предназначена для хранения сведений о терминале.

Таблица 9 - Структура таблицы «TERMINAL» («ТЕРМИНАЛ»)

№	Наименование поля	Тип поля	Размер	NULL	Назначение поля
1	ID_terminal	INTEGER		NULL	Первичный ключ. Хранит идентификатор терминала
2	Terminals_Bank	VARCHAR	20	NULL	Хранит информацию о наименовании терминала банка

Таблица «MODEL» («МОДЕЛЬ») предназначена для хранения данных о модели одежды. Структура таблицы указана ниже.

Таблица 10 - Структура таблицы «MODEL» («МОДЕЛЬ»)

№	Наименование поля	Тип поля	Размер	NULL	Назначение поля
1	ID_model	CHAR	18	NULL	Является первичным ключом, хранит информацию о номере занятия
2	ModelDefinition	VARCHAR	20	NULL	Хранит информацию об описании модели

Таблица «GOODS» («ТОВАР») предназначена для хранения данных об артикуле товара, наименования, стоимости, а также идентификаторов материала, категории, модели, бренда, страны-производства.

Таблица 11 - Структура таблицы «GOODS» («ТОВАР»)

№	Наименование поля	Тип поля	Размер	NULL	Назначение поля
1	Articul	VAR-CHAR	20	NULL	Первичный ключ, хранит артикул товара

2	Name	VAR-CHAR	20	NULL	Хранит наименование товара
3	Price	INTEGER		NULL	Хранит стоимость товара
4	ID_cloth	INTEGER		NOT NULL	Внешний ключ из таблицы «CLOTH», хранит идентификатор материала
5	ID_category_goods	INTEGER		NOT NULL	Внешний ключ из таблицы «GOODS_CATEGORY», хранит идентификатор категории товара
6	ID_model	CHAR	18	NULL	Внешний ключ из таблицы «MODEL», хранит идентификатор модели
7	ID_brand	INTEGER		NOT NULL	Внешний ключ из таблицы «BRAND», хранит идентификатор брэнда
8	ID_country	INTEGER		NOT NULL	Внешний ключ из таблицы «COUNTRY», хранит идентификатор страны

Таблица «GOODS_CATEGORY» («КАТЕГОРИЯ ТОВАРА») предназначена для хранения данных о идентификаторе и наименовании категории товара.

Таблица 12 - Структура таблицы «GOODS_CATEGORY» («КАТЕГОРИЯ ТОВАРА»)

№	Наименование поля	Тип поля	Размер	NULL	Назначение поля
---	-------------------	----------	--------	------	-----------------

1	ID_category_goods	INTEGER		NULL	Первичный ключ, хранит идентификатор категории
2	Name_of_cat_goods	VARCHAR	20	NULL	Хранит наименование категории товара

Таблица «COUNTRY» («СТРАНА») предназначена для хранения данных о стране производства.

Таблица 13 - Структура таблицы «COUNTRY» («СТРАНА»)

№	Наименование поля	Тип поля	Размер	NULL	Назначение поля
1	ID_country	INTEGER		NULL	Первичный ключ, хранит идентификатор страны
2	Name_Of_Country	VAR-CHAR	20	NULL	Хранит информацию о названии страны производства

Таблица «BRAND» («БРЕНД») предназначена для хранения данных о брэнде товара.

Таблица 14 - Структура таблицы «BRAND» («БРЕНД»)

№	Наименование поля	Тип поля	Размер	NULL	Назначение поля
1	ID_brand	INTEGER		NOT NULL	Первичный ключ, хранит идентификатор брэнда
2	Name_Of_Country	VAR-CHAR	20	NULL	Хранит информацию о названии брэнда

Таблица «CO_WORKER» («СОТРУДНИК») предназначена для хранения данных о полном имени сотрудника, опыт работы, банковский счет,

опыт работы, дата рождения, адрес проживания, ИИН, дата принятия на работу, контактный номер.

Таблица 15 - Структура таблицы «CO_WORKER» («СОТРУДНИК»)

№	Наименование поля	Тип поля	Размер	NULL	Назначение поля
1	ID_co_worker	NUMERIC	13	NULL	Первичный ключ, хранит идентификатор сотрудника
2	BankAccount	VAR-CHAR	20	NULL	Хранит информацию о банковском счете
3	FullName	VAR-CHAR	20	NULL	Хранит полное имя сотрудника
4	Experience	CHAR	18	NULL	Хранит опыт работы
5	ID_position	INTEGER		NOT NULL	Внешний ключ из таблицы «POSITION», хранит идентификатор должности
6	DateOfBirth	DATE		NULL	Хранит дату рождения
7	Address	VAR-CHAR	20	NULL	Хранит адрес проживания
8	IIN	INTEGER		NULL	Хранит ИИН сотрудника
9	DateOfAdoption	DATE		NULL	Хранит дату принятия на работу
10	Contact_Number	INTEGER		NULL	Хранит контактный номер сотрудника

Таблица «PURCHASE» («ПОКУПКА») предназначена для хранения данных о покупке.

Таблица 16 - Структура таблицы «PURCHASE» («ПОКУПКА»)

№	Наименование поля	Тип поля	Размер	NULL	Назначение поля
1	ID_check	INTEGER		NOT NULL	Внешний ключ из таблицы «FISCAL_RECEIPT», хранит идентификатор чека
2	Bar_Code	INTEGER		NOT NULL	Внешний ключ из таблицы «Size», хранит информацию о баркоде товара
3	Total_of_goods	INTEGER		NULL	Хранит информацию о количестве товаров в покупке

Таблица «FISCAL_RECEIPT» («ФИСКАЛЬНЫЙ ЧЕК») предназначена для хранения данных о чеке: тип оплаты, валюта, информация о возврате, дата покупки

Таблица 17 - Структура таблицы «FISCAL_RECEIPT» («ФИСКАЛЬНЫЙ ЧЕК»)

№	Наименование поля	Тип поля	Размер	NULL	Назначение поля
1	ID_check	INTEGER		NULL	Первичный ключ, хранит идентификатор чека
2	ID_co_worker_FK	NUMERIC	13	NOT NULL	Внешний ключ из таблицы «CO_WORKER», хранит информацию об идентификаторе сотрудника

3	BIN_FK	NUMERIC	12	NOT NULL	Внешний ключ из таблицы «SUPERMARKET», хранит БИН супермаркета
---	--------	---------	----	----------	--

Продолжение таблицы 17

4	Payment_type	VAR-CHAR	20	NULL	Хранит тип оплаты
5	Currency	VAR-CHAR	20	NULL	Хранит валюту оплаты
6	InfAboutReturn	VAR-CHAR	20	NULL	Хранит информацию о возврате товара
7	IDBonusCard	INTEGER		NULL	Хранит идентификатор бонусной карты, при его наличии
8	Date	DATE		NULL	Хранит дату покупки

Таблица «BANK_CHECK» («БАНКОВСКИЙ ЧЕК») предназначена для хранения данных о банковском чеке: номер транзакции, идентификатор терминала, номер карты, сумма, код ответа, БИН супермаркета.

Таблица 18 - Структура таблицы «BANK_CHECK» («БАНКОВСКИЙ ЧЕК»)

№	Наименование поля	Тип поля	Размер	NULL	Назначение поля
---	-------------------	----------	--------	------	-----------------

1	Number_of_Transaction	INTEGER		NULL	Первичный ключ, хранит номер транзакции
2	ID_terminal_FK	INTEGER		NOT NULL	Хранит информацию о названии бренда

Продолжение таблицы 18

3	Card-Number_FK	INTEGER		NOT NULL	Внешний ключ из таблицы «BANK_CARD», хранит номер карты
4	Total	INTEGER		NULL	Хранит информацию о сумме чека
5	Code_of_answer	VARCHAR	3	NULL	Хранит код ответа
6	BIN_FK	NUMERIC	12	NOT NULL	Внешний ключ из таблицы «SUPERMARKET», хранит БИН супермаркета

Таблица «BANKCARD» («БАНКОВСКАЯ КАРТА») предназначена для хранения данных о банковской карте: номер карты, тип оплаты, срок действия карты, ФИ держателя.

Таблица 19 - Структура таблицы «BANKCARD» («БАНКОВСКАЯ КАРТА»)

№	Наименование поля	Тип поля	Размер	NULL	Назначение поля
1	CardNumber	INTEGER		NULL	Первичный ключ, хранит номер карты

2	CardType	VAR-CHAR	20	NULL	Хранит тип карты
3	Validity	DATE		NULL	Хранит срок годности карты

Продолжение таблицы 19

4	NameOfHolder	VAR-CHAR	20	NULL	Хранит ФИ владельца
---	--------------	----------	----	------	---------------------

Таблица «ORDER» («ЗАКАЗ») предназначена для хранения данных заказе.

Таблица 20 - Структура таблицы «ORDER» («ЗАКАЗ»)

№	Наименование поля	Тип поля	Размер	NULL	Назначение поля
1	ID_order	INTEGER		NULL	Первичный ключ, хранит идентификатор заказа
2	OrderDate	DATE		NULL	Хранит информацию о дате заказа
3	ID_co_worker	NUMERIC	13	NOT NULL	Хранит идентификатор сотрудника

Таблица «SIZE» («НОМЕНКЛАТУРА») предназначена для хранения данных о номенклатуре: штрихкод товара, количество единиц, размер, цвет.

Таблица 21 - Структура таблицы «SIZE» («НОМЕНКЛАТУРА»)

№	Наименование поля	Тип поля	Размер	NULL	Назначение поля
---	-------------------	----------	--------	------	-----------------

1	Bar_Code	INTEGER		NOT NULL	Первичный ключ, хранит штрихкод товара
2	Articul_FK	VAR- CHAR	20	NOT NULL	Внешний ключ из таблицы «GOODS», хранит артикул товара

Продолжение таблицы 21

3	Size	INTEGER		NULL	Хранит информацию о размере товара
4	NumberOfUnits	INTEGER		NULL	Хранит количество единиц товара
5	Colour	INTEGER		NULL	Хранит информацию о цвете товара

Таблица «SIZE/ORDER» («НОМЕНКЛАТУРА/ЗАКАЗ») предназначена для хранения данных о количестве товара в заказе.

Таблица - 22 - Структура таблицы «SIZE/ORDER»

№	Наименование поля	Тип поля	Размер	NULL	Назначение поля
1	ID_order_FK	INTEGER		NOT NULL	Внешний ключ из таблицы «ORDER», хранит идентификатор заказа
2	Bar_Code_FK	INTEGER		NOT NULL	Внешний ключ из таблицы «SIZE», хранит штрихкод товара
3	Amount	INTEGER		NULL	Хранит информацию о количестве товара в заказе

3 Программная реализация

3.1 Обоснование выбора среды программирования

Информационная система «Supermarket» состоит из активностей(activity), фрагментов(fragment), отдельных классов(class) и макетов(layout). Макет состоит из отдельных объектов, которые позиционируются на экране и определяют пользовательский интерфейс. Макет создается из формализованного языка XML (eXtensible Markup Language) - расширяемый язык разметки.

Для создания модели базы данных использовано CASE-средство ERwin Data Modeler.

Данное CASE средство является лидером среди аналогичных систем, около 60% средств моделирования данных мирового рынка принадлежит AllFusion Erwin Data Modeler.

AllFusion Erwin Data Modeler не только помогает в дизайне (изображении) логической модели данных, он поддерживает дизайн соответствующей физической модели данных и автоматически генерирует структуру физической БД («Forward Engineering»). [16]

Данное CASE средство располагает инструментарием для генерации из функционирующей физической базы данных соответствующую ей модель данных («reverse engineering»), и поддерживает обе физическую и логическую модели данных. Вышеописанное позволяет поддерживать функционирующие базы данных и выполнять миграцию всей БД или ее части (подсхемы БД) на другие серверные платформы. [16]

В качестве СУБД использована MySQL. MySQL - популярная и хорошо распространенная система управления базами данных (СУБД). MySQL обладает надежностью, гибкостью, высокой рабочей скоростью. Работа с ней, как правило, не вызывает больших трудностей. Поддержка сервера MySQL автоматически включается в поставку PHP. Приложение на PHP, использующее для хранения информации базу данных (в частности MySQL) всегда работает быстрее приложения, построенного на файлах.

В качестве сервера для созданного приложения будет использован Open Server. Open Server Panel - это портативная серверная платформа и программная среда, созданная специально для веб-разработчиков с учётом их рекомендаций и пожеланий. В качестве веб-интерфейса для администрирования СУБД MySQL используется phpMyAdmin. Приложение пользуется большой популярностью, так как позволяет управлять СУБД MySQL без непосредственного ввода SQL команд, предоставляя дружелюбный интерфейс. Серверная часть приложения будет реализована на языке PHP. Основным преимуществом языка PHP – это его практичность. PHP должен предоставлять разработчику средства для своевременного и эффективного решения поставленных задач. Практичность языка PHP обуславливается его пятью важными характеристиками:

- традиционностью;
- простотой;
- эффективностью;
- безопасностью;
- гибкостью.

Проанализировав статьи по PHP, были выделены следующие достоинства [3, 4]:

- ориентация на Web-разработку - PHP создавался, развивался и поддерживается как язык для создания Web-сайтов. Многие конструкции и решения в нем созданы для удобства работы в Web-среде.;

- он поддерживает стандарт Unicode и, следовательно, может использоваться текст из любого набора символов;

- кроссплатформенность - PHP перенесен на все основные операционные системы: можно разрабатывать сайт в Windows, Mac OS X, а эксплуатировать на Linux-сервере. Сложности переноса будут минимальны и нивелироваться языком.;

- бесплатность - PHP является разработкой из мира свободного программного обеспечения, не требуется платить ни за сам язык, ни за большинство сопутствующих программ (редакторы, Web-серверы, базы данных). Вдобавок большинство программных продуктов, с которыми приходится иметь дело, имеют доступный для изучения и модификации исходный код. Вложения могут потребоваться при аренде доменного имени и сервера для публикации сайта в Интернете.

Также были выделены и недостатки [3, 4]:

- непоследовательный синтаксис - при изучении языка PHP, особенно старой части, основанной на функциях, можно заметить, что часть функций имеет префиксы array_, str_, часть не имеет. Параметры функций могут быть расположены не совсем логично и не так, как в другой функции этой же группы;

- PHP - уже довольно долго живущий язык. Когда язык только появляется, он довольно элегантный и внутренне согласованный. По мере жизненного цикла язык обрастает дополнительными ключевыми словами, артефактами, устаревшими конструкциями, которые вроде есть, работают, но которыми не рекомендуется пользоваться.

Интерфейс написан на HTML, CSS, JavaScript.

3.2 Структура программного обеспечения

Структура программного обеспечения ИС состоит из системного ПО, инструментального ПО и прикладного ПО. Для запуска веб-приложения необходимо системное программное обеспечение, включающее в себя операционную систему Windows и др. Инструментальное программное обеспечение включает в себя HTML, CSS, JavaScript, СУБД MySQL 8.0.17, и серверного языка программирования PHP 7.3. Схематично структура представлена на рисунке 3.1.

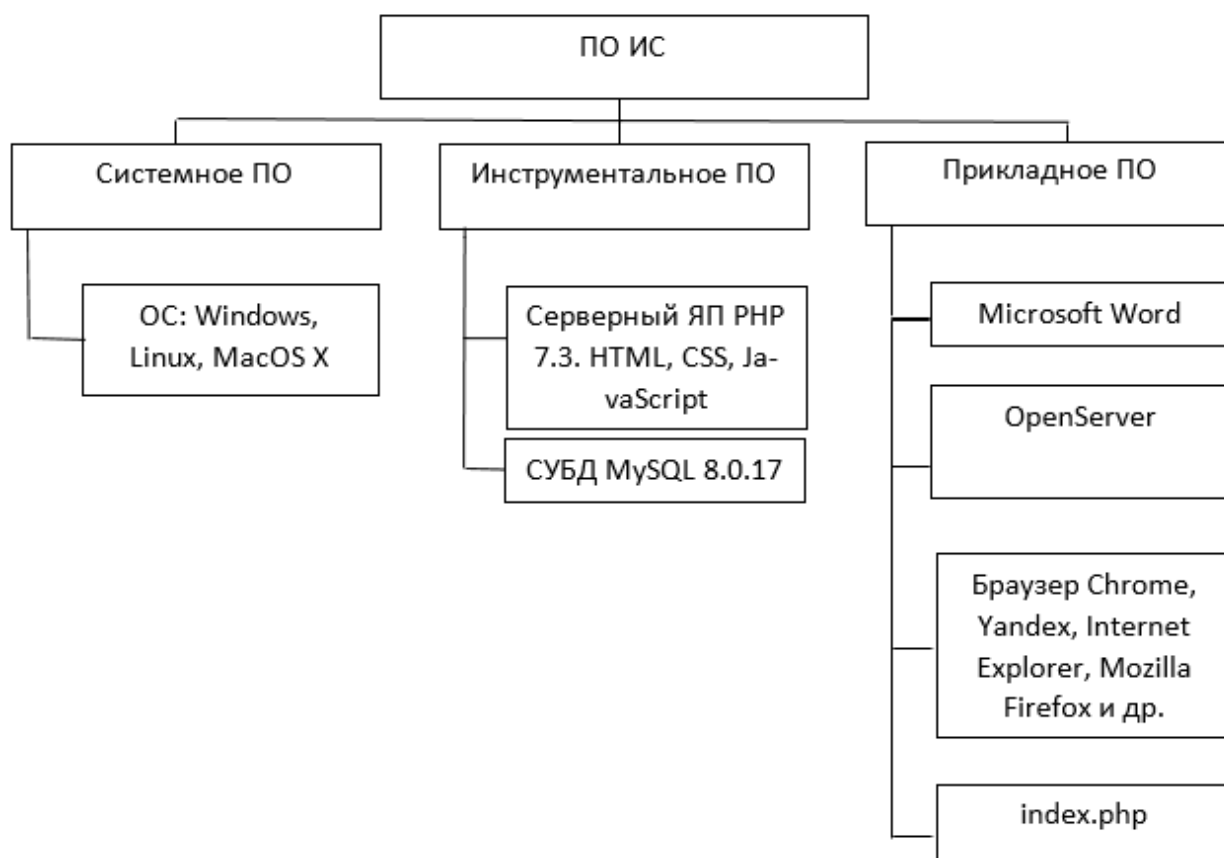


Рисунок 3.1 - Структура ПО

3.3 Разработка веб-приложения

Для начала создаем дампы базы данных и импортируем на сервер базы данных 127.0.0.1 via TCP/IP. Дамп делается при помощи утилиты mysqldump. Импорт базы данных предоставлен на рисунке 91.

