

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН
Некоммерческое акционерное общество
«АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ
им. ГУМАРБЕКА ДАУКЕЕВА»
Кафедра IT - инжиниринг

«ДОПУЩЕНА К ЗАЩИТЕ»
Зав. Кафедрой, PhD, доцент Досжанова А.А
_____ «__»

_____ 2020

ДИПЛОМНЫЙ ПРОЕКТ

На тему: Разработка программного обеспечения автоматизации поддержки процессов закупа и учета хранения товаров на складе
Специальность 5В070400 – «Вычислительная техника и программное обеспечение»

Выполнила: Асильбекова А.К. Группа: ВТ-16-2
Научный руководитель: д.т.н., профессор Казиев Г.З

Консультанты:

по экономической части: к.э.н., ассоц.профессор Габелашвили К.Р.
_____ «__» _____ 2020 г.

по безопасности жизнедеятельности: доцент Приходько Н.Г.
_____ «__» _____ 2020 г.

по программному обеспечению: ст.преп. Майкотов М.Н
_____ «__» _____ 2020 г.

Нормоконтролер: ст.преп. Абсатарова Б.Р.
_____ «__» _____ 2020 г.

Рецензент:

(учёная степень, звание, Ф.И.О.)

_____ «__» _____ 2020
г.

Алматы 2020

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

Институт систем управления и информационных технологий

Специальность 5В070400 – «Вычислительная техника и программное обеспечение»

Кафедра IT-инжиниринг

ЗАДАНИЕ

на выполнение дипломного проекта

Студенту Асильбековой Амине Хажмухановне

Тема проекта: Разработка программного обеспечения автоматизации поддержки процессов закупа и учета хранения товаров на складе

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): MySQL- среда разработки БД, PHP – среда программирования, Atom – интегрированная среда разработки.

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

- а) изучение технологии складского учета;
- б) технологии разработки приложения;
- в) функционал программного продукта;
- г) экономическая эффективность проекта;
- д) вопросы безопасности жизнедеятельности и охраны труда.

Перечень графического материала (с точным указанием обязательных чертежей): представлены 18 таблиц, 34 иллюстрации.

Основная рекомендуемая литература:

1 Святослав Куликов. “Работа с MySQL, MS SQL Server и Oracle в примерах” /Самиздат, 2017, 547 с.

2 Волгин В. "Склад. Стандарты управления: Практическое пособие. ".

3 “PHP&MySQL Novice to Ninja”, Tom Butler, Kevin Yank, 2014.

4 "PHP: A beginner’s guide ", Vikram Vaswani.

Консультация по проекту с указанием относящихся к ним разделов проекта

Раздел	Консультант	Сроки	Подпись
Экономическая часть	Габелашвили К.Р.	19.04.2020	
Безопасности жизнедеятельности	Приходько Н.Г.	22.04.2020	
Программная часть	Майкотов М.Н.	15.05.2020	
Нормоконтролер	Абсатарова Б.Р.	13.05.2020	

ГРАФИК

подготовки дипломного проекта

Наименование разделов, перечень разрабатываемых вопросов	Сроки представления научному руководителю	Примечания
Анализ предметной области	13.01.2020 – 20.01.2020	
Разработка и проектирование базы данных	21.01.2020 – 15.02.2020	
Создание приложения на PHP и интеграция с БД	16.02.2020 – 30.04.2020	
Экономическое обоснование	19.04.2020	
Безопасность жизнедеятельности	22.04.2020	

Дата выдачи задания « ____ » _____ 2020г.

Заведующий кафедрой _____ А.А.Досжанова

Научный руководитель проекта _____ Г.З.Казиев

Задание принял к исполнению студент _____ А.К.Асильбекова

Аңдатпа

Дипломдық жобаның тақырыбы: «Қоймада тауарларды сақтаудың сатып алу және есепке алу процестерін қолдаудың «Книга-НВ» ЖШС-ке арналған автоматтандырылуын бағдарламалық қамтамасыз етуін құру».

Бұл дипломдық жоба шағын кәсіпорынның қоймасындағы тауарлардың қозғалысын есепке алу жүйесін әзірлеуді жүзеге асырды, оның мақсаты тауарларды сатып алу және сақтауды есепке алу сияқты процестерді қолдаудың автоматтандырылған кешенін құру болып табылады.

Дипломдық жоба РНР негізінде өтінім түрінде орындалды. Дайындалған дипломдық жобаның мақсатына жету үшін MySQL дерекқоры және серверлік деректерді біріктіру сияқты технологиялар қолданылды.

Бағдарламаны әзірлеуге жұмсалған шығындар мен шығындардың экономикалық есебі, әзірленген жобаның экономикалық мақсаттылығы бағаланды.

Аннотация

Тема дипломного проекта: «Разработка программного обеспечения автоматизации поддержки процессов закупа и учета хранения товаров на складе» для ТОО «Книга-НВ».

В данном дипломном проекте реализована разработка системы учета движения товаров на складе малого предприятия, цель которого состоит реализации автоматизированного комплекса поддержки таких процессов как закуп и учет хранения товаров.

Дипломный проект был выполнен в формате приложения на базе PHP. Для достижения цели разрабатываемого дипломного проекта были применены такие технологии, как база данных MySQL и серверное объединение данных.

Также был проведен экономический расчет затрат и стоимости разработки программы, оценка экономической целесообразности разрабатываемого проекта.

Annotation

Theme of the graduation project: "Development of software for automation of purchase processes support and for accounting the storage of goods in the warehouse" for LLC "Kniga-NV".

This graduation project has implemented the development of a system for recording the movement of goods in a small enterprise's warehouse, the purpose of which is to implement an automated complex to support such processes as the purchase and accounting of storage of goods.

The graduation project was executed in the format of an application based on PHP. To achieve the goal of the developed graduation project, technologies such as the MySQL database and server data integration were applied.

An economic calculation of the costs and costs of developing the program, an assessment of the economic feasibility of the developed project was also carried out.

Содержание

Введение	11
1 Аналитическая часть	12
1.1 Задачи, решаемые складом	12
1.2 Понятие БД и СУБД	16
1.3 Кроссплатформенное ПО ХАМРР	22
2 Практическая часть	23
2.1 Разработка технического задания	23
2.2 Средства разработки кода приложения	25
3 Экспериментальная часть	34
3.1 Интерфейс приложения	34
3.2 Разработка функции закупа в РНР	41
4 Экономическая часть	44
4.1 Экономическое обоснование проекта	44
4.2 Расчет трудоёмкости разработки	44
4.3 Расчет затрат на разработку ПП	45
4.4 Определение возможной цены приложения	52
4.5 Сравнительный анализ эксплуатационных затрат	52
4.6 Вывод по экономической части	57
5 Безопасность жизнедеятельности	58
5.1 Анализ условий труда	58
5.2 Расчет естественного освещения	59
5.3 Расчет искусственного освещения	61
5.4 Расчет вентиляции	65
5.5 Вывод по разделу	68
Заключение	69
Список литературы	70
Приложение А. Техническое задание	71
Приложение Б. Листинг программы	72
Приложение В Акт внедрения	84

Введение

Любая автоматизация преследует определенные цели: минимизировать или полностью устранить человеческое взаимодействие в процессе. Производство товаров, поставка многих услуг – почти каждая отрасль претерпела изменения из-за автоматизации.

В наши дни всё большее количество субъектов бизнеса стремятся к использованию информационных технологий с целью автоматизации рабочих процессов.

Компании и фирмы, связывающие свою деятельность с производством и торговлей, имеют дело с временным хранением товара или продукции до момента реализации. Склад является местом хранения всех запасов компании. Автоматизация управления складом становится всё более актуальным вопросом, ввиду важности вопроса управления запасами на предприятии.

Автоматизация управления складом позволяет значительно снизить количество промахов и ошибок, которые делаются в процессе работы сотрудниками предприятия.

Учет склада – это всегда работа с большим объемом данных. Автоматизация же учета позволяет экономить время, деньги и человеческий ресурс предприятия. Выполнение программного обеспечения для управления складом в реальном времени может заметно минимизировать объем бумажной работы, которая обычно связана с функциями склада, а также обеспечивает своевременный и безупречный поток запасов и информации данных. Торговля является ключевой отраслью любой хозяйственной деятельности. Актуальность ведения складского учёта в данном программном обеспечении, вне всякого сомнения, велика, ведь компании и предприятия стремятся идти в ногу со временем и автоматизировать многие трудоёмкие процессы, без которых любое предприятие способно потерять значительную долю прибыли и свою эффективность. А главной задачей многих компаний является стабильность и главное рентабельность. Целью дипломной работы является изучение движения товаров на складе с последующей реализацией комплекса учета движения товаров: учета хранения, закупа и документации.

Одной из целей складского учета является сложение, поскольку оно связано с оценкой запасов. Оценка инвентарного счета несомненно важна, вследствие того, что вложенные корпорацией средства в свои товарные запасы обычно составляют крупную и значительную сумму. Инвентаризация бизнеса нередко является крупнейшим из его оборотных активов. Второй и по крайней мере, одинаково важной задачей можно считать правильное определение чистого дохода бизнеса за период, сопоставление соответствующей стоимости (проданный инвентарь) с соответствующим доходом (получение дохода от продажи инвентаря). Другими словами, процесс сопоставления требует, чтобы стоимость была назначена на те товары, которые были проданы в течение определенного времени, и те

товары, которые были еще в наличии и доступны для продажи в конце периода [2].

1 Аналитическая часть

В данном разделе рассматриваются задачи, которые будет выполнять разрабатываемая информационная система, изучаются аналоги и прототипы, выявляются их достоинства и недостатки.

1.1 Задачи, решаемые складом

Склад принимает партии товаров от поставщиков и отпускает его клиентам мелкими партиями. Поступление продукции идет непосредственно с заводов-изготовителей и путем реализации попадает к клиентам, имеющим необходимость в данном виде товаров. Передвижение продукции с производства на базу и складской учет, хранение и отгрузку клиентам осуществляется предприятием [11].

Основными задачами и функциями учета управления складом являются:

- прием всех товаров, участвующих в производственной деятельности компании, владеющей складом;
- контроль и инвентаризация хранимых продуктов;
- правильное хранение товара;
- подготовка заказов для отправки на региональные склады, клиентам или обоим;
- быстрая отправка заказов;
- оформление отчетных материалов с требуемой для этого аналитикой.

Целью автоматизации склада является повышение его эффективности. Ниже приведены некоторые из преимуществ, которые мы можем получить:

- бесперебойные операции, поскольку они не ограничены сменой персонала и возможностями;
- снижение затрат на рабочую силу, поскольку нам не нужно столько сотрудников для выполнения задач склада;
- меньше аварий и проблем со здоровьем, а также более безопасные условия труда для сотрудников склада и водителей;
- точное знание уровня запасов, избегая нехватки, затоваривания и устаревания;
- оптимизация складских площадей за счет применения высоких вертикальных систем;
- снижение затрат на техническое обслуживание за счет уменьшения событий, которые приводят к повреждению инфраструктуры или товаров;
- повышенная безопасность и меньшее загрязнение грузов, так как они больше не обрабатываются людьми;
- устранение человеческой ошибки в документации, которая может привести к задержкам при обработке товаров;
- улучшение сроков доставки и точности;
- более высокая специализация и обучение труда, что делает его более профессиональным, эффективным и продуктивным.

Отдельная классификация и оценка товарных запасов необходимы из-за их значимости в качестве основного ресурса для многих фирм, а также потому, что основа оценки запасов напрямую влияет на отчетный доход и представление потоков денежных средств. Оценка существующих и предлагаемых процедур измерения должна учитывать основную природу запасов в связи с деятельностью фирмы, а также цели и основные концепции складского учета. Одна из целей теории складского учета должна заключаться в предоставлении полезных руководств при поиске процедур оценки, которые обеспечат более точные измерения ресурсов инвентаризации и более точную информацию о потенциальных денежных потоках фирмы.

В ходе осуществления поставок на предприятие производится обработка и хранение большого объема информации, связанной с поставками, которая в себя включает:

- выявление расхождения фактического наличия и количества, указанного в сопроводительной документации;
- контроль соблюдения нормативов запаса всех товаров.

Для идентификации предметов, подлежащих транспортировке, сбору или выбрасыванию, материалы или товары принимаются с использованием транспортных средств в соответствии с планом производства, заказом клиента, заказом на работу и заказом на отправку. Товары транспортируются с места производства на склад или в указанное место. Происходит сортировка товаров предметы по размерам, формам, формам, типам и т. д., с целью определить количество и поместить их на полки или положить в коробки. Чтобы классифицировать товары в соответствии со стандартами компании, товар взвешивается и проводится инвентаризация количества и количество запасов записывается на текущую дату.

1.1.1 Методы управления запасами

Термин «запасы» включает в себя товары, предназначенные для продажи в ходе обычной деятельности, а также материалы и материалы, используемые в процессе производства для продажи. Из этой категории исключаются расходные материалы, которые будут потребляться в непроизводственных операциях, ценные бумаги, предназначенные для перепродажи, но связанные с деятельностью фирмы, а также используемые установки или оборудование, или ожидающие окончательного удаления после прекращения использования. В традиционном определении запасы являются текущими активами, потому что они, как правило, будут конвертированы в денежные средства или другие активы в рамках операционного цикла бизнеса [11]. Устаревшие и непригодные для продажи товары, однако, если материал в количестве должен быть исключен из этой классификации, если только он не может быть ликвидирован на доступных рынках в течение нормальных периодов продажи. Запасы обычно считаются запасами товаров, хотя учет поток товаров обычно считается более важным. Управление запасами, также

известное в менеджменте как «Управление цепочками поставок» или «Управление закупками», относится к контролю материальных потоков от поставщиков сырья до доставки готовой продукции покупателям. Поскольку запасы и капитал положительно связаны, важно понимать и применять различные методы управления запасами. Например, чем выше запас, тем выше капитальные затраты. Контроль затрат путем управления запасами приобретает первостепенное значение в любой производственной организации. Большинство методов управления запасами основаны на научных принципах и предполагают предварительное знание математических и вероятностных теорий. Управление запасами учитывает другие функции, такие как закупка, производство и маркетинг. Его методы направлены на уравнивание противоречивых целей. В силу того, что они должны основываться на научных теориях, эти методы лучше объясняются с помощью моделей, которые приобретают характеристики инструментов прогнозирования. Эти модели состоят из методов, которые прогнозируют спрос, заказы на покупку и постоянный мониторинг точек повторного заказа, чтобы инициировать предлагаемые заказы при достижении триггерных точек. Три из наиболее важных методов определены ниже.

Существуют различные типы инвентаря. Сырье - предметы, поставляемые Поставщиком, к которым не добавлялись трудовые ресурсы фирмой, получающей товары. Готовая продукция, которая является готовой продукцией, которая все еще находится во владении фирмы, которая их произвела. Незавершенное производство (WIP) - это позиции, которые были частично обработаны, но все еще не завершены. Эти три типа запасов должны контролироваться таким образом, чтобы минимизировать стоимость запасов при сохранении бесперебойного производственного потока.

1.1.2 Действующие аналоги программы «Складского учета»

В сегодняшнее время наблюдается тенденция пренебрежения системой складского учета многими крупными и мелкими компаниями по причине отсутствия необходимого программного обеспечения. На данный момент существуют аналоги данной программы.

«SLS – Склад Лайт» - простая программа для учета товаров на складе, однопользовательский вариант предоставляется бесплатно.

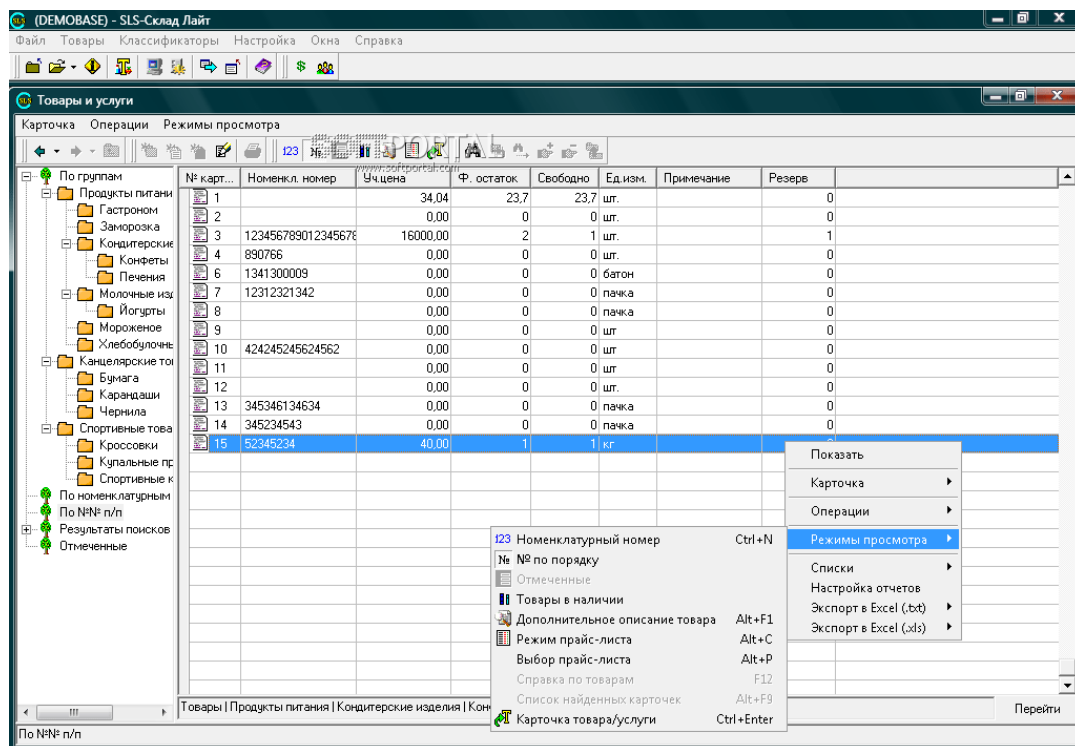


Рисунок 1.1.1 – «SLS – Склад Лайт»

Обладает ограниченным функционалом и не совсем удобным с точки зрения пользователя интерфейсом. Существуют также коммерческие многопользовательские версии, которые можно приобрести за определенную плату.

«Склад и Торговля» - очередная программа, направленная на учет товаров на складе. Обладает расширенным функционалом в сравнении с вышеописанной программой. Недостатком данного продукта можно указать чрезмерно высокую цену. Сама программа стоит 44990 тенге, но необходимо приобретать лицензию для каждого дополнительного компьютера за определенную плату, что не совсем выгодно в экономическом плане.

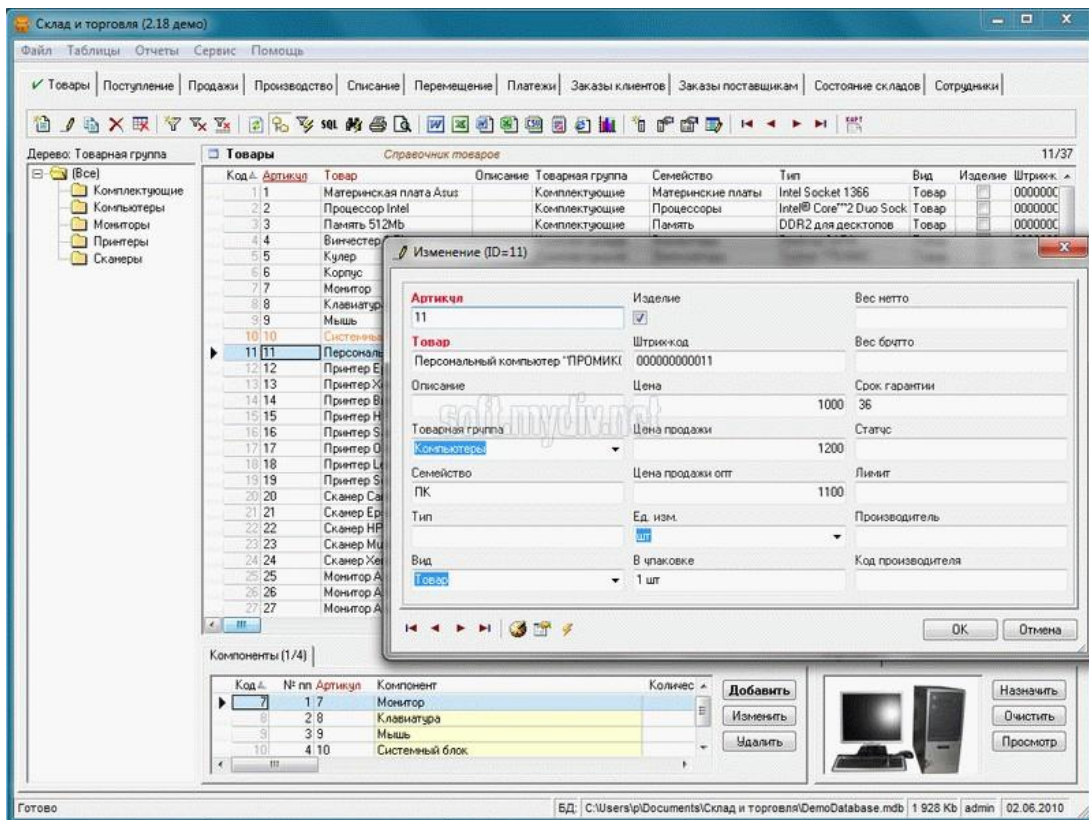


Рисунок 1.1.2 – «Склад и Торговля»

1.2. Понятие БД и СУБД

Складской учет и его основные идеи базируются на концепции баз данных (БД), так как включает в себя данные о поставщиках, товарах и операциях, связанных с упомянутыми данными.

