

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

РОСТОВСКИЙ ГОСУДАРСТВЕННЫЙ ЭКОНОМИЧЕСКИЙ УНИВЕРСИТЕТ
(РИНХ)

Факультет компьютерных технологий и информационной безопасности
Кафедра Информационных систем и прикладной информатики

ДОПУСТИТЬ К ЗАЩИТЕ

Зав. кафедрой _____

д.э.н., профессор Шполянская И.Ю.

« ____ » _____ 20__ г.

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

на тему:

**«РАЗРАБОТКА WEB-ПРИЛОЖЕНИЯ ДЛЯ АНАЛИЗА ЦЕН НА ТОВАРЫ
БЫТОВОЙ ТЕХНИКИ И ЭЛЕКТРОНИКИ»**

Выполнил

студент группы ПИЗС-341

С.Е. Мигулин

Направление

09.03.03 «Прикладная информатика»

Профиль

09.03.03.01 «Прикладная информатика в экономике»

Руководитель выпускной

квалификационной работы

к.э.н, доцент

С.А. Глушенко

Ростов-на-Дону, 2019

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

РОСТОВСКИЙ ГОСУДАРСТВЕННЫЙ ЭКОНОМИЧЕСКИЙ УНИВЕРСИТЕТ
(РИНХ)

Факультет компьютерных технологий и информационной безопасности
Кафедра Информационных систем и прикладной информатики

Зав. кафедрой _____
д.э.н., профессор Шполянская И.Ю.
« ____ » _____ 20__ г.

ЗАДАНИЕ НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ

Студенту Мигулину Сергею Евгеньевичу

код 12

Тема: Разработка web-приложения для анализа цен на товары бытовой
техники и электроники

Утверждено приказом РГЭУ 12.11.2018г. № 998/4к

Срок сдачи студентом законченного проекта 31.01.2019 г

Исходные данные по проекту: техническое задание, литературные источники

Содержание выпускной квалификационной работы (ВКР):

- а) Анализ и выбор метода решения задачи «Разработка web-приложения для анализа цен на товары бытовой техники и электроники»;
- б) Информационное, программное обеспечение задачи «Разработка web-приложения для анализа цен на товары бытовой техники и электроники»
- в) Надежность и эффективность применения программных средств, вычислительной техники.

Перечень графического материала (с точным указанием обязательных чертежей)
Структура предприятия; Диаграмма прецедентов; Диаграмма активности;
Диаграмма классов (логическая модель); Диаграмма классов (физическая модель);
Диаграмма компонентов; Диаграмма последовательности.

Дата выдачи задания 12.11.2018 г.

Календарный график
выполнения выпускной квалификационной работы

Наименование этапов ВКР	Срок выполнения этапов проекта	Примечание
1. Анализ и выбор метода решения задачи «Разработка web-приложения для анализа цен на товары бытовой техники и электроники»	20.11.2018 г.	
2. Информационное обеспечение задачи «Разработка web-приложения для анализа цен на товары бытовой техники и электроники»	25.12.2018 г.	
3. Разработка алгоритма, программы и технологического процесса решения задачи «Разработка web-приложения для анализа цен на товары бытовой техники и электроники»	20.01.2019 г.	
4. Надежность и эффективность применения программных средств и вычислительной техники	28.01.2019 г.	
5. Сдача на кафедру	31.01.2019 г.	

Руководитель проекта _____ С.А. Глушенко

Задание принял к исполнению _____ С.Е. Мигулин

Реферат

Страниц 79, рисунков 16, таблиц 13, источников 12, приложений 3.

АНАЛИЗ ЦЕН, ИНФОРМАЦИОННАЯ СИСТЕМА, WEB-ПРИЛОЖЕНИЕ, АВТОМАТИЗАЦИЯ, БАЗА ДАННЫХ, ПРИЛОЖЕНИЕ,

Выпускная квалификационная работа выполнена на тему «Разработка web-приложения для анализа цен на товары бытовой техники и электроники»

Целью ВКР является разработка автоматизированной системы мониторинга цен конкурентов для товаров из интернет-магазинов для компании ООО «А-Лаб Групп».

Источниками данных являлись книги, периодические издания, электронные ресурсы, нормативно-правовые акты и законы РФ, которые использовались в качестве теоретических основ рассматриваемой проблематики.

Результатом работы является внедрение системы автоматического анализа цен конкурентов.

Автоматизированная система посвящена сбору и анализу данных по ценам конкурентов для компании ООО «А-Лаб Групп».

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	8
1. АНАЛИТИЧЕСКАЯ ЧАСТЬ	11
1.1. Техничко-экономическая характеристика предметной области	11
1.1.1. Характеристика предприятия.....	11
1.1.2. Краткая характеристика подразделения и видов его деятельности..	13
1.2. Экономическая сущность задачи (комплекса задач)	14
1.3. Обоснование необходимости и цели использования вычислительной техники для решения задачи (комплекса задач).....	18
1.4. Постановка задачи.....	21
1.4.1. Цель и назначение автоматизированного варианта решения задачи	21
1.4.2. Общая характеристика организации решения задачи на ЭВМ	22
1.4.3. Формализация расчетов подзадач	23
1.5. Анализ существующих разработок, выбор и обоснование стратегии автоматизации для приобретения ИС. Обоснование выбора технологии проектирования	25
1.6. Обоснование проектных решений по видам обеспечения:.....	26
1.6.1. Обоснование проектных решений по техническому обеспечению (ТО).	26
1.6.2. Обоснование проектных решений по информационному обеспечению (ИО);	27
1.6.3. Обоснование проектных решений по программному обеспечению (ПО);	29
1.6.4. Обоснование проектных решений по технологическому обеспечению.....	30
2. ПРОЕКТНАЯ ЧАСТЬ.....	33

2.1. Информационное обеспечение задачи (комплекса задач, АРМ, информационной системы, подсистемы)	33
2.1.1. Информационная модель системы и ее описание	33
2.1.2. Характеристика первичных документов с нормативно- справочной и входной оперативной информацией	34
2.1.3. Характеристика базы данных.....	35
2.1.4. Характеристика результатной информации.....	40
2.2. Описание программных модулей	40
2.2.1. Общие положения (дерево функций и сценарий диалога)	43
2.2.2. Структурная схема программной системы (дерево вызова процедур и программных модулей)	44
2.2.3. Описание программных модулей	45
2.3. Технологическое обеспечение задачи.....	51
2.3.1. Организация технологии сбора, передачи, обработки и представления информации	51
3. ОБОСНОВАНИЕ ЭКОНОМИЧЕСКОЙ ЭФФЕКТИВНОСТИ ПРОЕКТА ИС	54
3.1. Эффективность применения информационных технологий	54
3.2. Расчет показателей экономической эффективности проекта ИС	56
ЗАКЛЮЧЕНИЕ	62
БИБЛИОГРАФИЧЕСКИЙ СПИСОК	63
ПЕРЕЧЕНЬ СОКРАЩЕНИЙ И УСЛОВНЫХ ОБОЗНАЧЕНИЙ	64
Приложения	65
ПРИЛОЖЕНИЕ А. ФОРМЫ ВХОДНЫХ ДАННЫХ	66
ПРИЛОЖЕНИЕ Б. ФОРМЫ ВЫХОДНЫХ ДАННЫХ	68
ПРИЛОЖЕНИЕ В. ЛИСТИНГ ПРОГРАММЫ	70

ВВЕДЕНИЕ

С появлением сети Интернет существенно сократились затраты времени на поиск необходимой информации. В настоящее время в сети Интернет можно найти практически любую интересующую пользователя информацию.

Широко известны различные интернет-магазины, которые предлагают приобрести свой товар пользователю, не выходя из дома. Сейчас это стало настолько обычным действием, что уже никого не удивит, покупая товары через интернет с доставкой на дом. Просто и удобно.

Намного сложнее работать в таких условиях владельцам интернет-магазинов. Если покупатель пришел в обычный магазин и нашел нужный ему товар, то он скорее всего купит его здесь, а не поедет искать такой же в другой магазин, находящийся, например, в другой части города. Ведь покупатель не знает, что там может быть дешевле, а в большинстве случаев именно цена товара играет важную роль в покупке.

В сети Интернет всё по-другому. Здесь пользователь может выбирать товар в различных интернет – магазинах, просто переходя на разные сайты, при этом выбирая товар по цене и ряду других параметров, например, удобствам доставки или приятным бонусам к основной покупке.

Владельцы интернет-магазинов в таком случае наиболее часто задумываются о конкурентоспособности цен для каждого товара. И в таком случае не обойтись без мониторинга цен конкурентов. Анализировать цены для каждого товара очень длительно, а если учесть, что конкурентов может быть много, то такая задача является практически невыполнимой.

Объектом выпускной квалификационной работы является компания ООО «А-Лаб Групп» г. Краснодар.

Предметом ВКР является web-приложение для анализа цен на товары бытовой техники и электроники.

Целью ВКР является разработка автоматизированной системы мониторинга цен конкурентов для товаров из интернет-магазинов для компании ООО «А-Лаб Групп».

В соответствии с целью поставлены следующие задачи:

- выполнить анализ организации деятельности компании ООО «А-Лаб Групп» г. Краснодар;
- изучить текущую деятельность сотрудника компании;
- рассмотреть систему работы ИТ-отдела;
- внедрить систему автоматического анализа цен конкурентов.

В рамках выпускной квалификационной работы поставлена задача разработать систему, способную автоматизировать процесс сбора цен конкурентов и проведение их дальнейшего анализа.

Все вышперечисленное позволяет сделать вывод о необходимости разработке системы, способной автоматизировать процесс сбора цен конкурентов и проведение их дальнейшего анализа.

В соответствии с поставленными целями ВКР изложен во введении, трех главах, заключении и приложениях.

В первой главе содержится технико-экономическая характеристика компании ООО «А-Лаб Групп», представлена деятельность, её существующее техническое и программное обеспечение. Также приведена характеристика входной и выходной информации.

Вторая глава посвящена информационному обеспечению задачи. Также в главе рассматриваются источники создания задачи, взаимосвязь данной задачи с другими задачами подсистемы, описывается перечень входных и выходных сообщений и описание структурных единиц информации, рассматриваются вопросы разработки алгоритмов, программ и технологического процесса обработки информации.

В третьей главе рассматриваются вопросы оценки надежности программного продукта и вопросы эффективности применения информационных технологий.

В заключении помещены выводы, сделанные по выпускной квалификационной работе и перспективы дальнейшего совершенствования задачи и практического использования результатов.

В библиографическом списке указана основная и дополнительная литература, при помощи которой была разработана данная квалификационная работа.

1. АНАЛИТИЧЕСКАЯ ЧАСТЬ

1.1. Техничко-экономическая характеристика предметной области

1.1.1. Характеристика предприятия

Компания ООО «А-Лаб Групп» (далее – компания) была основана в 2008 году. ООО «А-Лаб Групп» — это современная технологичная компания из финансового сектора. Центральный офис компании находится в г. Краснодаре. Продуктами компании пользуются клиенты из разных регионов России и стран Ближнего зарубежья.

Основные направления деятельности ООО «А-Лаб Групп»:

- инвестиционная деятельность и биржевые операции;
- разработка и внедрение финансового программного обеспечения;
- организация учебных курсов и семинаров дополнительного образования.

Помимо этой деятельности в компании создано еще одно направление – продажа бытовой техники в розницу. Основной канал продаж – интернет-магазин.

В штате компании около 30 сотрудников, в основном технических специальностей: разработчики, IT-специалисты и пр. Средний возраст сотрудников 26 лет.

На рисунке 1 показана сформированная организационная структура компании, которая построена в виде иерархии.

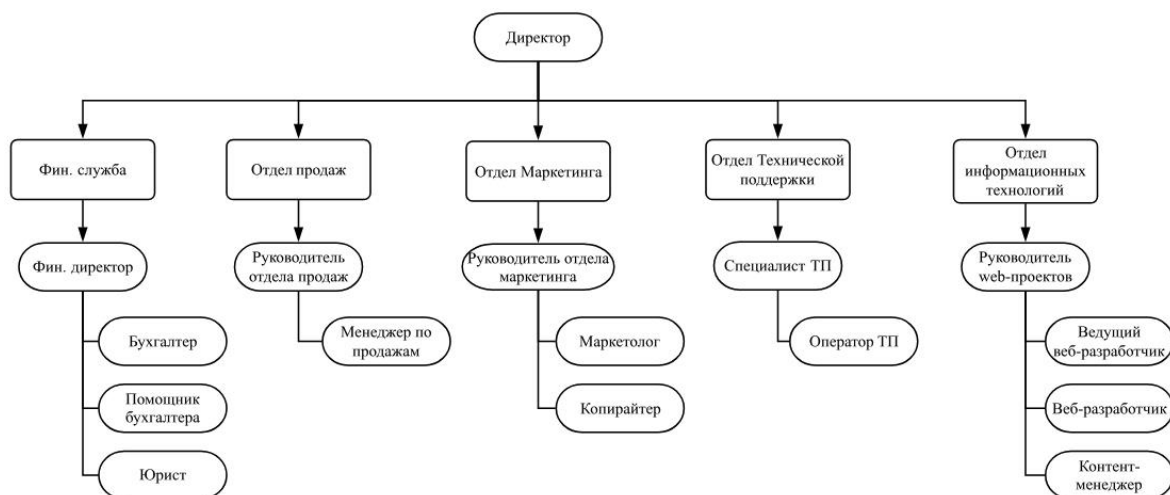


Рисунок 1 – Организационная структура компании

Руководителем компании является директор, у которого в подчинении находятся все сотрудники компании.