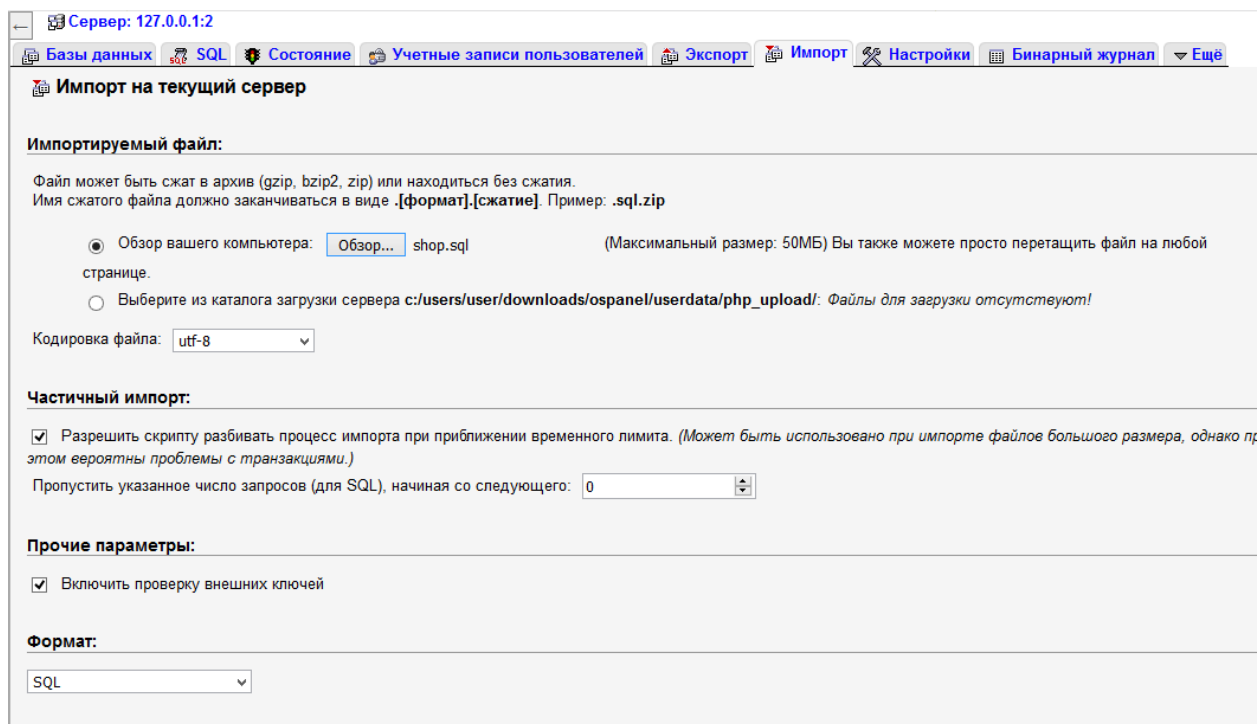


Рисунок 3.2 – Импорт базы данных

Далее будут описываться важные части кода на PHP и HTML.

Листинг 1. Форма для входа

```
<li role="presentation" class="active"><a href="#Section1" aria-controls="home" role="tab" data-toggle="tab">Вход в систему</a></li>
</ul>
<div class="tab-content tabs" style="margin-top: 20px;">
  <div role="tabpanel" class="tab-pane fade in active" id="Section1">
    <form action="check.php" method="POST" class="form-horizontal">
      <div class="form-group">
        <input type="text" name="login" class="form-control" placeholder="Логин" id="exampleInputEmail1">
      </div>
    </div>
  </div>
</div>
```

```

        <input type="password" name="password" class="form-control" placeholder="Пароль" id="exampleInputPassword1">
    </div>
    <div class="form-group">
        <button type="submit" name="submit" class="btn btn-default" style="outline: none;">войти</button>
    </div>
</form>
</div>

```

В листинге 1 представлен код для входа на страницу. В качестве обработчика выступает `check.php`, метод запроса POST.

Листинг 2. Создание сессий

```

<?php
    session_start();
    $_SESSION['user'] = $_POST['login'];
    $_SESSION['password'] = $_POST['password'];
    header("Location: coworker.php?table=coworker");
?>

```

В листинге 2 создаются две сессии. Функция `session_start()` создает сессию, либо возобновляет существующую, основываясь на идентификаторе сессии, переданном через GET- или POST-запрос, либо переданный через cookie. `header()` используется для отправки HTTP-заголовка. Он будет перенаправлять на страницу `main.php`.

Листинг 3. Подключение к базе данных

```

<?php
session_start();
$server = "localhost";
$username = $_SESSION['user'];
$password = $_SESSION['password'];
$db="shop";
if (empty($username) and empty($password)) {
    echo "<body text-align: center;>
        <div>
            <h1>Не введены логин и пароль для входа в систему
                <a href='index.php'>
                    <i style='width: 30px; height: 30px;'>
                        <img src='img/icon.png' style='width: 30px; height: 30px;'>
                    </i>
            </h1>
        </div>
    </body>";
}

```

```

        </a>
    </h1>
</div>
</body>";
die();
}
$connection = @mysqli_connect($server, $username, $password, $db);
if($connection == false) {
    echo "<body text-align: center;>
        <div>
            <h1>Для того, чтобы войти в систему введите корректные данные
                <a href='index.php'>
                    <i style='width: 30px; height: 30px;'>
                        <img src='img/icon.png' style='width: 30px; height: 30px;'>
                    </i>
                </a>
            </h1>
        </div>
</body>";
die(); }?>

```

В листинге 3 создается подключение к базе данных shop. Если введенные логин и пароль пустые, то выведется соответствующий HTML-код и сработает функция die(), если переменная connection будет равняться false, то также выведется соответствующий код и вызовется функция die(). Если же все условия будут не выполнены, то также выведется соответствующий HTML-код.

Листинг 4. Ссылка Сотрудники

```

if ($_GET['table'] == 'coworker') {
    $sql = "SELECT * FROM coworker";
    $result = mysqli_query($connection, $sql);
    ?>
        <div class="card mb-2 wow fadeIn" style="margin-bottom:
20px; padding-bottom: 20px;">
            <div class="card-body col-sm-12 d-flex justify-content-
between con">
                <div class="col-sm-4">
                    <a class="btn btn-primary fff"
href="coworker.php?add=coworker">Добавить сотрудника</a>
                </div>
                <form class="">
                    <div class="col-sm-4 d-flex justify-content-end"
style="float: right; display: flex; align-items: center;">

```

```

        <div class="col-sm-8">
            <input type="search" style="height: 32px; margin-right: 15px;" class="form-control" placeholder="dd-mm-yyyy" name="">
        </div>
        <div class="col-sm-2">
            <button class="mr-2 btn btn-primary btn-sm my-0" type="submit">
                <i class="fa fa-search"></i>
            </button>
        </div>
    </div>
</div>
</form>
</div>
</div>
<div class="card mb-2 wow fadeIn" style="padding: 5px; white-space: nowrap;">
    <table class="table table-striped">
        <thead class="thead-dark">
            <tr>
                <th>ID Сотрудника</th>
                <th>ID Должности</th>
                <th>Банковский счет</th>
                <th>Полное имя</th>
                <th>ИИН</th>
                <th>Дата принятия</th>
                <th>Контактный номер</th>
                <th><div class="trash-icon"></div></th>
            </tr>
        </thead>
        <?php
            while ($row = mysqli_fetch_assoc($result)) {
                $id = $row["ID_Co_Worker"];
                echo '<tr>
                    <td><a style="color: #17a2b8;" href="coworker.php?coworkerid='.$id.'">'.$row["ID_Co_Worker"].'</a></td>
                    <td>'.$row["ID_Position"].'</td>
                    <td>'.$row["Bank_Account"].'</td>
                    <td>'.$row["Full_Name"].'</td>
                    <td>'.$row["IIN"].'</td>
                    <td>'.$row["Date_Of_Adoption"].'</td>
                    <td>'.$row["Contact_Num"].'</td>
                    <td><a style="color: red" href="coworker.php?table=coworker&delete='.$id.'">удалить</a></td>
                </tr>
            </?php
    </div>

```

```

        </tr>';
    }
    echo '</table></div>';
?>

```

Если пользователь нажал на ссылку Сотрудники, то выполняется SQL-запрос и выводятся данные о всех сотрудниках.

Листинг 5. Ссылка Товары

```

if ($_GET['table'] == 'goods') { ?>
    <div class="card mb-2 wow fadeIn" style="margin-bottom:
20px; padding-bottom: 20px;">
        <div class="card-body col-sm-12 d-flex justify-content-be-
tween con">
            <div class="col-sm-4">
                <a href="goods.php?add=goods" class="btn btn-primary
waves-effect">Добавить товар</a>
            </div>
            <form class="">
                <div class="col-sm-4 d-flex justify-content-end"
style="float: right; display: flex; align-items: center;">
                    <div class="col-sm-8">
                        <input type="search" style="height: 32px; margin-
right: 15px;" class="form-control" placeholder="Articul" name="">
                    </div>
                    <div class="col-sm-2">
                        <button class="mr-2 btn btn-primary btn-sm my-0"
type="submit">
                            <i class="fa fa-search"></i>
                        </button>
                    </div>
                </div>
            </form>
        </div>
    </div>
    <?php
    $sql = "SELECT * FROM goods";
    $result = mysqli_query($connection, $sql); ?>
    <div class="card mb-2 wow fadeIn" style="padding: 5px; white-
space: nowrap;">
        <table class="table table-striped">
            <thead class="thead-dark">

```



```

<tr>
  <th>Артикул</th>
  <th>ID ткани</th>
  <th>ID категории</th>
  <th>Наименование</th>
  <th>Цена</th>
  <th>ID модели</th>
  <th>Бренд</th>
  <th>Страна</th>
</tr>
</thead>
<?php
while ($row = mysqli_fetch_assoc($result)) {
  echo '<tr>
    <td>'.$row["Articul"].'</td>
    <td>'.$row["ID_Cloth"].'</td>
    <td>'.$row["ID_Category_Goods"].'</td>
    <td>'.$row["Name"].'</td>
    <td>'.$row["Price"].'</td>
    <td>'.$row["ID_Model"].'</td>
    <td>'.$row["Brand"].'</td>
    <td>'.$row["ID_Country"].'</td>
  </tr>';
}
echo "</table></div>";
} ?>

```

Если пользователь нажал на ссылку Товары, то выводится HTML-код то выполняется SQL-запрос и выводятся данные о всех товарах. Если же он нажал на кнопку добавить товар, то появляется форма для добавления товара. Далее происходит процесс, который представлен в листинге 6: если была нажата кнопка Добавить, то выполняются SQL-запросы, которые проверяют существования тех или иных значений в ссылаемых таблицах базы данных. Если таких данных нет, то выводятся ошибки о некорректности. Если данные об артикуле уже существуют, то пользователю выведется эта ошибка. Если поля будут пустые, то также будет выведена ошибка. Иначе будет выполнен SQL-код, который добавит данные в таблицу Goods.

В листинге 7 показан следующий процесс: если пользователь нажимает на данную ссылку, то выводится форма и срабатывает SQL-код, который выводит всю информацию о номенклатуре. Однако, если же была нажата одна из кнопок, то выполняются запросы, представленные в конструкции if при нажатии на кнопку.

Листинг 6. Добавление товара

```

if (isset($_POST['addgoods'])) {
    $articul = $_POST['articul'];
    $idcloth = $_POST['idcloth'];
    $idcat = $_POST['idcat'];
    $name_ = $_POST['name_'];
    $price = $_POST['price'];
    $model = $_POST['model'];
    $brand = $_POST['brand'];
    $country = $_POST['country'];
    $sql6 = "SELECT * FROM goods WHERE `Articul` = '$ar-
articul'";

    $query6 = mysqli_query($connection, $sql6);
    $result6 = mysqli_num_rows($query6);
    if ($result6 == 1 ) {
        echo '<main class="pt-5 max-lg-5">
            <div class="container-fluid mt-5">
                <div class="add-form col-sm-4">
                    <div class="success-al alert alert-danger"
role="alert">
                        Данные о товаре с артикулом <span style="font-
weight: bold;"> ' . $articul. '</span> уже были добавлены!
                    </div>
                </div>
            </div>
        </main><br>';
        die();
    }
    if (empty($articul) OR empty($idcloth) OR empty($idcat)
OR empty($name_) OR empty($price) OR
empty($model) OR empty($brand) OR empty($country)) {
        echo '<div class="success-al alert alert-danger"
role="alert">
            Заполните пустые поля!
        </div>';
        die();
    }
    $query = "INSERT INTO goods VALUES('$articul',
'$idcloth', '$idcat', '$name_', '$price', '$model', '$brand', '$country')";
    $result = mysqli_query($connection, $query);
    echo ' <div class="success-al alert alert-success"
role="alert">
        Данные успешно добавлены
    </div>';
}

```

```
}
```

Листинг 7. Ссылка Номенклатура

```
if ($_GET['table'] == 'size') { ?>
    <div class="card mb-2 wow fadeIn" style="margin-bottom:
20px; padding-bottom: 20px;">
        <div class="card-body col-sm-12 d-flex justify-content-
between con">
            <div class="col-sm-4">
                <a href="size.php?add=size" class="btn btn-primary
fff">Добавить номенклатуру</a>
            </div>
            <form class="">
                <div class="col-sm-4 d-flex justify-content-end"
style="float: right; display: flex; align-items: center;">
                    <div class="col-sm-8">
                        <input type="search" style="height: 32px; mar-
gin-right: 15px;" class="form-control" placeholder="Articul" name="">
                    </div>
                    <div class="col-sm-2">
                        <button class="mr-2 btn btn-primary btn-sm my-
0" type="submit">
                            <i class="fa fa-search"></i>
                        </button>
                    </div>
                </div>
            </form>
        </div>
    </div>
    <?php
    $sql = "SELECT * FROM size";
    $result = mysqli_query($connection, $sql); ?>
    <div class="card mb-2 wow fadeIn" style="padding: 5px;
white-space: nowrap;">
        <table class="table table-striped table-dark" style="color:
#000">
            <thead class="thead-dark">
                <tr>
                    <th>Штрих-код</th>
                    <th>Размер</th>
                    <th>Количество</th>
                    <th>Артикул</th>
                    <th>Цвет</th>
                </tr>
            </thead>
        </table>
    </div>
}
```

```

        </tr>
    </thead>
<?php
    while ($row = mysqli_fetch_assoc($result)) {
        echo '<tr>
            <td>'.$row["Bar_Code"].'</td>
            <td>'.$row["Size"].'</td>
            <td>'.$row["Number_Of_Units"].'</td>
            <td>'.$row["Articul"].'</td>
            <td>'.$row["Colour"].'</td>
        </tr>';
    }
    echo "</table></div>";

```

Листинг 8. Ссылки Фискальные чеки и Заказы

```

if ($_GET['table'] == 'receipt') {
    $sql = "SELECT * FROM fiscalreceipt";
    $result = mysqli_query($connection, $sql);
    echo '<main class="pt-5 max-lg-5">
        <div class="container-fluid mt-5">
            <div class="card mb-2 wow fadeIn">
                <div class="card-body d-sm-flex justify-
content-between con">
                    </div>
                </div>
            <div class="card mb-2 wow fadeIn"
style="padding: 5px; white-space: nowrap;">
                <table class="table table-striped">
                    <thead class="thead-dark">
                        <tr>
                            <th>Номер чека</th>
                            <th>БИН</th>
                            <th>ID сотрудника</th>
                            <th>Дата</th>
                            <th>Бонусная карта</th>
                        </tr>
                    </thead>
                </table>
            </div>
        </div>
    </main>';
    while ($row = mysqli_fetch_assoc($result)) {
        echo '<tr>
            <td>'.$row["ID_Check"].'</td>
            <td>'.$row["BIN"].'</td>
            <td>'.$row["ID_Co_Worker"].'</td>

```

```

                <td>'.$row["Date"].'</td>
                <td>'.$row["ID_Bonus_Card"].'</td>
            </tr>;
        }
        echo "</table></div>";
    if ($_GET['table'] == 'order') {
        $sql = "SELECT * FROM ord JOIN ord_size ON ord.`ID_Order`
= ord_size.`ID_Order`";
        $result = mysqli_query($connection, $sql);
        echo '<main class="pt-5 max-lg-5">
<div class="container-fluid mt-5">
<div class="card mb-2 wow fadeIn">
<div class="card-body d-sm-flex justify-content-be-
tween con">
<a href="order.php?add=order" class="btn btn-pri-
mary waves-effect">Добавить заказ</a>
</div>
</div>
<div class="card mb-2 wow fadeIn" style="padding:
5px; white-space: nowrap;">
<table class="table table-striped">
<thead class="thead-dark">
<tr>
<th>Номер заказа</th>
<th>Номер сотрудника</th>
<th>Дата заказа</th>
<th>Штрих-код</th>
<th>Количество</th>
</tr>
</thead>
';
    while ($row = mysqli_fetch_assoc($result)) {
        echo '<tr>
<td>'.$row["ID_Order"].'</td>
<td>'.$row["ID_Co_Worker"].'</td>
<td>'.$row["Order_Date"].'</td>
<td>'.$row["Bar_Code"].'</td>
<td>'.$row["Amount"].'</td>
</tr>';
    }
    echo "</table></div>";
}

```

Аналогично выполняются запросы при нажатии на ссылки Фискальные чеки и Заказы: предоставляются все данные.

Листинг 9. Ссылка Дневные отчеты

```
if ($_GET['report'] == 'day') {
    echo '<div class="container-fluid mt-5">
        <div class="card mb-2 wow fadeIn">
            <div class="card-body d-sm-flex justify-content-between
con">
                <h4 class="mb-2 mb-sm-0 pt-1">
                    <span>Введите дату</span>
                </h4>
                <form
method="POST">
                    action="report-day.php?report=day"
                <div class="col-sm-12 d-flex justify-content-start"
style="display: flex-wrap;">
                    <div class="col-sm-6" style="display: flex">
                        <input type="search" style="height: 32px; margin-
right: 15px;" class="form-control" placeholder="yyuu-mm-dd" name="date">
                    </div>
                    <div class="col-sm-4" style="display: flex;">
                        <div>
                            <button class="btn-primary btn" type="submit"
name="coworker" style="margin-left: 5px; margin-right: 5px;">Сотрудники
                        </button>
                        </div>
                        <div>
                            <button class="btn-primary btn" type="submit"
name="receipt" style="margin-left: 5px; margin-right: 5px;">Чеки
                        </button>
                        </div>
                    </div>
                </div>
            </div>
        </div>
    </form>
</div>';
```

При нажатии на ссылку выполняется HTML-код, показанный в листинге 9. Далее, если нажата кнопка Сотрудники, то выводятся сумма продаж и данные о сотруднике за день, введенный в поле формы, если же нажата кнопка Чеки, то выводится сумма чека и основные данные о нем за день, введенный в поле формы. Данный код описан в листингах 10 и 11 соответственно для разных действий пользователя.