База данных (БД) – это поименованная совокупность структурированных данных, относящихся к определенной предметной области.

База данных является современной формой организации, хранения и доступа к информации. База данных должна быть хранилищем данных, необходимых для обработки данных организации. Эти данные должны быть точными, конфиденциальными и защищенными от повреждений. Он должен быть точным, чтобы данные могли использовать различные приложения с разными требованиями к данным. Разные прикладные программисты и разные конечные пользователи имеют разные взгляды на данные, которые должны быть получены из общей структуры данных. Их методы поиска и доступа к данным будут разными:

- база данных в значительной степени минимизирует избыточность данных;
- база данных может в значительной степени контролировать несогласованность данных;
- обмен данными также возможен с использованием базы данных;

- база данных обеспечивает соблюдение стандартов;
- использование баз данных может обеспечить безопасность данных;
- целостность может управляться с помощью базы данных [7].

Поскольку базу данных можно просматривать через три уровня абстракции, любое изменение на любом уровне может повлиять на схемы других уровней. Поскольку база данных продолжает расти, иногда могут происходить частые изменения. Это не должно приводить к перепроектированию и повторной реализации базы данных. Концепции независимости данных оказываются полезными в таких типах контекстов.

Система управления базами данных (СУБД) - это программный пакет, предназначенный для определения, обработки, извлечения и управления данными в базе данных. СУБД обычно манипулирует самими данными, форматом данных, именами полей, структурой записей и структурой файлов. Он также определяет правила для проверки и манипулирования этими данными [9].

СУБД освобождает пользователей от программ кадрирования для обслуживания данных. Языки запросов четвертого поколения, такие как SQL, используются вместе с пакетом СУБД для взаимодействия с базой данных. Система управления базами данных (СУБД) - это программный пакет, предназначенный для определения, обработки, извлечения и управления данными в базе данных. СУБД обычно манипулирует самими данными, форматом данных, именами полей, структурой записей и структурой файлов. Он также определяет правила для проверки и манипулирования этими данными.

СУБД освобождает пользователей от программ кадрирования для обслуживания данных. Языки запросов четвертого поколения, такие как SQL, используются вместе с пакетом СУБД для взаимодействия с базой данных.

1.2.1 Классификация баз данных и особенности СУБД

Базы данных и СУБД можно распределить по категориям в соответствии с поддерживаемыми ими моделями баз данных (например, реляционными или XML), типами компьютеров, на которых они работают (от кластера серверов до мобильного телефона), языком запросов (s) используется для доступа к базе данных (например, SQL или XQuery) и их внутреннего проектирования, что влияет на производительность, масштабируемость, устойчивость и безопасность.

Несколько критериев обычно используются для классификации СУБД. Первая - это модель данных, на которой основана СУБД. Основной моделью данных, используемой во многих современных коммерческих СУБД, является реляционная модель данных. Модель объектных данных была реализована в некоторых коммерческих системах, но не получила широкого применения. Многие устаревшие приложения по-прежнему работают в системах баз

данных на основе иерархических и сетевых моделей данных. Примерами иерархических СУБД являются IMS (IBM) и некоторые другие системы, такие как System 2K (SAS Inc.) и TDMS. IMS по-прежнему используется на государственных и промышленных объектах, включая больницы и банки, хотя многие из ее пользователей перешли на реляционные системы. Модель сетевых данных использовалась многими поставщиками и полученными продуктами, такими как IDMS (Cullinet - теперь Computer Associates), DMS 1100 (Univac - теперь Unisys), IMAGE (Hewlett-Packard), VAX-СУБД (Digital - затем Compaq и теперь HP), и SUPRA (Cincom) по-прежнему имеют следующие, и их группы пользователей имеют свои собственные активные организации. Если мы добавим к ним популярную файловую систему VSAM от IBM, мы можем легко сказать, что разумный процент компьютеризированных во всем мире данных все еще находится в этих так называемых устаревших системах баз данных. Реляционные СУБД постоянно развиваются и, в частности, включают в себя многие концепции, разработанные в объектных базах данных. Это привело к появлению нового класса СУБД, названного объектно-реляционными СУБД. Мы можем классифицировать СУБД на основе модели данных: реляционные, объектные, объектно-реляционные, иерархические, сетевые и другие.

В последнее время некоторые экспериментальные СУБД основаны на модели XML (eXtended Markup Language), которая представляет собой древовидную (иерархическую) модель данных. Они называются нативными XML-СУБД. Несколько коммерческих реляционных СУБД добавили XML-интерфейсы и хранилище в свои продукты.

Вторым критерием, используемым для классификации СУБД, является количество пользователей, поддерживаемых системой. Однопользовательские системы поддерживают только одного пользователя одновременно и в основном используются с ПК. Многопользовательские системы, которые включают большинство СУБД, поддерживают одновременное использование нескольких пользователей.

Третий критерий - количество сайтов, по которым распространяется база данных. СУБД централизована, если данные хранятся на одном компьютере. Централизованная СУБД может поддерживать несколько пользователей, но СУБД и база данных полностью находятся на одном компьютере. Распределенная СУБД (DDBMS) может иметь действующую базу данных и программное обеспечение СУБД, распределенное по многим объектам, соединенным компьютерной сетью. Однородные DDBMS используют одно и то же программное обеспечение СУБД на всех площадках, тогда как гетерогенные DDBMS могут использовать различное программное обеспечение СУБД на каждом сайте. Также возможно разработать программное обеспечение промежуточного программного обеспечения для доступа к нескольким автономным ранее существующим базам данных, хранящимся в разнородных СУБД. Это приводит к созданию интегрированной СУБД (или системы с несколькими базами данных), в

которой участвующие СУБД слабо связаны и имеют определенную степень локальной автономии. Встроенная система баз данных - это СУБД, тесно интегрированная с прикладным программным обеспечением, которое требует доступа к хранимым данным таким образом, что СУБД скрыта от конечных пользователей приложения и практически не требует постоянного обслуживания. Схема классификации баз данных приведена на рисунке 1.2.1.



Рисунок 1.2.1 – Классификация баз данных

Четвертый критерий - стоимость. Сложно предложить классификацию СУБД по стоимости. Сегодня у нас есть бесплатные СУБД с открытым исходным кодом, такие как MySQL и PostgreSQL, которые поддерживаются сторонними поставщиками дополнительных услуг. Гигантские системы продаются в модульной форме с компонентами для управления распределением, репликацией, параллельной обработкой, мобильными возможностями, а также с большим количеством параметров, которые должны быть определены для конфигурации. Кроме того, они продаются в форме лицензий - лицензии на сайты позволяют неограниченно использовать систему баз данных с любым количеством копий, запущенных на сайте клиента. Другой тип лицензии ограничивает количество одновременно работающих пользователей или количество пользовательских мест в местоположении. Автономные однопользовательские версии некоторых систем, таких как Microsoft Access, продаются по одной копии или включаются в общую конфигурацию настольного компьютера или ноутбука. Кроме того, функции хранилища данных и интеллектуального анализа данных, а также поддержка дополнительных типов данных предоставляются за дополнительную плату.

Можно классифицировать СУБД на основе типов параметров пути доступа для хранения файлов. Одно известное семейство СУБД основано на

инвертированных файловых структурах. Наконец, СУБД может быть общего или специального назначения. Когда производительность является основным фактором, СУБД специального назначения может быть спроектирована и построена для конкретного приложения; Такая система не может быть использована для других приложений без серьезных изменений. Многие системы бронирования авиабилетов и телефонных справочников, разработанные в прошлом, являются СУБД специального назначения. Они попадают в категорию систем онлайн-обработки транзакций (OLTP), которые должны поддерживать большое количество одновременных транзакций без чрезмерных задержек.

Модель данных объекта определяет базу данных с точки зрения объектов, их свойств и операций. Объекты с одинаковой структурой и поведением принадлежат классу, а классы организованы в иерархии (или ациклические графы). Операции каждого класса определяются в терминах предварительно определенных процедур, называемых методами. Реляционные СУБД расширяют свои модели для включения концепций объектных баз данных и других возможностей; эти системы называются объектно-реляционными или расширенными реляционными системами. Модель XML появилась в качестве стандарта для обмена данными через Интернет и использовалась в качестве основы для реализации нескольких прототипов собственных систем XML. XML использует иерархические древовидные структуры. Он объединяет понятия базы данных с понятиями из моделей представления документов. Данные представлены в виде элементов; с помощью тегов данные могут быть вложены для создания сложных иерархических структур. Эта модель концептуально напоминает объектную модель, но использует другую терминологию. Возможности XML были добавлены во многие коммерческие продукты СУБД.

1.2.2 Типы отношений

Существует три типа отношений:

- один к одному;
- один ко многим;
- многие ко многим;

Один-к-одному: обе таблицы могут иметь только одну запись по обе стороны отношения. Каждое значение первичного ключа относится только к одной (или нет) записи в связанной таблице. Они похожи на супругов - вы можете или не можете быть в браке, но если вы, то и у вас, и у вашего супруга есть только один супруг. Большинство отношений один-к-одному навязываются бизнес-правилами и не вытекают естественным образом из данных. При отсутствии такого правила вы обычно можете объединить обе таблицы в одну таблицу без нарушения каких-либо правил нормализации.

Один-ко-многим: таблица первичного ключа содержит только одну запись, которая не связана ни с одной, с одной или несколькими записями в

связанной таблице. Эти отношения похожи на отношения между вами и родителем. У вас есть только одна мать, но у вашей матери может быть несколько детей.

Многие ко многим: каждая запись в обеих таблицах может относиться к любому количеству записей (или вообще без записей) в другой таблице. Например, если у вас есть несколько братьев и сестер, то и ваши братья и сестры (имеют много братьев и сестер). Отношения «многие ко многим» требуют третьей таблицы, известной как таблица связывания или связывания, потому что реляционные системы не могут напрямую приспособлять отношения.

Система баз данных опирается на соответствующие значения, найденные в обеих таблицах, для формирования отношений. Когда совпадение найдено, система извлекает данные из обеих таблиц для создания виртуальной записи. Например, вы можете увидеть все книги, написанные конкретным автором. В этом случае система будет сопоставлять значения между таблицами Книг и Авторов. Важно помнить, что большую часть времени результирующая запись является динамической, что означает, что любое изменение, внесенное в виртуальную запись, обычно возвращается обратно к базовой таблице. Эти совпадающие значения являются значениями первичного и внешнего ключей. (Реляционная модель не требует, чтобы отношения основывались на первичном ключе. Вы можете использовать любой ключ-кандидат в таблице, но использование первичного ключа является принятым стандартом.) Вы узнали о первичных ключах в Части 2 - первичном Ключ однозначно идентифицирует каждую запись в таблице. Проще говоря, внешний ключ - это первичный ключ одной таблицы в другой таблице. Схема типов отношений и наглядного способа связывания данных приводится в рисунке 1.2.2.

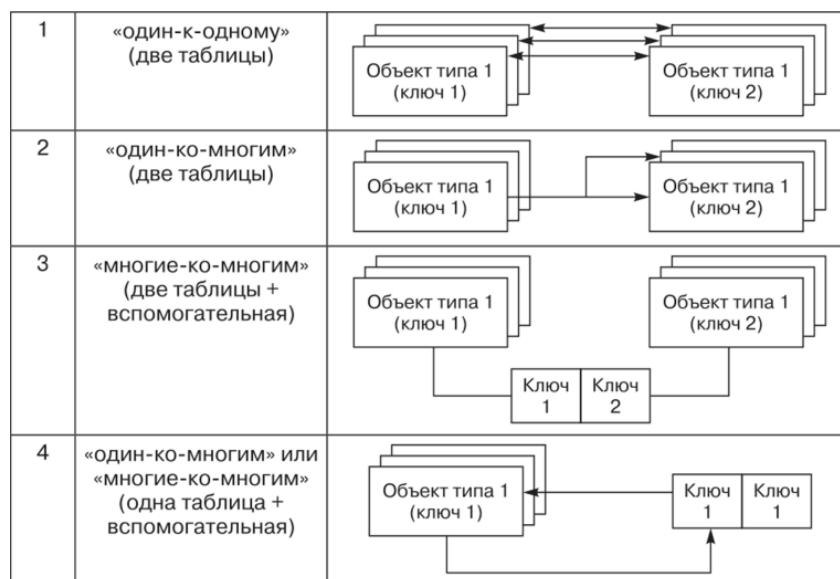


Рисунок 1.2.2 – Типы отношений

Единственное соображение заключается в том, что поле внешнего ключа должно иметь тот же тип данных, что и первичный ключ. Некоторые системы допускают одно исключение из этого правила и разрешают связь между числом и полем нумерации (например, AutoNumber в Access of Identity в SQL Server). Кроме того, значения внешнего ключа могут быть нулевыми, хотя рекомендуется не оставлять внешний ключ пустым без особых на то оснований.

1.3 Кроссплатформенное ПО ХАМРР

ХАМРР - кроссплатформенная сборка веб-сервера, содержащая Apache, MySQL, интерпретатор скриптов PHP, язык программирования Perl и большое количество дополнительных библиотек, позволяющих запустить полноценный веб-сервер. ХАМРР это акроним от «Х-любая ОС, Apache Сервер, MySQL система управления базами данных, PHP язык программирования, Perl язык программирования».

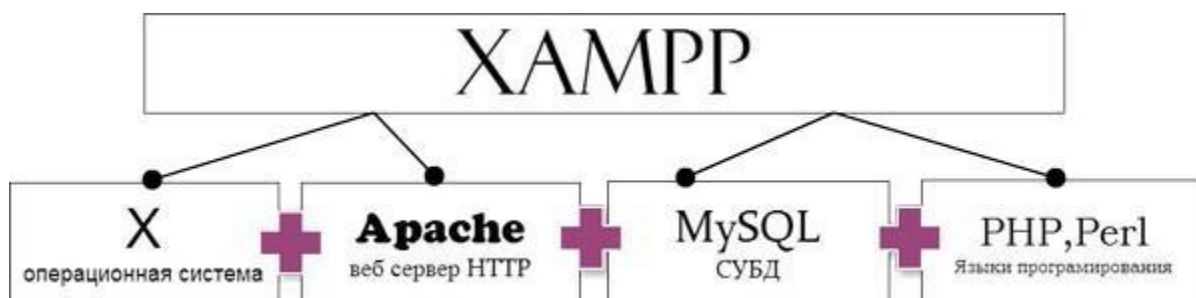


Рисунок 1.3.1 – Кроссплатформа ХАМРР

Полный пакет содержит:

- Web-сервер Apache с поддержкой SSL;
- СУБД MySQL;
- PHP;
- Perl;
- FTP-сервер FileZilla;
- POP3/SMTP сервер;
- утилиту phpMyAdmin.

ХАМРР работает со всеми 32-х разрядными ОС Microsoft (98/2000/XP/2003/Vista/7), а также с Linux, Mac OS X и Solaris. Программа свободно распространяется согласно лицензии GNU General Public License и является бесплатным, удобным в работе web-сервером, способным обслуживать динамические страницы. Количество скачанных пакетов ХАМРР в октябре 2008 г. — 775064 загрузок (33280 Гб). На сегодняшний день хамрр считается одной из лучших сборок веб-сервера, с помощью которой пользователь сможет быстро развернуть на своем устройстве полноценный и быстрый веб-сервер.

2 Практическая часть

2.1 Разработка технического задания

В качестве темы дипломного проекта была выбрана разработка программного обеспечения для автоматизации поддержки процессов учета хранения и закупа товаров на складе. Цель: разработка ИС для учета движения товаров на складе предприятия.

Исходя из цели, были поставлены следующие задачи:

- анализ предметной области;
- построение логической и физической моделей БД;
- реализация спроектированной БД на веб-странице;
- проектирование и разработка приложения для складского учёта.

Склад принимает партии товаров от поставщиков и отпускает его клиентам мелкими партиями в соответствии с документацией. Продукция поступает непосредственно с заводов-изготовителей и путем реализации попадает к клиентам, имеющим нужду в данном виде товаров. Предприятие осуществляет передвижение продукции с производства на базу, ее складской учет, хранение и отгрузку клиентам.

2.1.1 Построение структуры проекта

После корректного подбора выбора сущностей, задания атрибутов и подробного анализа связей между сущностями, можно переходить к проектированию концептуальной схемы базы данных, представленной в виде ER-диаграммы, сущности в которой принято обозначать прямоугольниками, а связи – ромбами.

Данная модель отражает ведение учета данных и наряду с этим указывает на особенности представления данных в разработанной системе (рисунок 2.1).

Процедуры нормализации применяются в целях уменьшения нежелательных характеристик БД к схемам отношений. Выделяют пять нормализованных форм (НФ), но практически достаточно, чтобы отношения удовлетворяли условиям 1НФ, 2НФ, 3НФ.