Организационная структура компании выбрана линейная: плюсы – единоначалие, четкая субординация с конкретным разграничением прав и обязанностей, минусы – необходима большая компетентность руководителя.

Так как обследуемая компания относится к финансовому сектору и напрямую связана с работой Московской биржи, то график работы в компании с 10.00 до 19.00. Таким образом, в соответствии с трудовым кодексом режим работы сотрудников установлен в виде пятидневной рабочей недели с часовым перерывом на обед.

Одна из деятельностей компании – продажа товаров (бытовой техники и электроники) через современные и менее затратные каналы, предпочтительно, через интернет-магазин. Для этого в компании имеются все необходимые средства: ресурсы, штат, коммуникация с поставщиками и хорошая логистика.

В компании существует пять подразделений, каждое из которых имеет свою роль и отвечает за определенный круг обязанностей. Рассмотрим каждое из них:

Финансовая служба в компании выполняет функции разработки предложений по установлению финансовой политики компании, а также её

реализации в различных проявлениях, контроля и составления финансовой отчетности. Руководителем финансовой службы является финансовый директор.

Отдел продаж отвечает за продажу товаров и услуг компании. Основной функционал, с которым работает отдел продаж реализован в CRM-системе компании. Ответственность за выполнение планов продаж и других показателей, связанных с данным отделом, лежит на Руководителе отдела продаж.

В компании существуют также отдел маркетинга и отдел технической поддержки. Отдел маркетинга отвечает за коммуникацию компании с клиентами, внешний имидж и бренд компании, привлечение новых клиентов в компанию, используя разные каналы. Отдел технической поддержки отвечает за работу с обращениями от клиентов, решение технических проблем, связанных с деятельностью компании

1.1.2. Краткая характеристика подразделения и видов его деятельности

В обязанности отдела информационных компании ООО «А-Лаб Групп» входит решение таких задач, как автоматизация бизнес-процессов компании, разработка нового программного обеспечения для клиентов компании и поддержание существующих проектов. Кроме этого, отдел информационных технологий работает с хранением электронной версии корпоративных документов, их доступностью и обработкой, а также решает технические проблемы, связанные с этой областью деятельности.

Для полной доступности информации, учитывая, что некоторые подразделения предприятия находятся в разных офисах, инфраструктура предприятия создана на SAAS (Soft as a Service) решении. Доступ к системе осуществляется через интернет с помощью авторизации по уникальным для каждого сотрудника данным. Данные шифруются и надежно защищены от стороннего проникновения. В отделе информационных технологий работают, в основном, сотрудники с техническим образованием. Рассмотрим основные функции веб-программиста.

Веб-программист относится к категории специалистов и должен подчиняться руководителю веб-проектов. В обязанности веб-программиста должно входить знание языков программирования таких как PHP, JavaScript, HTML и CSS. Он должен иметь понимание как работают протоколы HTTP, HTTPS, WebSocket и FTP, а также знание основ баз данных и таких СУБД, как MySQL и MSSQL [3]. Выполняя административные функции при работе с сервером компании, веб-программист должен знать принципы работы с операционной системой, установленной на данном сервере.

Для командной работы в отделе применяется система управления версиями GIT, веб-программист обязан уметь с ней работать.

На время отсутствия веб-программиста на рабочем месте, в связи с очередным отпуском, болезнью или другой причине, его обязанности должно выполнять лицо, назначенное в установленном порядке. Данное лицо приобретает соответствующие права и несет ответственность за надлежащее исполнение возложенных на него обязанностей.

1.2. Экономическая сущность задачи (комплекса задач)

Разработкой ценообразования в компании на товары в интернет-магазине занимается директор. Он устанавливает план продаж и минимальный порог цен, ниже которых продажа товара не допускается. Цикл выставления ценообразования директором составляет каждые полгода.

В каждом цикле, после первоначального выставления цен директором, за ценами следит контент-менеджер в отделе информационных технологий. Эта задача является одной из основных задач сотрудника этого отдела. Процесс называется «анализ цен конкурентов» и предполагает мониторинг ценовой информации с сайтов конкурентов с целью установки на сайте компании более привлекательной цены, чем у конкурентов.

Для того, чтобы быть конкурентоспособными на рынке электронной техники, контент-менеджер просматривает список товаров по конкурентам в их

интернет-магазинах и на основании формулы, которой он рассчитывает ценообразование, делает корректировку в интернет-магазине. Чтобы выполнить системное исследование процесса анализа цен конкурентов, необходимо изучить структуру происходящих внутри бизнес-процессов. Для этих целей строится модель «как есть». Для это был применен структурный подход и нотация IDEF0.

На рисунке 2 изображена контекстная диаграмма анализа цен контент-менеджером, с которой согласно нотации начинается моделирование.



Рисунок 2 – Контекстная диаграмма IDEF0

Рассмотрим более детально процесс сбора цен контент-менеджера. Для выполнения работы по мониторингу цен, сотрудник делает обращение к электронной таблице за URL-ссылкой на товар конкурента, далее он копирует эту URL-ссылку, вставляет её в web-браузер и переходит на страницу с товаром на сайте конкурента. После того, как работник нашел цену товара, он копирует её и записывает результат в электронную таблицу с ценами за определенную дату.

Основная сложность выполнения этой тривиальной задачи заключается в том, что цены меняются очень часто, так как сегмент рынка, в котором работает компания, очень разнообразен. Количество товаров, которые реализует компания, как через интернет-магазин превышает несколько сотен, но даже такое количество выставления цен является очень затруднительным и затратным в использовании времени. И хоть процент ключевых и более продающих товаров ниже от общего числа, этот процент достаточно велик и на него стоит часто обращать внимание.

После описания контекстной диаграммы проводится ее функциональная декомпозиция – это значит, что система разбивается на подсистемы и каждая подсистема описывается отдельно. Далее подсистемы разбиваются на более мелкие и так до достижения нужной степени подробности [8, с. 45].

В результате такого разбиения, каждый фрагмент системы изображается на отдельной диаграмме декомпозиции (рисунок 3).

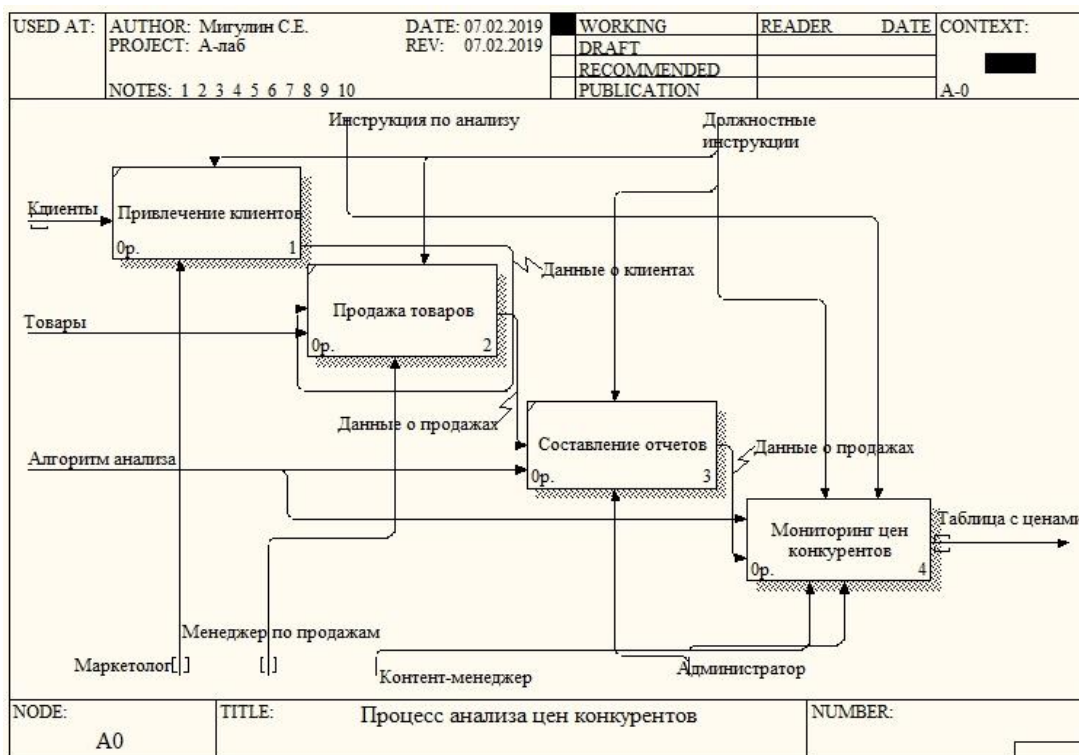


Рисунок 3 - Диаграмма Декомпозиции IDEF0 (Уровень 0).

Процесс, который рассматривается, можно разделить на четыре составляющих:

1) «Привлечение клиентов» иллюстрирует наработку новой и поддержание старой базы клиентов для их дальнейшего совершения покупок.

2) «Продажа товаров» иллюстрирует процесс от выбора товара клиентами до оплаты его им, а также последующей доставки товара (опционально).

3) «Составление отчетов» иллюстрирует формирование отчетов на основании продажи товаров за определенный период времени.

4) «Мониторинг цен конкурентов» иллюстрирует процесс сбора информации о ценах конкурентах для поддержания конкурентоспособной цены в своем сегменте.

Обработка цен, которые работник записал в электронную таблицу, вручную занимает колоссальные затраты времени (около 40 рабочих часов), а эффективность этой работы очень низкая. Алгоритм процесса сравнения товара работником компании изображен на рисунке 4.

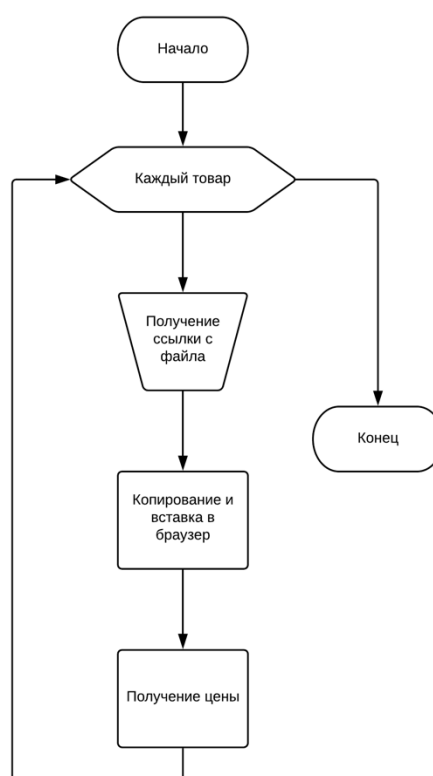


Рисунок 4 – Блок-схема алгоритма процесса сравнения товара

Хранить информацию в электронных таблицах очень затруднительно, за счет того, что товарных позиций в компании много, но цена меняется не у всех конкурентов и не на все товары одновременно. А необходимо не перезаписывать

данные о ценах, а сохранять несколько цен для каждого товара, чтобы производить расчеты и анализ поведения цен за некоторый период времени. Для этого нужно создавать новые электронные таблицы и связывать их, что доставляет сложность, а, следовательно, тормозит процесс анализа.

Актуальность ценовой политики важна для компании ООО «А-Лаб Групп», ведь на сегодняшний день в сегменте продаж бытовой техники и электроники большое количество конкурентов, которые и больше и более узнаваемы, чем компания. Поэтому компании необходима система для автоматического мониторинга и анализа цен конкурентов для принятия важных решений в вопросах ценовой политики.

1.3. Обоснование необходимости и цели использования вычислительной техники для решения задачи (комплекса задач).

В компании ООО «А-Лаб Групп» до текущего момента не было автоматизированной системы автоматического анализа. Анализ велся вручную на основе данных программы Excel, который поставляется в пакете Microsoft Office.