Листинг 10. Нажатие кнопки Сотрудники

```

if (isset($coworker)) {
    $sql = "SELECT `Full_Name`, fiscalre-
    ceipt.`ID_Co_Worker`,
    SUM(`Total_Of_Goods` * `Price`) AS `Total` FROM
    purchase JOIN size ON purchase.`Bar_Code` =
    size.`Bar_Code`
    JOIN goods ON size.`Articul` = goods.`Articul`
    JOIN fiscalreceipt ON fiscalreceipt.`ID_Check` = pur-
    chase.`ID_Check`
    JOIN coworker ON fiscalreceipt.`ID_Co_Worker` =
    coworker.`ID_Co_Worker`
    WHERE DATE(`Date`) = '$date'
    GROUP BY fiscalreceipt.`ID_Co_Worker`
    ORDER BY SUM(`Total_Of_Goods` * `Price`)";
    echo '<div class="card mb-2 wow fadeIn" style="pad-
    ding:10px;">
        <div style="padding-bottom: 10px; padding-top:
    40px;">Данные о продажах сотрудников за <span style="font-weight:
    bold;">'. $date. '</span></div>';
        $result = mysqli_query($connection, $sql);
        echo '
        <table class="table table-striped">
        <thead class="thead-dark">
        <tr>
        <th>Полное имя</th>
        <th>Номер сотрудника</th>
        <th>Сумма продаж</th>
        </tr>
        </thead>
        ';
        while ($row = mysqli_fetch_assoc($result)) {
            echo '<tr>
                <td>'. $row["Full_Name"]. '</td>
                <td>'. $row["ID_Co_Worker"]. '</td>
                <td>'. $row["Total"]. '</td>
            </tr>';
        }
        echo "</table></div>";
    }
}

```

Листинг 11. Нажатие кнопки Чеки

```

elseif (isset($receipt)) {

```

```

        $sql = "SELECT purchase.`ID_Check`, SUM(`Total_Of_Goods`* `Price`) AS `Toatal`,
        fiscalreceipt.`ID_Co_Worker`, coworker.`Full_Name`,
        fiscalreceipt.`Payment_Type`,
        SUM(`Total_Of_Goods`) AS `AmountInCheck`
        FROM purchase JOIN size ON size.`Bar_Code` = purchase.`Bar_Code`
        JOIN goods ON goods.`Articul` = size.`Articul`
        JOIN fiscalreceipt ON fiscalreceipt.`ID_Check` = purchase.`ID_Check`
        JOIN coWorker ON coworker.`ID_Co_Worker` = fiscalreceipt.`ID_Co_Worker`
        WHERE DATE(`Date`) = '$date'
        GROUP BY purchase.`ID_Check`
        ORDER BY fiscalreceipt.`ID_Check`";
        echo '<div class="card mb-2 wow fadeIn" style="padding:10px;">
        <div style="padding-bottom: 10px; padding-top: 40px;">Данные о чеках за <span style="font-weight: bold;">'. $date. '</span></div>';
        $result = mysqli_query($connection, $sql);
        echo '
        <table class="table table-striped">
        <thead class="thead-dark">
        <tr>
        <th>Номер чека</th>
        <th>Сумма чека</th>
        <th>ID сотрудника</th>
        <th>Имя сотрудника</th>
        <th>Тип оплаты</th>
        <th>Количество товара в чеке</th>
        </tr>
        </thead>
        '
        ;

```

При входе на страницу пользователю будут сразу отображаться данные о сотрудниках.

3.4 Описание интерфейса

На рисунке 3.41 показана страница входа. Пользователь вводит данные в форму.

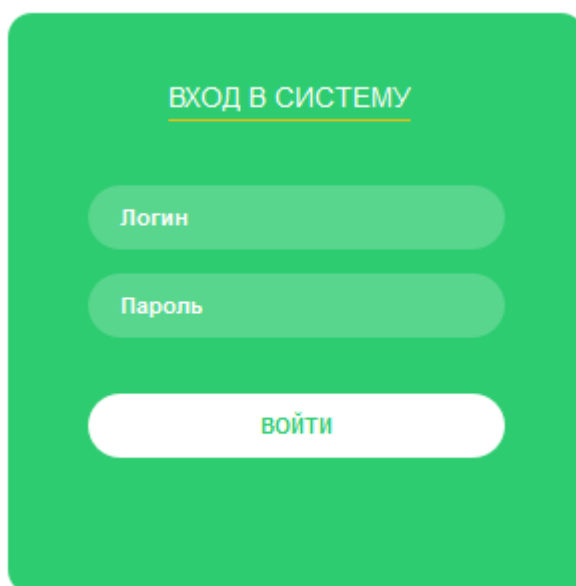


Рисунок 3.41 - Страница входа

Пользователь попадает в систему и ему выводятся данные о сотрудниках. Процесс показан на рисунке 3.42.

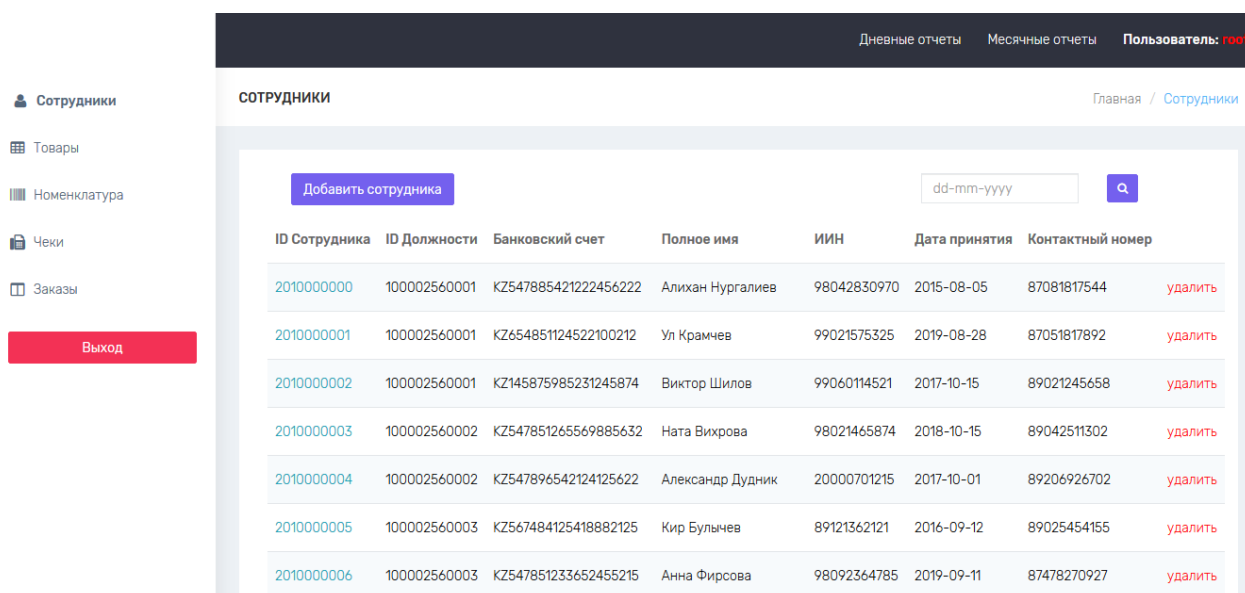


Рисунок 3.42 - Успешный вход в систему

На рисунке 3.42 проверяется работоспособность ссылки Сотрудники, на рисунке 3.44 - Товары, на рисунке 3.45 - Добавление товара, на рисунке 3.46 - Ссылка Номенклатура, на рисунке 3.48 - ссылка Заказы.

Форма добавления сотрудника

ID сотрудника	<input type="text"/>
Полное имя	<input type="text"/>
Дата рождения	<input type="text"/>
ИИН	<input type="text"/>
Должность	<input type="text" value="Продавец-кассир"/>
Банковский счет	<input type="text"/>
Опыт работы	<input type="text"/>
Адрес	<input type="text"/>
Дата принятия	<input type="text"/>
Номер телефона	<input type="text"/>

[Добавить](#)

Рисунок 3.43 - Форма добавления сотрудника

Дневные отчеты Месячные отчеты Пользователь: root

Сотрудники

Товары

Номенклатура

Чеки

Заказы

[Выход](#)

ТОВАРЫ Главная / Товары

[Добавить товар](#)

Артикул	ID ткани	ID категории	Наименование	Цена	ID модели	Бренд	Страна
10211 LEXUS 117055	1	1	Футболка	70000	1	1	487
10211 LEXUS 118	1	1	Футболка	70000	2	1	487
124 ROVER 111	1	1	Футболка	60000	1	1	487
458 LEXUS 124	2	1	Майка	25000	1	1	487
478 TORRO JL 444	6	1	Перчатки	11000	1	2	487
578 ВОHEM 145	1	1	Кардиган	50000	1	1	84
782 RICON 777	2	5	Шапка	18000	1	2	869

Рисунок 3.44 – Нажатие на ссылку Товары

Форма добавления товара

Артикул

Состав ткани ▼

Категория ▼

Наименование

Цена

Модель ▼

Бренд ▼

Страна ▼

Рисунок 3.45 – Добавление товара

Дневные отчеты Месячные отчеты Пользователь: root

НОМЕНКЛАТУРА Главная / Номенклатура

Штрих-код	Размер	Количество	Артикул	Цвет
200001245585	44	10	874 ART 456	Milky
200012145845	42	10	874 ART 456	Milky
200012456558	48	10	874 ART 456	Milky
200121212002	L	10	BARSELONA/70	White
201212000045	46	4	874 ART 456	Milky

Рисунок 3.46 – Ссылка Номенклатура

Форма добавления номенклатуры

Штрих-код

Размер

Количество

Артикул ▼

Цвет

Добавить

Рисунок 3.47 – Добавление номенклатуры

Дневные отчеты Месячные отчеты Пользователь: root

ЗАКАЗЫ Главная / Заказы

Добавить заказ

Номер заказа	Номер сотрудника	Дата заказа	Штрих-код	Количество
1	2010000000	2019-11-28	200012145845	10
1	2010000000	2019-11-28	200012456558	10
2	2010000000	2019-11-28	201212000045	10
2	2010000000	2019-11-28	300001254502	10

Рисунок 3.48 – Ссылка Заказы

Кроме этого, необходимо протестировать **Дневные отчеты**. Пользователь нажимает на ссылку и ему представляется форма для поиска. Результат представлен на рисунке 3.49.

ДНЕВНЫЕ ОТЧЕТЫ Главная / Дневные отчеты

Введите дату

Сотрудники **Чеки**

Рисунок 3.49 – Дневные отчеты

Далее пользователь вводит дату, допустим 2019-10-16, и нажимает на одну из кнопок. Данный процесс представлен на рисунке 3.5.

Введите дату

yyyy-mm-dd

Сотрудники Чеки

Данные о продажах сотрудников за **2019-10-16**

Полное имя	Номер сотрудника	Сумма продаж
Алихан Нургалиев	2010000000	18000
Ул Крамчев	2010000001	100000

Рисунок 3.5 – Данные о продажах сотрудников за день

Аналогично для Месячные отчеты. Пользователь выбирает месяц из выпадающего списка для получения отчета по чекам

МЕСЯЧНЫЕ ОТЧЕТЫ Главная / Месячные отчеты

Выберите дату

Октябрь

Сотрудники Чеки

Данные о чеках за

Номер чека	Сумма чека	ID сотрудника	Имя сотрудника	Тип оплаты	Количество товара в чеке
1	35970	2010000000	Алихан Нургалиев	Card	3
2	60000	2010000001	Ул Крамчев	Cash	1
3	149500	2010000007	Макар Горошко	Card	5
4	39960	2010000007	Макар Горошко	Cash	4
5	100000	2010000001	Ул Крамчев	Card	2
6	18000	2010000000	Алихан Нургалиев	Card	1
7	12990	2010000004	Александр Дудник	Cash	1

Рисунок 3.51 – Данные о чеках сотрудников за месяц

4 Экономическое обоснование проекта

Целью данного дипломного проекта является разработка информационной системы учета продаж супермаркета. Объектами автоматизации в ходе проектирования являются процесс оперативного получения информации о продажах сотрудников за день-месяц, процесс добавления новых товаров в базу данных, отказ от бумажной документации. Кроме того, информационная система имеет функции учета чеков, мониторинга товаров, добавление сотрудников, что позволяет администратору осуществлять контроль за деятельностью организации. Преимущества данного продукта – это наглядность, ИС отображает все продажи сотрудников, также отображает информацию о всех товарах на витринах. Основной задачей экономического раздела является определение объема и трудоемкости разработки программного обеспечения, а также расчет сроков окупаемости.

4.1 Определение объема и трудоемкости разработки программного обеспечения

Общий объем (V_0) программного продукта определяется исходя из количества и объема функций, реализуемых программой [9]. За уточненный объем ПО (V_y) учитывается «строка исходного кода» как одна, если в ней содержится лишь один оператор, в данном случае:

$$V_y = 1400 \text{ строк}$$

Так как проект является крупным, трудоемкость рассчитывается по стадиям разработки и затем складывается.

Трудоемкость стадий определяется с учетом сложности, новизны, степени использования в разработке стандартных модулей ПО и удельного веса трудоемкости каждой стадии в общей трудоемкости ПО [1]:

$$T_{yi} = T_n * d_{cmi} * K_c * K_m * K_n, \quad (4.1)$$

где T_{yi} – уточненная трудоемкость разработки ПО на i – й стадии;

T_n – нормативная трудоемкость;

d_{cmi} – удельный вес трудоемкости i – й стадии разработки ПО в общей трудоемкости разработки ПО;

K_c – коэффициент, учитывающий сложность ПО, вводится на всех стадиях;

K_m – коэффициент, учитывающий степень использования стандартных модулей ПО, вводится только на стадии рабочего проекта;

K_n – коэффициент, учитывающий степень новизны ПО, вводится на всех стадиях.

Нормативная трудоёмкость T_n определяем по таблице В.1 приложения В[1]:

$T_n = 48$ часов.

Коэффициент сложности рассчитывается по формуле [9]

$$K_c = 1 + \sum_{i=0}^n K_i, \quad (4.2)$$

где K_i – коэффициент, соответствующий степени повышения сложности ПО за счет конкретной характеристики;

n – количество учитываемых характеристик.

По таблице 1 и Г.1 приложения Г [9] программный продукт имеет следующие характеристики:

– интерактивный доступ ($K_c = 0,06$);

Следовательно,

$$K_c = 1 + 0,6 = 1,06.$$

Поправочный коэффициент, учитывающий степень использования готовых модулей, берется из таблицы 2 [9] и составляет $K_m = 0,6$, т.к в разработке задействован стандартный функционал веб-приложения phpMyAdmin и Sublime Text.

Коэффициент, определяющий степень новизны, находится по таблице 3 [9] и равен $K_n = 0,9$ т.к данный продукт имеет доступные аналоги, при это используется на разных родах ПК и на различных ОС. Значения коэффициентов удельных весов трудоёмкости стадий берутся из таблицы 4 [9], согласно степени новизны.

Занесем данные по этапам разработки и коэффициентам в таблицу 4.1.

Таблица 4.1

Стадии разработки программного продукта

№	Название	d_{cti} – удельный вес трудоёмкост и	K_c – коэффициен т сложности	T_n – нормативная трудоёмкост ь	K_n – коэффициен т новизны
1	Техническо е задание	0,10	1,06	48	0,9
2	Эскизный проект	0,08	1,06	48	0,9

Продолжение таблицы 4.1

3	Технический проект	0,09	1,06	48	0,9
4	Рабочий проект	0,58	1,06	48	0,9
5	Внедрение	0,15	1,06	48	0,9

Трудоемкость стадий ПО рассчитывается по следующим формулам [9]

$$\text{трудоемкость стадии ТЗ} \quad T_{уз} = T_{н} * d_{з} * K_{с} * K_{н} \quad (4.3)$$

$$\text{трудоемкость стадии ЭП} \quad T_{уэ} = T_{н} * d_{э} * K_{с} * K_{н} \quad (4.4)$$

$$\text{трудоемкость стадии ТП} \quad T_{ут} = T_{н} * d_{т} * K_{с} * K_{н} \quad (4.5)$$

$$\text{трудоемкость стадии РП} \quad T_{ур} = T_{н} * d_{р} * K_{с} * K_{н} * K_{т} \quad (4.6)$$

$$\text{трудоемкость стадии ВН} \quad T_{ув} = T_{н} * d_{в} * K_{с} * K_{н} \quad (4.7)$$

Тогда,

$$T_{уз} = 48 * 0,10 * 1,06 * 0,9 = 4,57 \text{ часа};$$

$$T_{уэ} = 48 * 0,08 * 1,06 * 0,9 = 3,66 \text{ часа};$$

$$T_{ут} = 48 * 0,09 * 1,06 * 0,9 = 4,12 \text{ часов};$$

$$T_{ур} = 48 * 0,58 * 1,06 * 0,9 * 0,6 = 15,93 \text{ часов};$$

$$T_{ув} = 48 * 0,18 * 1,06 * 0,9 = 8,24 \text{ часов.}$$

Общая трудоемкость будет равна

$$T_{у} = 4,57 + 3,66 + 4,12 + 15,93 + 8,24 = 36,52 \text{ часов} \approx 5 \text{ дней.}$$

4.2 Расчет затрат на разработку информационных технологий

Расчет полных затрат на разработку проектного решения осуществляется по формуле [9]:

$$C_{ni} = Z_{фот} + Z_{сзи} + Z_{сои} + M_i + P_{ci} + P_{ми} + P_{нki} + П_{зи} + P_{ни} + A + ВОСМС, \quad (4.8)$$

где $Z_{фот}$ – общий фонд оплаты труда разработчиков, тенге;

$Z_{сзи}$ – отчисления по социальному налогу, тенге;

$Z_{сои}$ – социальное отчисление, тенге;

M_i – затраты на материалы, тенге;

P_{ci} – затраты на специальные программные средства, необходимые для разработки проектного решения, тенге;

$P_{ми}$ – затраты, связанные с эксплуатацией техники, тенге;

$P_{нki}$ – затраты на научные командировки, тенге;

$П_{зи}$ – прочие затраты, тенге;

P_{ni} – накладные расходы, тенге.
А – амортизационные отчисления;
ВОСМС – отчисления по ОСМС.

Размер фонда оплаты труда разработчиков рассчитывается по формуле [9]:

$$З_{\text{ФОТ}} = З_{oi} + З_{\phi i}, \quad (4.9)$$

где $З_{oi}$ – основная заработная плата, тенге;
 $З_{\phi i}$ – дополнительная заработная плата, тенге.

Основная заработная плата рассчитывается по следующей формуле [9]

$$З_{oi} = T_y * СП / (t_{\text{ср}} * 8), \quad (4.10)$$

где T_y – суммарные затраты труда, вычисляемые по формуле;
 $t_{\text{ср}}$ – среднее число дней в месяце, равно 22 дню, умножается на количество часов в рабочем дне – 8;
СП – средняя заработная плата.
Т.к средняя заработная плата в Казахстане в 2020 году составляет 203900 тенге, то мы используем ее в формуле:

$$З_{oi} = 36,52 * \frac{203\,900}{22 * 8} = 42309 \text{ тенге.}$$

Дополнительная заработная плата составляет 10% от основной и рассчитывается по формуле [9]:

$$З_{\phi i} = З_{oi} * H_d / 100, \quad (4.11)$$

где H_d – коэффициент дополнительной заработной платы разработчиков.

Тогда,

$$З_{\phi i} = 42309 * \frac{15}{100} = 6346 \text{ тенге.}$$

ФОТ = 42309 + 6346 = 48655 тенге.

Отчисления на ОСМС рассчитываем по формуле:

$$\text{ВОСМС} = \text{ФОТ} * 2\% \quad (4.12)$$

Тогда,

$$\text{ВОСМС} = 48655 * 2\% = 973 \text{ тенге};$$

Социальный налог составляет (9,5%) (ст. 358 п.1 НК РК) от дохода работника, и рассчитывается по формуле [9]:

$$Z_{czi} = (\text{ФОТ} - \text{ПО} - \text{ВОСМС}) * 9,5\% - Z_{coi}, \quad (4.13)$$

где ПО – пенсионные отчисления, которые составляют 10% от ФОТ и социальным налогом не облагаются[9]:

$$\text{ПО} = \text{ФОТ} * 10\%. \quad (4.14)$$

Тогда,

$$\begin{aligned} \text{ПО} &= (42309 + 6346) * 10\% = 4865 \text{ тенге}; \\ Z_{czi} &= (48655 - 4865 - 973) * 9,5\% - 1532 = 2535 \text{ тенге}. \end{aligned}$$

Социальное отчисление составляет (3,5%) (ст. 358 п.1 НК РК) от дохода работника, и рассчитывается по формуле:

$$Z_{coi} = (\text{ФОТ} - \text{ПО}) * 3,5\%, \quad (4.15)$$

Тогда,

$$Z_{coi} = (48655 - 4865) * 3,5\% = 1532 \text{ тенге}.$$

Общая сумма налогов будет составлять:

$$\text{Налоги} = \text{СО} + \text{СН} + \text{ВОСМС} = 1532 + 2535 + 973 = 5040 \text{ тенге}$$

Величина затрат на материалы определяется по формуле [9]:

$$M_i = \frac{Z_{ocn} * H_{m3}}{100} \%, \quad (4.16)$$

где H_{m3} – норма расхода материалов от основной заработной платы (3-5%).