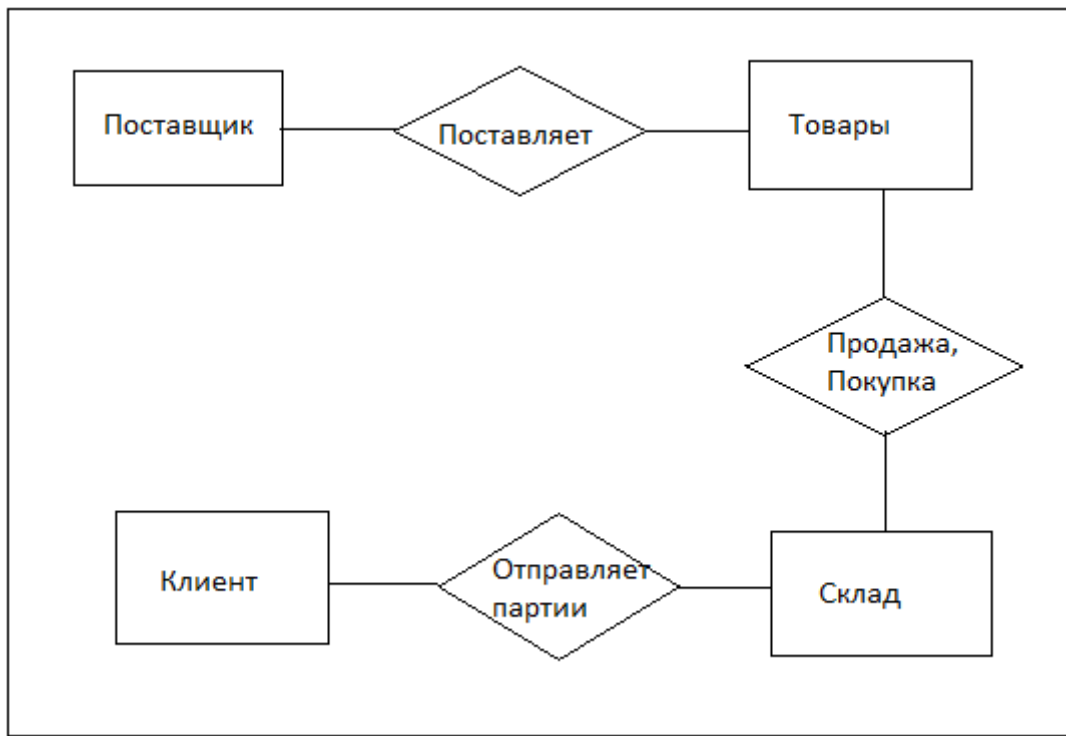


Рисунок 2.1.1 – Концептуальная схема проекта

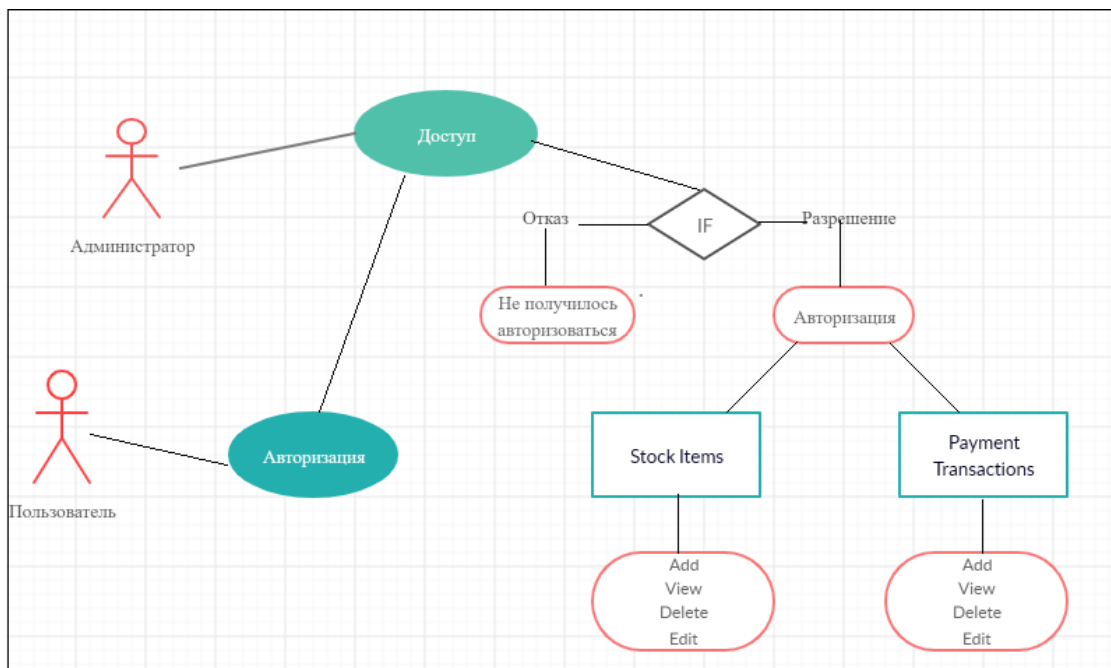


Рисунок 2.1.2 – Диаграмма прецедентов

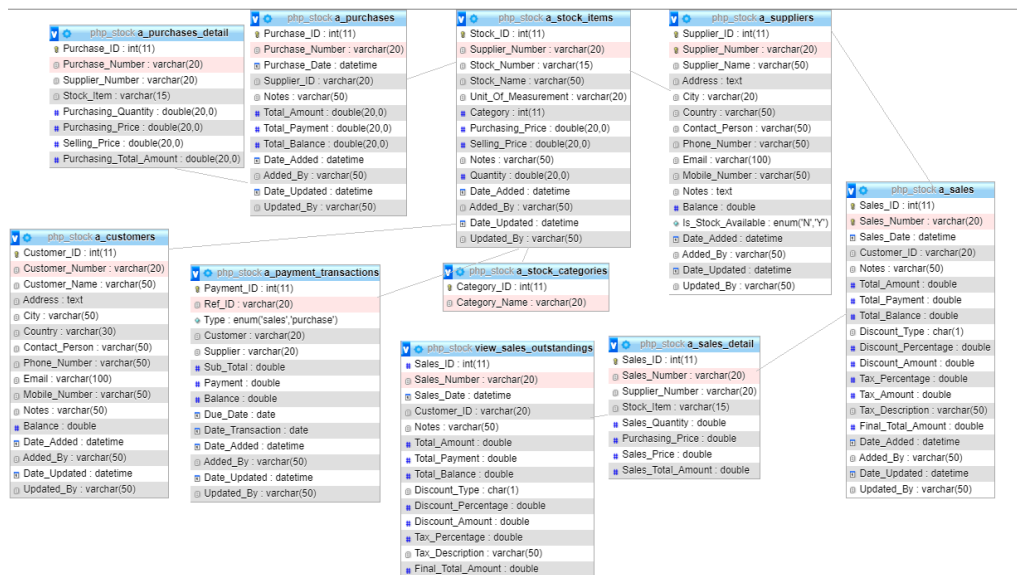


Рисунок 2.1.3 – Диаграмма классов

2.2 Средства разработки кода приложения

Для проектирования и разработки программного продукта необходимо разработать серверную и клиентскую части веб-приложения. Клиент-серверное приложение представляет собой приложение, состоящее из двух частей: клиент и сервер. Такие приложения в основном отображаются и взаимодействуют с пользователем через веб-браузер. Веб-браузер является проводником для отображения и взаимодействия с пользователем.

Клиент, это та часть приложения, которая отображается пользователю, выполняется в веб-браузере и взаимодействует визуально с пользователем. При проектировании этой стороны используются такие языки разметки, стилей и программирования как HTML, CSS и JavaScript [3].

Серверная часть характеризуется отсутствием визуального представления и необходимостью наличия веб-браузера для взаимодействия с пользователем. Название этой части вытекает из того, что все действия выполняются на сервере — специальном компьютере, который может быть расположен как за тысячи километров от браузера, так и в непосредственной близости, вплоть до одного устройства. На сервере располагается база данных и оперируют такие языки как Java, PHP, C# и т. д. Данное веб-приложение разрабатывается на динамическом языке программирования PHP. На рисунке 2.4 изображена схема взаимодействия клиента с сервером.



Рисунок 2.2.1 – Схема взаимодействия клиента с сервером

2.2.1 PHP

PHP - это популярный язык сценариев общего назначения, который особенно подходит для веб-разработки. Весомым преимуществом данного серверного языка является его доступность и практичность. Практическая природа PHP обосновывается пятью релевантными особенностями: традицией, простотой, эффективностью, безопасностью, эластичностью. PHP может работать на всех основных операционных системах, таких как Windows, Linux, Unix, Mac OS X и т. Д. Почти все основные серверы, доступные сегодня, такие как Apache, поддерживают PHP. PHP позволяет использовать широкий спектр баз данных. И самым важным фактором является то, что его можно свободно использовать и загружать, и любой желающий может загрузить PHP с его официального источника [14].

PHP-код обычно обрабатывается на веб-сервере интерпретатором PHP, реализованным в виде модуля, демона или исполняемого файла Common Gateway Interface (CGI). На веб-сервере результат интерпретируемого и исполняемого кода PHP - который может быть любым типом данных, таким как сгенерированные данные HTML или двоичные изображения - будет формировать весь или часть ответа HTTP. Существуют различные системы веб-шаблонов, системы управления веб-контентом и веб-структуры, которые можно использовать для организации или облегчения генерации такого ответа. Кроме того, PHP может использоваться для многих задач программирования вне веб-контекста, таких как автономные графические приложения и управление роботизированным дроном. Произвольный код PHP также может быть интерпретирован и выполнен через интерфейс командной строки (CLI).

Стандартный интерпретатор PHP, работающий на Zend Engine, является свободным программным обеспечением, выпущенным под лицензией PHP. PHP широко портирован и может быть бесплатно развернут на большинстве веб-серверов практически на всех операционных системах и платформах. Такие сайты, как www.facebook.com, www.yahoo.com, также построены на PHP.

Одна из основных причин этого заключается в том, что PHP может быть легко встроен в файлы HTML, а коды HTML также могут быть записаны в файл PHP [4].

Отличие PHP от клиентского языка, такого как HTML, заключается в том, что коды PHP выполняются на сервере, тогда как коды HTML отображаются непосредственно в браузере. PHP-коды сначала выполняются на сервере, а затем результат возвращается в браузер.

Единственная информация, которую знает клиент или браузер, - это результат, возвращаемый после выполнения сценария PHP на сервере, а не фактические коды PHP, присутствующие в файле PHP. Кроме того, файлы PHP могут поддерживать другие языки сценариев на стороне клиента, такие как CSS и JavaScript. На самом деле PHP может делать все, что связано с серверными сценариями или более широко известен как бэкэнд веб-сайта. Например, PHP может получать данные из форм, генерировать динамическое содержимое страницы, работать с базами данных, создавать сессии, отправлять и получать файлы cookie, отправлять электронные письма и т.д. В PHP также доступно множество хеш-функций для шифрования данных пользователя, что делает PHP безопасным и надежно использоваться в качестве языка сценариев на стороне сервера.

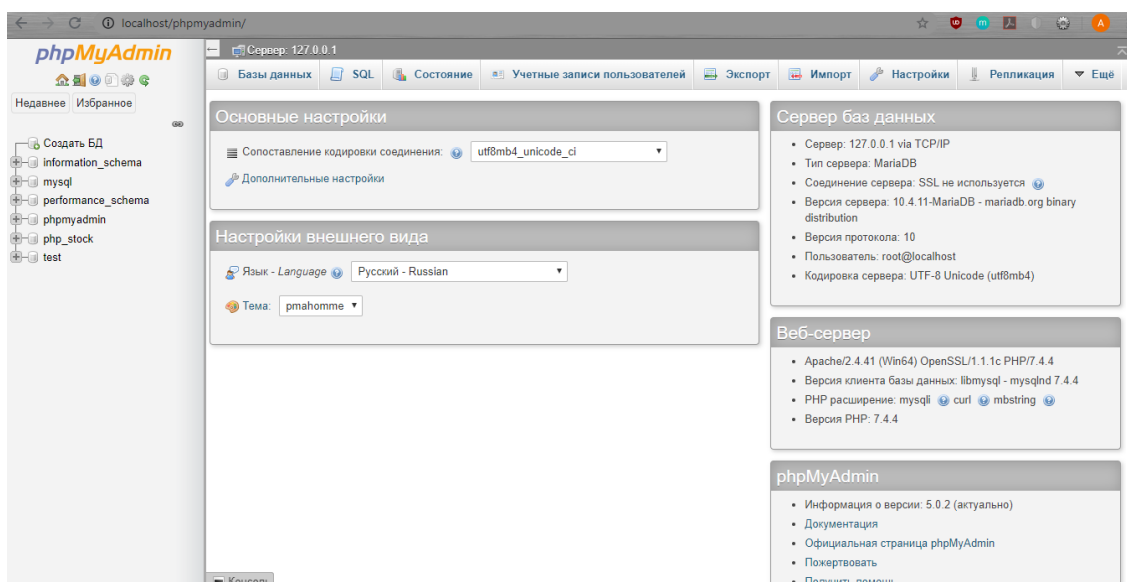


Рисунок 2.2.2 – Окно PHPMyAdmin

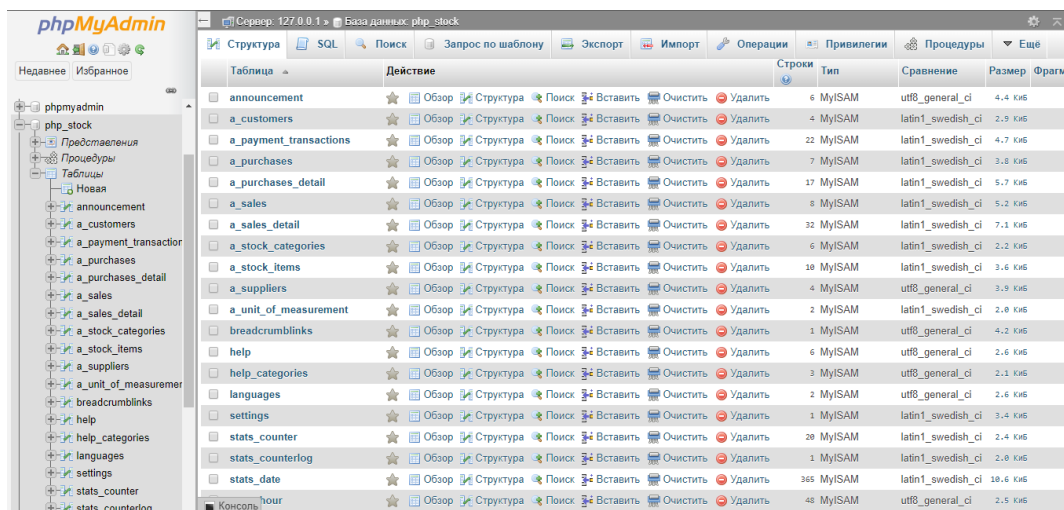


Рисунок 2.2.3 – База данных

2.2.2 Язык гипертекстовой разметки HTML

HTML расшифровывается как Hyper Text Markup Language. Он используется для разработки веб-страниц с использованием языка разметки. HTML - это сочетание гипертекста и языка разметки. Гипертекст определяет связь между веб-страницами. Язык разметки используется для определения текстового документа внутри тега, который определяет структуру веб-страниц. Этот язык используется для аннотирования (создания заметок для компьютера) текста, чтобы машина могла понимать его и соответствующим образом манипулировать текстом. Большинство языков разметки (например, HTML) читаются человеком. Язык использует теги, чтобы определить, какие манипуляции должны быть выполнены с текстом. Ни одна современная веб-страница не является полной без использования этой разметки. Для разработки предложенного проекта был использован стандарт последней версии-HTML5, который отличается от своих предшественников более строгими правилами, представляет собой продукт, сочетающий в себе свойства и синтаксические стандарты HTML и XHTML, а также нацелен на поддержку большего количества мультимедийных технологий [20].

2.2.3 Каскадные стили таблиц CSS

Каскадные таблицы стилей (CSS) - это язык таблиц стилей, используемый для описания представления документа, написанного на языке разметки, например HTML. CSS является краеугольной технологией Всемирной паутины, наряду с HTML и JavaScript.

Основная задача CSS - это выгрузки HTML-файла, так как перед логическим оформлением и описанием внешнего вида страница хранилась в одном файле и сильно затрудняла поддержку сайта и его дальнейшее развитие.

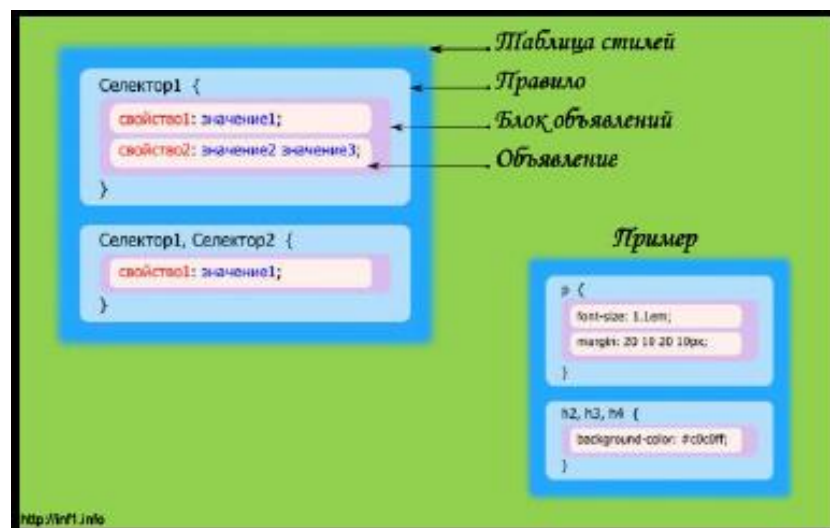


Рисунок 2.2.4 – Структура таблицы стилей

2.2.4 База данных MySQL

MySQL - это бесплатная база данных с открытым исходным кодом, которая обеспечивает эффективное управление базами данных, подключая их к программному обеспечению. Это стабильное, надежное и мощное решение с расширенными функциями, включая:

Безопасность данных - MySQL всемирно известен как самая безопасная и надежная система управления базами данных, используемая в популярных веб-приложениях, включая WordPress, Drupal, Joomla, Facebook и Twitter. Безопасность данных и поддержка обработки транзакций, которые сопровождают последнюю версию MySQL, могут принести большую пользу любому бизнесу, особенно если это бизнес электронной коммерции, который включает частые денежные переводы.

Масштабируемость по требованию. MySQL предлагает непревзойденную масштабируемость, чтобы упростить управление глубоко встроенными приложениями, используя меньшую площадь даже в больших хранилищах, в которых хранятся терабайты данных. Гибкость по требованию - главная особенность MySQL. Это решение с открытым исходным кодом позволяет полностью настроить предприятия электронной коммерции с уникальными требованиями к серверу баз данных [9].

Высокая производительность. MySQL имеет отличную структуру механизма хранения, которая помогает системным администраторам настраивать сервер базы данных MySQL для безупречной производительности. Будь то веб-сайт электронной коммерции, который получает миллион запросов каждый день, или высокоскоростная система транзакционной обработки, MySQL разработан для удовлетворения даже самых требовательных приложений, обеспечивая оптимальную скорость, полнотекстовые индексы и уникальные кэши памяти для повышения производительности.

Круглосуточная работа. MySQL обеспечивает круглосуточную работу без выходных и предлагает широкий спектр решений высокой доступности, включая специализированные кластерные серверы и конфигурации репликации master / slave.

Комплексная поддержка транзакций. MySQL возглавляет список надежных транзакционных баз данных, доступных на рынке. С такими функциями, как полная атомарная, согласованная, изолированная, надежная поддержка транзакций; поддержка многоверсионных транзакций; и неограниченная блокировка на уровне строк, это идеальное решение для полной целостности данных. Это гарантирует мгновенную идентификацию тупиковых ситуаций благодаря серверной ссылочной целостности.

Полный контроль рабочего процесса. Со средним временем загрузки и установки менее 30 минут MySQL означает удобство использования с первого дня. Независимо от того, используете ли вы платформу Linux, Microsoft, Macintosh или UNIX, MySQL - это комплексное решение с функциями самостоятельного управления, которые автоматизируют все: от расширения и настройки пространства до проектирования данных и администрирования базы данных [7].

Снижение общей стоимости владения. Перенос существующих приложений баз данных на MySQL позволяет предприятиям значительно экономить на новых проектах. Надежность и простота управления могут сэкономить время на устранение неполадок, которое в противном случае тратится на решение проблем простоя и производительности.

Гибкость открытого исходного кода. Все страхи и беспокойства, возникающие в решении с открытым исходным кодом, могут быть прекращены благодаря круглосуточной поддержке MySQL и корпоративной компенсации. Безопасная обработка и надежное программное обеспечение MySQL объединяются, чтобы обеспечить эффективные транзакции для проектов большого объема. Это делает техническое обслуживание, отладку и обновление быстрыми и легкими, расширяя возможности конечного пользователя.

2.2.5 Web-сервер. Apache

Главная задача веб сервера принимать HTTP-запросы от пользователей, обрабатывать их, переводить в цифровой компьютерный код. Затем выдавать HTTP-ответы, преобразуя их из миллионов нулей и единиц в изображения, медиа-поток, буквы, HTML страницы [1].

Программное обеспечение веб-сервера контролирует, как пользователь получает доступ к размещенным файлам. Доступ к нему осуществляется через доменные имена веб-сайтов и обеспечивает доставку контента сайта запрашивающему пользователю. В качестве аппаратного обеспечения веб-сервер - это компьютер, на котором хранится программное обеспечение веб-сервера и другие файлы, связанные с веб-сайтом, такие как документы HTML,

изображения и файлы JavaScript. Оборудование веб-сервера подключено к Интернету и позволяет обмениваться данными с другими подключенными устройствами. Процесс веб-сервера является примером модели клиент/сервер. Все компьютеры, на которых размещаются веб-сайты, должны иметь программное обеспечение веб-сервера. Ведущие веб-серверы включают в себя Apache, Microsoft Internet Information Server (IIS) и Nginx - ярко выраженный механизм X. Другие веб-серверы включают сервер Novell NetWare, Google Web Server (GWS) и семейство серверов Domino IBM.

В качестве веб-сервера выбран Apache, версии 2.4.41, поддерживаемый поддерживаемый платформой хатрр и позволяющий удобно обращаться к серверу. Apache - это мощная программа для веб-серверов, функции которой сравниваются с ее дорогостоящими конкурентами. HTTP-сервер Apache - это программное обеспечение (или программа), которое работает в фоновом режиме под соответствующей операционной системой, поддерживает многозадачность и предоставляет сервисы для других приложений, которые к нему подключаются, таких как клиентские веб-браузеры. Сначала он был разработан для работы с операционными системами Linux / Unix, но позже был адаптирован для работы под другими системами, включая Windows и Mac. Двоичный файл Apache, работающий под UNIX, называется HTTPd (сокращение от HTTP-демона), а под win32 - Apache.exe. Исходное ядро Apache довольно простое и содержит ограниченное количество функций. Его мощь, скорее всего, связана с дополнительными функциями, представленными во многих модулях, которые написаны программистами и могут быть установлены для расширения возможностей сервера. Чтобы добавить новый модуль, все, что вам нужно сделать, это установить его и перезапустить сервер Apache. Функциональность, которая вам не нужна или не нужна, может быть легко удалена, что на самом деле считается хорошей практикой, поскольку она делает сервер небольшим и легким, запускается быстрее, потребляет меньше системных ресурсов и памяти и делает сервер менее уязвимым для дыр в безопасности. Сервер Apache также поддерживает сторонние модули, некоторые из которых были добавлены в Apache 2 в качестве постоянных функций. Сервер Apache очень легко интегрируется с другими приложениями с открытым исходным кодом, такими как PHP и MySQL, что делает его еще более мощным, чем он есть. Сервер Apache предлагает ряд сервисов, которые могут использовать клиенты. Эти услуги предлагаются с использованием различных протоколов через разные порты и включают в себя: протокол передачи гипертекста (HTTP), обычно через порт 80, простой протокол передачи почты (SMTP), обычно через порт 25, службу доменных имен (DNS) для сопоставления доменных имен с их соответствующие IP-адреса, обычно через порт 53, и протокол передачи файлов (FTP) для загрузки и выгрузки файлов, обычно через порт 21.

2.2.6 XAMPP. Установка для Windows

XAMPP - сборка Web-сервера, в которой содержится набор приложений для полноценного функционирования сайта. При этом не требуется настройка каждого компонента по отдельности.

Для установки запускаем инсталлятор и на стартовом окне жмём кнопку «Next».