Входными документами являлись списки товаров с url-адресами конкурентов на аналогичные товары.

Директором компании было принято решение о автоматизации процесса анализа цен, так как, как уже говорилось ранее, скорость и гибкость в текущей реализации очень низкая и неэффективная.

Использование только ручного процесса не позволяло получать быстрый результат деятельности сотрудника, было сложно оценить его продуктивность из-за рутинных операций. Не существовало также и нужных аналитик, соответствующих направлениям деятельности компании, и из-за этого невозможно было правильно распределять затраты и выручку. На формирование таких данных уходило очень много времени, и из-за трудоемкости процесса не исключались ошибки.

Так как компания, в основном, для своей деятельности использует самостоятельно написанные SAAS-решения, а также имеет в своем штате высококвалифицированных сотрудников в отделе информационных технологий, то целесообразно использовать возможности создания своего гибкого, адаптированного под нужды компании, автоматизированного решения.

Вся деятельность компании ООО «А-Лаб Групп» ведется в созданной SAAS-системе, созданной специально для нужд компании. Кроме этого, на каждом из компьютеров установлено программное обеспечение в базовой конфигурации (для всех) и отдельное программное обеспечение для нужд конкретных сотрудников. Список программного обеспечения, установленного на всех компьютерах компании описан в таблице 1.

Таблица 1 – список программного обеспечения ООО «А-Лаб Групп».

Номер п/п	Наименование	Версия
1	Операционная система Windows	10
2	LibreOffice	6.1
3	Google Chrome	52.13
4	AcrobatReader	6.0 CE
5	Brackets	17.24

Для продуктивной и более качественной работы в компании ООО «А-Лаб Групп» используются следующие технические средства:

1) Системный блок ПК: процессор Intel Core i5 3.8 GHz, материнская плата Asus Mainboard 34.v, оперативная память Kingston hyper 8 или 16 gb, а также жесткий диск на 1024 Gb.

2) Монитор Asus диагональ 24.5”, 1920x1080 Full HD

3) Периферийные устройства: клавиатура, мышь и наушники с микрофоном A4Tech.

4) Многофункциональное устройство HP Sodi 24m. Максимальный формат печати – А4. Печать принтера осуществляется черно – белая.

5) Коммутатор TP-Link TL-SF1024M, 24 порта Ethernet 10/100/1000 Мбит/сек, uplink/SFP: 2, монтируется в стойку. Установлено программное обеспечение для управления трафиком Zabbix.

б) Точка доступа d-link dap-1360u для обеспечения беспроводного соединения в локальной сети и выхода в интернет. Защита данных для беспроводной сети шифруется с помощью современных протоколов, таких как WPA/WPA2 Enterprise.

Перечень используемых ППП в ООО «А-Лаб Групп»:

Веб-браузеры – это комплексные программы для обработки и отображения HTML-страниц по протоколу HTTPS (сейчас используется только безопасный тип соединения HTTPS).

Программа Google Chrome – веб-браузер, поддерживающий большинство современных веб-технологий и интернет-протоколов. У этого браузера хороший, продуманный интерфейс для пользователя, минималистичный дизайн и хорошая производительность.

LibreOffice – пакет программ для работы с офисными документами. Кроме стандартных для LibreOffice форматов хранения данных, можно успешно открывать и сохранять документы, созданные в других популярных офисных пакетах.

Текстовый процессор (LibreOffice Writer) – это аналог Microsoft Word. Он позволяет проектировать и создавать текстовые документы, содержащие изображения, таблицы или графики. После выполненной, в программе, работы, документ можно сохранить в любом популярном формате, таком как формат OpenDocument format (ODF), формат Microsoft Word (DOC, DOCX) или PDF.

Помимо текстового процессора в пакет LibreOffice входит электронная таблица LibreOffice Calc, предназначенная для работы с электронными таблицами. Libreoffice Impress для создания и демонстрации презентаций, векторный редактор Draw, редактор формул Math, а также система управления базами данных Base, которая поддерживает некоторые обычные файловые форматы баз данных, например, BASE. Кроме того, можно использовать

LibreOffice Base для подключения к внешним реляционным базам данных, например, к базам данных MySQL или Oracle [1].

1.4. Постановка задачи

1.4.1. Цель и назначение автоматизированного варианта решения задачи

Целью выпускной квалификационной работы является разработка автоматизированной системы мониторинга цен конкурентов для товаров из интернет-магазинов для компании ООО «А-Лаб Групп».

В результате анализа деятельности компании был сделан вывод, что существующий процесс сбора цен конкурентов является неэффективным и медленным, так как осуществляется одним специалистом вручную, которого явно недостаточно для быстрой, а главное, качественной работы.

Автоматизация такого бизнес-процесса, как анализ цен конкурентов позволит многократно повысить скорость обработки таблиц с товарами для контент-менеджера и освободит большую часть рабочего времени, которое он тратил на анализ цен конкурентов, чтобы он мог заниматься другими задачами в компании. Также автоматизация позволит решить следующие задачи:

- выполнить более глубокий анализ цен;
- осуществлять хранение истории цен за длительный период;
- оптимизировать рабочее время сотрудника;
- представить в более наглядной форме цены на информативных графиках и диаграммах;

На основании поставленных задач была спроектирована контекстная диаграмма, показывающаяся процесс «как должно быть». Необходимые товары и ссылки на адреса сайтов конкурентов на эти товары будут занесены в базу. Обход этих ссылок будет запланирован системой на «ежедневно». Полученные цены

будут обрабатываться и складываться в удобном виде в базу для дальнейшего анализа (рисунок 5).

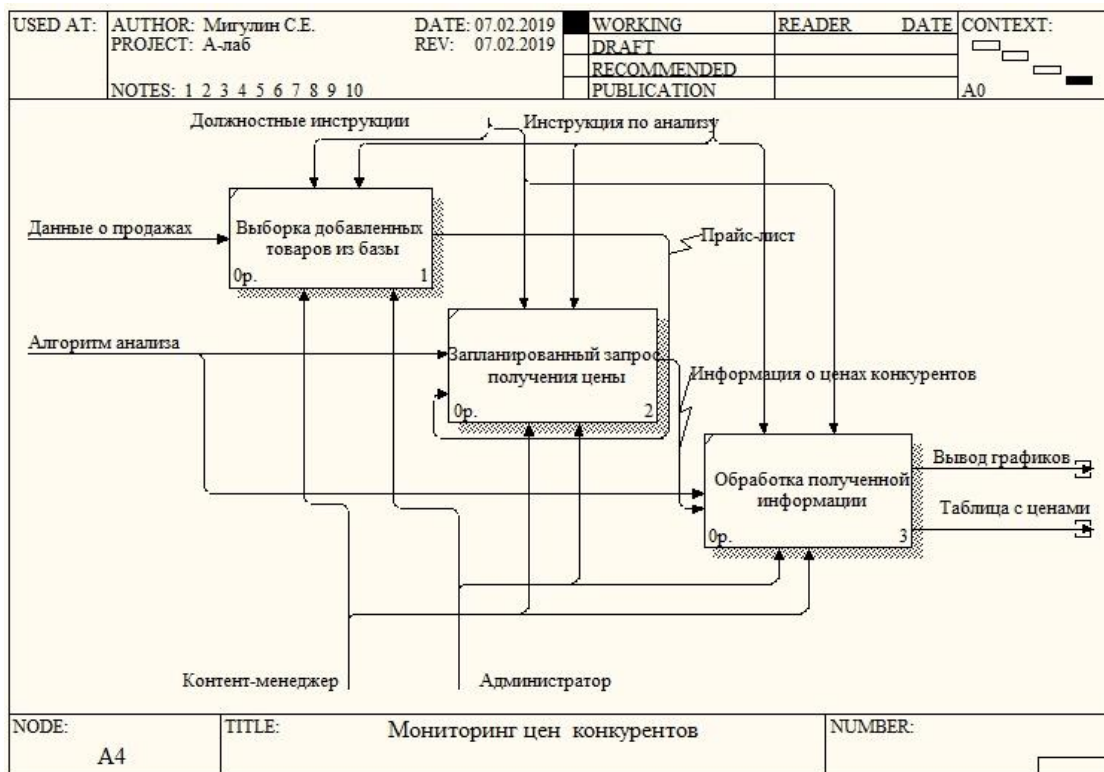


Рисунок 5 – Контекстная диаграмма процесса сбора цен.

1.4.2. Общая характеристика организации решения задачи на ЭВМ

Для решения задачи по автоматизации анализа цен конкурентов для компании «А-Лаб Групп» необходимо определить предполагаемую организационную архитектуру будущей аппаратной платформы.

После детального изучения объекта автоматизации, а также обоснования актуальности создания АИС, можно составить функционально-структурную схему АИС, изображенную на рисунке 6, отражающую функции, задачи и операции, которые будет выполнять автоматизированная система.

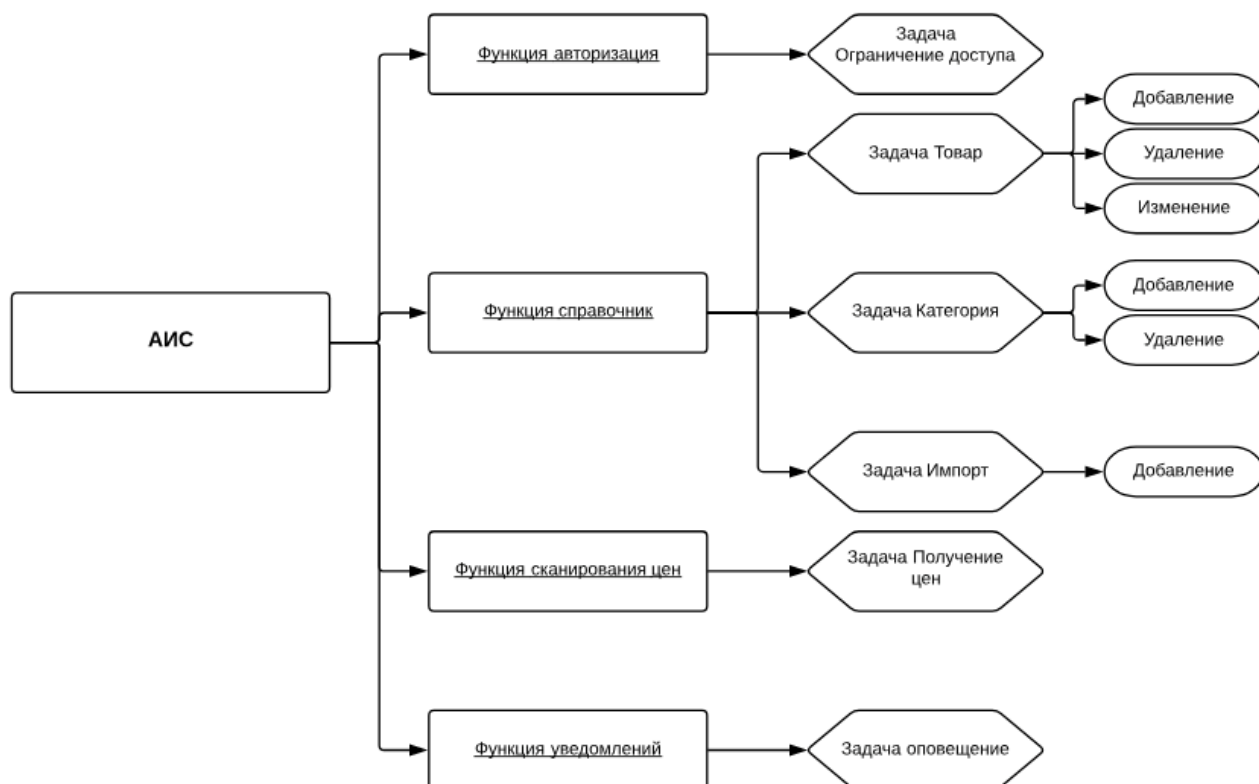


Рисунок 6 – функционально-структурная схема АИС.

В АИС выделены следующие функции:

- 1) функция «авторизация» – функция, которая ограничивает доступ к данным в целях сохранения коммерческой тайны компании.
- 2) функция «справочник» – функция необходима для работы АИС. В этой функции выполняется добавление, удаление и изменения товаров и категорий, а также добавление нескольких товаров одновременно с помощью импорта.
- 3) функция «сканирование цен» – основная функция, которая ежедневно делает запросы на URL-адреса сайтов конкурентов для сбора ценовой информации.
- 4) функция «уведомления» – оповещение пользователей системы об изменении цен товаров путем отправки электронных писем.