Следовательно,

$$M_i = 42309 * \frac{5}{100} = 2115 \text{ тенге}.$$

Расходы по статье «Спецоборудование» (P_{ci}) включают покупку дополнительной оперативной памяти 4 Гб стоимостью 35\$, а также расходы

на приобретение компьютерной мыши стоимостью 15\$ и пакета интернет трафика на сумму 50\$

$$P_{ci} = 100\$ = 42\,900 \text{ тенге.}$$

Амортизационные отчисления рассчитываются по формуле:

$$A = \frac{C_{\text{обор}} * N_a * N}{100 * 12 * t}, \quad (4.16)$$

где N_a – норма амортизации (25 %);

$C_{\text{обор}}$ – первоначальная стоимость оборудования;

t – количество рабочих дней в месяце ($t = 22$ дней);

N – время использования персонального компьютера;

$$N = 36,52 / 8 \approx 5 \text{ дней.}$$

Тогда,

$$A = \frac{2115 * 25 * 5}{100 * 12 * 22} = 10,49 \text{ тенге.}$$

Расходы по статье «Машинное время» (P_{mi}) включают оплату машинного времени, необходимого для разработки и отладки ПО, которое определяется по нормативам (в машино-часах) на 100 строк исходного кода машинного времени (H_{mv}) в зависимости от характера решаемых задач и типа ПК (Приложение Д) [9]:

$$P_{mi} = C_{mi} * \left(\frac{V_{oi}}{100} \right) * H_{mv}, \quad (4.17)$$

где C_{mi} – цена одного машино-часа (приблизительно в перерасчет на энергию равна 7,7 тенге);

V_{oi} – общий объем ПО (количество строк).

Тогда,

$$P_{mi} = 400 * \left(\frac{1400}{100} \right) * 12 = 67200 \text{ тенге.}$$

Расходы по статье «Научные командировки» (P_{nki}) на конкретное ПО определяются по нормативу, разрабатываемому в целом по организации, в процентах к основной заработной плате [9]:

$$P_{nki} = Z_{oi} * H_{pнк} / 100 \quad (4.18)$$

где $H_{pнк}$ – норматив расходов на командировки в целом по организации в (%), равен 30% [9].

Тогда,

$$P_{\text{нкi}} = 67200 * \frac{30}{100} = 20160 \text{ тенге.}$$

Расходы по статье «Прочие затраты» (P_{zi}) на конкретное ПО включают затраты на приобретение и подготовку специальной научно-технической информации и специальной литературы. Определяются по нормативу, разрабатываемому в целом по организации, в процентах к основной заработной плате [9]:

$$P_{zi} = Z_{oi} * H_{пз}/100 \quad (4.18)$$

где $H_{пз}$ – норматив прочих затрат в целом по организации в (%), равен 20%.

Тогда,

$$P_{zi} = 67200 * \frac{20}{100} = 13440 \text{ тенге.}$$

Затраты по статье «Накладные расходы» ($P_{\text{ни}}$), связанные с необходимостью содержания аппарата управления, вспомогательных хозяйств и опытных (экспериментальных) производств, а также с расходами на общехозяйственные нужды ($P_{\text{ни}}$), относятся на конкретное ПО по нормативу ($H_{\text{нр}}$) в процентном отношении к основной заработной плате исполнителей. Норматив устанавливается в целом по организации [9]:

$$P_{\text{ни}} = Z_{oi} * H_{\text{нр}}/100 \quad (4.19)$$

где $P_{\text{ни}}$ – накладные расходы на конкретную ПО (тыс. тенге);

$H_{\text{нр}}$ – норматив накладных расходов в целом по организации в (%), равен 70%.

Следовательно,

$$P_{\text{ни}} = 67200 * \frac{70}{100} = 47040 \text{ тенге.}$$

Таблица 4.2 Затраты на разработку информационных технологий

Затраты на разработку	Условное обозначение	Значение, тенге	В процентах от общей суммы
Фонд оплаты труда	$Z_{\text{фот}}$	48655	19,73
Социальный налог	$Z_{\text{сзи}}$	2535	1,03

Продолжение таблицы 4.2

Социальное отчисление	Z_{coi}	1532	0,62
Материалы	M_i	2115	0,86
Спецоборудование	P_{ci}	42900	17,40
Машинное время	P_{mi}	67200	27,25
Научные командировки	P_{nki}	20160	8,17
Прочие затраты	Π_{zi}	13440	5,45
Накладные расходы	P_{ni}	47040	19,08
Амортизационные отчисления	А	10	0,004
Отчисления по ОСМС	ВОСМС	973	0,40
Итого:	C_0	246560	100

Цена разрабатываемого программного обеспечения рассчитывается по формуле[9]:

$$C_p = C_0 + \text{НДС}, \quad (4.20)$$

где C_0 – первоначальная цена.

Первоначальная цена рассчитывается через рентабельность разрабатываемого проекта. Рентабельность равна 50%, воспользуемся следующей формулой:

$$C_n = C \times \left(1 + \frac{P}{100}\right), \quad (4.21)$$

где P – рентабельность (50%).

Подставив полученные значения, первоначальная цена согласно формуле составит:

$$C_0 = 246560 * \left(1 + \frac{50}{100}\right) = 369840 \text{ тенге.}$$

Цена готовой продукции рассчитывается по формуле:

$$C_p = C_0 + \text{НДС}, \quad (4.22)$$

где НДС – 12% от начальной цены готового продукта.

Рассчитаем НДС используя формулу (4.22)

$$\text{НДС} = 369840 * 0,12 = 44380 \text{ тенге.}$$

Следовательно, конечная итоговая цена программного продукта составит:

$$C_p = 369840 + 44380 = 414220 \text{ тенге.}$$

4.3 Расчет динамических показателей эффективности капитальных вложений

Прибыль от продукта для организации и разработчика будет идти за счет того, что отпадает необходимость в расходных материалах для бумажных отчетов, в среднем ежедневно на расходной материал и поддержку соответствующей аппаратуры расходовалась сумма равная 1215 тенге, а при условии изменения цен стоит предположить, что стоимость обслуживания будет повышаться на 100-120 тенге. Также стоит отметить благоприятный фактор от переноса данных в эл. вид: сократилось время для проведения отчетности продаж

Таблица 4.3 - Исходные данные для оценки проекта

Инвестиционные затраты, тг	Значения P_t , тг.		
	1 год	2 год	3 год
$T = 0$			
414220	443475	479975	487275

Чистый дисконтированный доход (NPV)

Коэффициент дисконтирования рассчитывается по формуле

$$\alpha_t = \frac{1}{(1+r)^t} \quad (4.23)$$

где α_t – коэффициент дисконтирования;

r – норма дисконта;

t – номер шага расчета.

Рассчитывается коэффициент дисконтирования для каждого года.

$$\text{за 1 год: } \alpha_1 = \frac{1}{(1+0,25)^1} = 0,8;$$

$$\text{за 2 год: } \alpha_2 = \frac{1}{(1+0,25)^2} = 0,64;$$

$$\text{за 3 год: } \alpha_3 = \frac{1}{(1+0,25)^3} = 0,512.$$

Чистая приведенная стоимость (Net Present Value – NPV) относится к группе методов дисконтирования денежных потоков или DCF-методов.

Пусть I_0 – сумма первоначальных затрат, т.е. сумма инвестиций на начало проекта;

PV – современная стоимость денежного потока на протяжении экономической жизни проекта.

Общая накопленная величина дисконтированных доходов рассчитывается по формуле

$$PV = \sum_{t=1}^n \frac{P_t}{(1+r)^t} = \sum_{t=1}^n P_t * \alpha_t \quad (4.24)$$

где r – норма дисконта;

n – число периодов реализации проекта;

P_t – чистый поток платежей в периоде t .

Рассчитывается величину дисконтированных доходов по каждому году.

Тогда,

$$PV_1 = 443475 * 0,8 = 354780 \text{ тенге};$$

$$PV_2 = 479975 * 0,64 = 307184 \text{ тенге};$$

$$PV_3 = 487275 * 0,512 = 249485 \text{ тенге}.$$

Общая накопленная величина дисконтированных доходов будет равна

$$PV = 354780 + 307184 + 249485 = 911449 \text{ тенге}.$$

Текущая стоимость затрат (I_0) сравнивается с текущей стоимостью доходов (PV). Разность между ними составляет чистую текущую стоимость проекта (NPV):

$$NPV = \sum_{t=1}^n \frac{CF_t}{(1+r)^t} - I_0 \quad (4.25)$$

$$NPV = 911449 - 414220 = 497229 > 0$$

Индекс прибыльности проекта (PI)

Индекс рентабельности (доходности) инвестиций (PI). Индекс рентабельности представляет собой отношение суммы приведенных эффектов к величине инвестиционных затрат. Индекс рентабельности (PI) рассчитывается по формуле

$$PI = \sum_k \frac{P_k}{(1+r)^k} / I \quad (4.26)$$

Тогда,

$$PI = \frac{911449}{414220} = 2,2 > 1.$$

Дисконтированный срок окупаемости (DPP)

Срок окупаемости инвестиций – один из самых простых и широко распространен в мировой практике, не предполагает временной упорядоченности денежных поступлений [9].

Общая формула расчета показателя DPP имеет вид

$DPP = t$, при котором $P_t > I$

$$DPP = t + \frac{\text{ост. долга инвестору}}{\text{денежный поток}(t+1)} \quad (4.27)$$

где P_t – чистый денежный поток доходов

$$DPP = 1 + \frac{59440}{307184} = 1,19 \text{ года (1 год и 2 месяца)}$$

Что округленно равно 1 год и 2 месяца

Заключение о принятии решения по проекту

В заключении получаем, что чистый дисконтированный доход (NPV) проекта выше нуля и составляет 497229 тенге. По расчету индекс прибыльности проекта (PI) > 1 и равняется 2,2. Срок окупаемости проекта составляет 1 год 2 месяца

5 Безопасность жизнедеятельности

Разработанная информационная система эксплуатируется в браузере. Во время разработки среда, в которой находится разработчик, представляет собой

помещение с организованным местом работы с наличием персонального компьютера. В процессе работы могут возникнуть различного вида риски для физической и умственной деятельности разработчика: плохой микроклимат помещения, несоответственно со стандартом оборудованное рабочее место, высокий уровень шума, блеклость/мерцание дисплея компьютера, плохая освещенность помещения, отсутствие кондиционирования помещения, высокий уровень статического электричества, повышенное электромагнитное поле от оборудования, опасность удара электрическим током от оборудования и другие неблагоприятные факторы.

5.1 Теоретическая часть

По санитарно-эпидемиологическим требованиям по эксплуатации помещений площадь для одного работника должна быть 6 кв.м. Во избежание запыленности в помещении должна ежедневно проводиться влажная уборка. Рабочее помещение должно оборудоваться кондиционером, системой отопления. Для поддержания влажности необходимо проветривание помещения или применение увлажнителей воздуха.

Продолжительность непрерывной работы за компьютером составляет не более 2-х часов. [19]

Согласно нормам микроклимата, при выполнении работ с нервным напряжением следует соблюдать величину температуры воздуха в пределах 22-25 градусов по Цельсию в холодный и теплый периоды года, при относительной влажности 60-40% [19]

Помещение оснащено кондиционером, что позволяет поддерживать вентиляцию, также система отопления позволит поддерживать благоприятную температуру

Освещенность рабочего места программиста рекомендуется в пределах 300–500 люкс. В поле зрения работающего за компьютером не должны быть окна и приборы освещения. Приборы освещения должны быть с рассеивателями, отражение на экране от источников света устраняется при помощи защитных экранов. Яркость света от светильников должна быть не меньше освещенности рабочей поверхности, т.к резкое изменение яркости при смене полей зрения должно быть минимизировано. Оконные проемы должны быть оснащены регулируемыми устройствами от света (жалюзи, занавеси, внешние козырьки и др.).

ЭВМ являются источниками таких излучений как: мягкого рентгеновского; ультрафиолетового 200 - 400 нм; видимого 400 - 700 нм, ближнего инфракрасного 700 - 1050 нм; радиочастотного 3 кГц - 30 МГц; электростатических полей.

При эксплуатации монитор компьютера излучает мягкое рентгеновское излучение. Опасность этого вида излучения связана с его способностью

проникать в тело человека на глубину 1-2 см и поражать поверхностный кожный покров.

Максимальный уровень рентгеновского излучения на рабочем месте программиста обычно не превышает 10 мкбэр/ч, а интенсивность ультрафиолетового и инфракрасного излучений от экрана монитора лежит в пределах 10...100мВт/м².

Ультрафиолетовое излучение в больших дозах приводит к дерматиту кожи, головной боли, рези в глазах.

Инфракрасное излучение приводит к перегреву тканей человека (особенно хрусталика глаза), повышению температуры тела.

Уровни напряженности электростатических полей должны составлять не более 20 кВ/м. Поверхностный электростатический потенциал не должен превышать 500 В. При повышенном уровне напряженности полей следует сократить время работы за компьютером, делать пятнадцатиминутные перерывы в течение полутора часов работы, обязательно применять защитные экраны, не размещать их концентрированно в рабочей зоне и выключать их, если на них не работают [18]

Рассмотрим наше помещение для работы: площадь 42 кв.м, что гораздо больше 6 кв.м, следовательно помещение пригодно для работы за компьютером.

Для безопасной работы на ЭВМ необходимо находиться на расстоянии не менее 50 см от экрана дисплея. Для снижения воздействия всех видов излучения рекомендуется применять мониторы с пониженным уровнем излучения, устанавливать защитные экраны, а также соблюдать регламентированные режимы труда и отдыха.

Наряду с этим нужно устанавливать в помещении, где работает программист, ионизаторы воздуха, чаще проветривать помещение и хотя бы один раз в течение рабочей смены очищать экран от пыли.

Для предотвращения образования и защиты от статического электричества необходимо использовать нейтрализаторы и увлажнители, а полы должны иметь антистатическое покрытие. Допустимые уровни напряженности электростатических полей не должны превышать 20 кВ в течение 1 часа. [18]

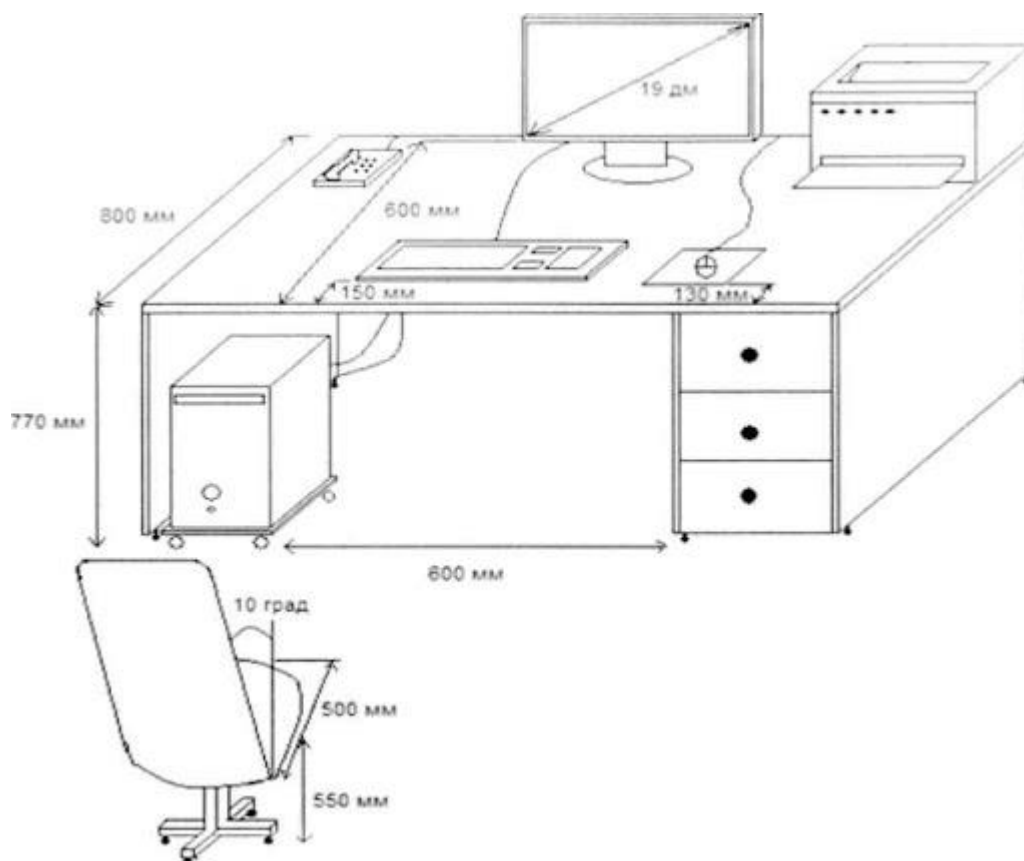


Рис. 5.1 – Рабочее место

На рисунке правильно обустроенное рабочее место.

5.2 Расчетная часть

5.2.1 Разработка системы вентиляции помещения

5.2.1.1 Условие задачи и исходные данные

- 1) Рассчитать тепловые нагрузки в помещении: внутренние и наружные.
- 2) Рассчитать количество воздуха, необходимое для подачи в помещение.
- 3) По найденному значению количества воздуха подобрать соответствующую модель кондиционера.
- 4) Привести основные характеристики выбранного кондиционера.
- 5) Привести схему расположения кондиционера в помещении и схему подачи воздуха.

Исходные данные

Город: Алматы;

Параметры помещения (Д x Ш x В), м: 6 x 7 x 3,5;

$$X_o = 0,39 \text{ Вт/м}^3 \text{ } ^\circ\text{C}$$

$t_{\text{Нрасч}}$ – наружная температура. Для холодного периода – средняя температура самого холодного месяца в 13 часов, для теплого периода – средней температуре самого жаркого месяца в 13 часов.

$t_{\text{Врасч}}$ – внутренняя температура, выбирается с учетом комфортных условий или технологических требований, предъявляемых к производственным процессам.

Для теплого времени года

$$t_{\text{Нрасч}} = 33 \text{ } ^\circ\text{C}$$

$$t_{\text{Врасч}} = 22 \text{ } ^\circ\text{C}$$

$$Q_{\text{огр}} = 147 * 0,39 * (33 - 22) = 630,63 \text{ Дж}$$

Для холодного времени года

$$t_{\text{Нрасч}} = -9 \text{ } ^\circ\text{C}$$

$$t_{\text{Врасч}} = 25 \text{ } ^\circ\text{C}$$

$$Q_{\text{огр}} = 128 * 0,39 * |(-9 - 25)| = 1697,28 \text{ Дж}$$

Избыточная теплота солнечного излучения в зависимости от типа стекла почти до 90% поглощается средой помещения, остальная часть отражается. Максимальная тепловая нагрузка достигается при максимальном уровне излучения, которое имеет прямую и рассеянную составляющие. Интенсивность излучения зависит от ширины местности, времени года и времени суток.