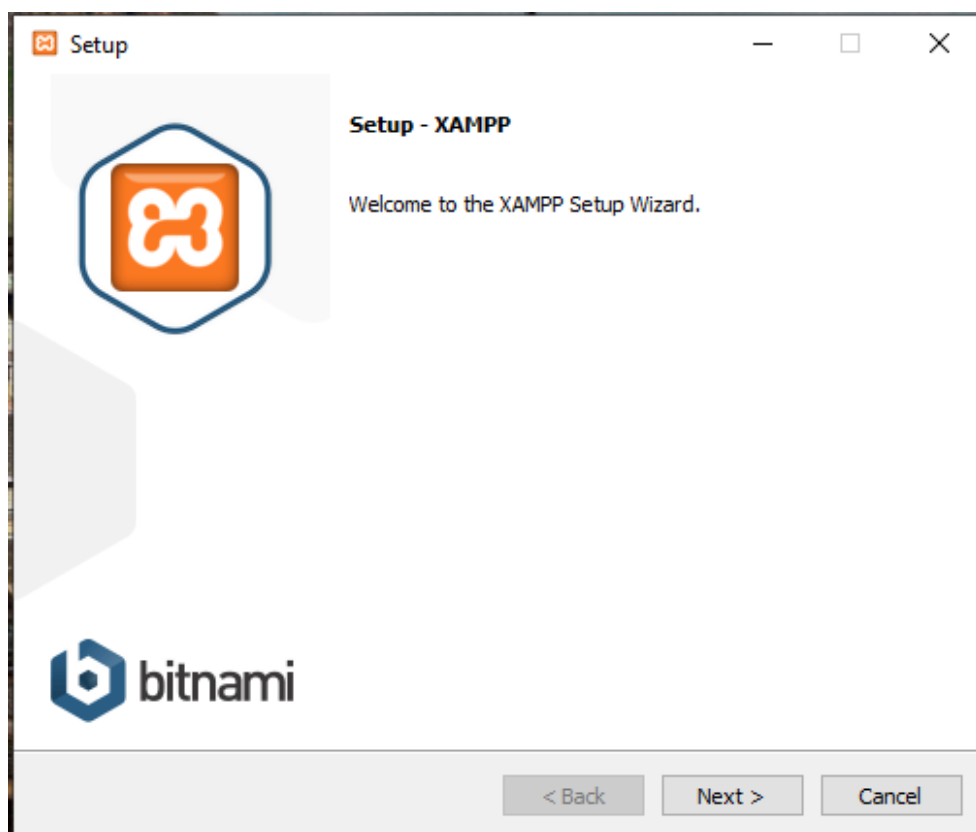


Рисунок 2.2.5 – Окно установки XAMPP.

Появится окно выбора устанавливаемых компонентов.

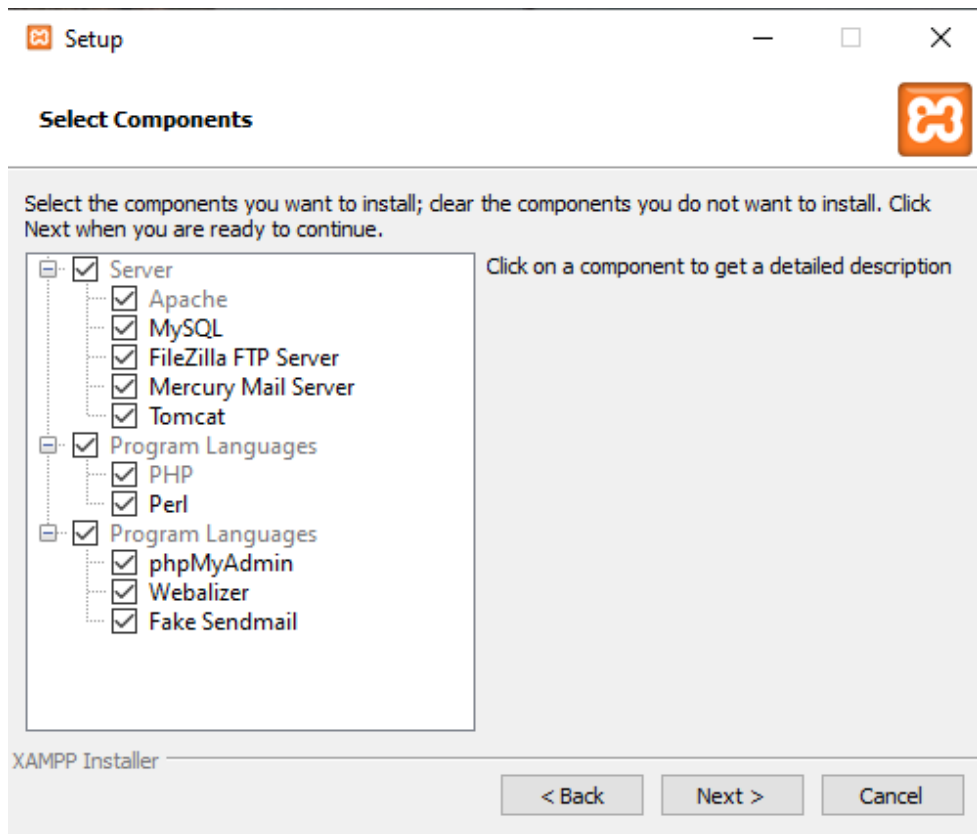


Рисунок 2.2.6 – Выбор компонентов

После установки открывается панель управления приложением.

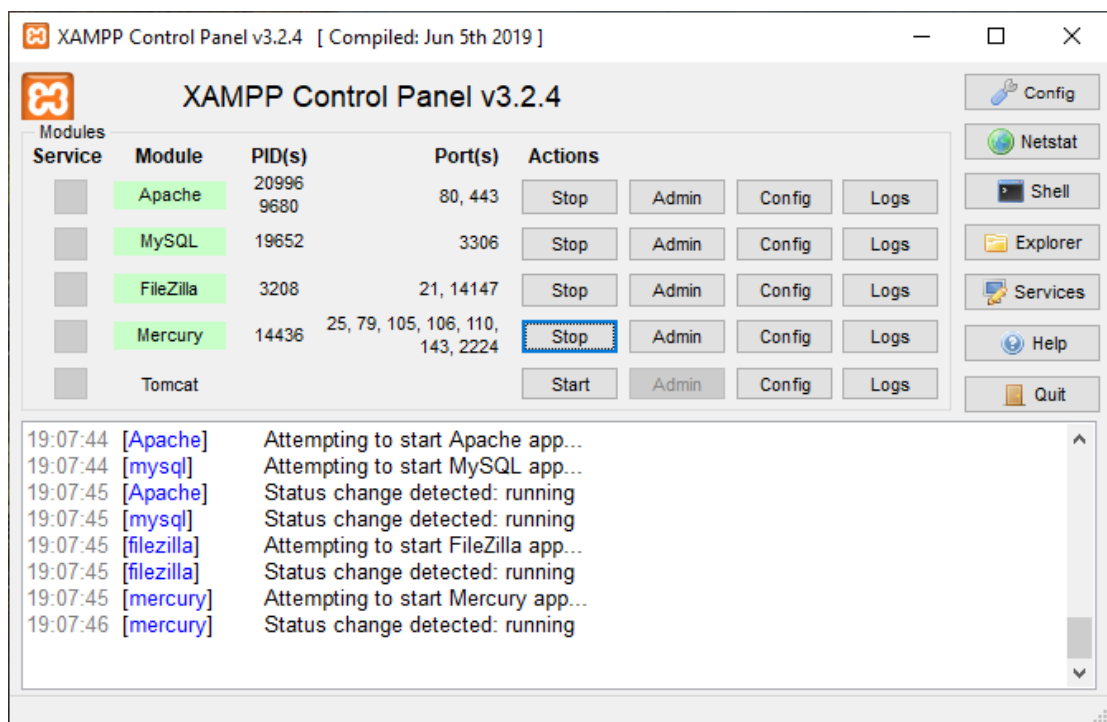


Рисунок 2.2.7 – Панель управления

3 Экспериментальная часть

3.1 Интерфейс приложения

Основной интерфейс включает в себя:

- окно входа;
- главную страницу с последними операциями;
- информацию о поставщиках, товарах, клиентах;
- настройки дизайна и темы;
- двуязычную форму.

На рисунке 3.1.1 представлено окно входа, состоящее из регистрационной формы и авторизации. Также есть функция восстановления пароля, при помощи имени пользователя.

Включает в себя:

- автоматическую авторизацию;
- сохранение пользователя;
- постоянное запрашивание данных.

Войти ?

Пользователь admin

Пароль

Действия ^

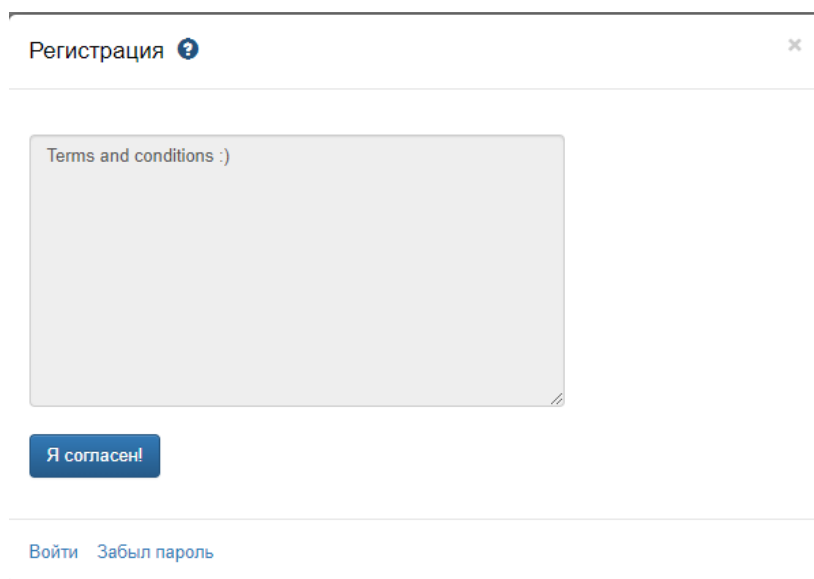
- Автоматическая авторизация
- Сохранить пользователя
- Всегда запрашивать логин и пароль

Войти Reset

[Забыл пароль](#) [Регистрация](#)

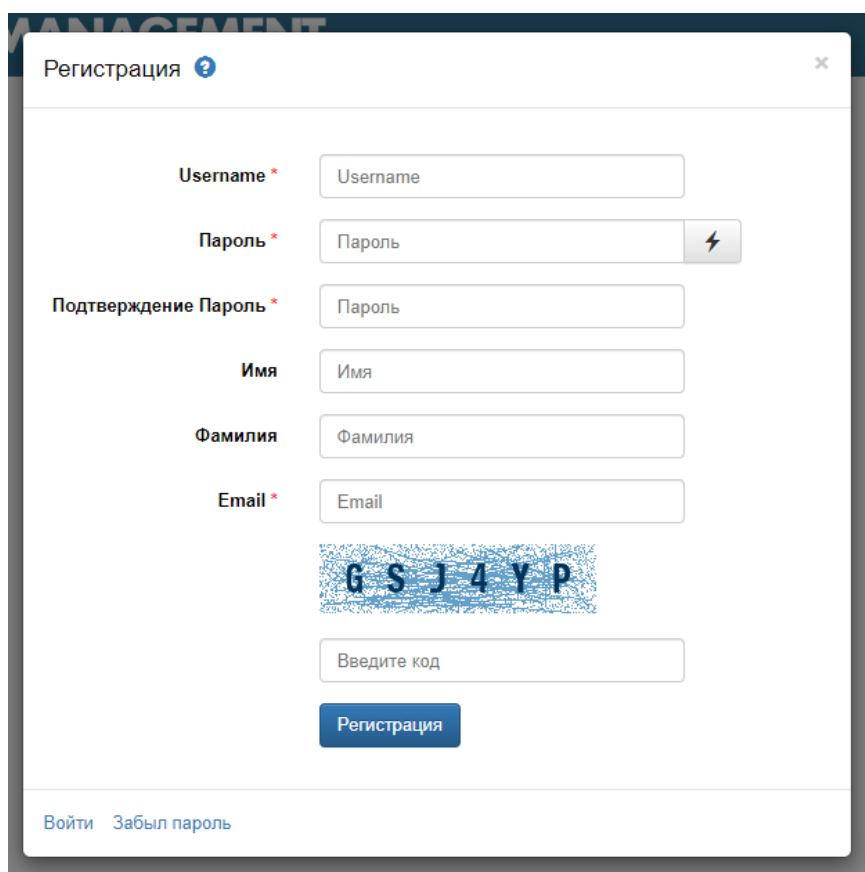
Рисунок 3.1.1. – Окно входа

На рисунке 3.1.2 и 3.1.3 показана форма регистрации с соглашением с условиями пользования. Есть возможность сгенерировать пароль.



The screenshot shows a registration window titled "Регистрация" with a help icon and a close button. Below the title is a large grey rectangular area containing the text "Terms and conditions :)". Below this area is a blue button labeled "Я согласен!". At the bottom of the window, there are two links: "Войти" and "Забыл пароль".

Рисунок 3.1.2 – Условия и положения



The screenshot shows a registration window titled "Регистрация" with a help icon and a close button. The form contains several input fields: "Username *" with a placeholder "Username"; "Пароль *" with a placeholder "Пароль" and a password generator icon; "Подтверждение Пароль *" with a placeholder "Пароль"; "Имя" with a placeholder "Имя"; "Фамилия" with a placeholder "Фамилия"; "Email *" with a placeholder "Email". Below these fields is a CAPTCHA image showing the characters "G S J 4 Y P" and a text input field with the placeholder "Введите код". A blue button labeled "Регистрация" is positioned below the CAPTCHA. At the bottom of the window, there are two links: "Войти" and "Забыл пароль".

Рисунок 3.1.3 – Регистрация

На рисунке 3.1.4 представлена Главная страница, включающая в себя меню в левой части приложения, кнопку перехода между языками и последние операции.

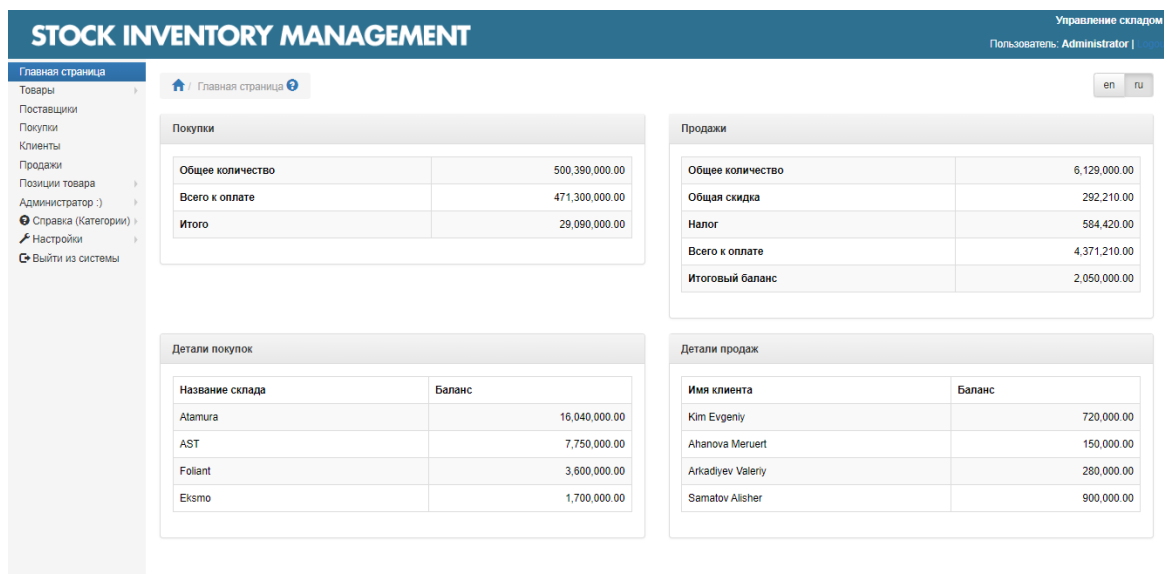


Рисунок 3.1.4 – Главная страница

Также меню можно расположить в верхней панели.

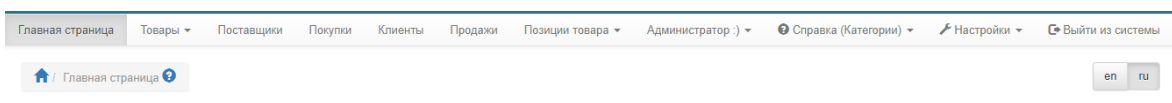


Рисунок 3.1.5. – Горизонтальное меню

Раздел «Товары» указан на рисунке 3.1.6. Здесь предоставляется информация о деталях купли/продажи товара, указано имя поставщика, количество товара. Функция «добавить товар» продемонстрирована на рисунке 3.1.7. Здесь можно указать категорию товара, данные о поставщике и о самом товаре.

Номер поставщика	Номер товара	Название товара	Цена покупки	Цена продажи	Количество
1. Atamura	Stock-01	Kazakhs	Tenge 100,000.00	Tenge 110,000.00	2,469
2. Atamura	Stock-02	History of Kazakhstan	Tenge 3,300.00	Tenge 3,500.00	25,702
3. AST	Stock-03	Pride and prejudice	Tenge 12,500.00	Tenge 15,000.00	2,432
4. Foliant	Stock-04	The Adventures of Sherlock Holmes	Tenge 200.00	Tenge 250.00	49,850
5. Foliant	Stock-05	Blood of elves	Tenge 1,500.00	Tenge 1,800.00	9,910
6. Foliant	Stock-06	The hearts of three	Tenge 3,000.00	Tenge 3,200.00	4,968
7. AST	Stock-07	Philosopher stone	Tenge 2,000.00	Tenge 2,300.00	12,834
8. AST	Stock-08	Poster	Tenge 5,000.00	Tenge 5,800.00	5,970
9. AST	Stock-09	Godfather	Tenge 7,000.00	Tenge 7,900.00	3,967
10. Eksmo	Stock-10	After you	Tenge 5,000.00	Tenge 5,400.00	1,956

Рисунок 3.1.6 – Товары

Категории * +
 Номер поставщика *
 Номер товара *
 Название товара *
 Единица измерения * +
 Цена покупки *
 Цена продажи *
 Заметки *

Рисунок 3.1.7 – Добавить товар

В разделе «Поставщики» предоставляется вся информация о складе: номер склада, его название и ответственное контактное лицо. Данные могут редактироваться, есть функция онлайн-закупа у поставщика, причем общая стоимость рассчитывается автоматически. Можно просмотреть товары, предоставляемые определенным поставщиком.

Поставщик

Разметка страницы: 10 Запись 1 с 4 до 4

				Номер склада	Название склада	Контактное лицо	Номер телефона	Мобильный телефон	Баланс	Товар в наличии		
<input type="checkbox"/>	Purchase Now			Покупки 3	Товары 2	Supplier-01	Atamura	John Mc. Enroe	022124415093	0824132048929	Tenge 16,040,000.00	Ya
<input type="checkbox"/>	Purchase Now			Покупки 2	Товары 4	Supplier-02	AST	Martina Navatrilova	02148272080	081232442840	Tenge 7,750,000.00	Ya
<input type="checkbox"/>	Purchase Now			Покупки 1	Товары 3	Supplier-03	Foliant	Joko Sentul	03142348293	081242009827	Tenge 3,600,000.00	Ya
<input type="checkbox"/>	Purchase Now			Покупки 1	Товары 1	Supplier-04	Eksmo	Siapa Sajalah	0213248290	081242932890	Tenge 1,700,000.00	Ya
									Tenge 29,090,000.00			

Рисунок 3.1.8 – Поставщики

Поставщик / Редактировать

Страница 1 до 4

ID Поставщика 1

Номер склада * Supplier-01

Название склада * Atamura

Адрес * Address for the first supplier

Город * Almaty

Страна * Kazakhstan

Контактное лицо * John Mc. Enroe

Номер телефона * 022124415093

Email * john.mcenroe@gmail.com

Мобильный телефон * 0824132048929

Заметки * Just a note

Баланс 16,040,000

Товар в наличии? Ya

Сохранить Отмена

Рисунок 3.1.9 – Редактирование информации

[/ Поставщик](#) / [Покупки](#) / [Добавить](#)

Номер покупки *	Purchase-0000000008
Дата покупки *	2020/04/28 11:13:42
ID Поставщик *	Atamura
Заметки	Заметки
Общее количество *	35000
Всего к оплате *	0
Итого *	35000

Детали покупки

+	Номер поставщика	Товар	Количество закупок	Закупочная цена	Продажная цена	Общая сумма покупки
🗑	Atamura	Kazakhs	5	7,000.00	5,000.00	35000
🗑	Выбрать	Выбрать	0	0	0	0
+						

[Добавить](#) [Отмена](#)

Рисунок 3.1.10 – Заказ у поставщика

[/ Поставщик](#) / [Товары](#)

Back to master record page

Номер склада	Название склада	Контактное лицо	Номер телефона	Мобильный телефон	Баланс	Товар в наличии?
Supplier-01	Atamura	John Mc. Enroe	022124415093	0824132048929	Tenge 16,040,000.00	Ya

Разметка страницы 10 Запись 1 с 2 до 2

			Номер поставщика	Номер товара	Название товара	Цена покупки	Цена продажи	Количество
☐	🔍	📄	1. Atamura	Stock-01	Kazakhs	Tenge 100,000.00	Tenge 110,000.00	2,469
☐	🔍	📄	2. Atamura	Stock-02	History of Kazakhstan	Tenge 3,300.00	Tenge 3,500.00	25,702

Рисунок 3.1.11 – Предоставляемые товары

На рисунке 3.1.12 продемонстрирована функция поиска, рисунок 3.1.13 показывает усовершенствованный поиск.