1.4.3. Формализация расчетов подзадач

Обобщенный алгоритм расчета средних значений, изображенный на рисунке 7, будет выполнять запрос к БД и получать список цен, с которыми в дальнейшем будут выполняться арифметические операции. Для получения среднего значения цены, каждый элемент списка складывается и делится на количество элементов в списке. Для получения максимальной и минимальной цены, алгоритм будет присваивать первые значения цен в переменную и начнет сравнение, если следующий элемент отличается от текущего в большую или меньшую сторону (в зависимости от алгоритма поиска значения), переменная перезаписывается. После прохождения каждого элемента выявляется максимальная и минимальная цена.

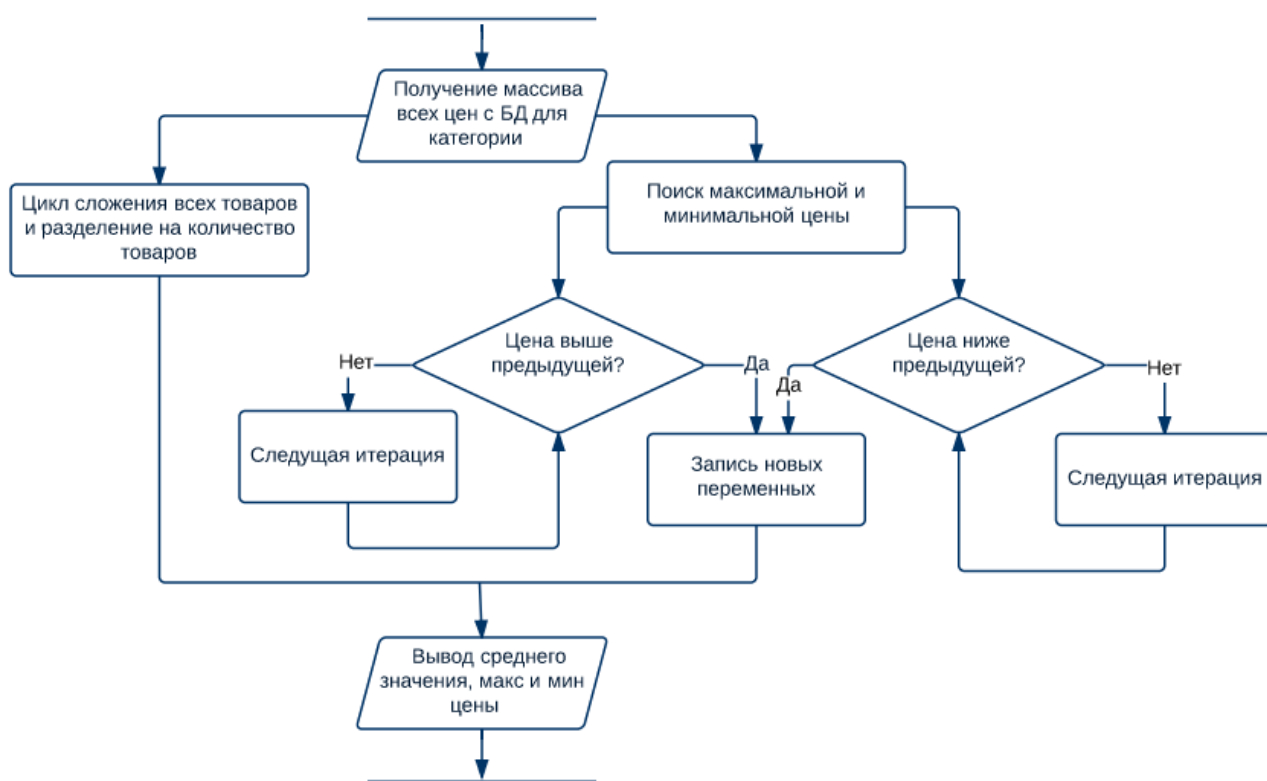


Рисунок 7 – Расчет среднего значения, а также минимальной и максимальной цены.

1.5. Анализ существующих разработок, выбор и обоснование стратегии автоматизации для приобретения ИС. Обоснование выбора технологии проектирования

В настоящее время существует небольшое количество автоматизированных информационных систем, осуществляющих решение задачи автоматизации сбора цен конкурентов.

Например, сервис Competera.ru. Это платформа для сбора цен, анализа рынка и автоматизации переоценки. В данной системе реализованы следующие возможности:

- мониторинг рынка;
- анализ конкурентов;
- управление ценой, ценовые модели;
- оптимизация ценообразования для ритейлеров;
- переоценка товаров;
- динамическое ценообразование.

Однако, автоматизированная система Competera является достаточно сложной и громоздкой. Система требует от пользователя высокой квалификации и, кроме того, требует достаточно сложной настройки и администрирования. Система не является гибкой, её сложно встроить в существующую модель компании.

Не мало важным является фактор высокой стоимости автоматизированной системы Competera, также на стоимость сильно повлияет необходимость обучения пользователей, развертывание системы, невозможность экстренной технической поддержки из-за сильной удаленности разработчика. Также нет необходимости в большинстве функций данной системы.

Таким образом, представляется нецелесообразным и дорогим использование автоматизированной системы Competera в ООО «А-Лаб Групп» для решения узко специализированной задачи создания автоматизированной информационной системы. Гораздо дешевле и быстрее представляется

написание отдельного приложения, решающего данную задачу, являющегося простым в настройке и использовании, дешевым и удобным. Легко приспособляемым к новым требованиям компании, а также возможность расширения функционала в будущем.

1.6. Обоснование проектных решений по видам обеспечения:

1.6.1. Обоснование проектных решений по техническому обеспечению (ТО)

Рассмотрим техническое обеспечение, необходимое для автоматизации бизнес-процесса по сбору и анализу цен конкурентов.

Техническое обеспечение – это средства для полноценной работы информационной системы и сопутствующая к ней документация. К техническому обеспечению можно отнести вычислительные машины, различное периферийное оборудование, а также каналы передачи данных.

На основании полученных данных сформирована задача для создания централизованного технического обеспечения. Вся архитектура будет находиться на сервере, а пользователь будет подключаться к серверу через интернет. На сервере будет развернута база данных в системе управления базами данных, также будет размещен сервис для получения запросов и передачи ответов пользователям. Предполагаемые характеристики сервера представлены в таблице 2.

Таблица 2 – характеристики сервера.

Характеристика	Значение
Процессор	Двухъядерный Intel Xeon тактовая частота 2,4 ГГц.
Оперативная память	Два слота оперативной памяти по 2 ГБ DDR-4
Жесткий диск	SSD диск 120 Гб.
Питание	Бесперебойное питание в дата-центре

Продолжение таблицы 2

IP адрес	Статичный IP-адрес v4
----------	-----------------------

Рассмотрим характеристики сервера подробнее:

- Двухъядерный процесс без особых усилий справится с нагрузкой, возложенной в результате обработки информации и позволит дополнительно эффективно распараллеливать задачи, которые на нем выполняются

- Оперативной памяти суммарным объемом 4 Гб достаточно для обработки больших объемов информации.

- Высокоскоростной SSD накопитель позволит моментально считывать и записывать информацию на диск.

Также необходимо заметить, что на сервере используется статичный IP-адрес для удаленного подключения к нему администратору.

Особых требований к ПК пользователя нет по причине того, что все вычисления проходят на сервере, а ПК лишь выводит информацию на монитор. Главной необходимостью является стабильное подключение к сети интернет.

1.6.2. Обоснование проектных решений по информационному обеспечению (ИО)

В предыдущем пункте дано определение АИС и сделан выбор по его техническому обеспечению. Информационное обеспечение занимает не менее важное место. Информационное обеспечение – это входные и выходные данные.

Входные документы – это набор справочных сведений, которые заносятся или используются в базе данных с помощью специально разработанных для этого форм.

В АИС используются следующие типы входной информации:

- 1) экранные формы - предназначены для управления системой, ввода информации, редактирования и поиска данных.

- 2) файлы – для импорта нескольких товаров одновременно в базу данных системы. Хранится в электронном виде на жестком диске ПК.

Входные сообщения представлены в таблице 3.

Таблица 3 – Входные сообщения

Наименование	ID	Вид представления	Источник	Задача
Логин и пароль	Z1	Экранная форма	Работник ИТ-отдела	Авторизация и вход в систему пользователя
Название товара	Z2	Экранная форма	Работник ИТ-отдела	Для идентификации товара пользователем
URL-адрес	Z3	Экранная форма	Работник ИТ-отдела	Отправка запросов по URL адресам для сканирования цены
Артикул	Z4	Экранная форма	Работник ИТ-отдела	Поиск нужного товара в системе
Изображение	Z5	Экранная форма	Работник ИТ-отдела	Вывод визуально товара для быстрого восприятия
Категория	Z6	Экранная форма	Работник ИТ-отдела	Создание категории для товаров
Маска поиска	Z7	Экранная форма	Работник ИТ-отдела	Часть HTML-кода на сайте для поиска цены

Первыми входными данными для начала работы системы будет авторизация. Пользователю необходимо задать логин и пароль, чтобы войти в систему и начать с ней работать.

Для добавления товара в систему необходимо ввести Название товара, а также URL-адрес хотя бы одного магазина для того, чтобы система могла собирать информацию с сайтов конкурентов. Помимо этого, по желанию пользователя и для его удобной работы, для каждого товара в системе можно вводить Артикул и Категорию.

Работа с информацией происходит посредством экранных форм. Форма – это важнейший визуальный компонент. При разработке экранных форм использовались разные инструментальные средства, позволяющие ускорить ввод данных, а именно:

- Кнопки, для организации более дружественного интерфейса;
- Выпадающие списки, для экономии места в окне приложения при неиспользовании данного типа справочника. Выпадающие списки представляют пользователю выбор из имеющихся значений;
- Таблицы, позволяют представить информацию в удобном и структурированном виде.

В АИС вся выводимая информация будет храниться базе данных. Выводимые сообщения представлены в таблице 4.

Таблица 4 – Выходные сообщения

Наименование	ID	Вид представления	Срок выдачи	Получатель	Задача
Данные о товаре	A1	Экранная форма	По запросу	Работник ИТ-отдела	Отображение информации о товаре
Данные о категории	A2	Экранная форма	По запросу	Работник ИТ-отдела	Отображение информации о категории
Название магазина	A3	Экранная форма	По запросу	Работник ИТ-отдела	Отображение магазина на основе URL
Цена	A4	Экранная форма	По запросу	Работник ИТ-отдела	Вывод из БД с различной выборкой
Дата	A5	Экранная форма	По запросу	Работник ИТ-отдела	Вывод из БД с различной выборкой

Выходные данные будут выводиться в разных модулях системы после выполнения системой определенных функций.

1.6.3. Обоснование проектных решений по программному обеспечению (ПО)

Программное обеспечение – это совокупность программ для реализации целей и задачи информационной системы АИС и нормального функционирования комплекса технического обеспечения. Основой любого автоматизированного рабочего места является организация данных, которые подвергаются обработке.

В целях экономии необходимо найти простое и бесплатное решение в виде программного обеспечения для его установки на сервер, который был описан в пункте 1.6.1. Операционная система Linux в оболочке Ubuntu 16.04 является свободно-распространяемой ОС, которая содержит в себе все необходимые компоненты для работы сервера. Ubuntu 16.04 много разнообразного софта и библиотек, которые подойдут для любого по сложности проекта.

В АИС будет использоваться реляционная база данных, управлять которой будет система управления базой данных (СУБД). Руководствуясь теми же критериями (гибкость и бесплатность) под ОС Ubuntu 16.04 будет установлена СУБД MySQL, которая является свободной в использовании.

MySQL – это свободная реляционная СУБД, которая хоть и является бесплатной, но поддерживается корпорацией Oracle, которая получила права на торговую марку. MySQL является решением для простых и средних по масштабу проектов. Гибкость СУБД MySQL обеспечивается поддержкой большого количества типов таблиц: пользователи могут выбрать как таблицы типа MyISAM, поддерживающие полнотекстовый поиск, так и таблицы InnoDB, поддерживающие транзакции на уровне отдельных записей. Более того, СУБД MySQL поставляется со специальным типом таблиц EXAMPLE, демонстрирующим принципы создания новых типов таблиц. Благодаря открытой архитектуре и GPL-лицензированию, в СУБД MySQL постоянно появляются новые типы таблиц.

1.6.4. Обоснование проектных решений по технологическому обеспечению

В реальных условиях проектирование – это поиск способа, который удовлетворяет требованиям функциональности системы средствами имеющихся технологий с учетом заданных ограничений. В настоящее время широкое распространение получила RAD-методология – методология быстрой разработки приложений. Данная методология охватывает все этапы жизненного цикла современных информационных систем [7].