Теплопоступление от солнечного излучения через остекление определяется по формуле:

$$Q_p = (q^I F_o^I + q^{II} F_o^{II}) \cdot \beta_{с.з.}$$

где q^I , q^{II} – тепловые потоки от прямой и рассеянной солнечной радиации, Вт/м²;

F_o^I , F_o^{II} – площади светового проема, облучаемые и необлучаемые прямой солнечной радиацией, м²;

$\beta_{с.з.}$ – коэффициент теплопропускания. По приложению 8 [18]:

$$\beta_{с.з.} = 0,15$$

При отсутствии наружных затеняющих козырьков, ребер и т. д. для периода облучения остекления солнцем, когда его лучи проникают через окно в помещение $F_o^I = F_o$; $F_o^{II} = 0$:

$$Q_p = q^I F_o \cdot \beta_{с.з.} = (q_{en} + q_{ep}) \cdot K_1^c \cdot K_2 \cdot \beta_{с.з.} \cdot n \cdot S_o, \text{ Вт} \quad (1.3)$$

$q_{вп}$; $q_{вр}$ – тепловые потоки от прямой и рассеянной радиации, Вт/м². По таблице 1 [18] для широты в 42⁰с. ш. в 13-14 ч. при расположении Ю:

$$q_{вп} = 222 \text{ Вт/м}^2; q_{вр} = 83 \text{ Вт/м}^2;$$

$F_o = nS_o = 2 \cdot 4 = 8 \text{ м}^2$ – площадь светового проема (n – число окон; S_o – площадь 1 окна);

K_1 – коэффициент затемнения остекления переплетами (K_1^C – для облученных проемов). По таблице 3 [18]:

$$K_1^C = 0,90;$$

K_2 – коэффициент загрязнения остекления. По таблице 4 [18]:

$$K_2 = 0,95.$$

Тогда:

$$Q_p = (222+83)*0,90*0,95*0,15*8 = 312,93 \text{ Дж}$$

По таблице 1 [18] для широты в 50⁰с.ш. до полудня в 11-12 ч. при расположении С:

$$q_{вп} = 0 \text{ Вт/м}^2; q_{вр} = 62 \text{ Вт/м}^2;$$

$F_o = nS_o = 4 \cdot 5 = 20 \text{ м}^2$ – площадь светового проема (n – число окон; S_o – площадь 1 окна);

Тогда:

$$Q_p = (0 + 62)*0,90*0,95*0,15*8 = 63,612 \text{ Дж}$$

Тогда общее тепlopоступление солнечного излучения с обеих окон равно:

$$Q_p = 312,93 + 63,612 = 376,542 \text{ Дж}$$

5.2.2.2 Внутренние тепловые нагрузки

Внутренние нагрузки в жилых, офисных или относящихся к сфере обслуживания помещениях слагаются в основном из тепла:

- выделяемого людьми;
- выделяемого лампами и осветительными, электробытовыми приборами;
- выделяемого компьютерами, печатающими устройствами фотокопировальными машинами пр.;

В производственных и технологических помещениях различного назначения дополнительными источниками тепловыделений могут быть: нагретое производственное оборудование, горячие материалы, в том числе жидкости и различного рода полуфабрикаты, продукты сгорания и химических реакций.

Теплопоступления от людей зависят от интенсивности выполняемой работы и параметров окружающего воздуха. Тепло, выделяемое человеком, складывается из ощутимого (явного), то есть передаваемого в воздух помещения путем конвекции и лучеиспусканий, и скрытого тепла, затрачиваемого на испарение влаги с поверхности кожи и из легких.

Летом при 22 °С один мужчина выделяет явного тепла 61 Вт, а общего - 102 Вт. Тогда выделение явного тепла в помещении составит:

$$Q_{л.я} = 61 \text{ Дж}$$

А выделение общего тепла:

$$Q_{л.я} = 102 \text{ Дж}$$

По таблице [4] зимой при 25 °С один мужчина выделяет явного тепла 85 Вт, а общего – 105 Вт

$$Q_{л.я} = 85 \text{ Дж}$$

А выделение общего тепла: ой кондиционер

$$Q_{л.я} = 105 \text{ Дж}$$

Теплопоступление от осветительных приборов, оргтехники и оборудования рассчитывается следующим образом. Теплопоступление от ламп определяется по формуле:

$$Q_{осв} = \eta \cdot N_{осв} \cdot F_{пол}, \text{ Вт} \quad (1.4)$$

где η – коэффициент перехода электрической энергии в тепловую (для ламп накаливания $\eta=0.95$);

$N_{осв}$ – установленная мощность ламп ($N=60 \text{ Вт/м}^2$);

$F_{пол}$ – площадь пола: $F_{пол} = 6 * 7 = 42 \text{ м}^2$

Тогда:

$$Q_{осв} = 0,95 * 60 * 42 = 2394 \text{ Дж}$$

Тепло, выделяемое производственным оборудованием, определяется по формуле:

$$Q_{об} = N_{уст} \cdot K \quad (1.5)$$

$$Q_{об} = 27,5 * 1 = 27,5 \text{ Дж}$$

Теплопритоки, возникающие за счет находящейся техники, - это 30% мощности оборудования:

$$Q_{\text{орг}} = 0,3 * 27,5 = 8,25 \text{ ДЖ}$$

5.2.1.3 Расчет теплового баланса помещения

На основании выполненных расчетов составим баланс теплопоступлений в помещении:

$$Q_{\text{вс}} = Q_{\delta} + Q_{\text{в}} + Q_{\text{тв}} + Q_{\text{тв}} + Q_{\text{тв}} + Q_{\text{тв}}$$

Лето:

$$Q_{\text{изб}} = 630,63 + 376,542 + 61 + 2394 + 27,5 + 8,25 = 3497,922 \text{ ДЖ}$$

Зима:

$$Q_{\text{изб}} = 1697,28 + 376,542 + 85 + 2394 + 27,5 + 8,25 = 4588,572 \text{ ДЖ}$$

Так как тепловой баланс для лета больше зимнего теплового баланса, то рассчитаем теплонапряженность воздуха по формуле:

$$Q_{\text{н}} = Q_{\text{изб.зима}} * 860 / V_{\text{пом}} = 4588,572 * 860 / 147 = 26,8 \text{ ккал/м}^3.$$

При $Q_{\text{н}} > 20 \text{ ккал/м}^3$, $\Delta t = 8 \text{ }^{\circ}\text{C}$,

Определение количества воздуха, необходимое для поступления в помещение:

$$L = Q_{\text{изб}} * 860 / (C * \Delta t * \gamma) = 4588,572 * 860 / (0,24 * 1000 * 8 * 1,206) = 1704 \text{ м}^3/\text{час}$$

где $C = 0,24 \text{ ккал/(кг}^{\circ}\text{C)}$ – теплоемкость воздуха,

$\gamma = 1,206 \text{ кг/м}^3$ – удельная масса приточного воздуха.

Определение кратности воздухообмена:

$$n = L / V_{\text{пом}} = 1704 / 147 = 11,59 \text{ час}^{-1}.$$

5.2.1.4 Выбор кондиционера. Схема расположения

Исходя из полученных данных, выберем кондиционер сплит-системы мультizonального типа.

Kentatsu KSGB26HFAN1 / KSRB26HFAN1



Рисунок 5.2 - Кондиционер *Kentatsu KSGB26HFAN1*

Основные характеристики кондиционера

Таблица 5.1 – Основные характеристики кондиционера

Разновидность	комплект
Площадь помещения	143 м ²
Напряжение	220 В
Мощность охлаждения	14.3 кВт
Потребляемая мощность при охлаждении	6.3 кВт
Режим обогрева	
Мощность обогрева	17.5 кВт
Потребляемая мощность при обогреве	5 кВт
Режим вентиляции	
Расход воздуха	3000 м ³ /ч
Макс. уровень шума (внутр. блок)	46 дБ
Макс. уровень шума (внеш. блок)	55 дБ
Хладагент	R410a
Ширина (внутр. блок)	1230 мм
Глубина (внутр. блок)	590 мм
Высота (внутр. блок)	380 мм
Вес (внутр. блок)	62 кг
Ширина (внеш. блок)	950 мм
Глубина (внеш. блок)	330 мм
Высота (внеш. блок)	1380 мм
Вес (внеш. блок)	105 кг
Страна-производитель	Ю. Корея

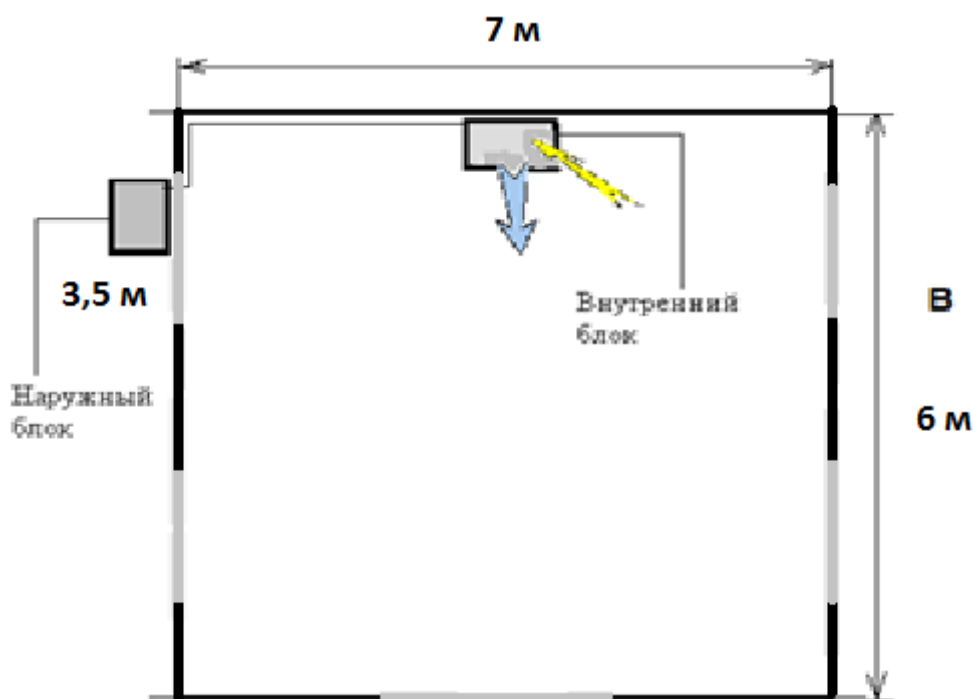


Рисунок 5.2 – Схема расположения кондиционера

Вывод: по проведенными расчетам был подобран кондиционер *Kentatsu KSGB26HFAN1 / KSRB26HFAN1* удовлетворяющий условиям. Изображена схема расположения кондиционера

5.2.1 Расчет категории пожарной безопасности

Помещение № 313 архива площадью 42 кв. м.

Высота потолка - 3.5 м..

В помещении хранятся документы в папках, , ноутбук, стул и стол из дерева, стеллажи и шкаф.

В результате пожарно-технического обследования помещения установлено:

Максимальная горючая загрузка может составлять:

- 1) Бумага – 4 кг.
- 2) Стеллаж-1, стул-1, стол-1. – 90 кг.

В первую очередь определяется пожарная нагрузка.

$Q=m \cdot H_c$, где m – масса , H_c – количество энергии, которая выделяется при сгорании древесины, это величина стандартная (табличная). [18]

В итоге получается: $90 \cdot 13,8 = 1242$ МДж.

Площадь размещения пожарной нагрузки равна площади, которую занимают стеллаж. А так как их размер 6 кв.м и стол 2 кв.м, то площадь будет составлять

8 м². А так как этот размер не превышает 10 м², то за расчетную площадь опасности принимается именно 10 м².

Теперь можно рассчитать удельную нагрузку. Вот ее формула:

$$q=Q/S=1242/10=124,2 \text{ МДж/м}^2. [18]$$

В1. Здесь «q» более 2200 МДж/м².

В2. «q» находится в диапазоне от 1410 до 2200.

В3. Удельная нагрузка – это пределы: 181-1401.

В4. От 1 до 181

По данному параметру опасности помещение, для которого производился расчет, относится к категории «В4» - отличающиеся малой опасностью. [4] В них не требуется системы пожарной сигнализации и пожаротушения

Заключение

В результате выполнения дипломного проекта были выполнены поставленные в цели и задачи. Было разработано техническое задание проекта и написана постановка задачи. В ходе работы было спроектировано информационное обеспечение программного проекта: выбрана СУБД для хранения данных, построена физическая и логическая модели базы данных, на основе которого были созданы и описаны структуры таблиц БД, описаны связи между ними.

Кроме того, была спроектированы диаграммы UML: диаграммы вариантов использования для различных акторов, а также диаграммы последовательности; разработаны основные бизнес-модели и показаны бизнес-объекты приложения.

В ходе проектирования программного обеспечения была описана структура ПО, обоснован выбор инструментально ПО (MySQL, серверный язык PHP), разработан интерфейс и написаны основные классы и методы для нормального функционирования информационной системы. В конце работы были произведены тестирование и отладка, в ходе которых не были выявлены никакие ошибки.

Разработанная информационная система позволит вести учет продаж сотрудников супермаркета, получать отчет по продажам сотрудников, по чекам, за определенные дни, либо за определенный месяц. Также информационная система позволяет вести учет товаров, разработан функционал добавления/удаления товаров, номенклатуры, заказов, сотрудников из созданной базы данных.

Список литературы

- 1 Кузнецов С.Д. Базы данных. – М.: Academia, 2012. – 496 с.
- 2 Моргунов Е.П. Язык SQL. Базовый курс: учеб.-практ. пособие – М.: Postgress Professional, 2017. – 256 с.
- 3 Кузнецов М.В., Симдянов И.В. MySQL 5 – СПб.: БХВ-Петербург, 2010. – 1024 с.
- 4 Горбаченко В.И., Убиенных Г.Ф., Бобрышева Г.В. Проектирование информационных систем с СА ERwin Modeling Suite 7.3: учебное пособие – П.: Издательство ПГУ, 2012. – 154 с.
- 5 Маклафлин Б. PHP и MySQL. Исчерпывающее руководство. 2-е изд. — СПб.: Питер, 2014. — 544 с.: ил. — (Серия «Бестселлеры O'Reilly»).
- 6 Сайфуллин Э. Основные бизнес-процессы в магазине // CRYSTALS.RU: автоматизация розничной торговли, кассовое оборудование для магазина. 1994. URL: <http://www.crystals.ru/articles/kak-upravljat-magazinom>.
- 7 Справочное руководство по MySQL // MYSQL.RU: одобрено лучшими российскими программистами. URL: http://www.mysql.ru/docs/man/My-SQL_Database_Administration.html.
- 8 Официальный мануал «MySQL::MySQL 8.0 Reference Manual». // DEV.MYSQL.COM: Developer Zone. URL: <https://dev.mysql.com/doc/ref-man/8.0>.
- 9 Г.Ш. Боканова. Методические указания по выполнению экономической части дипломных работ для студентов специальностей 5В070400 – Вычислительная техника и программное обеспечение, 5В070300 – «Информационные системы», 5В060200 – «Информатика». – Алматы: АУЭС, 2020 – 35 с.
- 10 Программа учета для супермаркетов URL: http://usu.kz/uchet_v_super-markete.php
- 11 Программа для контроля торговли URL: <https://cloudshop.ru/>
- 12 СанПиН «Санитарно-эпидемиологические требования к условиям работы с источниками физических факторов (компьютеры и видеотерминалы), оказывающих воздействие на человека»
- 13 Кириченко, Дубовик: Динамические сайты на HTML, CSS, JavaScript и Bootstrap. Издательство – Наука и Техника, 2018. – 272 с.
- 14 Дженнифер Нидерст Роббинс: HTML5, CSS3 и JavaScript. Исчерпывающее руководство, 4-ое издание. Издательство – Эксмо, 2014. – 516 с.
- 15 Никсон Р.: Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5. 4-е изд. Издательство – Питер-Трейд, 2016. – 768 с.

16 Пилецкий И.И: Проектирование, разработка и сопровождение баз данных с использованием CASE-средств – Минск, 116 с.

(https://www.bsuir.by/m/12_100229_1_90142.pdf)

17 В. И. Грекул, Н.Л. Коровкина, Г.А. Левочкина. – Проектирование информационных систем : учебник и практикум для среднего профессионального образования – Москва : Издательство Юрайт, 2019. – 385 с.

18. Хакимжанов Т.Е. Расчет аспирационных систем. Дипломное проектирование. Для студентов всех форм обучения всех специальностей. – Алматы: АИЭС, 2002. – 30 с.

19. СНиП РК 2.04-01-2001. Общие строительные нормы и правила устройства систем вентиляции и кондиционирования воздуха. Административные и бытовые здания.

Приложение А

Техническое задание

1 Общие сведения

1.1 Наименование системы

Полное наименование системы:

Информационная система учета продаж супермаркета

1.2 Сроки начала и окончания работ

Дата начала: 17.02.2020

Дата окончания: 09.05.2020

2 Назначение и цели создания системы

2.1 Назначение системы

Информационная система позволяет вести учет по продажам товаров, сотрудников, вести отчетность.

3 Рекомендации к разработке программы

Информационная система может быть разработана в виде веб-приложения.

4 Требование к внешнему виду системы

Конкретных требований к виду интерфейса не имеется. Желательно понятный интерфейс.

4 Технические требования

Специальных технических требований для ПК для работы веб-приложения не имеются, наличие браузера и установка прикладного ПО.

5 Экономические требования

- стоимость разработки продукта составила 414220 тг.