[/ Товары](#)

Поиск [Поиск](#)

Разметка страницы 10

Автопоиск

- Полное совпадение
- Все ключевые слова
- Любые ключевые слова

Рисунок 3.1.12 – Поиск

[Товары](#) / [Поиск](#)

ID товара	=	ID товара
Категории	=	Выбрать
Номер поставщика	=	Выбрать
Номер товара	=	Номер товара
Название товара	=	Название товара
Единица измерения	=	Выбрать
Цена покупки	=	Цена покупки
Цена продажи	=	Цена продажи
Количество	=	Количество
Заметки	=	Заметки
Данные добавлены	=	Данные добавлены
Добавлены кем	=	Добавлены кем
Данные обновлены	=	Данные обновлены
Обновлены кем	=	Обновлены кем

Рисунок 3.1.13 – Усовершенствованный поиск

Доступен экспорт данных в разных форматах.

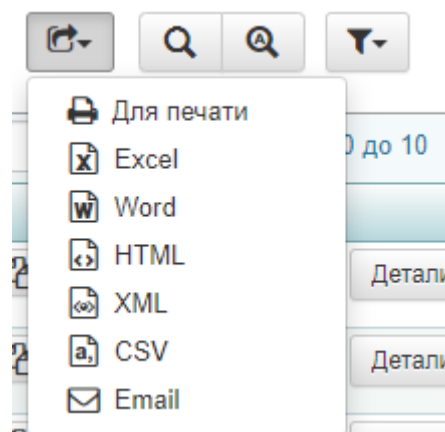


Рисунок 3.1.14 – Экспорт данных

ID	Supplier_Number	Stock_Number	Stock_Name	Purchasing_Price	Selling_Price	Quantity
1.	Atamura	Stock-01	Kazakhs	Tenge 100,000.00	Tenge 110,000.00	2,469
2.	Atamura	Stock-02	History of Kazakhstan	Tenge 3,300.00	Tenge 3,500.00	25,702
3.	AST	Stock-03	Pride and prejudice	Tenge 12,500.00	Tenge 15,000.00	2,432
4.	Foliant	Stock-04	The Adventures of Sherlock Holmes	Tenge 200.00	Tenge 250.00	49,850
5.	Foliant	Stock-05	Blood of elves	Tenge 1,500.00	Tenge 1,800.00	9,910
6.	Foliant	Stock-06	The hearts of three	Tenge 3,000.00	Tenge 3,200.00	4,968
7.	AST	Stock-07	Philosopher stone	Tenge 2,000.00	Tenge 2,300.00	12,834
8.	AST	Stock-08	Poster	Tenge 5,000.00	Tenge 5,800.00	5,970
9.	AST	Stock-09	Godfather	Tenge 7,000.00	Tenge 7,900.00	3,967
10.	Eksmo	Stock-10	After you	Tenge 5,000.00	Tenge 5,400.00	1,956

Рисунок 3.1.15 – Версия для печати

Supplier_Number	Stock_Number	Stock_Name	Purchasing_Price	Selling_Price	Quantity
Atamura	Stock-01	Kazakhs	100000	110000	2469
Atamura	Stock-02	History of Kazakhstan	3300	3500	25702
AST	Stock-03	Pride and prejudice	12500	15000	2432
Foliant	Stock-04	The Adventures of Sherlock Holmes	200	250	49850
Foliant	Stock-05	Blood of elves	1500	1800	9910
Foliant	Stock-06	The hearts of three	3000	3200	4968
AST	Stock-07	Philosopher stone	2000	2300	12834
AST	Stock-08	Poster	5000	5800	5970
AST	Stock-09	Godfather	7000	7900	3967
Eksmo	Stock-10	After you	5000	5400	1956

Рисунок 3.1.16 – Экспорт в doc-формате

3.2 Разработка функции закупа в PHP

Фрагмент кода из PHP:

```
<?php } ?>
</span>
<?php echo $a_sales->Tax_Amount->CustomMsg ?></div></div>
```



```

    </div>
<?php } ?>
<?php if ($a_sales->Tax_Description->Visible) { // Tax_Description ?>
    <div id="r_Tax_Description" class="form-group">
        <label id="elh_a_sales_Tax_Description" for="x_Tax_Description"
class="col-sm-4 control-label ewLabel"><?php echo $a_sales->Tax_Description-
>FldCaption() ?></label>
        <div class="col-sm-8"><div<?php echo $a_sales->Tax_Description-
>CellAttributes() ?>>
<span id="el_a_sales_Tax_Description">
<input type="text" data-table="a_sales" data-field="x_Tax_Description"
name="x_Tax_Description" id="x_Tax_Description" size="30" maxlength="50"
placeholder="<?php echo ew_HtmlEncode($a_sales->Tax_Description-
>getPlaceholder()) ?>" value="<?php echo $a_sales->Tax_Description->EditValue
?>"<?php echo $a_sales->Tax_Description->EditAttributes() ?>>
</span>
<?php echo $a_sales->Tax_Description->CustomMsg ?></div></div>
    </div>
<?php } ?>
<?php if ($a_sales->Final_Total_Amount->Visible) { // Final_Total_Amount ?>
    <div id="r_Final_Total_Amount" class="form-group">
        <label id="elh_a_sales_Final_Total_Amount"
for="x_Final_Total_Amount" class="col-sm-4 control-label ewLabel"><?php echo
$a_sales->Final_Total_Amount->FldCaption() ?></label>
        <div class="col-sm-8"><div<?php echo $a_sales-
>Final_Total_Amount->CellAttributes() ?>>
<span id="el_a_sales_Final_Total_Amount">
<input type="text" data-table="a_sales" data-field="x_Final_Total_Amount"
name="x_Final_Total_Amount" id="x_Final_Total_Amount" size="30"
placeholder="<?php echo ew_HtmlEncode($a_sales->Final_Total_Amount-
>getPlaceholder()) ?>" value="<?php echo $a_sales->Final_Total_Amount-
>EditValue ?>"<?php echo $a_sales->Final_Total_Amount->EditAttributes() ?>>
<?php if (!$a_sales->Final_Total_Amount->ReadOnly && !$a_sales-
>Final_Total_Amount->Disabled && @$a_sales->Final_Total_Amount-
>EditAttrs["readonly"] == "" && @$a_sales->Final_Total_Amount-
>EditAttrs["disabled"] == "") { ?>
<script type="text/javascript">
$( "#x_Final_Total_Amount" ).autoNumeric( 'init', { aSep: ',', aDec: '.', mDec: '2',
aForm: false } );
</script>

function ValidateFinalTotalAmount() {
    if ( $( "#x_Final_Total_Amount" ).autoNumeric( 'get' ) < 0 ) {
        $( "#x_Final_Total_Amount" ).val( 0 );

```

```

    } else if ($("#x_Final_Total_Amount").val() == "") {
        $("#x_Final_Total_Amount").val(0);
    }
}

function GetBalanceTotalFinal() {
    var final_total_amount = $("#x_Final_Total_Amount").autoNumeric('get');
    var total_payment = $("#x_Total_Payment").autoNumeric('get');
    var total_balance = final_total_amount - total_payment;
    $("#x_Total_Balance").val(total_balance);
    if (total_balance < 0) {

        $("#x_Total_Payment").val($("#x_Final_Total_Amount").autoNumeric('get')
);
        $("#x_Total_Balance").val("0");
    }
}

```

```

function CalculateGrid(event) {
    var elm_name = $(event.target).attr('name');
    var start_pos = elm_name.indexOf('x') + 1;
    var end_pos = elm_name.indexOf('_', start_pos);
    var idx = elm_name.substring(start_pos, end_pos)
    $("#x" + idx + "_Sales_Total_Amount").val($("#x" + idx +
"_Sales_Quantity").autoNumeric('get') *
$("#x" + idx + "_Sales_Price").autoNumeric('get'));
    $("#x_Total_Amount").val(GetSalesTotal());
}

```

```

function GetSalesTotal() {
    var fobj = fa_sales_detailgrid.GetForm(), $fobj = $(fobj);
    fa_sales_detailgrid.PostAutoSuggest();
    if ($fobj.find("#a_confirm").val() == "F")
        return true;
    var val_elm, addcnt = 0;
    var $k = $fobj.find("#" + fa_sales_detailgrid.FormKeyCountName);
    var rowcnt = ($k[0]) ? parseInt($k.val(), 10) : 1;
    var startcnt = (rowcnt == 0) ? 0 : 1;
    var gridinsert = $fobj.find("#a_list").val() == "gridinsert";
    var total_amount = 0;
    for (var i = startcnt; i <= rowcnt; i++) {
        var infix = ($k[0]) ? String(i) : "";
        $fobj.data("rowindex", infix);
    }
}

```

```

        var checkrow = (gridinsert) ? !fa_sales_detailgrid.EmptyRow(infix) :
true;
        if (checkrow) {
            addcnt++;
            val_elm = $("#x" + infix +
"_Sales_Total_Amount").autoNumeric('get');
            total_amount += +(val_elm);
            ew_ElementsToRow(fobj);
        }
    }
    return total_amount;
}
</script>
<?php if (MS_ENTER_MOVING_CURSOR_TO_NEXT_FIELD) { ?>

```

4 Экономическая часть

В данном разделе приводится экономическое обоснование выбора проекта и расчеты эффективности внедрения данного продукта.

4.1 Экономическое обоснование проекта

Целью дипломного проекта является создание программного продукта для складского учета для ТОО «Книга-НВ». Сегодня электронная коммерция стремительно развивается. Есть также предприниматели, которые ведут бумажную документацию работы склада, но не понимают преимуществ автоматизированного складского учета. Складской учет - очень эффективный инструмент для увеличения производительности труда и упрощения составления и ведения отчетности. В данном проекте разработана база данных и веб-приложение для доступа к данным полученным в ходе проведения научно-исследовательской работы.

База данных дает возможность увеличить скорость поиска и доступа к необходимым данным, упорядочивает и систематизирует их. Как следствие это оказывает влияние на производительность труда в примерно в 3 - 4 раза, не необходимости поиска и работы с данными на бумажных носителях.

Онлайн-учет книг и сопутствующего товара позволяет решить ряд задач:

- снизить общую стоимость;
- создать базу данных поставщиков и клиентов, у которых закупается нужный товар.
- оптимизация рабочего процесса;
- поддержание порядка на складе и контроль над осуществлением отгрузок;
- принятие товара и его учет;
- повышается конкурентоспособность бизнеса, любые бизнес-операции становятся более эффективными.

В этом разделе представлен обзор экономической составляющей проекта, которая отражает временные, трудовые и финансовые затраты проекта.

4.2 Расчет трудоемкости разработки

Трудоёмкость – это экономический показатель, характеризующий затраты рабочего времени на производство единицы продукции или на выполнение конкретной технологической операции; один из показателей производительности труда. В программировании программного обеспечения по мере того, как проектирование программного обеспечения реализуется, число элементов и их взаимосвязей постепенно становится огромным, что становится слишком трудным для понимания сразу.

Сложность проектирования программного обеспечения трудно оценить без использования метрик и показателей сложности. Единицей измерения трудоемкости является время(в часах, днях и т.п). Иногда ее измеряют в человеко часах. Определение трудоемкости разработки программного продукта начнем с составления перечня основных этапов и видов работ, выполненные в ходе разработки. Опытно-статистический метод лежит в основе расчета трудоёмкости программного продукта [16].

Таблица 4.1.1 – Распределение работ по этапам и видам и оценка их трудоемкости

Этапы разработки ПП	Вид работы	Трудоемкость, чел. час.
Этап 1	Определение цели	10
Этап 2	Выбор средств для разработки ПП	15
Этап 3	Оформление блок-схемы для ПП	30
Этап 4	Составление таблиц и связей	10

Этап 5	Разработка графического дизайна ПП	40
Этап 6	Создание функционала продукта	70
Этап 7	Отладка программной части продукта	35
Этап 8	Налаживание работы	30
Итого: трудоемкость выполнения дипломного проекта		240

4.3 Расчет затрат на разработку ПП

В данный этап включены затраты:

- на необходимые материальные ресурсы;
- затраты на оплату труда;
- затраты на расход электроэнергии;
- общие затраты на амортизационные отчисления;
- на социальный налог;
- прочие расходы.

Расчет затрат на материальные ресурсы описан в таблице 4.2.1, которая представлена ниже.

Определяем общую сумму затрат на различные материальные ресурсы (Z_M), по формуле, которая представлена ниже:

$$Z_M = \sum_{i=1}^n P_i * C_i, \quad (4.1)$$

где P_i - расход i -го вида материального ресурса, натуральные единицы;

C_i - цена за единицу i -го вида материального ресурса, тг;

i - вид материального ресурса;

n - количество видов материальных ресурсов.

Таблица 4.2.1– Затраты на материальные ресурсы

Наименование материального ресурса	Единица измерения	Количество израсходованного материала	Цена за единицу, тг	Сумма, тг
Бумага А4 для блок схем	Штук	1	1750	1750,
Тетрадь 96 л.	Штук	1	340	340
Ручки	Штук	3	150	450
ИТОГО затраты на материальные ресурсы				2540,00

$$Z_{\text{общее}} = Z_{\text{бумага}} + Z_{\text{тетрадь}} + Z_{\text{ручки}} = 2540,00 \text{ тг.}$$

Расчет затрат на необходимое оборудование приведены в таблице 4.2.2.

Таблица 4.2.2 – Расчет затрат на оборудование и ПО, необходимое для проекта

Наименование материала	Марка	Ед. измерения	Количество	Цена за ед. в тенге	Сумма в тенге
Ноутбук	HP Pavilion 14-5GZ09EA	Штук	1	149000	149000
Принтер	HP Laser Jet Pro M15a	Штук	1	39890	39890
Компьютерная мышь беспроводная	Logitech M170	Штук	1	4200	4200
Итого					193090

$$Z_{\text{общее}} = 149000 + 39890 + 4200 = 193090 \quad (4.2)$$

Ввиду того, что для разработки программного продукта используются техника, которая потребляет электроэнергию, необходимо рассчитать затраты на электроэнергию. Расчеты, связанные с электроэнергией представлены в Таблице 4.2.3.

Таблица 4.2.3 – Затраты на электроэнергию

Наименование оборудования	Паспортная мощность, кВт	Коэффициент использования мощности	Время работы оборудования для разработки ПП, ч	Цена электроэнергии, тг/кВт*ч	Сумма, тг
Ноутбук HP Pavilion 14-5GZ09EA	0,6	0,7	240	19,17	1932
Принтер HP Laser Jet Pro M15a	0,5	0,7	6	19,17	40,3
ИТОГО затраты на электроэнергию					1972,3

Общая сумма затрат на электроэнергию (Z_3) рассчитывается по формуле:

$$Z_3 = \sum_{i=1}^n M_i * K_i * T_i * Ц, \quad (4.3)$$

где M_i - паспортная мощность i -го электрооборудования, кВт;
 K_i - коэффициент использования мощности i -го электрооборудования (принимается $K_i=0.7, 0.9$);
 T_i - время работы i -го оборудования за весь период разработки ПП ч;
 $Ц$ - цена электроэнергии, тг/кВт×ч;
 i - вид электрооборудования;
 n - количество электрооборудования.

$$Z_{3, \text{общие}} = (0,6 \times 0,7 \times 240 \times 19,17) + (0,5 \times 0,7 \times 6 \times 19,17) = 1972,3 \text{ тенге} \quad (4.4)$$

Для статьи «Затраты на оплату труда» необходимо включить расходы по оплате труда всех работников (студента, дипломного руководителя и консультантов, привлеченных к дипломному проекту). Которые были вовлечены и заняты разработкой и проектированием самого программного продукта. Средний заработок программиста, начинающего по представленным данным составляет 140000 тенге, средний заработок руководителя проекта составляет примерно 180000 не учитывая всякие бонусы, относящиеся к его званию кандидата технических наук и профессора.

Затраты на оплату труда рассчитываются по форме, приведенной в Таблице 4.2.4.

Таблица 4.2.4 – Затраты на оплату труда

Категория работника	Квалификация	Трудоемкость разработки ПП, чел.×ч	Часовая ставка, тг/ч	Сумма, тг
Служащий	Программист	240	833,3	200000
Руководитель	Руководитель проекта	100	1071,43	107143
ИТОГО затраты на оплату труда				307143

Общая сумма затрат на оплату труда ($Z_{тр}$) определяется по формуле:

$$Z_{тр} = \sum_{i=1}^n ЧС_i * T_i, \quad (4.5)$$

Где $ЧС_i$ - часовая ставка i -го работника, тг;
 T_i - трудоемкость разработки ПП, чел.×ч;
 i - категория работника;

n - количество работников, занятых разработкой ПП.

Часовая ставка работника может быть рассчитана по формуле:

$$\text{ЧС}_i = \frac{\text{ЗП}_i}{\text{ФРВ}_i}, \quad (4.6)$$

Фонд рабочего времени при 40 часовой неделе в 5 рабочих дней равняется 21 дням в месяц по 8 часов:

$$\text{ФРВ}_i = 21 \times 8 = 168 \text{ ч.}$$

$$\text{ЧС}_i = \frac{\text{ЗП}_i}{\text{ФРВ}_i} = \frac{140\,000}{168} = 833,3 \quad (4.7)$$

$$\text{ЧС}_{\text{рук}} = \frac{\text{ЗП}_{\text{рук}}}{\text{ФРВ}_{\text{рук}}} = \frac{180\,000}{168} = 1071,43 \quad (4.8)$$

$$\text{З}_{\text{тр.общее}} = (833,3 \times 250) + (11071,43 \times 100) = 307143 \quad (4.9)$$

Трудоемкость разработки ПП определяется по данным таблицы 1.

В статью «Социальный налог» включается сумма с учетом обязательного пенсионного взноса, составляющего 10% от заработной платы, которая вычисляется по формуле 4.10:

$$\text{СО} = (\text{ЗП} - \text{ОПВ}) \times 3,5 \% \quad (4.10)$$

Тогда социальные отчисления будут равны:

$$\text{СО} = (307143 - 30714,3) \times 0,035 = 9675 \text{ тг.}$$

Социальный налог вычисляется по формуле 4.11:

$$\text{СН} = (\text{ЗП} - \text{ОПВ} - \text{ВОСМС}) \times 9,5\% - \text{СО}, \quad (4.11)$$

где ВОСМС – взнос на ОСМС, 1%

Социальный налог составит:

$$\text{СН} = (307143 - 30714,3 - 3071,43) \times 0,095 - 9675 = 25968,84 - 9675 = 16293,94 \text{ тг.}$$

Общие отчисления на социальные нужды вычисляются по формуле 4.12:

$$OT = CO + BOСМС + CH, \quad (4.12)$$

где ВОСМС – взнос на ОСМС, 2%

Отчисления на социальные нужды составят:

$$OT = 9675 + 6142,86 + 16293,94 = 32111 \text{ тг.}$$

В статью «Амортизация основных фондов» включается сумма амортизационных отчислений от стоимости оборудования и программного обеспечения (ПО), используемых при разработке ПП. Амортизационные отчисления рассчитываются по форме, приведенной в таблице.

Общая сумма амортизационных отчислений определяется по формуле:

$$Z_{AM} = \sum_{i=1}^n \frac{\Phi_i * H_{Ai} * T_{НИРi}}{100 * T_{Э\Phi i}} \quad (4.13)$$

где Φ_i - стоимость i -го ОФ, тг;

H_{Ai} - годовая норма амортизации i -го ОФ, %;

$T_{НИРi}$ - время работы i -го ОФ за весь период разработки ПП, ч;

$T_{Э\Phi i}$ - эффективный фонд времени работы i -го ОФ за год, ч/год;

i - вид ОФ;

n - количество ОФ.

При определении стоимости ОФ необходимо учесть также затраты на доставку и монтаж, установку ПО. Эти затраты могут быть приняты в размере 10-25 % от затрат на приобретение ОФ.

Годовые нормы амортизации ОФ принимаются по налоговому кодексу РК или определяются, исходя из возможного срока полезного использования ОФ:

$$H_{Ai} = \frac{100}{T_{Ni}} \quad (4.14)$$

$$H_{Ai} = \frac{100}{5} = 20\%$$

$$H_{Ai} = \frac{100}{7} = 14,3\%$$

где T_{Ni} - возможный срок использования i -го ОФ, год;

Возможный срок полезного использования ОФ может быть принят: 5 лет для ноутбука и 7 лет для принтера (ОФ может быть принят от 3 до 10 лет).

$$Z_{AM1} = \frac{149\,000 \times 20 \times 240}{100 \times 1825} = 3918,90 \text{ тенге} \quad (4.15)$$

$$Z_{AM2} = \frac{39\,890 \times 14,3 \times 6}{100 \times 1825} = 21,83 \text{ тенге} \quad (4.16)$$

$$Z_{AMобщие} = 3918,90 + 21,83 = 3940,73 \text{ тенге} \quad (4.17)$$

Таблица 4.2.5 - Амортизация основных фондов (ОФ)

Наименование оборудования и ПО	Стоимость оборудования и ПО, тг	Годовая норма амортизации, %	Эффективный фонд времени работы оборудования и ПО, ч/год	Время работы оборудования и ПО для разработки ПП, ч	Сумма, тг
Ноутбук HP Pavilion 14-5GZ09EA	149 000	20	1825	240	3918,90
Принтер HP Laser Jet Pro M15a	39 890	14,3	1825	6	21,83
ИТОГО					3940,73

В статью «Прочие затраты» включаются расходы на коммунальные платежи, затраты на лицензирование и сертификацию, расходы на рекламу, канцелярские и прочие хозяйственные расходы.