RAD-технология позволяет при разработке и создании информационной системы применять спиральную модель жизненного цикла информационной системы, т.е. использовать прототипы. Возможность согласования прототипов системы с непосредственным будущим пользователем системы позволяет не терять общей картины проектируемой системы, контролировать требования заказчика и оперативно вносить необходимые корректировки функциональных возможностей системы. Визуальные инструменты RAD позволяют создавать современные пользовательские интерфейсы с минимальным написанием кодов программ [9].

Программные средства RAD-разработки делятся на универсальные и специализированные. Специализированные средства разработки ориентированы только на создание приложений баз данных и, как правило, привязаны к вполне определенным системам управления баз данных. Среди универсальных систем визуального программирования наибольшее распространение получили: Microsoft Visual Studio, Azure DevOps, с помощью которых можно разработать приложения практически любого типа, причем программы, разработанные такими средствами в состоянии взаимодействовать практически с любой системой управления базами данных. Возможности таких систем при разработке пользовательского интерфейса слились с созданием прототипов системы. При активном участии пользователя системы, способного корректировать как результаты расчетов, так и собственные требования к системе, сроки изготовления программы сокращаются. Наследованное универсальными системами событийного программирования предполагает описание различных событий различных объектов, установленных на форме-контейнере элементов

управления. С помощью событийного программирования можно как обрабатывать информацию, хранящуюся в базе данных, так и создавать, редактировать, модифицировать, удалять объекты баз данных. Как и любая технология, методология RAD имеет ряд своих недостатков и ограничений. Но при разработке относительно небольших систем, разрабатываемых под конкретную организацию, она достаточно эффективна.

2. ПРОЕКТНАЯ ЧАСТЬ

2.1. Информационное обеспечение задачи (комплекса задач, АРМ, информационной системы, подсистемы).

Создание АИС позволит обеспечить:

- Автоматизировать сбор данных цен конкурентов;
- Проведение анализа цен для выявления более конкурентного предложения на рынке;
- Создания единой системы доступа к данным;
- Снижение трудозатрат на сбор, хранение и анализ данных.

После создания и внедрения АИС должны быть улучшены следующие показатели:

- Многократное уменьшение времени на сбор и обработку информации;
- Увеличение качества анализа данных;
- Уменьшение ошибок (сведение их к низкому показателю) при получении данных в отличии от выполнения этой задачи пользователем.

2.1.1. Информационная модель системы и ее описание

В АИС выполнены следующие функции:

- 1) функция «авторизация» – функция, которая ограничивает доступ к данным в целях сохранения коммерческой тайны компании.
- 2) функция «справочник» – функция необходима для работы АИС. В этой функции выполняется добавление, удаление и изменения товаров и категорий, а также добавление нескольких товаров одновременно с помощью импорта.
- 3) функция «сканирование цен» – основная функция, которая ежедневно делает запросы на URL-адреса сайтов конкурентов для сбора ценовой информации.

4) функция «уведомления» – оповещение пользователей системы об изменении цен товаров путем отправки электронных писем.

На рисунке 8 изображен алгоритм поиска цен: система разбивает общий массив на два, каждый из которых имеет одинаковую дату за сегодня и вчера. Условие проверяет, отличается ли цена сегодня от цены вчера. Если нет, то алгоритм переходит к следующей итерации. Если да, то рассчитывается разница между ценами и данные выводятся на экран, после этого алгоритм также переходит к следующей итерации.

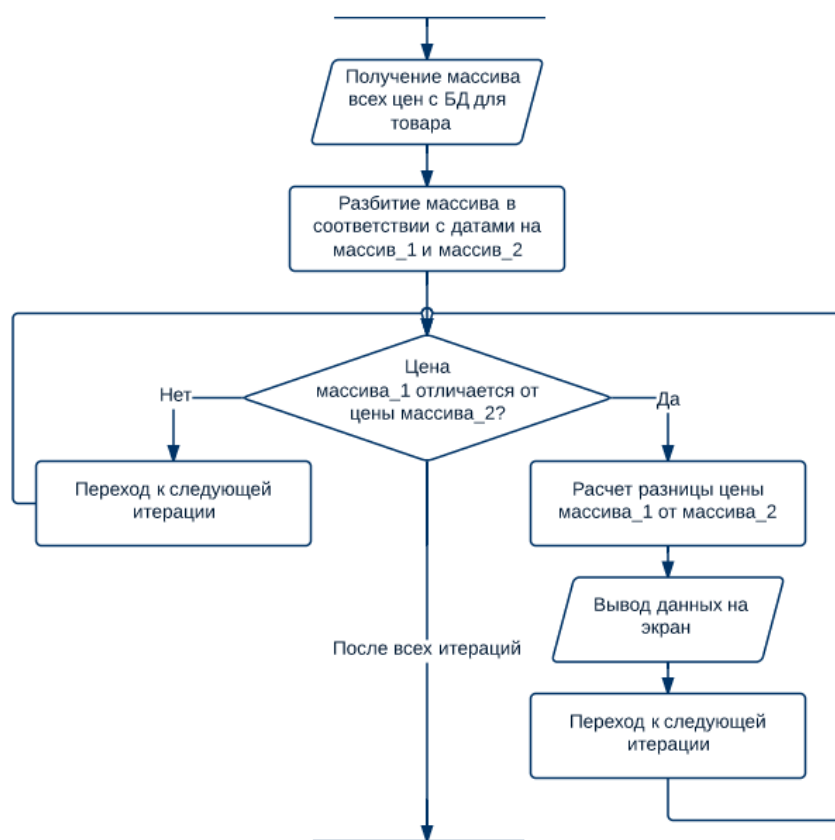


Рисунок 8 – алгоритм поиска цен

2.1.2. Характеристика первичных документов с нормативно-справочной и входной оперативной информацией

Структурными единицами в базе данных являются таблицы, а также запросы и формы.

Таблицы – это объекты, в которые вносятся данные. Каждая таблица – это сущность, которая отвечает за собственные данные.

Запросы – это объекты, с помощью которых извлекаются нужные данные по определенным критериям, также эти данные могут фильтроваться или производиться расчеты.

Формы предназначены для работы с данными из таблиц баз данных. Через формы данные можно вводить, редактировать и удалять или же только просматривать.

Все таблицы создаются на основе информационной модели. Ключевые поля будут соответствовать первичным ключам сущностей.

2.1.3. Характеристика базы данных

Схема данных, изображенная на рисунке 9, является графическим образом БД. С помощью схемы определяются связи между несколькими таблицами. Также схема обеспечивает целостность взаимосвязанных данных при корректировке таблиц.

Связь между таблицами устанавливается отношения между одинаковыми значениями в ключевых полях. Практически во всех случаях с ключевым полем одной таблицы, являющимся уникальным идентификатором каждой записи, связывается внешний ключ другой таблиц. При связывании полей одинаковый тип и формат является обязательным условием для связи.

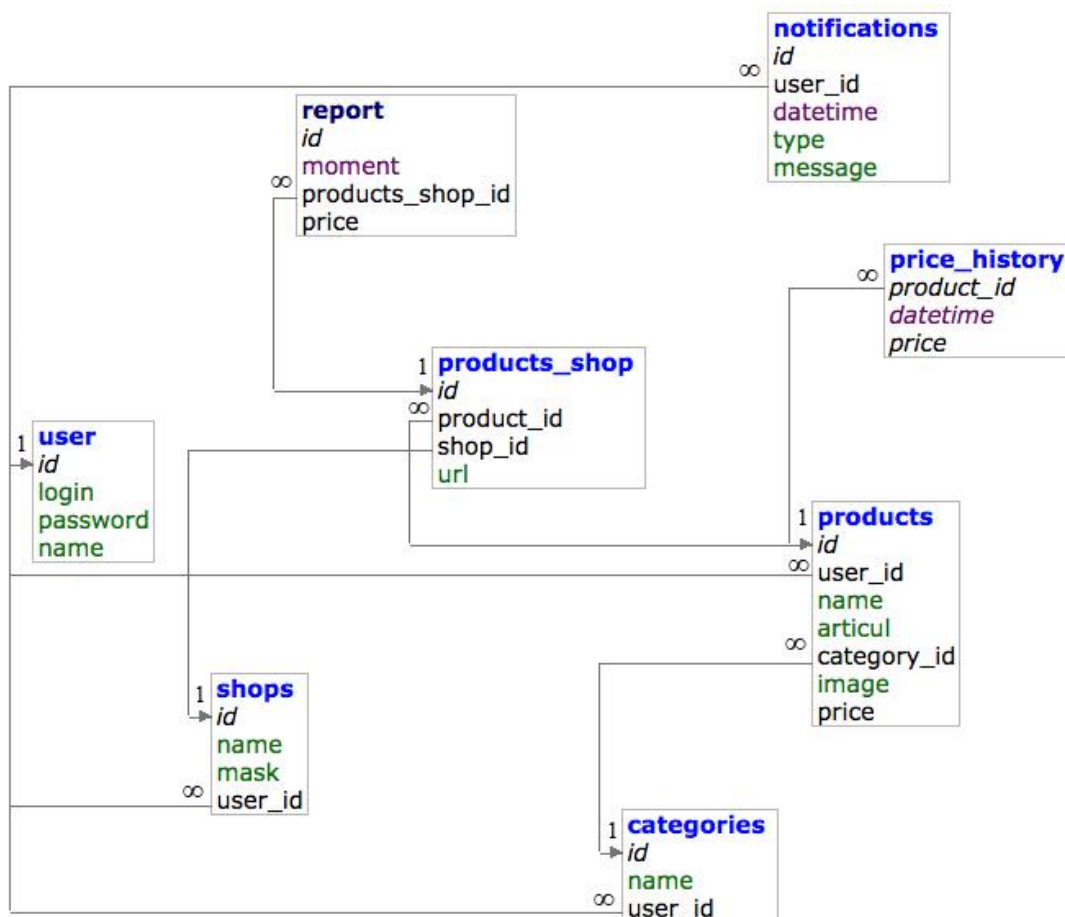


Рисунок 9 – Схема данных в СУБД.

База данных организована в виде реляционной табличной структуры и содержит следующие таблицы:

В системе используется восемь сущностей, которые имеют между собой связи. Сущность «User» будет содержать информационные данные для авторизации пользователя. В сущностях «Products», «Products_shop» и «Categories» содержатся справочные данные о товарах и категориях соответственно. Сущность «Report» содержит информацию о ценах для каждого товара за разный период времени. Сущность «Shops» содержит данные необходимые для поиска цен на сайтах конкурентов. Сущности «Notifications» и «Price_history» содержат справочные данные о уведомлениях и истории изменения цены внутри компании.

Сущность «User» имеет данные, представленные в таблице 5.

Таблица 5 – Данные о пользователе

Название атрибута	Тип данных	Длина в знаках	Ключевое поле
ID	Счетчик	11	Да
Login	Текстовый	50	Нет
Password	Текстовый	50	Нет
Name	Текстовый	50	Нет

Сущность «Products» имеет данные, представленные в таблице 6.

Таблица 6 – Данные о товаре

Название атрибута	Тип данных	Длина в знаках	Ключевое поле
ID	Счетчик	11	Да
User_id	Подстановка (внешний ключ)	11	Нет
Name	Текстовый	200	Нет
Articul	Текстовый	30	Нет
Category_id	Подстановка (внешний ключ)	11	Нет
Image (url)	Текстовый	1024	Нет
Price	Денежный	16,2	Нет

Сущность «Categories» имеет данные, представленные в таблице 7.

Таблица 7 – Данные о категории

Название атрибута	Тип данных	Длина в знаках	Ключевое поле
ID	Счетчик	11	Да

Продолжение таблицы 7

User_id	Подстановка (внешний ключ)	11	Нет
Name	Текстовый	200	Нет

Сущность «Shops» имеет данные, представленные в таблице 8.

Таблица 8 – Данные о магазине

Название атрибута	Тип данных	Длина в знаках	Ключевое поле
ID	Счетчик	11	Да
User_id	Подстановка (внешний ключ)	11	Нет
Name	Текстовый	200	Нет
Mask	Текстовый	1024	Нет

Сущность «Products_shop» имеет данные, представленные в таблице 9.

Таблица 9 – Данные о ссылках на товар для каждого магазина

Название атрибута	Тип данных	Длина в знаках	Ключевое поле
ID	Счетчик	11	Да
Product_id	Подстановка (внешний ключ)	11	Нет
Shop_id	Подстановка (внешний ключ)	11	Нет
Url	Текстовый	1024	Нет

Сущность «Price_history» имеет данные, представленные в таблице 10.

Таблица 10 – Данные о истории изменения цены для товара в компании

Название атрибута	Тип данных	Длина в знаках	Ключевое поле
Product_id	Подстановка (внешний ключ)	11	Нет
Datetime	Дата/время	-	Нет
Price	Денежный	16,2	Нет

Сущность «Report» имеет данные, представленные в таблице 11.