Приложение Б
(листинг программы)

Листинг 1. goods.php

```
<?php include "includes/connection.php"; ?>
<!DOCTYPE html>
<html lang="en">
<?php include "includes/header.html"?>
<body class="fix-header">
  <!--
=====
=== -->
  <!-- Preloader -->
  <!--
=====
=== -->
  <div class="preloader">
    <svg class="circular" viewBox="25 25 50 50">
      <circle class="path" cx="50" cy="50" r="20" fill="none" stroke-width="2"
stroke-miterlimit="10" />
    </svg>
  </div>
  <!--
=====
=== -->
  <!-- Wrapper -->
  <!--
=====
=== -->
  <div id="wrapper">
    <!--
=====
=== -->
  <!-- Подключение навбара -->
  <!--
=====
=== -->
```



```

<?php include "includes/navbar.html"?>
<!--

=====

=== -->
<!-- Подключение сайдбара -->
<!--

=====

=== -->
<?php include "includes/sidebar.html"?>
<!--

=====

=== -->
<!-- Блок контента -->
<!--

=====

=== -->
<div id="page-wrapper">
  <div class="container-fluid">
    <div class="row bg-title">
      <div class="col-lg-3 col-md-4 col-sm-4 col-xs-12">
        <h4 class="page-title">Товары</h4>
      </div>
      <div class="col-lg-9 col-sm-8 col-md-8 col-xs-12">
        <ol class="breadcrumb">
          <li><a href="/">Главная</a></li>
          <li class="active">Товары</li>
        </ol>
      </div>
      <!-- /.col-lg-12 -->
    </div>
    <div class="row">
      <div class="col-md-12">
        <div class="white-box">
          <?php
            if ($_GET['table'] == 'goods') { ?>
              <div class="card mb-2 wow fadeIn" style="margin-bottom:
20px; padding-bottom: 20px;">
                <div class="card-body col-sm-12 d-flex justify-content-be-
tween con">
                  <div class="col-sm-4">

```

Продолжение приложения Б.

```
<a href="goods.php?add=goods" class="btn btn-primary
waves-effect">Добавить товар</a>
</div>
<form class="">
  <div class="col-sm-4 d-flex justify-content-end"
style="float: right; display: flex; align-items: center;">
    <div class="col-sm-8">
      <input type="search" style="height: 32px; margin-
right: 15px;" class="form-control" placeholder="Articul" name="">
    </div>
    <div class="col-sm-2">
      <button class="mr-2 btn btn-primary btn-sm my-
0" type="submit">
        <i class="fa fa-search"></i>
      </button>
    </div>
  </div>
</form>
</div>
</div>
<?php
$sql = "SELECT * FROM goods";
$result = mysqli_query($connection, $sql); ?>
<div class="card mb-2 wow fadeIn" style="padding: 5px;
white-space: nowrap;">
  <table class="table table-striped">
    <thead class="thead-dark">
      <tr>
        <th>Артикул</th>
        <th>ID ткани</th>
        <th>ID категории</th>
        <th>Наименование</th>
        <th>Цена</th>
        <th>ID модели</th>
        <th>Бренд</th>
        <th>Страна</th>
      </tr>
    </thead>
  </table>
</div>
<?php
```

```

while ($row = mysqli_fetch_assoc($result)) {
    echo '<tr>
        <td>'.$row["Articul"].'</td>
        <td>'.$row["ID_Cloth"].'</td>
        <td>'.$row["ID_Category_Goods"].'</td>
        <td>'.$row["Name"].'</td>
        <td>'.$row["Price"].'</td>
        <td>'.$row["ID_Model"].'</td>
        <td>'.$row["Brand"].'</td>
        <td>'.$row["ID_Country"].'</td>
    </tr>';
}
echo "</table></div>";
} ?>
<?php
if ($_GET['add'] == 'goods') {
    $fetchClothQuery = "SELECT * FROM `cloth`";
    $fetchCategoryQuery = "SELECT * FROM `goodscategory`";
    $fetchBrandQuery = "SELECT * FROM `brand`";
        Продолжение приложения Б.

    $fetchModelQuery = "SELECT * FROM `model`";
    $fetchCountryQuery = "SELECT * FROM `production`";
    $clothes = mysqli_query($connection, $fetchClothQuery);
    $categories = mysqli_query($connection, $fetchCatego-
ryQuery);

    $brands = mysqli_query($connection, $fetchBrandQuery);
    $models = mysqli_query($connection, $fetchModelQuery);
    $countries = mysqli_query($connection, $fetchCountryQuery);
?>
<main class="pt-5 max-lg-5">
    <div class="container-fluid mt-3">
        <div class="good-add-form col-sm-8">
            <h4 class="coworker-form-title text-center mb-4">Форма
добавления товара</h4>
            <form action="goods.php?add=goods" method="POST"
id="form">
                <div class="form-group row">
                    <label for="staticEmail" class="col-sm-4 col-form-la-
bel">Артикул</label>
                    <div class="col-sm-8">

```

```

        <input name="articul" type="text" class="form-control"
id="inputPassword" required>
    </div>
</div>
<div class="form-group row">
    <label for="staticEmail" class="col-sm-4 col-form-la-
bel">Состав ткани</label>
    <div class="col-sm-8">
        <select name="idcloth" class="form-control" id="exam-
pleFormControlSelect1" required>;
            <?php
                foreach ($clothes as $row) {
                    echo '<option value="'. $row["ID_Cloth"] ."'>' .
$row["Definition_Of_Cloth"] . '</option>';
                };?>
            echo '</select>
        </div>
    </div>
<div class="form-group row">
    <label for="staticEmail" class="col-sm-4 col-form-la-
bel">Категория</label>
    <div class="col-sm-8">
        <select name="idcat" class="form-control" id="example-
FormControlSelect1" required>;
            <?php
                foreach ($categories as $row) {
                    echo '<option value="'. $row["ID_Category_Goods"]
.'">' . $row["Name_Of_Cat_Goods"] . '</option>';
                };
            ?>
        </select>
    </div>
</div>
<div class="form-group row">
    <label for="staticEmail" class="col-sm-4 col-form-la-
bel">Наименование</label>
    <div class="col-sm-8">
        <input name="name_" type="text" class="form-control"
id="inputPassword" required>
    </div>
</div>

```

```

<div class="form-group row">
  <label for="staticEmail" class="col-sm-4 col-form-label">Цена</label>
  <div class="col-sm-8">
    <input name="price" type="text" class="form-control"
id="inputPassword" required>
  </div>
</div>

```

Продолжение приложения Б.

```

<div class="form-group row">
  <label for="staticEmail" class="col-sm-4 col-form-label">Модель</label>
  <div class="col-sm-8">
    <select name="model" class="form-control" id="exampleFormControlSelect1" required>;
    <?php
      foreach ($models as $row) {
        echo '<option value="'. $row["ID_Model"] ."'>' .
$row["Model_Definition"] . '</option>';
      };
    ?>
  </select>
</div>
</div>

```

```

<div class="form-group row">
  <label for="staticEmail" class="col-sm-4 col-form-label">Бренд</label>
  <div class="col-sm-8">
    <select name="brand" class="form-control" id="exampleFormControlSelect1" required>;
    <?php
      foreach ($brands as $row) {
        echo '<option value="'. $row["ID_Brand"] ."'>' .
$row["Name_Of_Brand"] . '</option>';
      };
    ?>
  </select>
</div>
</div>
<div class="form-group row">

```

```

        <label for="staticEmail" class="col-sm-4 col-form-label">Страна</label>
        <div class="col-sm-8">
            <select name="country" class="form-control" id="exampleFormControlSelect1" required>;
                <?php
                    foreach ($countries as $row) {
                        echo '<option value="'. $row["ID_Country"] ."'>' .
                            $row["Country"] . '</option>';
                    };
                ?>
            </select>
        </div>
        <div class="d-flex justify-content-end">
            <button type="submit" class="btn btn-primary"
name="addgoods">Добавить</button>
        </div>
    </form>
</div>
</div>
</s>
<?php
if (isset($_POST['addgoods'])) {
    $articul = $_POST['articul'];
    $idcloth = $_POST['idcloth'];
    $idcat = $_POST['idcat'];
    $name_ = $_POST['name_'];
    $price = $_POST['price'];
    $model = $_POST['model'];
    $brand = $_POST['brand'];
    $country = $_POST['country'];
    $sql6 = "SELECT * FROM goods WHERE `Articul` = '$ar-
articul'";

```

Продолжение приложения Б.

```

$query6 = mysqli_query($connection, $sql6);
$result6 = mysqli_num_rows($query6);
if ($result6 == 1 ) {
    echo <main class="pt-5 max-lg-5">
        <div class="container-fluid mt-5">

```

```

        <div class="add-form col-sm-4">
            <div class="success-al alert alert-danger"
role="alert">
                Данные о товаре с артикулом <span style="font-
weight: bold;"> ' . $articul. '</span> уже были добавлены!
            </div>
        </div>
    </div>
</main><br>';
    die();
}
if (empty($articul) OR empty($idcloth) OR empty($idcat)
OR empty($name_) OR empty($price) OR
empty($model) OR empty($brand) OR empty($country)) {
    echo '<div class="success-al alert alert-danger"
role="alert">
        Заполните пустые поля!
    </div>';
    die();
}
$query = "INSERT INTO goods VALUES('$articul',
'idcloth', '$idcat', '$name_', '$price', '$model', '$brand', '$country')";
$result = mysqli_query($connection, $query);
echo '<div class="success-al alert alert-success"
role="alert">
        Данные успешно добавлены
    </div>';
}
echo '</div>';
}??>
</div>
</div>
</div>
</div>
<!-- /.container-fluid -->
<!-- Подключение футер -->
<!--
=====
=== -->
<?php include "includes/footer.html"?>

```

```

<!--
=====
=== -->
</div>
<!--
=====
=== -->
<!-- End Page Content -->
<!--
=====
=== -->
</div>
<!-- /#wrapper -->
<!-- jQuery -->
<script src="../../plugins/bower_components/jquery/dist/jquery.min.js"></script>
<!-- Bootstrap Core JavaScript -->
<script src="bootstrap/dist/js/bootstrap.min.js"></script>
<!-- Menu Plugin JavaScript -->
<script src="../../plugins/bower_components/sidebar-nav/dist/sidebar-
nav.min.js"></script>
<!--slimscroll JavaScript -->
<script src="js/jquery.slimscroll.js"></script>
<!--Wave Effects -->

```

Продолжение приложения Б.

```

<script src="js/waves.js"></script>
<!-- Custom Theme JavaScript -->
<script src="js/custom.min.js"></script>
</body>

</html>

```

Листинг 2. coworker.php

```

<?php include "includes/connection.php"; ?>
<!DOCTYPE html>
<html lang="en">
<?php include "includes/header.html"?>
<body class="fix-header">

```



```

<!--
=====
==== -->
<!-- Preloader -->
<!--
=====
==== -->
<div class="preloader">
  <svg class="circular" viewBox="25 25 50 50">
    <circle class="path" cx="50" cy="50" r="20" fill="none" stroke-width="2"
stroke-miterlimit="10" />
  </svg>
</div>
<!--
=====
==== -->
<!-- Wrapper -->
<!--
=====
==== -->
<div id="wrapper">
  <!--
=====
==== -->
<!-- Подключение навбара -->
<!--
=====
==== -->
<?php include "includes/navbar.html"?>
<!--
=====
==== -->
<!-- Подключение сайдбара -->
<!--
=====
==== -->
<?php include "includes/sidebar.html"?>
<!--
=====
==== -->
<!-- Блок контента -->

```

```

<!--
=====
=== -->
<div id="page-wrapper">
  <div class="container-fluid">
    <div class="row bg-title">
      <div class="col-lg-3 col-md-4 col-sm-4 col-xs-12">
        <h4 class="page-title">Сотрудники</h4>
      </div>
      <div class="col-lg-9 col-sm-8 col-md-8 col-xs-12">
        <ol class="breadcrumb">
          <li><a href="/">Главная</a></li>
          <li class="active">Сотрудники</li>
        </ol>
      </div>
      <!-- /.col-lg-12 -->
    </div>
    <div class="row">
      <div class="col-md-12">
        <div class="white-box">
          <?php
            Продолжение приложения Б.

            if ($_GET['table'] == 'coworker') {
              $sql = "SELECT * FROM coworker";
              $result = mysqli_query($connection, $sql);
            ?>
            <div class="card mb-2 wow fadeIn" style="margin-bottom:
20px; padding-bottom: 20px;">
              <div class="card-body col-sm-12 d-flex justify-content-
between con">
                <div class="col-sm-4">
                  <a class="btn btn-primary fff"
href="coworker.php?add=coworker">Добавить сотрудника</a>
                </div>
                <form class="">
                  <div class="col-sm-4 d-flex justify-content-end"
style="float: right; display: flex; align-items: center;">
                    <div class="col-sm-8">
                      <input type="search" style="height: 32px; mar-
gin-right: 15px;" class="form-control" placeholder="dd-mm-yyuy" name="">

```

```

        </div>
        <div class="col-sm-2">
            <button class="mr-2 btn btn-primary btn-sm
my-0" type="submit">
                <i class="fa fa-search"></i>
            </button>
        </div>
    </div>
</form>
</div>
<div class="card mb-2 wow fadeIn" style="padding: 5px;
white-space: nowrap;">
    <table class="table table-striped">
        <thead class="thead-dark">
            <tr>
                <th>ID Сотрудника</th>
                <th>ID Должности</th>
                <th>Банковский счет</th>
                <th>Полное имя</th>
                <th>ИИН</th>
                <th>Дата принятия</th>
                <th>Контактный номер</th>
                <th><div class="trash-icon"></div></th>
            </tr>
        </thead>
        <?php
            while ($row = mysqli_fetch_assoc($result)) {
                $id = $row["ID_Co_Worker"];
                echo <tr>
                    <td><a style="color: #17a2b8;"
href="coworker.php?coworkerid='.$id.'">'.$row["ID_Co_Worker"].'</a></td>
                    <td>'.$row["ID_Position"].'</td>
                    <td>'.$row["Bank_Account"].'</td>
                    <td>'.$row["Full_Name"].'</td>
                    <td>'.$row["IIN"].'</td>
                    <td>'.$row["Date_Of_Adoption"].'</td>
                    <td>'.$row["Contact_Num"].'</td>
                    <td><a style="color: red"
href="coworker.php?table=coworker&delete='.$row["ID_Co_Worker"].' "
>удалить</a></td>

```

```

        </tr>;
    }
    Продолжение приложения Б.

    echo '</table></div>';
?>

<?php
if ($_GET['delete']) {
    if(isset($_GET['delete'])) {
        $id = $_GET['delete'];
        $sql = "SELECT * FROM `coworker` WHERE
`ID_Co_Worker` = $id";
        $result = mysqli_query($connection, $sql);
        while ($row = mysqli_fetch_assoc($result)) {
            echo '
                <div class="my-modal">
                    <div class="modal-dialog" role="document">
                        <div class="modal-content">
                            <div class="modal-header">
                                <h5 class="modal-title" id="exampleModalLa-
bel">Удаление сотрудника</h5>
                                <button type="button" class="close close-
modal">
                                    <span aria-hidden="true">&times;</span>
                                </button>
                            </div>
                            <div class="modal-body">
                                Вы уверены что хотите удалить сотрудника
                                <span style="color: #17a2b8"> ! $row['Full_Name'] .' </span>
                            </div>
                            <div class="modal-footer justify-content-end"
                                style="display: flex">
                                <div>
                                    <button type="button" class="close-modal btn
                                btn-secondary">Нет</button> </div> <div>
                                    <form action="coworker.php?ta-
                                ble=coworker&delete=' . $row["ID_Co_Worker"]. '" method="POST">
                                        <button type="submit" name="deleteCo-
                                worker" style="margin-left: 10px;" class="btn btn-primary">Да</button>
                                    </form>
            '
        }
    }
}

```

```

        </div>
    </div>
</div>
</div>';

if (isset($_POST['deleteCoworker'])) {
    $sql = "DELETE FROM `coworker` WHERE
`ID_Co_Worker` = '$id'";
    if ($result = mysqli_query($connection, $sql)) {
        header("Location: coworker.php?ta-
ble=coworker");
        echo("<meta http-equiv='refresh' content='1'>");
    }
}
}??>
<?php }
}
}
elseif ($_GET['add'] == 'coworker') {
    $fetchPositionsQuery = "SELECT * FROM `position`";
    $positions = mysqli_query($connection, $fetchPosi-
tionsQuery);
    echo '<main class="pt-5 max-lg-5">
    <div class="container-fluid mt-5">
    <div class="coworker-add-form col-sm-8">

```

Продолжение приложения Б.

```

    <h4 class="coworker-form-title text-center mb-
4">Форма добавления сотрудника</h4>
    <form action="coworker.php?add=coworker"
method="POST">
        </select>
        <div class="form-group row">
            <label for="staticEmail" class="col-sm-4 col-
form-label">ID сотрудника</label>
            <div class="col-sm-8">
                <input name="coworkerId" type="text"
class="form-control" id="inputPassword" required>
            </div>
        </div>
    </div>

```

```

        <div class="form-group row">
        <label for="staticEmail" class="col-sm-4 col-
form-label">Полное имя</label>
        <div class="col-sm-8">
        <input name="fullName" type="text"
class="form-control" id="inputPassword" required>
        </div>
        </div>
        <div class="form-group row">
        <label for="staticEmail" class="col-sm-4 col-
form-label">Дата рождения</label>
        <div class="col-sm-8">
        <input name="dateOfBirth" type="text"
class="form-control" id="inputPassword" required>
        </div>
        </div>
        <div class="form-group row">
        <label for="staticEmail" class="col-sm-4 col-
form-label">ИИН</label>
        <div class="col-sm-8">
        <input name="iin" type="text" class="form-con-
trol" id="inputPassword" required>
        </div>
        </div>
        <div class="form-group row">
        <label for="staticEmail" class="col-sm-4 col-
form-label">Должность</label>
        <div class="col-sm-8">
        <select name="positionId" class="form-control"
id="exampleFormControlSelect1">;
        foreach ($positions as $row) {
        echo '<option value="'. $row["ID_Position"]
.'">' . $row["Name_Of_Position"] . '</option>';
        };
        echo '</select>
        </div>
        </div>
        <div class="form-group row">
        <label for="staticEmail" class="col-sm-4 col-
form-label">Банковский счет</label>
        <div class="col-sm-8">

```

```

        <input name="bankAccount" type="text"
class="form-control" id="inputPassword" required>
    </div>
</div>

<div class="form-group row">
    <label for="staticEmail" class="col-sm-4 col-
form-label">Опыт работы</label>
    <div class="col-sm-8">
        Продолжение приложения Б.
        <input name="expirence" type="text"
class="form-control" id="inputPassword" required>
    </div>
</div>
<div class="form-group row">
    <label for="staticEmail" class="col-sm-4 col-
form-label">Адрес</label>
    <div class="col-sm-8">
        <input name="address" type="text" class="form-
control" id="inputPassword" required>
    </div>
</div>
<div class="form-group row">
    <label for="staticEmail" class="col-sm-4 col-
form-label">Дата принятия</label>
    <div class="col-sm-8">
        <input name="dateOfAdoption" type="text"
class="form-control" id="inputPassword" required>
    </div>
</div>
<div class="form-group row">
    <label for="staticEmail" class="col-sm-4 col-
form-label">Номер телефона</label>
    <div class="col-sm-8">
        <input name="phoneNumber" type="text"
class="form-control" id="inputPassword" required>
    </div>
</div>
<div class="d-flex justify-content-end">

```

```

        <button style="float: right" type="submit"
class="addcoworker btn btn-primary" name="addcoworker">Добавить</button>
    </div>
</form>
</div>
</div>
</main>';
if (isset($_POST['addcoworker'])) {
    $expiryBonus = '0';
    $coworkerId = $_POST['coworkerId'];
    $positionId = $_POST['positionId'];
    $bankAccount = $_POST['bankAccount'];
    $iin = $_POST['iin'];
    $expiry = $_POST['expiry'];
    $fullName = $_POST['fullName'];
    $address = $_POST['address'];
    $dateOfBirth = $_POST['dateOfBirth'];
    $dateOfAdoption = $_POST['dateOfAdoption'];
    $phoneNumber = $_POST['phoneNumber'];
    $query = "INSERT INTO `coworker` VAL-
UES('$coworkerId', '$positionId', '$bankAccount', '$fullName', '$expiry',
$dateOfBirth', '$address', '$iin', '$dateOfAdoption', '$expiryBonus', '$pho-
neNumber')";

    if ($result = mysqli_query($connection, $query)) {
        echo '
<main class="pt-5 max-lg-5">
<div class="container-fluid mt-5">
<div class="add-form col-sm-4">
<div class="success-al alert alert-success"
role="alert">

```

Продолжение приложения Б.

```

        Данные успешно сохранены!
    </div>
</div>
</div>
</main>';
    }
}
}