Таблица 4.2.6 – Затраты на интернет

Цена за 1 месяц, тг	Количество месяцев	Сумма, тг
4 500	1	4 500

На основании полученных данных по отдельным статьям составляется смета затрат на разработку (материальные затраты, затраты на оплату труда, отчисления на социальные нужды, амортизация основных фондов, прочие затраты) ПП по форме, приведенной в таблице 4.2.7.

Таблица 4.2.7 – Смета затрат на разработку программы

Статьи затрат	Сумма, тг
1. Материальные затраты, в том числе:	
- материалы	2540,00
- оборудование и ПО	193090,00
- электроэнергия	1972,3
2. Затраты на оплату труда.	307143,00
3. Отчисления на социальные нужды.	32111,8

4. Амортизация основных фондов.	3940,73
5. Прочие затраты.	4500,00
ИТОГО по смете	545297,83



Диаграмма 1 – Смета затрат на разработку программы

4.4 Определение возможной (договорной) цены ПП

Величина возможной (договорной) цены программы должна устанавливаться с учетом эффективности, качества и сроков ее выполнения на уровне, отвечающем экономическим интересам заказчика (потребителя) и исполнителя. Договорная цена ($Ц_d$) для прикладных ПП рассчитывается по формуле:

$$Ц_d = Z_{\text{НИР}} \times \left(1 + \frac{P}{100}\right), \quad (4.18)$$

где $Z_{\text{НИР}}$ - затраты на разработку ПП, тг;

P - средний уровень рентабельности ПП. % (принимается в размере 40 %, по согласованию с консультантом по экономической части).

$$\begin{aligned} Ц_d &= Z_{\text{НИР}} \times \left(1 + \frac{P}{100}\right) = 545297,83 + 545297,83 \times 0,4 = \\ &= 545297,83 + 218119,13 = 763416,96 \text{ тенге} \end{aligned} \quad (4.19)$$

Далее определяется цена реализации с учетом налога на добавленную стоимость (НДС), ставка НДС устанавливается законодательно Налоговым

Кодексом РК. На 2020 год ставка НДС установлена в размере 12%. Цена реализации с учетом НДС рассчитывается по формуле:

$$C_p = C_d + C_d \times \text{НДС}$$

$$C_p = 763416,96 + 763416,96 \times 0,12 = 855026,99 \text{ тенге} \quad (4.20)$$

Таким образом, себестоимость программного продукта составляет – 545297,83 тенге.

Прибыль составляет – 218119,13 тенге.

Договорная цена – 855026,99 тенге.

Рассмотрим вопрос эффективности внедрения и необходимости приобретения данной разработки с экономической стороны.

В связи с тем, что мой проект не влечет за собой огромные капиталовложения, следовательно, у него очень короткий срок окупаемости, что очень выгодно.

4.5 Сравнительный анализ эксплуатационных затрат до и после внедрения

Статьи затрат после внедрения программного обеспечения:

- основная заработная плата программиста и руководителя проекта;
- материальные ресурсы;
- амортизация основных фондов;
- накладные расходы.

Статьи затрат до внедрения:

- основная заработная плата 1 заведующего складом, 1 бухгалтера и 2 рабочих склада;
- материальные ресурсы;
- накладные расходы.

4.5.1 Расчет затрат до внедрения ПО

Таблица заработной платы и отчисляемых налогов представлена в таблице 4.2.8.

Таблица 4.2.8- Затраты на заработную плату

Сотрудники	Кол-во, чел	Заработная плата в месяц, тг	Заработная плата в год, тг	Социальное отчисление	Социальный налог	ВОС МС
Заведующий складом	1	180 000	2 160 000	68 040	112 536	43 200
Бухгалтер	1	160 000	1 920 000	60 480	100	38 400

р					032	
Рабочий скалада	2	100 000 (200 000)	1 200 000 (2 400 000)	37 800 (75 600)	62 250 (125 040)	24 000 (48 000)
К оплате	7 151 328					

Расчет затрат на необходимое для продукта ПО и оборудование представлен в таблице 4.2.9

Таблица 4.2.9 – Расчет затрат на оборудование и ПО, необходимое для проекта

Наименование материала	Марка	Ед. измерения	Количество	Цена за ед. в тенге	Сумма в тенге
Ноутбук	HP Pavilion 14-5GZ09EA	Штук	3	149000	447 000
Принтер	HP Laser Jet Pro M15a	Штук	1	39890	39890
Беспроводная мышь	Logitech M170	Штук	3	4 200	12600
Итого	499490				

Затраты, приуроченные к амортизационным отчислениям представлены в таблице 4.2.10

Таблица 4.2.10-Амортизационные отчисления

Оборудование	Количество	Стоимость, тенге	Отчисления
Ноутбук HP Pavilion 14-5GZ09EA	3	447 000	11 756,7
Принтер HP Laser Jet Pro M15a	1	39 890	21,83
Итого	4	486 890	11 778,5

Общие накладные расходы рассчитываются по формуле:

$$Z_{н.р.} = 7\,151\,328 * 0,2 = 1\,430\,265,6 \text{ тенге.}$$

Итого, общие эксплуатационные расходы после внедрения программного продукта будут равны,

$$3 = 7\,151\,328 + 499\,490 + 11\,778,5 + 1\,430\,265,6 = 9\,092\,852 \text{ тенге.}$$

4.5.2 Расчет затрат до внедрения ПО

Таблица заработной платы и отчисляемых налогов представлена в таблице 4.2.11

Таблица 4.2.11 - Затраты на заработную плату

Сотрудник	Кол-во, чел	Заработная плата в месяц одного, тг	Заработная плата в год, тг	Социальное отчисление	Социальный налог	ВОСМС
Программист	1	200 000	2 400 000	75 600	125 040	48 000
Руководитель	1	107 143	1 071 430	3375	5582,2	2 142,9
К оплате	2 718 883,1					

Расчет затрат на необходимое для продукта ПО и оборудование представлен в таблице 4.2.12

Таблица 4.2.12 – Расчет затрат на оборудование и ПО, необходимое для проекта

Наименование материала	Марка	Ед. измерения	Количество	Цена за ед. в тенге	Сумма в тенге
Ноутбук	HP Pavilion 14-5GZ09EA	Штук	1	149000	149 900
Принтер	HP Laser Jet Pro M15a	Штук	1	39890	39890
Компьютерная мышь беспроводная	Logitech M170	Штук	1	4 200	4200
Итого					193090

Затраты, приуроченные к амортизационным отчислениям представлены в таблице 4.2.13

Таблица 4.2.13 - Амортизационные отчисления

Оборудование	Количество	Стоимость, тенге	Отчисления
Ноутбук HP Pavilion 14- 5GZ09EA	1	149 000	3918,90
Принтер HP Laser Jet Pro M15a	1	39 890	21,83
Итого	4	188 890	3940,73

Общие накладные расходы рассчитываются по формуле:

$$З_{н.р.} = 2\,718\,883,1 * 0,2 = 543\,776,62 \text{ тенге.}$$

Итого, общие эксплуатационные расходы до внедрения программного продукта будут равны,

$$З = 2\,718\,883,1 + 193\,090 + 3940,73 + 543\,776,2 = 3\,459\,690 \text{ тенге.}$$

Для наглядности сведем все данные в аналитическую таблицу 4.3, в которой представлен сравнительный анализ годовых эксплуатационных затрат до и после внедрения программного продукта.

Таблица 4.3 - Годовые эксплуатационные затраты

Статьи	До внедрения ПП	После внедрения ПП
Годовая заработная плата	7 151 328	2 718 883,1
Оборудование	499 940	193 090
Амортизационные отчисления	11 778,5	3940,73
Накладные расходы	1 430 256,6	543 776,2
Всего	9 092 852	3 459 690

Годовая экономия денежных средств будет составлять:

$$9\,092\,852 - 3\,459\,690 = 5\,633\,152 \text{ тенге.}$$

Необходимо определить экономический эффект от внедрения программного продукта.

Технико-экономические показатели позволяют определить целесообразность проведения разработки и ее внедрения, а также оценить реальную выгоду, как для разработчика системы, так и для ее пользователя.

Можно определить срок окупаемости программного продукта в месяцах по формуле:

$$P_{\text{окуп}} = 12 * \frac{Z_{\text{разработка}}}{E_{\text{годовая}}} \quad (4.21)$$

где $P_{\text{окуп}}$ – период окупаемости в месяцах;

$Z_{\text{разр.}}$ – затраты на разработку программы;

$E_{\text{годовая}}$ – годовая экономия.

$$P_{\text{окуп}} = 12 * \frac{545\,297,83}{5\,633\,152} = 12 * 0,1358 = 1,1 \text{ мес.}$$

Зная годовую экономию и годовые эксплуатационные затраты с внедрением программного продукта можно определить коэффициент эффективности ($K_{\text{эф}}$) по формуле:

Расчетный коэффициент экономической эффективности капитальных вложений составляет:

$$E_p = \frac{E_{\text{уг}}}{K}$$

$$E_p = \frac{5\,633\,152}{545\,297,83} * 100\% = 1033\% \text{ или } 10,33 \text{ тенге.}$$

Коэффициент эффективности показывает сколько экономии в тенге приходится на каждый тенге затрат. Следовательно, на каждый тенге затрат приходится 10,33 тенге экономии. [18]

Сравнивая данные показатели видно, что внедрение программного продукта способствует прибыльности предприятия, поскольку внедрение способствует уменьшению кадров, что в свою очередь уменьшит выплаты заработной платы и налогов. Кроме того, уменьшится количество оборудования и амортизационных отчислений вместе с потребляемой энергией. Программный продукт сделает работу более автоматизированной, тем самым повлечет за собой уменьшение материальных расходов и увеличит производительность труда.

4.6 Вывод по экономической части

В данной работе рассматривается пример экономической оценки проекта по разработке программного продукта для автоматизации поддержки процессов закупа и учета хранения товаров на складе, в частности книг и сопутствующего товара. Приведен расчет затрат, связанных с закупкой оборудования и программного обеспечения, расчет эксплуатационных расходов: фонд оплаты труда, социальный налог, расходы на электроэнергию, амортизационные отчисления.

Стоимость разработки приложения составила 545297,83 тенге, в которую заложены все возможные затраты при разработке программного продукта. Это приложение имеет много применений. Данный склад может быть использован для учета любого вида и категорий товаров.

Разработка складского учета признана дорогим проектом, требующих больших интеллектуальных и финансовых затрат, а также обязательного использования компьютерной техники. Расчеты показали, что реализация проекта будет экономически целесообразна при условии высокого качества продукта и хорошего знания окончательного продукта, позволяя получить прибыль по истечению определенного времени, что позволяет сделать вывод, что производство и реализация данного программного продукта можно считать экономически выгодным, эффективным и рентабельным.

5 Безопасность жизнедеятельности

Пользователями программного обеспечения, разработанного для складского учёта являются служащий и научный руководитель, которые работают в офисе, поэтому необходимо обеспечить безопасное условие труда.

5.1 Анализ условий труда

Профессия программиста связана прежде всего с работой на компьютере, все профессиональные услуги выполняются только на компьютере. В этой главе описаны основные вредные факторы при работе с компьютером. Вредные факторы при работе с компьютером включают в себя: длительное сидение, вид нагрузки, боли в суставах рук, возможность респираторных заболеваний, аллергию, шум, напряжение глаз и другие. Рабочее место разработчиков должно обеспечивать комфортное положение для работы, обеспечивая высокую производительность труда. Работа над дипломным проектом велась в офисе, в котором работают 2 служащих. Офисные работники проводят много часов в день, сидя за столом, работая за компьютером, что приводит к эргономическим нагрузкам и другим травмам,

связанным с осанкой и повторяющимися движениями. Эти типы опасностей могут быть трудно обнаружить [19].

Разнообразные регулируемые стулья, столы, клавиатуры и т. д. должны предлагаться для удовлетворения самых разных стилей работы. Сотрудники должны быть осведомлены, как настроить и использовать настраиваемое оборудование для наилучшего соответствия рабочей станции.

По требованиям ГОСТ (ГОСТ 12.2.032-78. "Профессиональные стандарты безопасности. Рабочее место в сидячем положении. Общие эргономические условия"), при работе в сидячем положении были соблюдены ряд условия и поэтому организация исполняет свою деятельность на базе современных требования эргономики. Столы и кресла позволяют индивидуально настраивать высоту и угол соответственно росту человека, создавая удобное положение. Часто используемые вещи и устройства ввода (клавиатура и мышь) находятся в оптимальном рабочем диапазоне [18].

Требования к рабочему стулу пользователя персональным компьютером:

- стул должен быть снабжен поворотным и подъемным механизмом;
- высота должна регулироваться;
- 380 мм глубины сиденья;
- 400 мм ширины сиденья;
- регулировать угол наклона спинки стула до 90-110.

Согласно СН РК 2.04-02-2011, в помещении кабинетов вычислительных центров требуется применять комбинированную систему освещения.

Так как работа выполняется в офисе на ноутбуке с хорошим шумоподавлением, то работа с шумом исключается. Из этого исходит, что в офисе основная работа над дипломным проектом ведется без проблем с шумом. В офисе находится 3 светильника ЛД65 и имеется одно окно(3,5x1,5). Недостаток освещения неблагоприятно сказывается на здоровье и, как правило, на работе разработчика и в этой главе описаны основные вредные факторы при работе с компьютером, приведены расчеты. Также организовано рабочее место программиста. Также приведены расчеты по решению проблемы с вентиляцией.

5.2 Расчет естественного освещения

Недостаточная освещенность, пониженная контрастность могут привести к напряженности зрительного анализатора, что вследствие может повлечь ухудшение зрения.

В данном случае, под работой над проектом предполагается чтение, письмо, а также непосредственная работа с компьютером. Из этого следует, что освещенность, которая необходима для комфортного выполнения работы в данном помещении составляет 400 лк.

Необходимо произвести расчет площади боковых световых проемов офиса, которая предназначена для создания нормируемой освещенности на рабочих местах.

Тип помещения – офис.

Характеристики помещения:

длина $L = 8$ м, ширина $B = 4$ м, высота $H = 3$ м.

Высота рабочей поверхности над уровнем пола $h_{рп}=0,725$, окна начинаются с высоты $h_{рп}=0,725$ м, высота окна $h_0=2$ м.

Рабочее помещение находится в IV часовом поясе – в г. Алматы (пояс светового климата – IV северной широты и южнее).

Рабочие места расположены в $l_{рт} = 0,5$ м, от наружной стены помещения, где проектируются оконные проемы. Минимальная освещенность будет в точке, отстоящей на расстояние 4 м от оконного проема.

Общую площадь окон $S_0, м^2$, можно определить по формулам 5.1 и 5.2

$$100S_0 = e_n \times \eta_0 \times K_{зд} \times K_3 \quad (5.1)$$

$$S_0 = \frac{S_n \times e_n \times \eta_0 \times K_{зд} \times K_3}{\tau_0 \times r_1} \quad (5.2)$$

где S_n – площадь помещения; $S_n = 6 \times 4 = 24$ м²;

e_n – нормируемое значение КЕО;

K_3 – коэффициент запаса;

m – коэффициент светового климата;

Учитывая заданный световой пояс, ориентация световых проемов направлена на Север, определим по формуле 5.3.

$$e_x^{IV} = e_n \times m \times c \quad (5.3)$$

где $m = 0,7$;

$c = 0,75$ в наружных стенах зданий;

$e_n = 1,2$ для работ высокой точности III разряда зрительной работы;

$$e_x^{IV} = 1,2 \times 0,7 \times 0,75 = 0,63 \quad (5.4)$$

Учитывая тип помещения и находим коэффициент $k_3 = 1,2$ (учебные помещения, лаборатории, конструкторские бюро).

τ_0 – общий коэффициент светопропускания,

равный $\tau_0 = \tau_1 \times \tau_2 \times \tau_3 \times \tau_4$;

$\tau_1 = 0,5$ (пустотелые стеклянные блоки);

$\tau_2 = 0,6$ (деревянные двойные раздельные переплеты);

$\tau_3 = 0,8$ (железобетонные фермы и арки); $\tau_4 = 1$ (регулируемые жалюзи и шторы);

η_0 – световая характеристика окон.

$$\tau_0 = 0,5 \cdot 0,6 \cdot 0,8 \cdot 1 = 0,24 \quad (5.5)$$

Определим по формуле 5.4

$$L = B - 1$$

$$h_{\text{расч}} = h_{\text{но}} + h_0 - h_{\text{рп}} \quad (5.6)$$

$$L = 4 - 1 = 3 \text{ м}$$

$$\frac{L}{1} = \frac{L}{B-1} = \frac{6}{3} = 2$$

$$h_{\text{расч}} = 0,8 + 1,5 - 0,725 = 1,575$$

$$\frac{B}{h_{\text{расч}}} = \frac{4}{1,575} = 2,54$$

По таблице определим $\eta_0 = 10,5$.

r_1 – коэффициент, учитывающий повышение КЕО при боковом освещении благодаря свету, отраженному от поверхностей помещения и подстилающего слоя.

Средний коэффициент отражения в помещении $\rho_{\text{ср}} = 0,5$, принимаем одностороннее боковое освещение.

$$\frac{B}{h_{\text{расч}}} = \frac{0,5}{4} = 0,125$$

Тогда $r_1 = 1,05$.

$k_{\text{зд}}$ – коэффициент, учитывающий затенение окон противостоящими зданиями.

Поскольку затеняющих зданий поблизости нет, $k_{\text{зд}} = 1$. Вычислим общую площадь окон:

$$S_0 = \frac{24 \times 1,2 \times 10,5 \times 1 \times 0,63}{100 \times 0,24 \times 1,05} = 7,56 \text{ м}^2$$

Площадь световых проемов равна $S_{\text{сп}} = 7,56 \text{ м}^2$

$$l_{\text{ок}} = \frac{S_0}{h_{\text{ок}}} = \frac{7,56}{3} = 2,5 \quad (5.7)$$

Исходя из выше продемонстрированных расчетов, площадь световых проемов ($5,75 < 7,56 \text{ м}^2$) никак не способна обеспечить рабочие места необходимыми условиями труда.

Основной целью является создание благоприятных условий труда в помещении с площадью $S=24 \text{ м}^2$, следовательно естественное освещение должно использоваться наряду с искусственным освещением.

5.3 Расчет искусственного освещения

Разряд зрительной работы – V. Нормируемая освещённость – 400 лк. В качестве светильника был взят ЛЗЗ. Длина светильника 1514 мм, мощность – 65 Вт. В помещении имеется 2 светильника.

В таблице 5.3.1 изображены технические характеристики используемых светильников.

Схема освещенности представлена на рисунке 5.3.1.