Таблица 11 – Данные о истории изменения цены для товара

Название атрибута	Тип данных	Длина в знаках	Ключевое поле
Id	Счетчик	11	Да
Moment	Дата/время	-	Нет
Products_shop_id	Подстановка (внешний ключ)	11	Нет
Price	Денежный	16,2	Нет

Сущность «Notifications» имеет данные, представленные в таблице 12.

Таблица 12 – Данные об уведомлениях для пользователя

Название атрибута	Тип данных	Длина в знаках	Ключевое поле
Id	Счетчик	11	Да
User_id	Подстановка (внешний ключ)	11	Нет
Datetime	Дата/время	-	Нет
Type	Текстовый	50	Нет

Message	Текстовый	200	Нет
---------	-----------	-----	-----

2.1.4. Характеристика результатной информации

После получения и последующий обработки входной информации, делается запись в таблицы, которые были описаны предыдущем разделе. На основании этих данных информация представлена в виде графиков и диаграмм. Таким образом можно сказать, что обработанная информация отображается графически и на основе этого отображения проводится анализ показателей, который наиболее удобен, а также нагляден для пользователей системы (Приложение Б).

В нашем случае, для построения графиков и диаграмм будет использована сторонняя JS библиотека Highchart. Она является бесплатной, очень гибко настраивается под любые данные, а также имеет мощный функционал управления.

Кроме этого, данная библиотека отлично вписывается в интерфейс разработанного приложения.

2.2. Описание программных модулей

Так как АИС будет делать запросы для поиска ценовой информации на сайты конкурентов, которые располагаются в сети интернет, то требуется реализовать программную часть системы как интернет-приложение. Таким образом, работник ИТ-отдела должен иметь на своем ПК только Web-браузер и выход в интернет для работы с системой. Стоит отметить, что приложение в сети интернет позволяет не привязываться к определенному ПК и если работнику необходимо проанализировать ситуацию на рынке, находясь вне офиса, например в командировке, он сможет легко это сделать.

Теперь необходимо выбрать программные средства для разработки системы, а также СУБД для хранения информации.

Для обработки данных, получаемых с БД и генерируемых страниц для пользователя целесообразно использовать язык серверных сценариев. Выбор языка стоит между ASP.NET от компании Майкрософт и PHP, который распространяется под собственной лицензией [12].

В ASP.NET можно писать код, используя практически любые языки программирования, входящие в комплект .NET Framework [4]. ASP.NET имеет преимущество в скорости по сравнению со скриптовыми технологиями, так как при первом обращении код компилируется и помещается в специальный кэш, и впоследствии только исполняется, не требуя затрат на парсинг, оптимизацию и т.д.

Преимущества ASP.NET:

- 1) компилируемый код выполняется быстрее, большинство ошибок отлавливается еще на стадии разработки;
- 2) ASP.NET опирается на многоязычные возможности .NET, что позволяет писать код страниц для VB.NET, Delphi.NET, Visual C#;
- 3) возможность кэширования всей страницы или её части для увеличения производительности;
- 4) расширяемая модель обработки запросов;
- 5) встроенная поддержка AJAX.

PHP представляет собой работающий на стороне сервера кроссплатформенный интерпретируемый язык программирования. Это один из популярнейших языков, благодаря своей простоте, скорости выполнения, богатой функциональности и распространению исходных кодов [5]. PHP поддерживает большинство известных баз данных среди которых Access и MySQL, различные интернет-протоколы, работу с файлами и множество других полезных функций.

Преимущества PHP:

- 1) автоматическое извлечение POST и GET-параметров, а также переменных окружения web-сервера и предопределенные массивы;

2) взаимодействие с большим количеством различных систем управления базами данных;

3) Работа с cookie и session.

4) Обработка файлов, загружаемых на сервер;

5) Работа с HTTP протоколом.

Из систем управления базами данных выбор стоит организовать между Microsoft SQL SERVER и MySQL. Каждая из этих СУБД по-своему хороша и имеет ряд преимуществ перед другим продуктом. Рассмотрим каждую из них.

Microsoft SQL SERVER в качестве языка запросов использует версию SQL, получившую название Transact-SQL (сокращённо T-SQL), являющуюся реализацией SQL-92 с множественными расширениями. T-SQL позволяет использовать дополнительный синтаксис для хранимых процедур и обеспечивает поддержку транзакций. Microsoft SQL SERVER используется для работы с базами данных размером от персональных до крупных баз данных масштаба предприятия.

MySQL является решением для малых и средних приложений. Гибкость СУБД MySQL обеспечивается поддержкой большого количества типов таблиц: пользователи могут выбрать как таблицы типа MyISAM, поддерживающие полнотекстовый поиск, так и таблицы InnoDB, поддерживающие транзакции на уровне отдельных записей. Более того, СУБД MySQL поставляется со специальным типом таблиц EXAMPLE, демонстрирующим принципы создания новых типов таблиц. Благодаря открытой архитектуре и GPL-лицензированию, в СУБД MySQL постоянно появляются новые типы таблиц.

Из приведенных выше программных средств и условий поставленной задачи был выбран язык серверных сценариев PHP. Главными причинами выбора этого языка была возможность хорошей работы с HTTP протоколами, работа с cookie и session, а также отличное взаимодействие с системами управления базами данными, в частности с MySQL. Также стоит отметить, что язык PHP работает по своей лицензии и он бесплатен, а это, несомненно, еще одно преимущество [11].

В выборе СУБД главным критерием была простота создания и управление таблицами и хорошая скорость при обмене данными с web-приложением. С этим отлично справится СУБД MySQL, которая отлично взаимодействует с PHP и может хранить в себе огромное количество записей, а также выполнять сложные SQL -запросы и быстро обрабатывать их [2].

2.2.1. Общие положения (дерево функций и сценарий диалога)

Построение АИС будет основано на трехуровневой архитектуре клиент-сервер.

Пользователь системы, должен получать через Интернет доступ к компонентам АИС на сервере, используя web-браузер, установленный на его компьютере (клиентская часть системы). В роли серверной части системы приложения будет выступать удаленный сервер, на котором установлено все необходимое ПО для получения, обработки и передачи информации клиенту приложения.

На рисунке 10 изображена архитектура приложения. Клиент с помощью веб-браузера будет делать запросы к серверу через формы приложения. Данные передаются через протокол https на сервер, где на основании полученных запросов с помощью CGI-обработчика обрабатываются и отправляются на сервер БД для получения необходимой информации [6]. Получив и обработав информацию от базы данных, сервер отправляет результат клиенту, который отображается в веб-браузере.

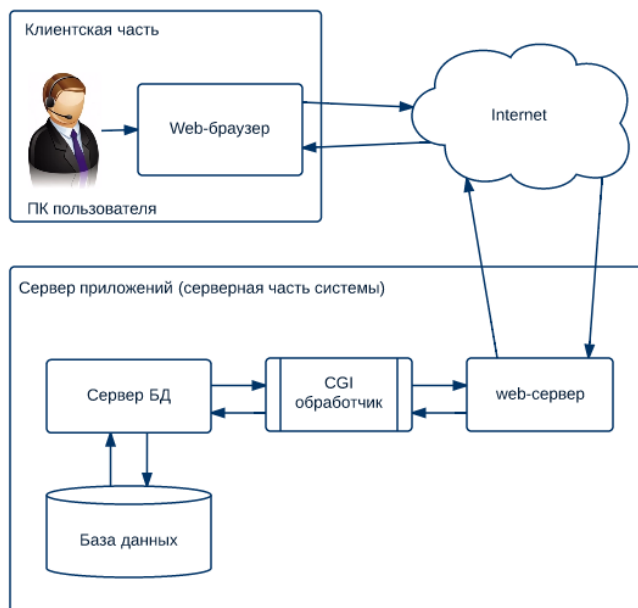


Рисунок 10 – Архитектура приложения клиент-сервер

2.2.2. Структурная схема программной системы (дерево вызова процедур и программных модулей)

Работа системы начинается с авторизации пользователя. В случае успешного прохождения авторизации пользователю открывается главная страница, с помощью которой можно получить доступ ко всем страницам системы и их модулям.

На рисунке 11 представлен общий алгоритм работы программного компонента системы.

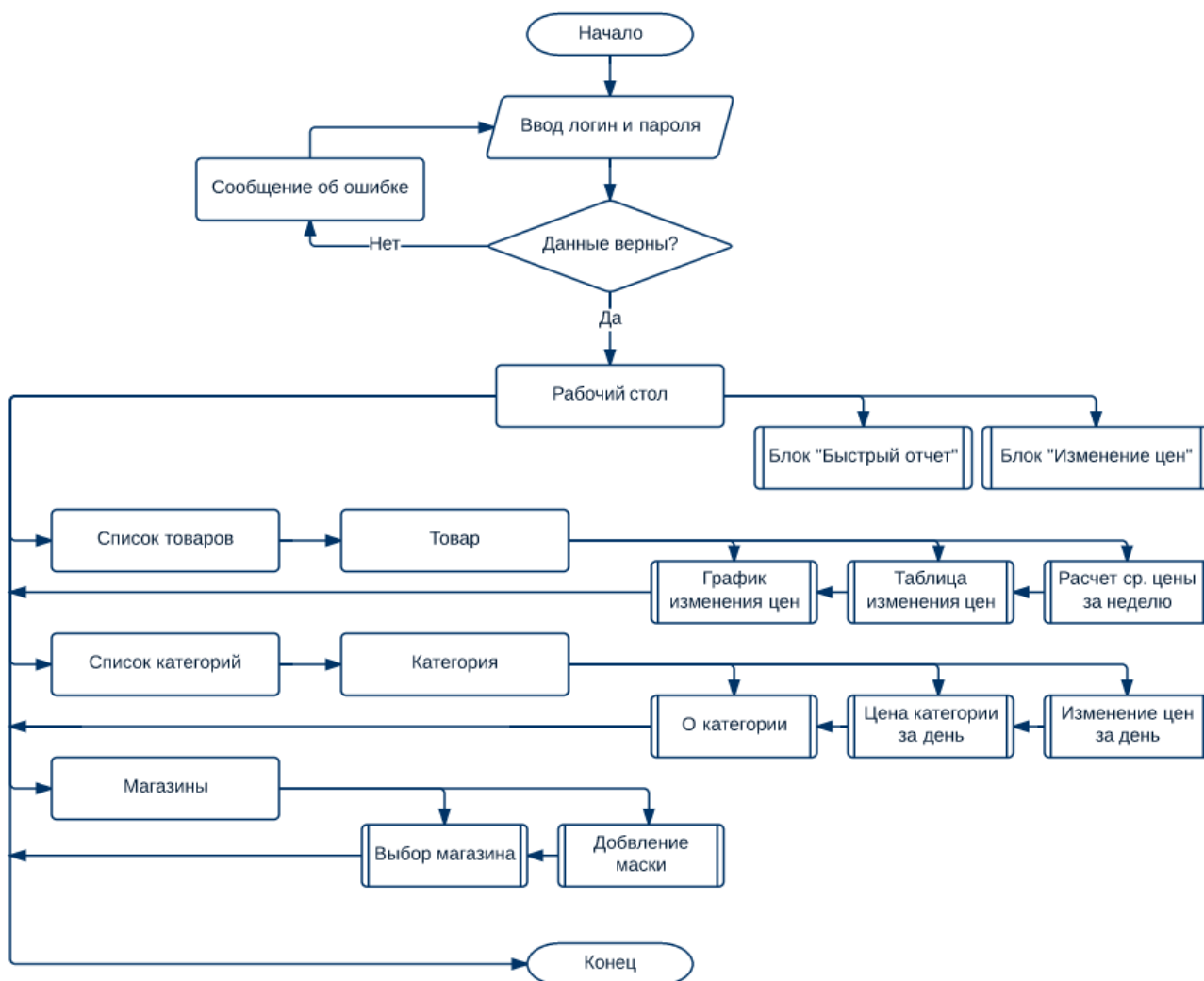


Рисунок 11 – алгоритм работы приложения

2.2.3. Описание программных модулей

В качестве основы проектирования лежат модули, благодаря которым приложение умеет собирать, обрабатывать и выдавать результаты пользователю. Рассмотрим некоторые из них:

Для авторизации пользователя в системе необходимо ввести логин и пароль. Если данные введены успешно, то будет произведен вход в личный

кабинет [10]. Если данные введены неверно, то будет выведено сообщение об ошибке. Форма модуля «Авторизация» изображена на рисунке 12.

The image shows a login form with a light gray background. At the top, there is a header with the text "Пожалуйста, войдите в систему". Below the header are two input fields: the first is labeled "Логин" and the second is labeled "Пароль". At the bottom of the form is a large green button with the text "Войти" in white.