```



```

elseif ($_GET['coworkerid']) {
    if(isset($_GET['coworkerid'])) {
        $id = $_GET['coworkerid'];
        $sql = "SELECT * FROM coworker WHERE
`ID_Co_Worker` = '$id'";
        $fetchPositionsQuery = "SELECT * FROM `posi-
tion`";
        $positions = mysqli_query($connection,
$fetchPositionsQuery);
        $result = mysqli_query($connection, $sql);
        while ($row = mysqli_fetch_assoc($result)) {
            echo '<main class="pt-5 max-lg-5">
<div class="container-fluid mt-5">
<div class="coworker-add-form col-sm-8">
<h4 class="coworker-form-title text-center
mb-4">Форма изменения сотрудника</h4>
<form action="coworker.php?coworkerid='.
$id.'" method="POST">
<div class="form-group row">
<label for="staticEmail" class="col-sm-4
col-form-label">Полное имя</label>
<div class="col-sm-8">
<input name="fullName" type="text"
class="form-control" id="inputPassword" value="'. $row['Full_Name'] .'" re-
quired>
</div>
</div>
<div class="form-group row">
<label for="staticEmail" class="col-sm-4
col-form-label">Дата рождения</label>
<div class="col-sm-8">
<input name="dateOfBirth" type="text"
class="form-control" id="inputPassword" value="'. $row["Date_Of_Birth"] .'" re-
quired>
</div>
</div>
<div class="form-group row">
<label for="staticEmail" class="col-sm-4
col-form-label">ИИН</label>
<div class="col-sm-8">

```

```

        <input name="iin" type="text"
class="form-control" id="inputPassword" value="" . $row["IIN"] . "" required>
    </div>
</div>
<div class="form-group row">
    <label for="staticEmail" class="col-sm-4
col-form-label">Должность</label>
    <div class="col-sm-8">
        <select name="positionId" class="form-
control" id="exampleFormControlSelect1">;
            foreach ($positions as $value) {
                echo '<option value="" . $row["ID_Positi-
tion"] . "">' . $value["Name_Of_Position"] . '</option>';
            };
            echo '</select>

```

Продолжение приложения Б.

```

    </div>
</div>
<div class="form-group row">
    <label for="staticEmail" class="col-sm-4
col-form-label">Банковский счет</label>
    <div class="col-sm-8">
        <input name="bankAccount" type="text"
class="form-control" id="inputPassword" value="" . $row["Bank_Account"] . "" re-
quired>
    </div>
</div>
<div class="form-group row">
    <label for="staticEmail" class="col-sm-4
col-form-label">Опыт работы</label>
    <div class="col-sm-8">
        <input name="expirence" type="text"
class="form-control" id="inputPassword" value="" . $row["Experience"] . "" re-
quired>
    </div>
</div>
<div class="form-group row">
    <label for="staticEmail" class="col-sm-4
col-form-label">Адрес</label>

```

```

        <div class="col-sm-8">
            <input name="address" type="text"
class="form-control" id="inputPassword" value="" . $row["Address"] . "" required>
        </div>
    </div>
    <div class="form-group row">
        <label for="staticEmail" class="col-sm-4
col-form-label">Дата принятия</label>
        <div class="col-sm-8">
            <input name="dateOfAdoption"
type="text" class="form-control" id="inputPassword" value="" .
$row["Date_Of_Adoption"] . "" required>
        </div>
    </div>
    <div class="form-group row">
        <label for="staticEmail" class="col-sm-4
col-form-label">Номер телефона</label>
        <div class="col-sm-8">
            <input name="phoneNumber" type="text"
class="form-control" id="inputPassword" value="" . $row["Contact_Num"] . "" re-
quired>
        </div>
    </div>
    <div class="d-flex justify-content-end">
        <button type="submit" style="float: right;"
class="btn btn-primary" name="updateCoworker">Сохранить</button>
    </div>
</form>
</div>
</div>
</main>'; ?>
<?php
if (isset($_POST['updateCoworker'])) {
    $expiryBonus = '0';
    $coworkerId = $_GET['coworkerid'];
    $positionId = $_POST['positionId'];
    $bankAccount = $_POST['bankAccount'];

```

Продолжение приложения Б.

```

    $iin = $_POST['iin'];
    $expiry = $_POST['expiry'];

```

```

        $fullName = $_POST['fullName'];
        $address = $_POST['address'];
        $dateOfBirth = $_POST['dateOfBirth'];
        $dateOfAdoption = $_POST['dateOfAdoption'];
        $phoneNumber = $_POST['phoneNumber'];
        $query = "UPDATE `coworker` SET
        `ID_Position` = '$positionId',
        `Bank_Account` = '$bankAccount',
        `Full_Name` = '$fullName',
        `Experience` = '$expiarence',
        `Date_Of_Birth` = '$dateOfBirth',
        `Address` = '$address',
        `IIN` = '$iin',
        `Date_Of_Adoption` = '$dateOfAdoption',
        `Experience_Bonus` = '$expiarenceBonus',
        `Contact_Num` = '$phoneNumber' WHERE
`ID_Co_Worker` = '$coworkerId'";

        if ($result = mysqli_query($connection, $query)) {
?>
            <main class="pt-5 max-lg-5">
                <div class="container-fluid mt-5">
                    <div class="add-form col-sm-4">
                        <div class="success-al alert alert-success"
role="alert">
                            Данные успешно сохранены!
                        </div>
                    </div>
                </div>
            </main>
            <?php
                }
                echo("<meta http-equiv='refresh' con-
tent='1'>");
            }
        }
    }
?>
</div>
</div>

```

```

        </div>
    </div>
    <!-- /.container-fluid -->
    <!-- Подключение футер -->
    <!--

=====
=== -->
    <?php include "includes/footer.html"?>
    <!--

=====
=== -->
    </div>
    <!--

=====
=== -->
    <!-- End Page Content -->
    <!--

=====
=== -->
    </div>
    <!-- /#wrapper -->
    <!-- jQuery -->
    <script src="../../plugins/bower_components/jquery/dist/jquery.min.js"></script>
    <!-- Bootstrap Core JavaScript -->
    <script src="bootstrap/dist/js/bootstrap.min.js"></script>
        Продолжение приложения Б.

    <!-- Menu Plugin JavaScript -->
    <script src="../../plugins/bower_components/sidebar-nav/dist/sidebar-
nav.min.js"></script>
    <!--slimscroll JavaScript -->
    <script src="js/jquery.slimscroll.js"></script>
    <!--Wave Effects -->
    <script src="js/waves.js"></script>
    <!-- Custom Theme JavaScript -->
    <script src="js/custom.min.js"></script>

    <script src="js/toastr.js"></script>
    <script src="js/custom.js"></script>
</body>

```

```
</html>
```

Листинг 3. order.php

```
<?php include "includes/connection.php"; ?>
<!DOCTYPE html>
<html lang="en">
<?php include "includes/header.html"?>
<body class="fix-header">
  <!--
=====
=== -->
  <!-- Preloader -->
  <!--
=====
=== -->
  <div class="preloader">
    <svg class="circular" viewBox="25 25 50 50">
      <circle class="path" cx="50" cy="50" r="20" fill="none" stroke-width="2"
stroke-miterlimit="10" />
    </svg>
  </div>
  <!--
=====
=== -->
  <!-- Wrapper -->
  <!--
=====
=== -->
  <div id="wrapper">
    <!--
=====
=== -->
    <!-- Подключение навбара -->
    <!--
=====
=== -->
    <?php include "includes/navbar.html"?>
    <!--
=====
=== -->
```

```

<!-- Подключение сайдбара -->
<!--
=====
=== -->
<?php include "includes/sidebar.html"?>
<!--
=====
=== -->
<!-- Блок контента -->
<!--
=====
=== -->
<div id="page-wrapper">
  <div class="container-fluid">
    <div class="row bg-title">
      <div class="col-lg-3 col-md-4 col-sm-4 col-xs-12">
        <h4 class="page-title">Заказы</h4>
      </div>
      <div class="col-lg-9 col-sm-8 col-md-8 col-xs-12">
        <ol class="breadcrumb">
          <li><a href="/">Главная</a></li>
          <li class="active">Заказы</li>
          Продолжение приложения Б.
        </ol>
      </div>
    </div>
    <!-- /.col-lg-12 -->
  </div>
  <div class="row">
    <div class="col-md-12">
      <div class="white-box">
        <?php
        if ($_GET['table'] == 'order') {
          $sql = "SELECT * FROM ord JOIN ord_size ON ord.`ID_Order` =
ord_size.`ID_Order`";
          $result = mysqli_query($connection, $sql);
          echo <main class="pt-5 max-lg-5">
            <div class="container-fluid mt-5">
              <div class="card mb-2 wow fadeIn">
                <div class="card-body d-sm-flex justify-content-between
con">

```

```

        <a href="order.php?add=order" class="btn btn-primary
waves-effect">Добавить заказ</a>
    </div>
</div>
<div class="card mb-2 wow fadeIn" style="padding: 5px;
white-space: nowrap;">
    <table class="table table-striped">
        <thead class="thead-dark">
            <tr>
                <th>Номер заказа</th>
                <th>Номер сотрудника</th>
                <th>Дата заказа</th>
                <th>Штрих-код</th>
                <th>Количество</th>
            </tr>
        </thead>
        ;
        while ($row = mysqli_fetch_assoc($result)) {
            echo <tr>
                <td>'.$row["ID_Order"].'</td>
                <td>'.$row["ID_Co_Worker"].'</td>
                <td>'.$row["Order_Date"].'</td>
                <td>'.$row["Bar_Code"].'</td>
                <td>'.$row["Amount"].'</td>
            </tr>;
        }
        echo "</table></div>";
    }

    elseif ($_GET['add'] == 'order') {
        echo <main class="pt-5 max-lg-5">
            <div class="container-fluid mt-3">
                <div class="good-add-form col-sm-8">
                    <h4 class="coworker-form-title text-center mb-4">Форма
добавления заказа</h4>
                    <form action="order.php?table=goods&add=goods"
method="POST" id="form">
                        <div class="form-group row">
                            <label for="staticEmail" class="col-sm-4 col-form-la-
bel">Номер заказа</label>
                            <div class="col-sm-8">

```



```

        <input type="password" class="form-control" id="in-
putPassword" >
        </div>
    </div>
    <div class="form-group row">
        <label for="staticEmail" class="col-sm-4 col-form-la-
bel">Номер сотрудника</label>
        Продолжение приложения Б.

        <div class="col-sm-8">
            <input type="password" class="form-control" id="in-
putPassword" >
            </div>
        </div>
        <div class="form-group row">
            <label for="staticEmail" class="col-sm-4 col-form-la-
bel">Дата заказа</label>
            <div class="col-sm-8">
                <input type="password" class="form-control" id="in-
putPassword" >
                </div>
            </div>
            <div class="form-group row">
                <label for="staticEmail" class="col-sm-4 col-form-la-
bel">Штрих-код</label>
                <div class="col-sm-8">
                    <input type="password" class="form-control" id="in-
putPassword" >
                    </div>
                </div>
                <div class="form-group row">
                    <label for="staticEmail" class="col-sm-4 col-form-la-
bel">Количество</label>
                    <div class="col-sm-8">
                        <input type="password" class="form-control" id="in-
putPassword" >
                        </div>
                    </div>
                    <div class="d-flex justify-content-end">
                        <button type="submit" class="btn btn-primary"
name="addgoods">Добавить</button>

```

```

        </div>
    </form>
</div>
</div>
</main>;
    }?>
    </div>
</div>
</div>
</div>
<!-- /.container-fluid -->
<!-- Подключение футер -->
<!--
=====
=== -->
    <?php include "includes/footer.html"?>
    <!--
=====
=== -->
    </div>
    <!--
=====
=== -->
    <!-- End Page Content -->
    <!--
=====
=== -->
    </div>
    <!-- /#wrapper -->
    <!-- jQuery -->
    <script src="../../plugins/bower_components/jquery/dist/jquery.min.js"></script>
    <!-- Bootstrap Core JavaScript -->
    <script src="bootstrap/dist/js/bootstrap.min.js"></script>
    <!-- Menu Plugin JavaScript -->
    <script src="../../plugins/bower_components/sidebar-nav/dist/sidebar-
nav.min.js"></script>
    <!--slimscroll JavaScript -->
    <script src="js/jquery.slimscroll.js"></script>
    <!--Wave Effects -->
    <script src="js/waves.js"></script>
    <!-- Custom Theme JavaScript -->

```

Продолжение приложения Б.

```
<script src="js/custom.min.js"></script>
</body>

</html>
```

Листинг 4. receipt.php

```
<?php include "includes/connection.php"; ?>
<!DOCTYPE html>
<html lang="en">
<?php include "includes/header.html"?>
<body class="fix-header">
  <!--
=====
=== -->
  <!-- Preloader -->
  <!--
=====
=== -->
  <div class="preloader">
    <svg class="circular" viewBox="25 25 50 50">
      <circle class="path" cx="50" cy="50" r="20" fill="none" stroke-width="2"
stroke-miterlimit="10" />
    </svg>
  </div>
  <!--
=====
=== -->
  <!-- Wrapper -->
  <!--
=====
=== -->
  <div id="wrapper">
    <!--
=====
=== -->
  <!-- Подключение навбара -->
```

```

<!--
=====
=== -->
<?php include "includes/navbar.html"?>
<!--

=====
=== -->
<!-- Подключение сайдбара -->
<!--

=====
=== -->
<?php include "includes/sidebar.html"?>
<!--

=====
=== -->
<!-- Блок контента -->
<!--

=====
=== -->
<div id="page-wrapper">
  <div class="container-fluid">
    <div class="row bg-title">
      <div class="col-lg-3 col-md-4 col-sm-4 col-xs-12">
        <h4 class="page-title">Фискальные чеки</h4>
      </div>
      <div class="col-lg-9 col-sm-8 col-md-8 col-xs-12">
        <ol class="breadcrumb">
          <li><a href="/">Главная</a></li>
          <li class="active">Фискальные чеки</li>
        </ol>
      </div>
      <!-- /.col-lg-12 -->
    </div>
    <div class="row">
      <div class="col-md-12">
        <div class="white-box">
          <?php if ($_GET['table'] == 'receipt') {
            $sql = "SELECT * FROM fiscalreceipt";
            $result = mysqli_query($connection, $sql);
            Продолжение приложения Б.
          }
        </div>
      </div>
    </div>
  </div>
</div>

```

```

echo '<main class="pt-5 max-lg-5">
    <div class="container-fluid mt-5">
        <div class="card mb-2 wow fadeIn">
            <div class="card-body d-sm-flex justify-content-
between con">

                </div>
            </div>
            <div class="card mb-2 wow fadeIn" style="padding: 5px; white-space: nowrap;">
                <table class="table table-striped">
                    <thead class="thead-dark">
                        <tr>
                            <th>Номер чека</th>
                            <th>БИН</th>
                            <th>ID сотрудника</th>
                            <th>Дата</th>
                            <th>Бонусная карта</th>
                        </tr>
                    </thead>
                    ;
                    while ($row = mysqli_fetch_assoc($result)) {
                        echo '<tr>
                            <td>'.$row["ID_Check"].'</td>
                            <td>'.$row["BIN"].'</td>
                            <td>'.$row["ID_Co_Worker"].'</td>
                            <td>'.$row["Date"].'</td>
                            <td>'.$row["ID_Bonus_Card"].'</td>
                        </tr>';
                    }
                    echo "</table></div>";
                }?>
            </div>
        </div>
    </div>
</div>
<!-- /.container-fluid -->
<!-- Подключение футер -->
<!--
=====
=== -->
<?php include "includes/footer.html"?>

```

```

<!--
=====
=== -->
</div>
<!--
=====
=== -->
<!-- End Page Content -->
<!--
=====
=== -->
</div>
<!-- /#wrapper -->
<!-- jQuery -->
<script src="../../plugins/bower_components/jquery/dist/jquery.min.js"></script>
<!-- Bootstrap Core JavaScript -->
<script src="bootstrap/dist/js/bootstrap.min.js"></script>
<!-- Menu Plugin JavaScript -->
<script src="../../plugins/bower_components/sidebar-nav/dist/sidebar-
nav.min.js"></script>
<!--slimscroll JavaScript -->
<script src="js/jquery.slimscroll.js"></script>
<!--Wave Effects -->
<script src="js/waves.js"></script>
<!-- Custom Theme JavaScript -->
<script src="js/custom.min.js"></script>

```

Продолжение приложения Б.

```

</body>

</html>

```

Листинг 5. report-day.php

```

<?php
    if ($_GET['report'] == 'day') {
        echo '<div class="container-fluid mt-5">
            <div class="card mb-2 wow fadeIn">
                <div class="card-body d-sm-flex justify-content-between
con">
                    <h4 class="mb-2 mb-sm-0 pt-1">

```

```

        <span>Введите дату</span>
    </h4>
    <form action="report-day.php?report=day"
method="POST">
        <div class="col-sm-12 d-flex justify-content-start"
style="display: flex-wrap;">
            <div class="col-sm-6" style="display: flex">
                <input type="search" style="height: 32px; margin-
right: 15px;" class="form-control" placeholder="yyyy-mm-dd" name="date">
            </div>
            <div class="col-sm-4" style="display: flex;">
                <div>
                    <button class="btn-primary btn" type="submit"
name="coworker" style="margin-left: 5px; margin-right: 5px;">Сотрудники
                </button>
                </div>
                <div>
                    <button class="btn-primary btn" type="submit"
name="receipt" style="margin-left: 5px; margin-right: 5px;">Чеки
                </button>
                </div>
            </div>
        </div>
    </form>
</div>
</div>;
$coworker = $_POST['coworker'];
$receipt = $_POST['receipt'];
$date = $_POST['date'];
if (isset($coworker)) {
    $sql = "SELECT `Full_Name`, fiscalre-
ceipt.`ID_Co_Worker`,
SUM(`Total_Of_Goods` * `Price`) AS `Total` FROM
purchase JOIN size ON purchase.`Bar_Code` =
size.`Bar_Code`
JOIN goods ON size.`Articul` = goods.`Articul`
JOIN fiscalreceipt ON fiscalreceipt.`ID_Check` = pur-
chase.`ID_Check`
JOIN coworker ON fiscalreceipt.`ID_Co_Worker` =
coworker.`ID_Co_Worker`

```

```

WHERE DATE(`Date`) = '$date'
GROUP BY fiscalreceipt.`ID_Co_Worker`
ORDER BY SUM(`Total_Of_Goods` * `Price`);
echo '<div class="card mb-2 wow fadeIn" style="padding:10px;">
    <div style="padding-bottom: 10px; padding-top: 40px;">Данные о продажах сотрудников за <span style="font-weight: bold;">'. $date. '</span></div>;
    $result = mysqli_query($connection, $sql);
echo '
    Продолжение приложения Б.

    <table class="table table-striped">
    <thead class="thead-dark">
    <tr>
    <th>Полное имя</th>
    <th>Номер сотрудника</th>
    <th>Сумма продаж</th>
    </tr>
    </thead>
    ';
while ($row = mysqli_fetch_assoc($result)) {
    echo '<tr>
        <td>'. $row["Full_Name"]. '</td>
        <td>'. $row["ID_Co_Worker"]. '</td>
        <td>'. $row["Total"]. '</td>
        </tr>';
    }
echo "</table></div>";
}
elseif (isset($receipt)) {
    $sql = "SELECT purchase.`ID_Check`, SUM(`Total_Of_Goods` * `Price`) AS `Toatal`,
        fiscalreceipt.`ID_Co_Worker`, coworker.`Full_Name`,
        fiscalreceipt.`Payment_Type`,
        SUM(`Total_Of_Goods`) AS `AmountInCheck`
        FROM purchase JOIN size ON size.`Bar_Code` = purchase.`Bar_Code`
        JOIN goods ON goods.`Articul` = size.`Articul`
        JOIN fiscalreceipt ON fiscalreceipt.`ID_Check` = purchase.`ID_Check`";
}