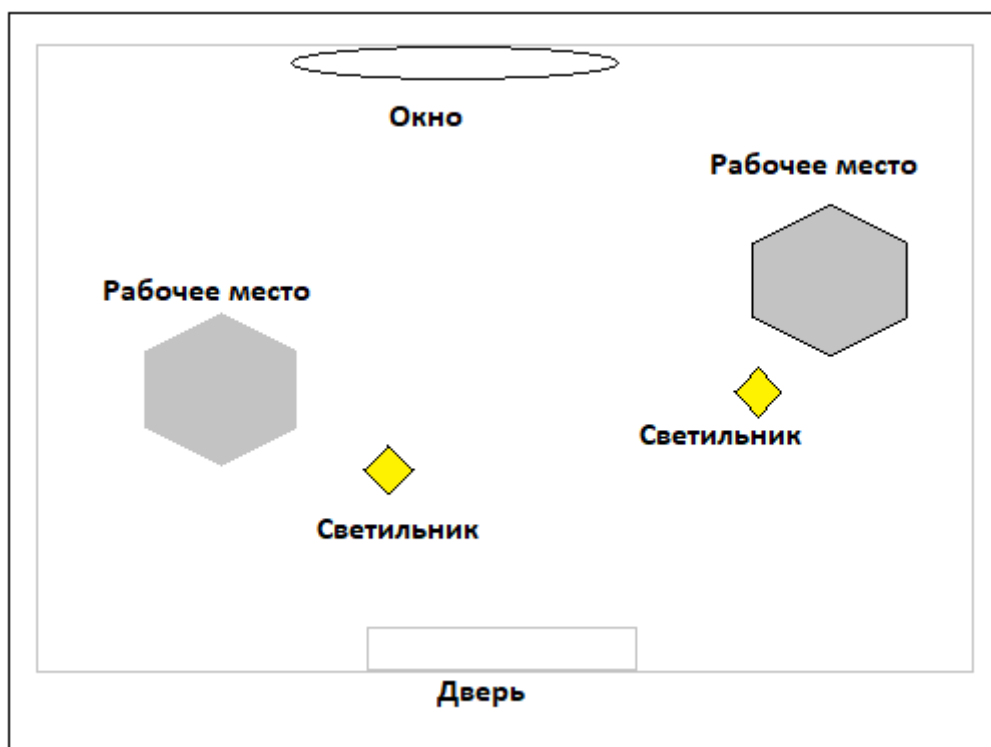


Рисунок 5.3.1 – Схема расчета освещенности

Таблица 5.3.1 – Технические характеристики ЛЗЗ

Тип лампы	Мощность, Вт	Напряжение, В	Световой поток, лм	Длина L, не более, мм	Диаметр D, мм	Тип цоколя
ЛЗЗ	65	110	3750	1514	38	G13d

Для начала идет проверка соответствия минимальной заданной освещенности при имеющихся 2 светильниках.

$$E = \frac{N \cdot n \cdot \Phi_{\text{л}} \cdot \eta}{k_z \cdot S_{\text{ос}} \cdot Z} \quad (5.8)$$

Где $S_{\text{ос}}$ – площадь помещения;

k_z – коэффициент запаса;

N – Количество светильников;

Z – коэффициент неравномерности освещения, $Z = 1,1$;

n – количество ламп в светильнике;

$\Phi_{\text{л}}$ – световой поток выбранной лампы, $\Phi_{\text{л}} = 3750$ лм

η – коэффициент использования, $\eta = 65\%$

$$E = \frac{2 \cdot 2 \cdot 3750 \cdot 0,65}{1,3 \cdot 24 \cdot 1,1} \approx 284 \text{ лк}$$

При 2 светильниках минимальная освещенность равняется примерно 284 лк, что не соответствует условиям труда. В следствие чего возникает необходимость, увеличить количество светильников для обеспечения приемлемой освещенности. Для этого необходимо произвести расчеты для реконструкции:

Коэффициенты отражения от потолка стен и пола соответственно равны:

$$\rho_{\text{пот}} = 70 \%$$

$$\rho_{\text{ст}} = 50 \%$$

$$\rho_{\text{пол}} = 30 \%$$

Вычислим высоту подвеса светильника над рабочей поверхностью по формуле 5.7:

$$H = h - h_p - h_c \quad (5.9)$$

Где, h_c – расстояние от светильника до перекрытия, $h_c = 0,1$ м;

h_p – высота рабочей поверхности над полом, $h_p = 0,725$ м;

h – высота помещения, $h = 3$ м.

$$H = 3 - 0,1 - 0,725 = 2,175 \text{ м}$$

Расстояние между рядами светильников:

$$L_b = \lambda \cdot H \quad (5.10)$$

$$L_b = 0,8 \cdot 2,175 = 1,74 \text{ м}$$

Расстояние между светильниками:

$$L_a = L_b - 0,1 = 1,74 - 0,1 = 1,64 \text{ м}$$

Расстояние от стены до ближайшего светильника, когда работа у стены не проводится, определяем по формуле 5.9

$$l_1 = (0,4 \div 0,8) \cdot L \quad (5.11)$$

$$l_1 = 0,6 \cdot 1,74 = 1,044 \text{ м}$$

Определяем индекс помещения по формуле 5.10

$$i = \frac{L \times B}{h_p \times (L+B)} \quad (5.12)$$
$$i = \frac{4 \times 2}{2.175 \times (4+2)} = 1,319$$

Коэффициент использования в данном случае равен $\eta = 65\%$, коэффициент запаса равен $k_z = 1,2$

Определим количество люминесцентных ламп по формуле 5.11

$$N = \frac{E \times k_z \times S_{oc} \times Z}{n \times \Phi_{л} \times \eta} \quad (5.13)$$

Где S_{oc} – площадь помещения;

k_z – коэффициент запаса;

E – заданная минимальная освещенность, $E = 400$ лк;

Z – коэффициент неравномерности освещения, $Z = 1,4$;

n – количество ламп в светильнике;

$\Phi_{л}$ – световой поток выбранной лампы, $\Phi_{л} = 3570$ лм;

η – коэффициент использования, $\eta = 65\%$.

$$N = \frac{400 \times 1,2 \times 24 \times 1,4}{2 \times 3570 \times 0,65} \approx 4$$

Для создания нормируемой освещенности 400 лк необходимо 4 люминесцентных лампы серии ЛД, мощность каждой лампы должна быть не

меньше 65 Вт, что соответствует действительности, а значит имеющегося в наличии освещения достаточно для соответствия санитарным нормам.

Для решения данной проблемы следует провести реконструкцию офиса и добавить 2 люминесцентных лампы серии ЛЗЗ для обеспечения необходимого освещения в помещении. На рисунке 5.3.2 изображен офис после реконструкции.

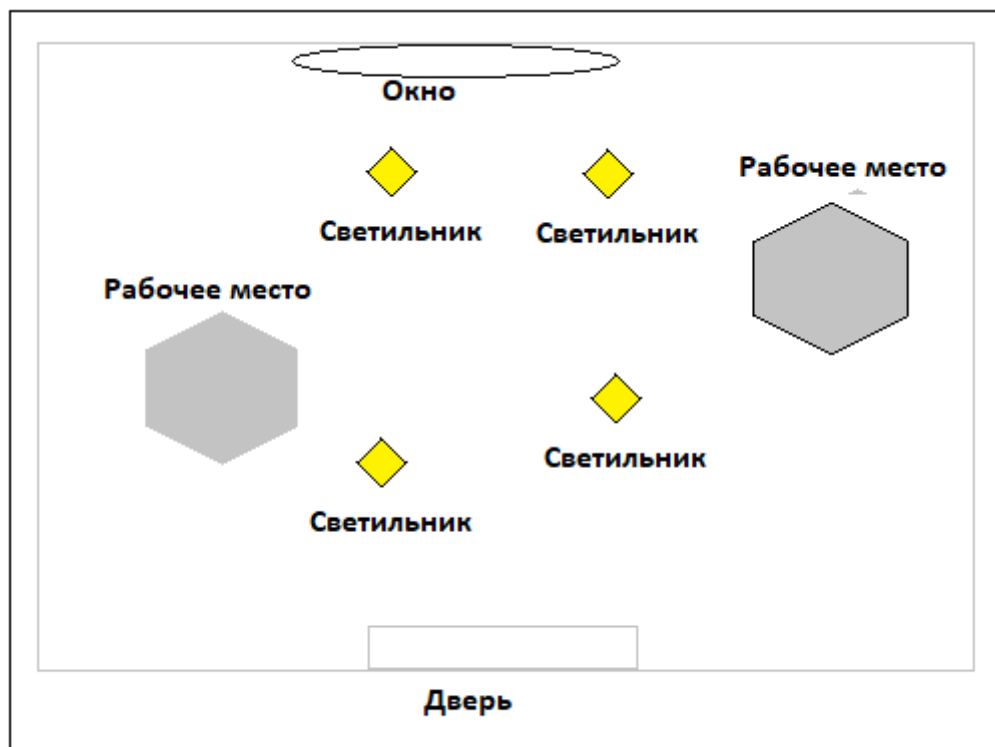


Рисунок 5.3.2 – Окончательный вариант расстановки светильников

5.4 Расчет вентиляции

Распространенность плохого качества воздуха в помещениях способствует росту профессиональной астмы и других респираторных расстройств, химической чувствительности и аллергии. Некоторыми из причин плохого качества воздуха являются неадекватные системы вентиляции; переполненность офиса; наличие чистящих химикатов и пестицидов; повреждение водой и рост плесени; конструкция кабины, которая блокирует поток воздуха в рабочие зоны; слишком большая или слишком низкая влажность; и плохое ведение домашнего хозяйства, что приводит к грязной рабочей среде. Качество воздуха в офисе может быть значительно улучшено путем надлежащего технического обслуживания, очистки и фильтрации системы вентиляции, отопления и кондиционирования воздуха. Это поможет уменьшить респираторные раздражители, инфекции и болезни. Температура воздуха не должна быть отделена друг от друга на 5 градусов на уровне эскиза лица и уровня головы человека. Основным параметром,

который характеризует характеристики системы вентиляции, является многократное вращение, которым является количество изменений воздуха в помещении. Оптимальные и допускаемые общепризнанных мерок характеристик локального климата в помещении оператора показаны в таблице 5.4.1

Таблица 5.4.1 – Допустимые и оптимальные нормы

Период года	Температура воздуха, С	Относительная влажность воздуха	Скорость движения воздуха
Холодный	22 - 24	40 -60	0,1
Теплый	23 - 25	40 -60	0,1

Мы рассчитываем кратность воздушного потока по формуле

$$K = V_B / V_k , \quad (5.14)$$

где V_B - объем воздуха, подаваемого в помещение
 V_k - объем помещения

Мы определяем объем воздуха, необходимый для замены теплового баланса V-баланса.

$$V_B = (3600 * Q_t) / C * (t_o - t_p) * Y, \quad (5.15)$$

где Q_t - избыток тепла;
 $C = 1000$ - парциальная теплопроводность воздуха;
 $Y = 1,2$ - плотность воздуха.

Температура отработанного воздуха рассчитывается по формуле

$$t_o = t_{км} + (H-2) * t, \quad (5.16)$$

где t - изменение температуры в помещении на высоте 1 метра, принимает значения от 1 до 5 градусов

t_p - температура на рабочем месте, 23 градуса
 H - высота помещения -4 метра
 $t_{км}$ – температура воздуха, равная 18 градусам,

Температура вытяжного воздуха по формуле 5.17 составляет:

$$T_o = 23 + (3 - 2) \cdot 2 = 25$$

Рассчитать избыток тепла по формуле

$$Q_t = Q_{t1} + Q_{t2} + Q_{t3} \quad (5.17)$$

где Q_{t1} - избыток тепла от электрооборудования и освещения, Вт;

Q_{t2} - тепло от солнечного света, Вт;

Q_{t3} - теплопередача человека, Вт.

Рассчитаем избыточное тепло от электрооборудования и освещения по формуле 5.18

$$Q_{t1} = E * P, \quad (5.18)$$

где E - коэффициент рассеяния энергии рассеяния тепла;
 P - мощность электрооборудования, Вт.

Коэффициент энергопотребления принимает следующие значения:

$E = 0,55$ для освещения, $E = 0,4$ для персонального компьютера.

Рассчитаем общую мощность осветительного и электрооборудования по следующей формуле 5.19

$$P = P_{pcs} * K, \quad (5.19)$$

где P_{pcs} - измерение мощности одного элемента оборудования, Вт;
 K - количество единиц оборудования.

Таким образом, по формуле 5.19:

$$P_{\text{освещения}} = 150 * 3 = 450 \text{ Вт},$$
$$P_{\text{компьютеров и т. Д.}} = 600 * 2 + 700 = 1900 \text{ Вт}.$$

Мы рассчитываем избыток тепла от электрооборудования и освещения по формуле 5.17:

$$Q_{t1} = 0,55 * 450 + 0,4 * 1900 = 1007,5 \text{ Вт}.$$

Рассчитайте избыточное тепло от искусственного освещения по следующей формуле 5.20

$$Q_{t2} = m * S * k * Q_t \quad (5.20),$$

где m - количество окон;

S - площадь одного окна, м²;

k - коэффициент, учитывающий остекление.

Коэффициент двойного остекления $k = 0,8$;

Q_t - тепло от окна, равное 127 Вт / м.

Рассчитать теплоту сверх искусственного освещения по формуле 5.7:

$$Q_{t2} = 1 * 2 * 0,8 * 127 = 203 \text{ Вт.}$$

Рассчитать теплопередачу человека по следующей формуле

$$Q_{tч} = n * q, \quad (5.21)$$

где q - теплоотдача одного человека, 80; n - количество человек, равное 1.

Рассчитываем теплопередачу человека по формуле 5.21:

$$Q_{tч} = 180 = 80 \text{ Вт.}$$

Рассчитываем избыток тепла по формуле 5.17:

$$Q_t = 1007,5 + 203 + 80 = 1290,5 \text{ Вт.}$$

Таким образом, рассчитываем объем воздуха, подлежащий переносу по формуле 5.15:

$$V_B = 3600 * 1290,5 / 1000 * (25-18) * 1,2 = 553,07.$$

Из этого следует, что для создания оптимальных рабочих условий необходимо установить кондиционер. Мой выбор пал на кондиционер Royal Clima RC-E25HN . Этот кондиционер быстро охлаждает и нагревает воздух до желаемой температуры, удаляет из воздуха вредные примеси: аммиак, формальдегид, сероводород. Антиаллергенный фильтр с активированным углем улавливает мельчайшие частицы пыли и удаляет их. Есть функции автоматического режима и осушения воздуха. Технические характеристики устройства приведены в таблице 5.4.2.

Таблица 5.4.2 – Технические характеристики

Мощность в режиме охлаждения	2.50 кВт
Мощность в режиме обогрева	2.62 кВт
Расход воздуха (внутренний блок)	480 куб.м/ч
Min уровень шума	32 дБ(А)
Антиаллергенный фильтр	Есть
Срок службы	10 лет

5.5 Вывод по разделу безопасность жизнедеятельности

В данном разделе был произведён анализ условий труда в офисе и выполнен расчет искусственного и естественного освещения, а также

вентиляции. Уровень условий труда признан допустимым, и данные, полученные из расчетов, удовлетворяют требованиям стандартов безопасности жизнедеятельности.

Заключение

Преимущества автоматизации управления складом не только экономические. Автоматизированное складирование может обеспечить широкий спектр преимуществ, начиная от экономии места, снижения затрат на строительство, повышения производительности, повышения эффективности материальных потоков, сокращения персонала и обеспечения безопасности операций, до сокращения запасов, снижения эксплуатационных расходов, повышения рентабельности инвестиций и снижения затрат в течение жизненного цикла. Преимущества заключаются в том, что так много предприятий все больше внедряют автоматизацию и робототехнику. Инструменты автоматизации могут упростить весь процесс, от выбора и получения предметов до их проталкивания по сборочным линиям и распространения. Инструменты могут организовывать продукты, управлять штрих-кодами и отправлять автоматические оповещения заинтересованным сторонам, когда конкретные задачи выполнены. Это позволяет сотрудникам склада тратить меньше времени на организацию и управление различными частями своих процессов. Автоматизация складов снижает потребность в ручном труде и повышает гибкость в работе.

Стоимость автоматизации является основной причиной того, что все склады не автоматизированы. Первоначальные затраты очень высоки, поэтому ограничивают инвестиции, необходимые для внедрения автоматизации, но в долгосрочной перспективе это окупается, что заставляет многих операторов складских помещений переходить на решение такого типа. Его преимущества, обобщенные в повышении производительности и эффективности, обещают снижение эксплуатационных и трудовых затрат и повышение прибыльности. Его принятие, хотя и медленное, станет важным для компаний с распределительными центрами, которые стремятся повысить свою производительность и конкурентоспособность.

В данной работе рассмотрено создание приложения, которое применимо к любому складу и каждый желающий может модифицировать его под себя. Также произведены расчеты освещения помещения и вентиляции, необходимые для комфортных условий труда.

Список литературы

- 1 "Web programming for everybody". Web Developers Official Website. May 15, 2019. Retrieved May 30, 2019.
- 2 "Warehouse storage solutions". Warehouse Manual. January 20, 2020. Retrieved April 3, 2020.
- 3 Alan Forbes, How Much Do Average Apps Make.
- 4 Alan Forbes, The Joy of PHP programming
- 5 "Download Options". php.net. Google. Retrieved 08 February 2019.
- 6 "PHP&MySQL Novice to Ninja", Tom Butler, Kevin Yank, 2014.
- 7 «MySQL по максимуму», Шварц Б., Зайцев П., Питер, 2018, 864 с.
- 8 "MySQL. Справочник по языку", Дэйв К., MySQL Press, 2010, 432 с.
- 9 Святослав Куликов. "Работа с MySQL, MS SQL Server и Oracle в примерах" /Самиздат, 2017, 547 с.
- 10 "PHP: A beginner's guide ", Vikram Vaswani.
- 11 Волгин В. "Склад. Стандарты управления: Практическое пособие. ".
- 12 "Логистика. Персонал, технологии, практика", Евгений Панасенко.
- 13 "Google Launches Android Studio And New Features For Developer Console, Including Beta Releases And Staged Rollout". VentureBeat. December 8, 2014. Retrieved December 9, 2014.
- 14 "Planet PHP. PHP Developers Blog. Google. Retrieved June 21, 2018.
- 15 "Frontend Focus". Web Developers. Google. Retrieved May 30, 2015.
- 16 Боканова Г.Ш. – Методические указания по выполнению экономической части дипломных работ для студентов специальностей 5В070400 – «Вычислительная техника и программное обеспечение», 5В060200 – «Информатика», 5В070300 – «Информационные системы». Алматы, АУЭС, 2020. - 35 с.
- 17 "Economic Analysis on cost-effectiveness". Google.
- 18 СНиП РК 2.02-05-2019 Пожарная безопасность.
- 19 Санатова Т.С. Безопасность жизнедеятельности. – Учебное пособие для всех специальностей, 2017. - 407 с.
- 20 "HTML and CSS". Google. Google. May 25 2012. Retrieved August 19, 2014

Приложение А

Техническое задание

1. Основные требования:

- название разрабатываемого программного обеспечения: поддержка процессов закупа и учета хранения товаров на складе.

- цель разработки: реализация программного продукта, способного автоматизировать движение товаров на складе (учет, хранение и закуп);

Возможного платформы для разработки программы (на выбор разработчика):

- PHP;
- Python;
- Perl;
- Java;
- ASP.NET;
- MySQL;

общий объем программной части системы: не более 50 Мб.

2. Технические требования:

- платформа работы приложения: PHP;

3. Экономические требования.

Расчет сметы разработки программного продукта (подлежит обсуждению):

- стоимость готового продукта 855026,99 тг;
- стоимость разработки 545297,83;

Аудитория направлена на компании, стремящиеся эффективно управлять складом.

Приложение Б

Листинг программы

```
<?php
if (session_id() == "") session_start(); // Initialize Session data
ob_start(); // Turn on output buffering
?>
<?php include_once "ewcfg12.php" ?>
<?php include_once ((EW_USE_ADODB) ? "adodb5/adodb.inc.php" :
"ewmysql12.php") ?>
<?php include_once "phpfn12.php" ?>
<?php include_once "a_customersinfo.php" ?>
<?php include_once "usersinfo.php" ?>
<?php include_once "a_salesgridcls.php" ?>
<?php include_once "userfn12.php" ?>
<?php

// Page class
$a_customers_add = NULL; // Initialize page object first

class ca_customers_add extends ca_customers {

    // Page ID
    var $PageID = 'add';

    // Project ID
    var $ProjectID = "{B36B93AF-B58F-461B-B767-5F08C12493E9}";

    // Table name
    var $TableName = 'a_customers';

    // Page object name
    var $PageObjName = 'a_customers_add';

    // Page name
    function PageName() {
        return ew_CurrentPage();
    }

    // Page URL
    function PageUrl() {
        $PageUrl = ew_CurrentPage() . "?";
    }
}
```

```
        if ($this->UseTokenInUrl) $PageUrl .= "t=" . $this->TableVar .
"&"; // Add page token
        return $PageUrl;
    }

    // Message
    function getMessage() {
```


Продолжение приложения Б

```
function getFailureMessage() {
    return @$_SESSION[EW_SESSION_FAILURE_MESSAGE];
}

function setFailureMessage($v) {

ew_AddMessage($_SESSION[EW_SESSION_FAILURE_MESSAGE], $v);
}

function getSuccessMessage() {
    return @$_SESSION[EW_SESSION_SUCCESS_MESSAGE];
}

function setSuccessMessage($v) {

ew_AddMessage($_SESSION[EW_SESSION_SUCCESS_MESSAGE], $v);
}

function getWarningMessage() {
    return @$_SESSION[EW_SESSION_WARNING_MESSAGE];
}

function setWarningMessage($v) {

ew_AddMessage($_SESSION[EW_SESSION_WARNING_MESSAGE],
$v);
}

// Methods to clear message
function ClearMessage() {
    $_SESSION[EW_SESSION_MESSAGE] = "";
}

function ClearFailureMessage() {
    $_SESSION[EW_SESSION_FAILURE_MESSAGE] = "";
}

function ClearSuccessMessage() {
    $_SESSION[EW_SESSION_SUCCESS_MESSAGE] = "";
}

function ClearWarningMessage() {
    $_SESSION[EW_SESSION_WARNING_MESSAGE] = "";
}
```

```
}  
  
function ClearMessages() {  
    $_SESSION[EW_SESSION_MESSAGE] = "";  
    $_SESSION[EW_SESSION_FAILURE_MESSAGE] = "";  
    $_SESSION[EW_SESSION_SUCCESS_MESSAGE] = "";  
    $_SESSION[EW_SESSION_WARNING_MESSAGE] = "";  
}
```