Рисунок 12 – форма авторизации

После ввода логина и пароля пользователь кликает на кнопку «Войти», после чего запускается обработчик `auth()`, код этого события приведен ниже:

```
public function auth($login, $password) {
    $identity = new UserIdentity($login, trim($password));
    if ($identity->authenticate()) {
        Yii::app()->user->login($identity, Yii::app()->user->allowAutoLogin ?
        $duration : 0);
    }
    return Yii::app()->createUrl("/dashboard/");
}

public function authenticate() {
    try {
        $user = Yii::app()->userService->getUserByLogin($this->username);
        $user->verifyPassword($this->password);
    } catch (CdbException $ex) {
        throw new Exception('База данных недоступна. Попробуйте войти
        позже.');
```

```
    } catch (Exception $ex) {
        throw new Exception('Неправильный логин или
        пароль.<br>Проверьте раскладку клавиатуры и режим "Caps Lock".');
```

```

$this->_id = $user->getId();
$this->errorCode = self::ERROR_NONE;
return true;
}

```

После успешной авторизации пользователю доступна форма добавления товара. Добавив товар один раз, система будет ежедневно пытаться получить цену по этому товару в ссылках, указанных в форме. Форма добавления товара изображена на рисунке 13.

Добавить товар

Рисунок 13 – форма для добавления товара

После нажатия на кнопку «Добавить товар» приложение обрабатывает введенные данные программным кодом, который описан ниже:

```

public function create($name, $category, $articul, $image, $price, $urls) {
    $urls = (array) $urls;
    $urls = array_filter($urls, function($value) { return $value !== ""; });
    if(empty($urls)) {
        throw new Exception("Не одна ссылка на магазин не найдена!");
    }
    $category_id = $category->id;
}

```

```

        $product      =      $this->createEntity("Products")->create(trim($name),
$category_id, $articul, $image, floatval($price));
        foreach ($urls as $url) {
            $this->createShopForProduct($url, $product->getId());
        }
        return $product;
    }

    public function create($name, $category, $articul, $image, $price) {
        $this->name = $name;
        $this->category = $category;
        $this->articul = $articul;
        $this->image = $image ?? NULL;
        $this->user = Yii::app()->user->id;
        $this->setPrice($price);
        return $this->modify();
    }

    public function createShopForProduct($url, $id) {
        $shop = ShopHelper::getOrCreate($url);
        ShopHelper::createShopProduct($shop->getId(), (int) $id, $url);
    }

    public static function createShopProduct($shop_id, $product_id, $url) {
        return Yii::app()->db->createCommand("INSERT IGNORE INTO
products_shop (`product_id`, `shop_id`, `url`) VALUES ($product_id, $shop_id,
'$url')")->execute();
    }

```

Модуль «Сканирование цен» - главный модуль системы. Делает запросы на URL-ссылки товаров и извлекает цену и добавляет её в базу данных системы для последующей обработки и вывода информации пользователю.

```

protected function execute() {
    # Начинаем пробегать по каждому товару.
    foreach($this->products as $product) {
        #mask
        if(!$product->mask) {
            $this->logger("id" . $product->id. " нет маски. Пропущен");
            continue;
        }
        # curl
        $curl = curl_init($product->url);

```



```

        curl_setopt($curl, CURLOPT_RETURNTRANSFER, 1);
        curl_setopt($curl, CURLOPT_FOLLOWLOCATION, 1);
//следование 302 redirect
        $html = curl_exec($curl);
        curl_close($curl);

        phpQuery::newDocument($html);
        $get = pq($product->mask->text());
        $price = (float) trim(preg_replace('/[^0-9,\.\.]/', '', $get));
        Yii::app()->db->createCommand("INSERT INTO report (`moment`,
products_shop_id, `price`) VALUES (NOW(), {$product->id}, '$price')")->execute();
        $this->logger("id" . $product->id. " получена цена.");
        phpQuery::unloadDocuments();
    }
    Yii::consolelog(implode("", $this->logger));
}

```

Вывод цен в виде графика возможен после получения и обработки из базы данных. Для этого используется следующий программный код:

```

        $data = Yii::app()->db->createCommand("(SELECT DATE(moment) as `date`,
price, name, s.id FROM report as r
        JOIN products_shop as ps on r.products_shop_id = ps.id
        JOIN shops as s ON s.id = ps.shop_id
        WHERE ps.product_id = :id ORDER BY 1 ASC)
        UNION ALL
        SELECT DATE(`datetime`) as `date`, price, "" . Yii::app()->params['name']
. "" as name, 0 as id FROM price_history WHERE product_id = :id and `datetime` in
(select max(`datetime`) FROM price_history WHERE product_id = :id GROUP BY
date(`datetime`))")->bindValue(":id", $id, PDO::PARAM_INT)->queryAll();
        if(!$data) {
            throw new Exception("Нет данных для визуализации");
        }
        $minDate = $maxDate = Date::create("now")->format('Y-m-d');
        foreach($data as $row) {
            if($minDate > $row['date']) $minDate = $row['date'];
            if($maxDate < $row['date']) $maxDate = $row['date'];
        }
        $period = new DateTimeRange($minDate, $maxDate);
        $shops = [];
        foreach($data as $v) {
            $shops[] = $v['name'];
        }
    }
}

```

```

$shops = array_unique($shops);

$result = [];
foreach($shops as $shop) {
    $lastPrice = 0;
    $prices = ArrayHelper::filterColumn($data, ['name' => $shop]);
    $result[$shop]['name'] = $shop;
    $result[$shop]['tooltip'] = ['pointFormat' => " {series.name}: {point.y}
рублей"];
    for($cur = $period->getDateStart()->cloneable(); $cur->getTimestamp()
<= $period->getDateEnd()->getTimestamp(); $cur->modify("+1 day")) {
        $price = $lastPrice;
        $today = ArrayHelper::filterColumn($prices, ['date' => $cur-
>format('Y-m-d')]);
        if(!empty($today)) {
            $price = $lastPrice = $today[0]['price'];
        }
        $result[$shop]['data'][$cur->format('d.m')] = ["name" => $cur-
>format('d.m'), "y" => (float) $price];
    }
    ksort($result[$shop]['data']);
    $result[$shop]['data'] = array_values($result[$shop]['data']);
}
return array_values($result);

```

Результатом выполнения этого кода является форма, изображенная на рисунке 14, выполненная в виде графика для отображения изменения цены на товар за период времени.

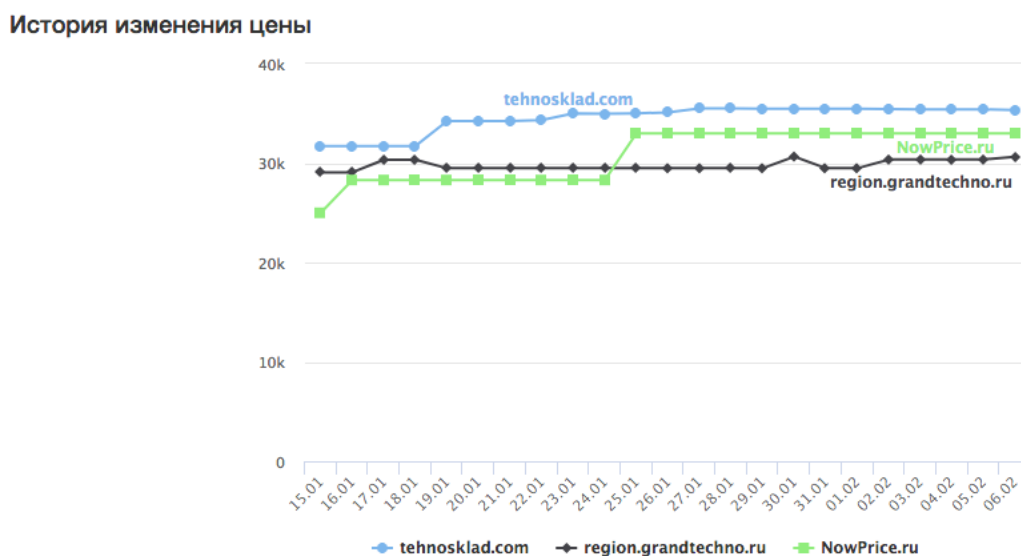


Рисунок 14 – Форма графика изменения цены

2.3. Технологическое обеспечение задачи

Данный пункт включает в себя описание организации технологии сбор, передачи, обработки, а также выдачи информации. Кроме этого, отражается последовательность операций, начиная от способа сбора информации и заканчивая формированием результатной информацией и способами его передачи.

2.3.1. Организация технологии сбора, передачи, обработки и представления информации

Для организации сбора информации используются формы ввода в приложении. Затем информация обрабатывается и заносится в БД. Если информация предполагает сравнение данных ввода с иными данными, например авторизация, то для такой информации по ключу извлекаются записи из таблиц, сравниваются с введенными данными в форме и выводится результат на страницу пользователя.

Модуль сбора информации для поиска изменения цен выполняет главную роль в системе. Ежедневно обходя сайты конкурентов и заносая полученные данные в БД. В основе алгоритма лежит работа со списками. Блок-схема выполнения алгоритма описана в рисунке 15.

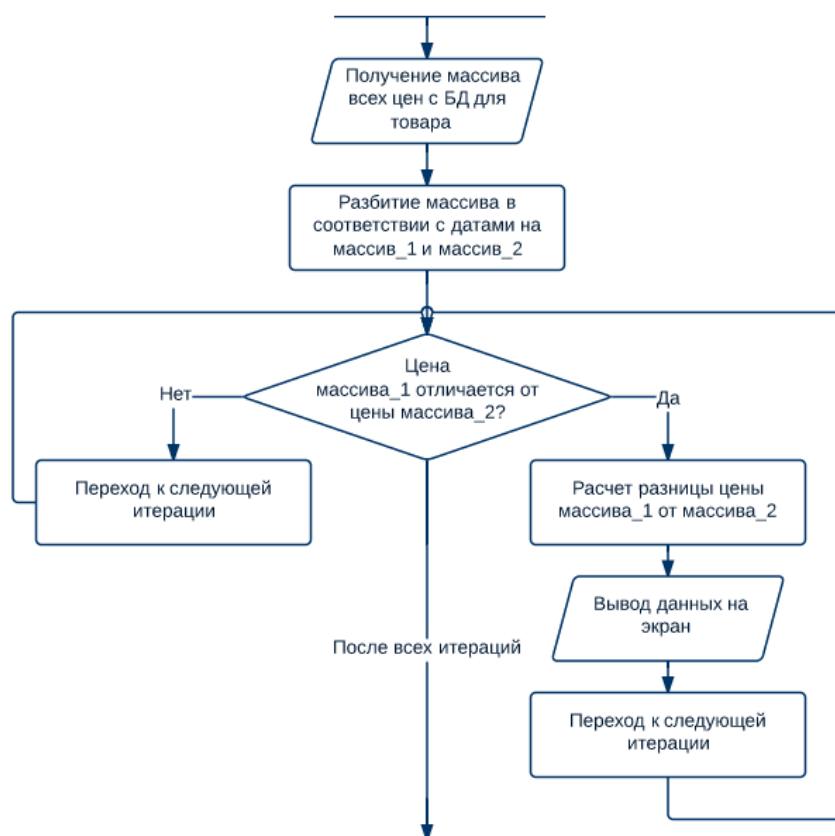


Рисунок 15 – алгоритм поиска изменения цен.

Модуль разбивает общий массив на два, каждый из которых имеет одинаковую дату за сегодня и вчера. Условие проверяет, отличается ли цена сегодня от цены вчера. Если нет, то алгоритм переходит к следующей итерации. Если да, то рассчитывается разница между ценами и данные выводятся на экран, после этого алгоритм также переходит к следующей итерации.

Для вывода графиков используется модуль вывода графиков. После запроса к базе данных, алгоритм обрабатывает полученные данные с ценами и начинает выполнять условие: если цена принадлежит компании-владельцу системы, то алгоритм формирует данные отдельно, рассчитывая разницу между другими магазинами и компанией-владельцем, иначе данные формируются без расчета разницы. Алгоритм работы описан на рисунке 16.