```



```

        JOIN coWorker ON coworker.`ID_Co_Worker` = fis-
calreceipt.`ID_Co_Worker`
        WHERE DATE(`Date`) = '$date'
        GROUP BY purchase.`ID_Check`
        ORDER BY fiscalreceipt.`ID_Check`;
        echo '<div class="card mb-2 wow fadeIn"
style="padding:10px;">
        <div style="padding-bottom: 10px; padding-top:
40px;">Данные о чеках за <span style="font-weight:
bold;">'.$date.'</span></div>;
        $result = mysqli_query($connection, $sql);
        echo '
<table class="table table-striped">
    <thead class="thead-dark">
        <tr>
            <th>Номер чека</th>
            <th>Сумма чека</th>
            <th>ID сотрудника</th>
            <th>Имя сотрудника</th>
            <th>Тип оплаты</th>
            <th>Количество товара в чеке</th>
        </tr>
    </thead>
    ';
        while ($row = mysqli_fetch_assoc($result)) {
            echo '<tr>
                <td>'.$row["ID_Check"].'</td>
                <td>'.$row["Toatal"].'</td>
                <td>'.$row["ID_Co_Worker"].'</td>
                <td>'.$row["Full_Name"].'</td>
                <td>'.$row["Payment_Type"].'</td>
                <td>'.$row["AmountInCheck"].'</td>
            </tr>';
        }
        Продолжение приложения Б.

        echo "</table></div>";
    }
    echo '</main>';
}??>

```

Листинг 6. report-month.php

```
<?php if ($_GET['report'] == 'month') {
    echo '<main class="pt-5 max-lg-5">
        <div class="container-fluid mt-5">
            <div class="card mb-2 wow fadeIn">
                <div class="card-body d-sm-flex justify-content-between
con">
                    <h4 class="mb-2 mb-sm-0 pt-1">
                        <span>Выберите дату</span>
                    </h4>
                    <form class="d-flex justify-content-center" action="re-
port-month.php?report=month" method="POST">
                        <div class="col-sm-4">
                            <select class="form-control" id="exampleFormCon-
trolSelect1" required name="month">
                                <option selected>Выберите месяц</option>
                                <option value="01">Январь</option>
                                <option value="02">февраль</option>
                                <option value="03">Март</option>
                                <option value="04">Апрель</option>
                                <option value="05">Май</option>
                                <option value="06">Июнь</option>
                                <option value="07">Июль</option>
                                <option value="08">Август</option>
                                <option value="09">Сентябрь</option>
                                <option value="10">Октябрь</option>
                                <option value="11">Ноябрь</option>
                                <option value="12">Декабрь</option>
                            </select>
                        </div>
                        <!--<input type="text" class="form-control" place-
holder="yyyy-mm-dd" name="month">-->
                        <button class="btn-primary btn my-0" type="submit"
name="coworker" style="margin-left: 5px; margin-right: 5px;">Сотрудники
                        </button>
                        <button class="btn-primary btn my-0" type="submit"
name="receipt" style="margin-left: 5px; margin-right: 5px;">Чеки
                        </button>
                    </form>
                </div>
            </div>
        </div>
    </main>
}
```

```

</div>
</div>';
$coworker = $_POST['coworker'];
$receipt = $_POST['receipt'];
$month = $_POST['month'];
if (isset($coworker)) {
    $sql = "SELECT `Full_Name`, fiscalre-
ceipt.`ID_Co_Worker`,
        SUM(`Total_Of_Goods` * `Price`) AS `Total` FROM
purchase JOIN size ON purchase.`Bar_Code` =
size.`Bar_Code`
        JOIN goods ON size.`Articul` = goods.`Articul`
        JOIN fiscalreceipt ON fiscalreceipt.`ID_Check` = pur-
chase.`ID_Check`
        JOIN coworker ON fiscalreceipt.`ID_Co_Worker` =
coworker.`ID_Co_Worker`
        WHERE MONTH(`Date`) = '$month'
        GROUP BY fiscalreceipt.`ID_Co_Worker`
        ORDER BY SUM(`Total_Of_Goods` * `Price`)";
    echo '<div class="card mb-2 wow fadeIn" style="pad-
ding: 5px; white-space: nowrap;">
        <div style="padding-bottom: 10px; padding-top:
10px;">Данные о продажах сотрудников за <span style="font-weight:
bold;">'. $date. '</span></div>';
    $result = mysqli_query($connection, $sql);
    echo '
    <table class="table table-striped">
    <thead class="thead-dark">
    <tr>
    <th>Полное имя</th>
    <th>Номер сотрудника</th>
    <th>Сумма продаж</th>
    </tr>
    </thead>
    ';
    while ($row = mysqli_fetch_assoc($result)) {
        echo '<tr>
            <td>'. $row["Full_Name"]. '</td>
            <td>'. $row["ID_Co_Worker"]. '</td>

```

```

                <td>'.$row["Total"].'</td>
            </tr>;
        }
        echo "</table></div>";
    }
    elseif (isset($receipt)) {
        $sql = "SELECT purchase.`ID_Check`, SUM(`Total_Of_Goods`*`Price`) AS `Toatal`,
            fiscalreceipt.`ID_Co_Worker`, coworker.`Full_Name`,
            fiscalreceipt.`Payment_Type`,
            SUM(`Total_Of_Goods`) AS `AmountInCheck`
            FROM purchase JOIN size ON size.`Bar_Code` = purchase.`Bar_Code`
            JOIN goods ON goods.`Articul` = size.`Articul`
            JOIN fiscalreceipt ON fiscalreceipt.`ID_Check` = purchase.`ID_Check`
            JOIN coWorker ON coworker.`ID_Co_Worker` = fiscalreceipt.`ID_Co_Worker`
            WHERE MONTH(`Date`) = '$month'
            GROUP BY purchase.`ID_Check`
            ORDER BY fiscalreceipt.`ID_Check`";
        echo '<div class="card mb-2 wow fadeIn" style="padding: 5px; white-space: nowrap;">
            <div style="padding-bottom: 10px; padding-top: 10px; padding-left: 15px;">Данные о чеках за <span style="font-weight: bold;">'.$date.'</span></div>;
        $result = mysqli_query($connection, $sql);
        echo '
        <table class="table table-striped">
            <thead class="thead-dark">
                <tr>
                    <th>Номер чека</th>
                    <th>Сумма чека</th>
                    <th>ID сотрудника</th>
                    <th>Имя сотрудника</th>
                    <th>Тип оплаты</th>
                    <th>Количество товара в чеке</th>
                </tr>
            </thead>
        ';
    }

```

Продолжение приложения Б.

```

        while ($row = mysqli_fetch_assoc($result)) {
            echo '<tr>
                <td>'.$row["ID_Check"].'</td>
                <td>'.$row["Toatal"].'</td>
                <td>'.$row["ID_Co_Worker"].'</td>
                <td>'.$row["Full_Name"].'</td>
                <td>'.$row["Payment_Type"].'</td>
                <td>'.$row["AmountInCheck"].'</td>
            </tr>';
        }
        echo "</table></div>";
    } echo '</main>';
}??>

```

Листинг 7. size.php

```

<?php
    if ($_GET['table'] == 'size') { ?>
        <div class="card mb-2 wow fadeIn" style="margin-bottom:
20px; padding-bottom: 20px;">
            <div class="card-body col-sm-12 d-flex justify-content-
between con">
                <div class="col-sm-4">
                    <a href="size.php?add=size" class="btn btn-primary
fff">Добавить номенклатуру</a>
                </div>
                <form class="">
                    <div class="col-sm-4 d-flex justify-content-end"
style="float: right; display: flex; align-items: center;">
                        <div class="col-sm-8">
                            <input type="search" style="height: 32px; mar-
gin-right: 15px;" class="form-control" placeholder="Articul" name="">
                        </div>
                        <div class="col-sm-2">
                            <button class="mr-2 btn btn-primary btn-sm
my-0" type="submit">
                                <i class="fa fa-search"></i>
                            </button>
                        </div>
                    </div>
                </div>
            </div>

```

```

        </form>
    </div>
</div>
<?php
$sql = "SELECT * FROM size";
$result = mysqli_query($connection, $sql); ?>
<div class="card mb-2 wow fadeIn" style="padding: 5px;
white-space: nowrap;">
    <table class="table table-striped table-dark" style="color:
#000">
        <thead class="thead-dark">
            <tr>
                <th>Штрих-код</th>
                <th>Размер</th>
                <th>Количество</th>
                <th>Артикул</th>
                <th>Цвет</th>
            </tr>
        </thead>
<?php
    Продолжение приложения Б.

    while ($row = mysqli_fetch_assoc($result)) {
        echo '<tr>
            <td>'.$row["Bar_Code"].'</td>
            <td>'.$row["Size"].'</td>
            <td>'.$row["Number_Of_Units"].'</td>
            <td>'.$row["Articul"].'</td>
            <td>'.$row["Colour"].'</td>
        </tr>';
    }
    echo "</table></div>";
} ?>
<?php
if ($_GET['add'] == 'size') {
    $fetchGoodsQuery = "SELECT * FROM `goods`";
    $goods = mysqli_query($connection, $fetchGoodsQuery);
?>

    <main class="pt-5 max-lg-5">
        <div class="container-fluid mt-3">
            <div class="good-add-form col-sm-8">

```

```

<h4 class="coworker-form-title text-center mb-4">Форма добавления номенклатуры</h4>
<form method="POST" id="form">
  <div class="form-group row">
    <label for="staticEmail" class="col-sm-4 col-form-label">Штрих-код</label>
    <div class="col-sm-8">
      <input type="text" name="barcode"
class="form-control" id="inputPassword" required>
    </div>
  </div>
  <div class="form-group row">
    <label for="staticEmail" class="col-sm-4 col-form-label">Размер</label>
    <div class="col-sm-8">
      <input type="text" name="size" class="form-control" id="inputPassword" required>
    </div>
  </div>
  <div class="form-group row">
    <label for="staticEmail" class="col-sm-4 col-form-label">Количество</label>
    <div class="col-sm-8">
      <input type="text" name="number_of_units"
class="form-control" id="inputPassword" required>
    </div>
  </div>
  <div class="form-group row">
    <label for="staticEmail" class="col-sm-4 col-form-label">Артикул</label>
    <div class="col-sm-8">
      <select name="articul" class="form-control"
id="exampleFormControlSelect1" required>;
      <?php foreach ($goods as $row) {
        echo '<option value="'. $row["Articul"] ."'>'
. $row["Articul"] . '</option>';
      };?>
    </select>
  </div>
</div>
<div class="form-group row">

```

```

        <label for="staticEmail" class="col-sm-4 col-
form-label">Цвет</label>
        <div class="col-sm-8">
            Продолжение приложения Б.
            <input type="text" name="colour" class="form-
control" id="inputPassword" required>
        </div>
    </div>
    <div class="d-flex justify-content-end">
        <button type="submit" class="btn btn-primary"
name="addsize">Добавить</button>
    </div>
</form>
</div>
</div>
</main>
<?php
if (isset($_POST['addsize'])) {
    $barcode = $_POST['barcode'];
    $size = $_POST['size'];
    $number_of_units = $_POST['number_of_units'];
    $articul = $_POST['articul'];
    $colour = $_POST['colour'];

    $sql6 = "SELECT * FROM goods WHERE `Articul`
= '$articul'";

    $query6 = mysqli_query($connection, $sql6);
    $result6 = mysqli_num_rows($query6);

    if (empty($articul) OR empty($barcode) OR
empty($size) OR empty($number_of_units) OR empty($colour)) {
        echo '<div class="success-al alert alert-danger"
role="alert">
            Заполните пустые поля!
        </div>';
        die();
    } else {

```



```

        $query = "INSERT INTO `size` VALUES('$barcode', '$size', '$number_of_units', '$articul', '$colour')";
        $result = mysqli_query($connection, $query);
        echo '<div class="success-al alert alert-success"
role="alert">
                Данные успешно добавлены
            </div>';
        }
    }
    echo '</div>';
}?'>

```

Листинг 8. connection.php

```

<?php
session_start();
$server = "localhost";
$username = $_SESSION['user'];
$password = $_SESSION['password'];
$db="shop";
if (empty($username) and empty($password)) {
    echo "<body text-align: center;>
        Продолжение приложения Б.

        <div>
            <h1>Не введены логин и пароль для входа в систему
            <a href='index.php'>
                <i style='width: 30px; height: 30px;'>
                    <img src='img/icon.png' style='width: 30px; height: 30px;'>
                </i>
            </a>
            </h1>
        </div>
    </body>";
    die();
}
$connection = @mysqli_connect($server, $username, $password, $db);
if($connection == false) {
    echo "<body text-align: center;>
        <div>

```

```

<h1>Для того, чтобы войти в систему введите корректные данные
  <a href='index.php'>
    <i style='width: 30px; height: 30px;'>
      <img src='img/icon.png' style='width: 30px; height: 30px;'>
    </i>
  </a>
</h1>
</div>
</body>";
die();
}
?>

```

Листинг 9. index.php

```

<!DOCTYPE html>
<html>
<?php include "includes/header.html"?>
  <style type="text/css">
    body {
      background-color: #fff;
font-family: source-sans-pro, sans-serif;
    }
    .login {
      text-align: center;
    }
    a:hover,a:focus{
      outline: none;
      text-decoration: none;
    }
    .tab{
background-color: #2ecc71;
      background: -webkit-linear-gradient(to bottom, #6f0000, #200122);
/*background: linear-gradient(to bottom, #6f0000, #200122);*/
      padding: 40px 50px;
      position: relative;
    }
    .tab .nav-tabs{
      border-bottom: none;

```

Продолжение приложения Б.

```

padding: 0 20px;
position: relative;
}
.tab .nav-tabs li{ margin: 0 30px 0 0; }
.tab .nav-tabs li a{
font-size: 18px;
color: #fff;
border-radius: 0;
text-transform: uppercase;
background: transparent;
padding: 0;
margin-right: 0;
border: none;
border-bottom: 2px solid transparent;
opacity: 0.5;
position: relative;
transition: all 0.5s ease 0s;
}
.tab .nav-tabs li a:hover{ background: transparent; }
.tab .nav-tabs li.active a,
.tab .nav-tabs li.active a:focus,
.tab .nav-tabs li.active a:hover{
border: none;
background: transparent;
opacity: 1;
border-bottom: 2px solid #eec111;
color: #fff;
}
.tab .tab-content{
padding: 20px 0 0 0;
margin-top: 40px;
background: transparent;
z-index: 1;
position: relative;
}
.form-bg{ background: #ddd; }
.form-horizontal .form-group{
margin: 0 0 15px 0;
position: relative;
}
.form-horizontal .form-control{

```

```
height: 40px;
background: rgba(255,255,255,0.2);
border: none;
border-radius: 20px;
box-shadow: none;
padding: 0 20px;
font-size: 14px;
font-weight: bold;
color: #000;
transition: all 0.3s ease 0s;
}
.form-horizontal .form-control:focus{
box-shadow: none;
outline: 0 none;
}
```

Продолжение приложения Б.

```
.form-control::placeholder {
color: #fff;
}
.form-horizontal .form-group label{
padding: 0 20px;
color: #fff;
text-transform: capitalize;
margin-bottom: 10px;
}
.form-horizontal .main-checkbox{
width: 20px;
height: 20px;
background: #eec111;
float: left;
margin: 5px 0 0 20px;
border: 1px solid #eec111;
position: relative;
}
.form-horizontal .main-checkbox label{
width: 20px;
height: 20px;
position: absolute;
top: 0;
left: 0;
```

```

    cursor: pointer;
}
.form-horizontal .main-checkbox label:after{
    content: "";
    width: 10px;
    height: 5px;
    position: absolute;
    top: 5px;
    left: 4px;
    border: 3px solid #fff;
    border-top: none;
    border-right: none;
    background: transparent;
    opacity: 0;
    -webkit-transform: rotate(-45deg);
    transform: rotate(-45deg);
}
.form-horizontal .main-checkbox input[type=checkbox]{ visibility: hidden; }
.form-horizontal .main-checkbox input[type=checkbox]:checked + label:af-
ter{ opacity: 1; }
.form-horizontal .text{
    float: left;
    font-size: 14px;
    font-weight: bold;
    color: #fff;
    margin-left: 7px;
    line-height: 20px;
    padding-top: 5px;
    text-transform: capitalize;
}
.form-horizontal .btn{
    width: 100%;
    background: #fff;

```

Продолжение приложения Б.

```

padding: 10px 20px;
border: none;
font-size: 14px;
font-weight: bold;
color: #2ecc71;
border-radius: 20px;

```

```

text-transform: uppercase;
margin: 20px 0 30px 0;
}
.form-horizontal .btn:hover {
  background-color: #27ae60;
  color:#ffffff;
}
.form-horizontal .forgot-pass{
border-top: 1px solid #615f6c;
margin: 0;
text-align: center;
}
.form-horizontal .forgot-pass .btn{
width: auto;
background: transparent;
margin: 30px 0 0 0;
color: #615f6c;
text-transform: capitalize;
transition: all 0.3s ease 0s;
}
.form-horizontal .forgot-pass .btn:hover{ color: #eec111; }
@media only screen and (max-width: 479px){
.tab{ padding: 40px 20px; }
}
.border {
  border-radius: 10px;
  background-color: #615f6c;
  padding: 5px;
  text-align: center;
  transition: all 0.3s ease 0s;
}
.border:hover {
  background-color: #eec111;
}
</style>
<body>
<div class="container">
  <div class="row" style="margin-top: 100px;">
    <div class="col-sm-4 col-md-offset-4">
      <div class="tab" role="tabpanel" style="border-radius: 15px;">

```

```

        <ul class="nav nav-tabs" role="tablist" style="padding-left: 50px;
padding-right: 0;">
            <li role="presentation" class="active"><a href="#Section1" aria-
controls="home" role="tab" data-toggle="tab">Вход в систему</a></li>
        </ul>
        <div class="tab-content tabs" style="margin-top: 20px;">
            <div role="tabpanel" class="tab-pane fade in active" id="Sec-
tion1">
                <form action="check.php" method="POST" class="form-hori-
zontal">
                    <div class="form-group">
                        <input type="text" name="login" class="form-control"
placeholder="Логин" id="exampleInputEmail1">
                            Продолжение приложения Б.
                    </div>
                    <div class="form-group">
                        <input type="password" name="password" class="form-con-
trol" placeholder="Пароль" id="exampleInputPassword1">
                    </div>
                    <div class="form-group">
                        <button type="submit" name="submit" class="btn btn-de-
fault" style="outline: none;">ВОЙТИ</button>
                    </div>
                </form>
            </div>
        </div>
    </div>
</div>
</div>
</div>
</div>
<!-- /#wrapper -->
<!-- jQuery -->
<script src="../../plugins/bower_components/jquery/dist/jquery.min.js"></script>
<!-- Bootstrap Core JavaScript -->
<script src="bootstrap/dist/js/bootstrap.min.js"></script>
<!-- Menu Plugin JavaScript -->
<script src="../../plugins/bower_components/sidebar-nav/dist/sidebar-
nav.min.js"></script>
<!-- slimscroll JavaScript -->
<script src="js/jquery.slimscroll.js"></script>

```

```
<!--Wave Effects -->
<script src="js/waves.js"></script>
<!-- Custom Theme JavaScript -->
<script src="js/custom.min.js"></script>
</body>
</html>
```

Листинг 10. check.php

```
<?php
    session_start();
    $_SESSION['user'] = $_POST['login'];
    $_SESSION['password'] = $_POST['password'];
    header("Location: coworker.php?table=coworker");
?>
```

Листинг 11. logout.php

```
<?php
    session_start();
    session_destroy();
    header("Location: index.php");
?>
```

Приложение В Акт внедрения