Продолжение приложения Б

```
function ShowMessage() {

    // $hidden = TRUE;
    $hidden = MS_USE_JAVASCRIPT_MESSAGE;
    $html = "";

    // Message
    $sMessage = $this->getMessage();
    $this->Message_Showing($sMessage, "");
    if ($sMessage <> "") { // Message in Session, display
        if (!$hidden)
            $sMessage = "<button type=\"button\" class=\"close\"
data-dismiss=\"alert\">&times;</button>" . $sMessage;
            $html .= "<div class=\"alert alert-info ewInfo\">" . $sMessage .
"</div>";
            $_SESSION[EW_SESSION_MESSAGE] = ""; // Clear message
in Session
        }

    // Warning message
    $sWarningMessage = $this->getWarningMessage();
    $this->Message_Showing($sWarningMessage, "warning");
    if ($sWarningMessage <> "") { // Message in Session, display
        if (!$hidden)
            $sWarningMessage = "<button type=\"button\"
class=\"close\" data-dismiss=\"alert\">&times;</button>" . $sWarningMessage;
            $html .= "<div class=\"alert alert-warning ewWarning\">" .
$sWarningMessage . "</div>";
            $_SESSION[EW_SESSION_WARNING_MESSAGE] = ""; //
Clear message in Session
        }

    // Success message
    $sSuccessMessage = $this->getSuccessMessage();
    $this->Message_Showing($sSuccessMessage, "success");
    if ($sSuccessMessage <> "") { // Message in Session, display

        // if (!$hidden)
        //     $sSuccessMessage = "<button type=\"button\"
class=\"close\" data-dismiss=\"alert\">&times;</button>" . $sSuccessMessage;
        // $html .= "<div class=\"alert alert-success ewSuccess\">" .
$sSuccessMessage . "</div>";
    }
}
```

// Begin of modification Auto Hide Message, by Masino Sinaga,
January 24, 2013

```
if (@MS_AUTO_HIDE_SUCCESS_MESSAGE) {

    // $sSuccessMessage = "<button type=\"button\"
class=\"close\" data-dismiss=\"alert\">&times;</button>";
    $html .= "<p class=\"alert alert-success
msSuccessMessage\" id=\"ewSuccessMessage\">" . $sSuccessMessage . "</p>";
    } else {

        Продолжение приложения Б
        $_SESSION[EW_SESSION_SUCCESS_MESSAGE] = ""; // Clear message in
        Session
    }

    // Failure message
    $sErrorMessage = $this->getFailureMessage();
    $this->Message_Showing($sErrorMessage, "failure");
    if ($sErrorMessage <> "") { // Message in Session, display
        if (!$hidden)
            $sErrorMessage = "<button type=\"button\"
class=\"close\" data-dismiss=\"alert\">&times;</button>" . $sErrorMessage;
            $html .= "<div class=\"alert alert-danger ewError\">" .
            $sErrorMessage . "</div>";
            $_SESSION[EW_SESSION_FAILURE_MESSAGE] = ""; //
            Clear message in Session
        }

        // echo "<div class=\"ewMessageDialog\" . (($hidden) ? "
style=\"display: none;\" : \"\") . ">" . $html . "</div>";
        if (@MS_AUTO_HIDE_SUCCESS_MESSAGE ||
MS_USE_JAVASCRIPT_MESSAGE==0) {
            echo $html;
        } else {
            if (MS_USE_ALERTIFY_FOR_MESSAGE_DIALOG) {
                if ($html <> "") {
                    $html = str_replace("", "\", $html);
                    echo "<script
type='text/javascript'>alertify.alert(\".$html.\" , function (ok) { }).set('title',
ewLanguage.Phrase('AlertifyAlert'));</script>";
                }
            } else {
                echo "<div class=\"ewMessageDialog\" . (($hidden) ? "
style=\"display: none;\" : \"\") . ">" . $html . "</div>";
            }
        }
    }
}
```

```

        }
    }
}
var $PageHeader;
var $PageFooter;

// Show Page Header
function ShowPageHeader() {
    $sHeader = $this->PageHeader;
    $this->Page_DataRendering($sHeader);
    if ($sHeader <> "") { // Header exists, display
        echo "<p>" . $sHeader . "</p>";
    }
}

// Show Page Footer
function ShowPageFooter() {
    $sFooter = $this->PageFooter;
    $this->Page_DataRendered($sFooter);
    if ($sFooter <> "") { // Footer exists, display
        Продолжение приложения Б
    }
}

// Validate page request
function IsPageRequest() {
    global $objForm;
    if ($this->UseTokenInUrl) {
        if ($objForm)
            return ($this->TableVar == $objForm-
>GetValue("t"));
        if (@$_GET["t"] <> "")
            return ($this->TableVar == $_GET["t"]);
    } else {
        return TRUE;
    }
}

var $Token = "";
var $TokenTimeout = 0;
var $CheckToken = EW_CHECK_TOKEN;
var $CheckTokenFn = "ew_CheckToken";
var $CreateTokenFn = "ew_CreateToken";

// Valid Post
function ValidPost() {
    if (!$this->CheckToken || !ew_IsHttpPost())
        return TRUE;
    if (!isset($_POST[EW_TOKEN_NAME]))

```

```

        return FALSE;
    $fn = $this->CheckTokenFn;
    if (is_callable($fn))
        return $fn($_POST[EW_TOKEN_NAME], $this-
>TokenTimeout);
    return FALSE;
}

// Create Token
function CreateToken() {
    global $gsToken;
    if ($this->CheckToken) {
        $fn = $this->CreateTokenFn;
        if ($this->Token == "" && is_callable($fn)) // Create
token
            $this->Token = $fn();
            $gsToken = $this->Token; // Save to global variable
    }
}

//
// Page class constructor
//
function __construct() {
    global $conn, $Language;
    global $UserTable, $UserTableConn;
    $GLOBALS["Page"] = &$this;
    $this->TokenTimeout = ew_SessionTimeoutTime();
}

```

Продолжение приложения Б

```

$UserProfile = new cUserProfile();

// Security
$Security = new cAdvancedSecurity();
if (IsPasswordExpired())
    $this->Page_Terminate(ew_GetUrl("changepwd.php"));
if (!$Security->IsLoggedIn()) $Security->AutoLogin();
if ($Security->IsLoggedIn()) $Security->TablePermission_Loading();
$Security->LoadCurrentUserLevel($this->ProjectID . $this-
>TableName);
if ($Security->IsLoggedIn()) $Security->TablePermission_Loaded();
if (!$Security->CanAdd()) {
    $Security->SaveLastUrl();
}

```

```

        $this->setFailureMessage($Language-
>Phrase("NoPermission")); // Set no permission
        if ($Security->CanList())
            $this-
>Page_Terminate(ew_GetUrl("a_customerslist.php"));
        else
            $this->Page_Terminate(ew_GetUrl("login.php"));
    }

    // Begin of modification Auto Logout After Idle for the Certain Time,
by Masino Sinaga, May 5, 2012
    if (IsLoggedIn() && !IsSysAdmin()) {

        // Begin of modification by Masino Sinaga, May 25, 2012 in
order to not autologout after clear another user's session ID whenever back to
another page.
        $UserProfile->LoadProfileFromDatabase(CurrentUserName());

        // End of modification by Masino Sinaga, May 25, 2012 in order
to not autologout after clear another user's session ID whenever back to another
page.

        // Begin of modification Save Last Users' Visitted Page, by
Masino Sinaga, May 25, 2012

        $lastpage = ew_CurrentPage();
        if ($lastpage!='logout.php' && $lastpage!='index.php') {
            $lasturl = ew_CurrentUrl();
            $sFilterUserID = str_replace("%u",
ew_AdjustSql(CurrentUserName(), EW_USER_TABLE_DBID),
EW_USER_NAME_FILTER);
            ew_Execute("UPDATE ".EW_USER_TABLE." SET
Current_URL = ".$lasturl." WHERE ".$sFilterUserID.", $UserTableConn);
        }

        // End of modification Save Last Users' Visitted Page, by Masino
Sinaga, May 25, 2012
        $LastAccessDateTime = strval(@$UserProfile-
>Profile[EW_USER_PROFILE_LAST_ACCESSED_DATE_TIME]);
        $nDiff = intval(ew_DateDiff($LastAccessDateTime,
ew_StdCurrentDateTime(), "s"));
        $nCons =
intval(MS_AUTO_LOGOUT_AFTER_IDLE_IN_MINUTES) * 60;
        Продолжение приложения Б
        if ($UserProfile->IsValidUser(CurrentUserName(), session_id())) {

```

```

        // Do nothing since it's a valid user! SaveProfileToDatabase has
        been handled from IsValidUser method of UserProfile object.
        } else {

            // Begin of modification How to Overcome "User X already
            logged in" Issue, by Masino Sinaga, July 22, 2014
            // echo $Language->Phrase("UserProfileCorrupted");

            header("Location: logout.php");

            // End of modification How to Overcome "User X already logged
            in" Issue, by Masino Sinaga, July 22, 2014
        }
        if (@MS_USE_CONSTANTS_IN_CONFIG_FILE == FALSE) {

            // Call this new function from userfn*.php file
            My_Global_Check();
        }

        // Create form object
        $objForm = new cFormObj();
        $this->CurrentAction = (@$_GET["a"] <> "") ? $_GET["a"] :
        @$_POST["a_list"]; // Set up current action

        // Global Page Loading event (in userfn*.php)
        Page_Loading();

// Begin of modification Disable/Enable Registration Page, by Masino Sinaga, May
14, 2012
// End of modification Disable/Enable Registration Page, by Masino Sinaga, May
14, 2012

        // Page Load event

        $this->Page_Load();

        // Check token
        if (!$this->ValidPost()) {
            echo $Language->Phrase("InvalidPostRequest");
            $this->Page_Terminate();
            exit();
        }
        if (ALWAYS_COMPARE_ROOT_URL == TRUE) {
            if ($_SESSION['php_stock_Root_URL'] <> Get_Root_URL()) {

```



```

                header("Location: " .
$_SESSION['php_stock_Root_URL']);
            }
        }
// Process auto fill
    if (@$_POST["ajax"] == "autofill") {

        // Process auto fill for detail table 'a_sales'
        Продолжение приложения Б
        if (!isset($GLOBALS["a_sales_grid"])) $GLOBALS["a_sales_grid"] = new
ca_sales_grid;

                $GLOBALS["a_sales_grid"]->Page_Init();
                $this->Page_Terminate();
                exit();
            }
$results = $this->GetAutoFill(@$_POST["name"], @$_POST["q"]);
    if ($results) {

        // Clean output buffer
        if (!EW_DEBUG_ENABLED && ob_get_length())
            ob_end_clean();
        echo $results;
        $this->Page_Terminate();
        exit();
    }
}

// Create Token
$this->CreateToken();
}

//
// Page_Terminate
//
function Page_Terminate($url = "") {
    global $gsExportFile, $gTmpImages;

    // Page Unload event
    $this->Page_Unload();

    // Global Page Unloaded event (in userfn*.php)
    Page_Unloaded();

    // Export

```

```

        global $EW_EXPORT, $a_customers;
        if ($this->CustomExport <> "" && $this->CustomExport ==
$this->Export && array_key_exists($this->CustomExport, $EW_EXPORT)) {
            $sContent = ob_get_contents();
            if ($gsExportFile == "") $gsExportFile = $this->TableVar;
            $class = $EW_EXPORT[$this->CustomExport];
            if (class_exists($class)) {
                $doc = new $class($a_customers);

resqIv[curq].setBackground(getResources().getDrawable(R.drawable.qcorrect));
        $doc->Text = $sContent;
            if ($this->Export == "email")
                echo $this->ExportEmail($doc->Text);
            else
                $doc->Export();
            ew_DeleteTmpImages(); // Delete temp images
            exit();
        }
        Продолжение приложения Б
$this->Page_Redirecting($url);

```

```

        // Close connection
        ew_CloseConn();

        // Go to URL if specified
        if ($url <> "") {
            if (!EW_DEBUG_ENABLED && ob_get_length())
                ob_end_clean();
            header("Location: " . $url);
        }
        exit();
    }
    var $FormClassName = "form-horizontal ewForm ewAddForm";
    var $DbMasterFilter = "";
    var $DbDetailFilter = "";
    var $StartRec;
    var $Priv = 0;
    var $OldRecordset;
    var $CopyRecord;

    //
    // Page main
    //
    function Page_Main() {
        global $objForm, $Language, $gsFormError;

```

```

// Process form if post back
if (@$_POST["a_add"] <> "") {
    $this->CurrentAction = $_POST["a_add"]; // Get form action
    $this->CopyRecord = $this->LoadOldRecord(); // Load old
recordset
    $this->LoadFormValues(); // Load form values

    // End of modification Permission Access for Export To Feature,
by Masino Sinaga, May 5, 2012
    } else { // Not post back

        // Load key values from QueryString
        $this->CopyRecord = TRUE;
        if (@$_GET["Customer_ID"] != "") {
            $this->Customer_ID-
>setQueryStringValue($_GET["Customer_ID"]);
this->setKey("Customer_ID", $this->Customer_ID->CurrentValue); // Set up key
        } else {
            $this->setKey("Customer_ID", ""); // Clear key
            $this->CopyRecord = FALSE;
        }
        if ($this->CopyRecord) {
            $this->CurrentAction = "C"; // Copy record
        }
}

```

Продолжение приложения Б

```

$this->Customer_Number->CurrentValue = NULL;
    $this->Customer_Number->OldValue = $this->Customer_Number-
>CurrentValue;
    $this->Customer_Name->CurrentValue = NULL;
    $this->Customer_Name->OldValue = $this->Customer_Name-
>CurrentValue;
    $this->Address->CurrentValue = NULL;
    $this->Address->OldValue = $this->Address->CurrentValue;
    $this->City->CurrentValue = NULL;
    $this->City->OldValue = $this->City->CurrentValue;
    $this->Country->CurrentValue = NULL;
    $this->Country->OldValue = $this->Country->CurrentValue;
    $this->Contact_Person->CurrentValue = NULL;
    $this->Contact_Person->OldValue = $this->Contact_Person-
>CurrentValue;
    $this->Phone_Number->CurrentValue = NULL;
    $this->Phone_Number->OldValue = $this->Phone_Number-
>CurrentValue;

```

```

$this->_Email->CurrentValue = NULL;
$this->_Email->OldValue = $this->_Email->CurrentValue;
$this->Mobile_Number->CurrentValue = NULL;
$this->Mobile_Number->OldValue = $this->Mobile_Number-
>CurrentValue;
$this->Notes->CurrentValue = NULL;
$this->Notes->OldValue = $this->Notes->CurrentValue;
$this->Date_Added->CurrentValue = ew_CurrentDateTime();
$this->Added_By->CurrentValue = CurrentUserName();
$this->Date_Updated->CurrentValue = NULL;
$this->Date_Updated->OldValue = $this->Date_Updated-
>CurrentValue;
$this->Updated_By->CurrentValue = NULL;
$this->Updated_By->OldValue = $this->Updated_By->CurrentValue;
}

```

// Load form values

```

function LoadFormValues() {

    // Load from form
    global $objForm;
    if (!$this->Customer_Number->FldIsDetailKey) {
        $this->Customer_Number->setFormValue($objForm-
>GetValue("x_Customer_Number"));
    }
    if (!$this->Customer_Name->FldIsDetailKey) {
        $this->Customer_Name->setFormValue($objForm-
>GetValue("x_Customer_Name"));
    }
    if (!$this->Address->FldIsDetailKey) {
        $this->Address->setFormValue($objForm-
>GetValue("x_Address"));
    }
    if (!$this->City->FldIsDetailKey) {
        $this->City->setFormValue($objForm->GetValue("x_City"));
    }
    if (!$this->Country->FldIsDetailKey) {
        $this->Country->setFormValue($objForm-
>GetValue("x_Country"));
    }
}

```

Продолжение приложения Б

```

$this->Contact_Person->setFormValue($objForm-
>GetValue("x_Contact_Person"));
}

```

```

        if (!$this->Phone_Number->FldIsDetailKey) {
            $this->Phone_Number->setFormValue($objForm-
>GetValue("x_Phone_Number"));
        }
        if (!$this->_Email->FldIsDetailKey) {
            $this->_Email->setFormValue($objForm-
>GetValue("x__Email"));
        }
        if (!$this->Mobile_Number->FldIsDetailKey) {
            $this->Mobile_Number->setFormValue($objForm-
>GetValue("x_Mobile_Number"));
        }
        if (!$this->Notes->FldIsDetailKey) {
            $this->Notes->setFormValue($objForm->GetValue("x_Notes"));
        }
        if (!$this->Date_Added->FldIsDetailKey) {
            $this->Date_Added->setFormValue($objForm-
>GetValue("x_Date_Added"));
            $this->Date_Added->CurrentValue =
ew_UnFormatDateTime($this->Date_Added->CurrentValue, 0);
        }
        if (!$this->Added_By->FldIsDetailKey) {
            $this->Added_By->setFormValue($objForm-
>GetValue("x_Added_By"));
        }
        if (!$this->Date_Updated->FldIsDetailKey) {
            $this->Date_Updated->setFormValue($objForm-
>GetValue("x_Date_Updated"));
            $this->Date_Updated->CurrentValue =
ew_UnFormatDateTime($this->Date_Updated->CurrentValue, 0);
        }
        if (!$this->Updated_By->FldIsDetailKey) {
            $this->Updated_By->setFormValue($objForm-
>GetValue("x_Updated_By"));
        }
    }

    // Restore form values
    function RestoreFormValues() {
        global $objForm;
        $this->LoadOldRecord();
        $this->Customer_Number->CurrentValue = $this->Customer_Number-
>FormValue;
    }

```

```

    $this->Customer_Name->CurrentValue = $this->Customer_Name-
>FormValue;
    $this->Address->CurrentValue = $this->Address->FormValue;
    $this->City->CurrentValue = $this->City->FormValue;
    $this->Country->CurrentValue = $this->Country->FormValue;
    $this->Contact_Person->CurrentValue = $this->Contact_Person-
>FormValue;
    $this->Phone_Number->CurrentValue = $this->Phone_Number-
>FormValue;
    $this->_Email->CurrentValue = $this->_Email->FormValue;

```

Продолжение приложения Б

```

?>
<script type="text/javascript">

// Write your table-specific startup script here
// document.write("page loaded");

</script>
<?php if (MS_ENTER_MOVING_CURSOR_TO_NEXT_FIELD) { ?>
<script type="text/javascript">
    $(document).ready(function(){ $("#fa_customersadd:first
*:input[type!=hidden]:first").focus(),$("#input").keydown(function(i){ if(13==i.whic
h){ var
e=$(this).closest("form").find(":input:visible:enabled"),n=e.index(this);n==e.length-
1||(e.eq(e.index(this)+1).focus(),i.preventDefault())}else
113==i.which&&$("#btnAction").click()}),$("#select").keydown(function(i){ if(13=
=i.which){ var
e=$(this).closest("form").find(":input:visible:enabled"),n=e.index(this);n==e.length-
1||(e.eq(e.index(this)+1).focus(),i.preventDefault())}else
113==i.which&&$("#btnAction").click()}),$("#radio").keydown(function(i){ if(13==
i.which){ var
e=$(this).closest("form").find(":input:visible:enabled"),n=e.index(this);n==e.length-
1||(e.eq(e.index(this)+1).focus(),i.preventDefault())}else
113==i.which&&$("#btnAction").click()}));
    </script>
    <?php } ?>
    <?php if ($a_customers->Export == "") { ?>
    <script type="text/javascript">
        $("#btnAction").attr('onclick', 'return alertifyAdd(this)'); function
alertifyAdd(obj) { <?php global $Language; ?> if (fa_customersadd.Validate() ==
true ) { alertify.confirm("<?php echo $Language->Phrase('AlertifyAddConfirm');
?>", function (e) { if (e) { $(window).unbind('beforeunload');
alertify.success("<?php echo $Language->Phrase('AlertifyAdd'); ?>");
$("#fa_customersadd").submit(); } }).set("title", "<?php echo $Language-

```

```
>Phrase('AlertifyConfirm'); ?>").set("defaultFocus", "cancel").set('oncancel',  
function(closeEvent){ alertify.error('<?php echo $Language-  
>Phrase('AlertifyCancel'); ?>');}).set('labels', {ok:'<?php echo $Language-  
>Phrase("MyOKMessage"); ?>!', cancel:'<?php echo $Language-  
>Phrase("MyCancelMessage"); ?>'}); } return false; }  
    </script>  
    <?php } ?>  
    <?php include_once "footer.php" ?>  
    <?php  
    $a_customers_add->Page_Terminate();  
    ?>
```

Приложение В АКТ Внедрения



Исх. № 02-09/345

«5» мая 2020г.

Акт внедрения складского учёта «SIM»

Настоящий Акт свидетельствует, что складской учет «Stock Inventory Management(SIM)», разработанный Асильбековой Аминой, внедрено в ТОО «Книга-НВ».

Процесс внедрения проходил с 25 апреля по 5 мая 2020 г.

Заявленные характеристики системы предполагали наличие следующих основных возможностей:

- заполнение и редактирование данных
- функция закупа и учета товаров
- модификация приложения по усмотрению пользователя
- отслеживание выполненных операций.

В ходе опытной эксплуатации складского учёта подтверждено, что приложение обладает всеми заявленными возможностями.

На момент подписания настоящего Акта система установлена и эксплуатируется сотрудниками данной компании.

Исполнительный директор ТОО «Книга-НВ» Шарипова А.Р.



*Толе би, 285/5
Ауэзовский район, Алматы, 050031*

Scanned with CamScanner