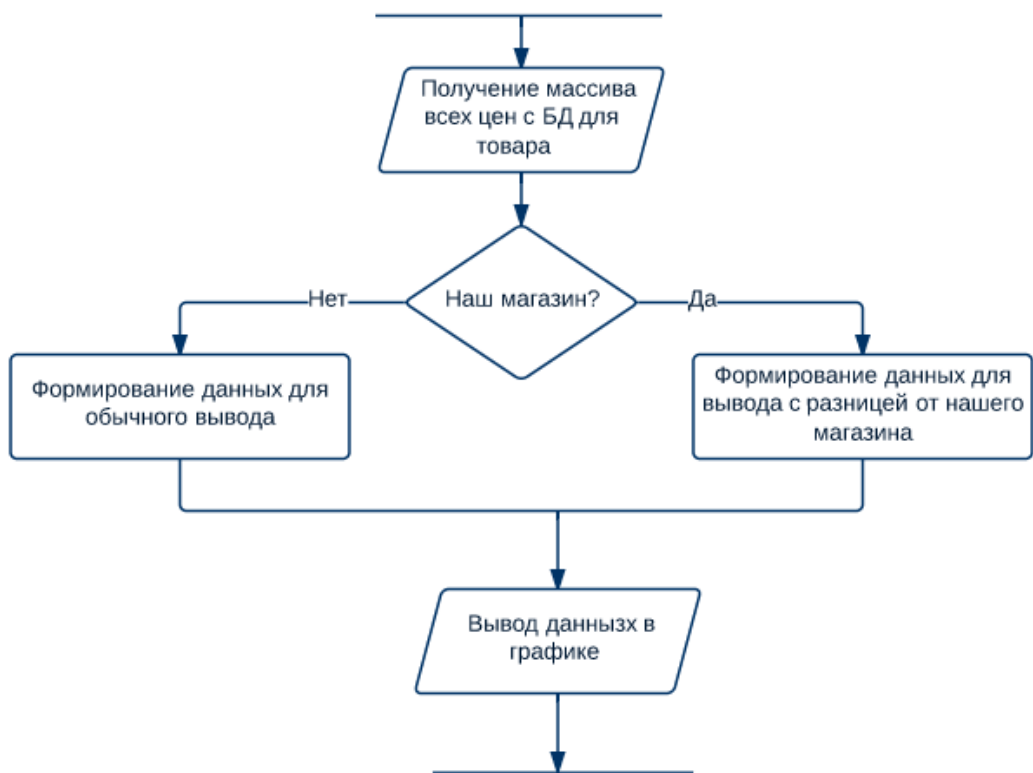


Рисунок 16 – блок-схема работы алгоритма вывода графика.

3. ОБОСНОВАНИЕ ЭКОНОМИЧЕСКОЙ ЭФФЕКТИВНОСТИ ПРОЕКТА ИС

3.1. Эффективность применения информационных технологий

Под эффективностью мы понимаем отношение произведенных действий к определенным целям. Эффект – это результат внедрения какого-либо мероприятия, выраженный в стоимостной форме в виде экономии от его осуществления. Сегодня информационные технологии внедряются практически везде, и они имеют большое количество отличительных средств.

Чтобы иметь возможность судить об эффективности применения того или иного программного средства необходимо иметь численный показатель или критерий, характеризующий степень качества выполнения системой своих функций. Общими требованиями к критерию эффективности относят:

- а) вычисляемость, то есть получение значения в виде числа;
- б) простота, то есть легкость и доступность понимания;
- в) наглядность, то есть оценка должна быть очевидна.

При этом критерий должен быть:

- а) полным, то есть охватывать все важнейшие аспекты проблемы;
- б) минимальным, то есть используется минимум информации;
- в) действенным, то есть способствовать достижению поставленной цели;
- г) разложимым, то есть обеспечивать получение оценки путем разбиения

на части.

Все критерии эффективности можно условно разделить на 2 группы:

- а) критерии комплексной экономической оценки для больших и сложных систем, такие как отрасль экономики;
- б) критерии, представляющие собой соотношение, полученной от внедрения какого-либо объекта прибыли к тем или иным затратам.

В большой части методик расчета экономической эффективности от применения информационных технологий используются критерии второй группы, так как они более наглядны, а методика их расчета менее трудоемка. Оценка эффективности производится путем сопоставления затрат труда и средств производства, с одной стороны, и получаемых результатов, с другой. Основным принципом расчета экономической эффективности в этом дипломном проекте будет являться сравнение двух вариантов обработки информации – ручной обработки данных и обработки данных с использованием ЭВМ. Основными факторами, выгодно отличающие автоматизированную обработку данных от ручной могут являться:

- а) высокая скорость выполнения операций по сбору, передаче, обработке и выдаче информации;
- б) своевременное получение полной и достоверной информации о состоянии объекта управления;
- в) применение совершенных методов планирования и управления, экономико-математических методов и ЭВМ;
- г) повышение качества управления объектом за счет устранения дублирования и несопоставимости данных.

От перечисленных выше факторов зависит снижение трудовых, материальных и финансовых затрат, повышение эффективности функционирования экономического объекта.

Эффект от применения ИТ может быть прямым, поддающимся прямому счету, и косвенным, не поддающимся прямому счету и выявляемому опосредствовано. Источники эффективности могут быть прямыми и косвенными. Они в свою очередь делятся на измеримые и неизмеримые.

3.2. Расчет показателей экономической эффективности проекта ИС

Программные средства – это очень специфический продукт. Они характеризуются высокими темпами морального старения, высокими затратами на разработку и незначительными затратами на их распространение и тиражирование, поэтому при расчете эффективности и также выборе различных коэффициентов эти свойства следует учитывать. Срок окупаемости ($T_{ок}$) в годах и коэффициент эффективности капитальных вложений (E) связаны обратно пропорциональной зависимостью.

Методика оценки эффективности основана на расчете прямого сравнительного анализа от использования. Сравнивается ручной способ обработки данных и использование сайта, то есть машинная обработка информации. Предполагается произвести расчет следующих основных показателей:

- а) годовая экономия затрат на обработку информации при использовании программно–технического комплекса;
- б) единовременные затраты на модернизацию и внедрение программно–технического комплекса назначения выплат пенсий;
- в) срок окупаемости.

Экономия затрат на обработку информации, связанная с разработкой и внедрением ПС ВТ, определяются по формуле:

$$C = C_p - C_m, \quad (3.1)$$

Определим каждое значение:

- C - годовая экономия затрат на обработку информации, связанная с внедрением ППП;
- C_p - затраты на подготовку и обработку информации в базовом варианте, т.руб.;

- C_m - затраты на обработку информации при внедрении ПС ВТ, т.руб.

Показатели, входящие в предыдущую формулу, определяются следующим образом:

$$C_p = (Q_{вх,б} + Q_{вых,б}) \times C_p \times \Gamma_d / N_v, \quad (3.2)$$

Определим каждое значение:

- $Q_{вх,б}$, $Q_{вых,б}$ - объем входной и выходной информации, обрабатываемой в базовом варианте соответственно, в тыс.зн.;
- C_p - стоимости одного часа ручной обработки информации, руб./час;
- Γ_d - коэффициент, учитывающий дополнительные затраты времени на логические операции при ручной обработке информации;
- N_v - норма выработки, зн./час.

Стоимость одного часа ручной обработки информации определяется следующим образом:

$$C_p = Z_m \times W / T, \quad (3.3)$$

Определим каждое значение:

- Z_m - среднемесячная заработная плата пользователя, руб.;
- W - коэффициент начисления на заработную плату;
- T - среднемесячный фонд рабочего времени, час.

Затраты на обработку информации при использовании внедренного ПС ВТ определяют следующим образом:

$$C_m = C_p + C_{об}, \quad (3.4)$$

Определим каждое значение:

- C_p - затраты на подготовку информации для реализации функций, автоматизированных в ПС ВТ, т.руб;
- $C_{об}$ - затраты на машинное время для реализации функций, автоматизированных в ПС ВТ, т.руб.

Компоненты, входящие в последнюю формулу, определяются следующим образом:

$$C_{\Pi} = (C_{\text{р}} + C_{\text{мч}}) \times Q_{\text{вх, н}} / N_{\text{выр, а}}, \quad (3.5)$$

Определим каждое значение:

- $Q_{\text{вх, н}}$ - объем входной информации, обрабатываемой с помощью ПС ВТ (новый вариант), т.зн;
- $N_{\text{выр, а}}$ - норма выработки пользователя при подготовке и вводе информации в ЭВМ, тыс. зн./час;
- $C_{\text{мч}}$ - стоимость одного машинного часа работы ЭВМ, руб.

$$C_{\text{об}} = T_{\text{м}} \times C_{\text{мч}}, \quad (3.6)$$

Определим каждое значение:

$T_{\text{м}}$ - затраты машинного времени, необходимого для реализации функций, автоматизированных с помощью ПС ВТ, час.

$$T_{\text{м}} = (Q_{\text{вх, н}} + Q_{\text{вых, н}}) \times T_{\text{з}}, \quad (3.7)$$

Определим каждое значение:

- $Q_{\text{вых, н}}$, $Q_{\text{вх, н}}$ - объем выходной и входной информации, получаемой при использовании ПС ВТ, т.зн.;
- $T_{\text{з}}$ - среднее количество часов для обработки 1000 знаков с использованием ПС ВТ, час/т.зн.

Единовременные затраты на создание и внедрение ПС ВТ рассчитываются по формуле:

$$K = K_{\text{п}} + K_{\text{к}}, \quad (3.8)$$

Определим каждое значение:

- K - единовременные затраты на создание и внедрение ПС ВТ с учетом фактора времени, т. руб.;
- $K_{\text{п}}$ - производственные затраты (затраты на разработку или поставку и привязку ПС ВТ), т. руб.;
- $K_{\text{к}}$ - капитальные вложения, необходимые для реализации ПС ВТ, т.руб.

Внедрение ПС ВТ будет эффективным, если расчетный коэффициент эффективности капитальных вложений (E_p) будет больше выбранного граничного значения:

$$E_p = C/K, \quad (3.9)$$

Срок окупаемости капитальных вложений ($T_{ок}$) рассчитывается по формуле:

$$T_{ок} = K/C. \quad (3.10)$$

Произведем расчет экономической эффективности внедрения программно–технического комплекса анализа цен конкурентов. Исходные данные приведены в таблице 13.

Таблица 13 – Данные для расчета экономической эффективности

№ п/п	Наименование показателя	Базовый вариант	Новый вариант
1	2	3	4
1	Предпроизводственные затраты, тыс. руб. - K_p		30.00
2	Капитальные вложения, тыс. руб. - K_k		70.00
	Годовой объем информации, тыс. зн.:		
3	входной - $Q_{вх}$	712.00	950.00
4	выходной - $Q_{вых}$	125.00	160.00
5	Норма выработки при ручной обработке информации, зн./час - $N_{выр.р}$	700.00	
6	Норма выработки при подготовке данных на машинный носитель, тыс. зн./час - $N_{выр.а}$		5.00
7	Коэффициент, учитывающий дополнительные затраты времени на логические операции при ручкой обработке информации - Γ_d	1.10	
8	Среднемесячная зарплата пользователя, руб. - Z_m	25000.00	28000.00
9	Коэффициент начисления на заработную плату - W	1.20	1.20
10	Коэффициент, учитывающий накладные расходы - K_n	1.80	1.80
11	Среднемесячный фонд рабочего времени, час - T	167.80	167.80
12	Затраты времени для обработки 1000 зн. и вывода информации с использованием ПС ВТ, час/тыс. зн. - T_z		0.50

13	Стоимость одного машинного часа работы ЭВМ, руб./час - Ц _м		50.00
14	Нормативный коэффициент эффективности капитальных вложений - Е		1.5

По формуле 3.3 произведем расчет стоимости одного часа ручной обработки информации (Ц_р):

$$Ц_r = 25000 \times 1,2 * 1,8 / 167.8 = 321.8 \text{ руб/час.}$$

По формуле 3.2 определим затраты на подготовку и обработку информации в базовом варианте (С_р):

$$C_p = (712,0+125,0) \times 321.8 \times 1,1 / 700 = 423.25 \text{ т. руб.}$$

По формуле 3.5 определим затраты на подготовку информации и ввод информации для реализации функций, автоматизированных в ПС ВТ (С_п):

$$C_{п} = (321.80 + 50,0) \times 950,0 / 5,0 = 70642 \text{ руб.}$$

По формуле 3.7 определим затраты машинного времени, необходимые для реализации функций, автоматизированных с помощью ПС ВТ (Т_м):

$$T_m = (950,0 + 160,0) \times 0,5 = 555 \text{ час.}$$

По формуле 3.6 затраты на машинное время для реализации функций, автоматизированных в ПС ВТ (С_{об}):

$$C_{об} = 555 \times 50,0 = 27750.00 \text{ руб.}$$

По формуле 3.4 определим затраты на обработку информации при использовании внедренного ПС ВТ (С_м):

$$C_m = 70642 + 27750 = 98392. \text{ руб.} = 98.392 \text{ т. руб.}$$

По формуле 3.1 определим годовую экономию затрат на обработку информации, связанную с внедрением ПС ВТ (С):

$$C = 423.25 - 98.392 = 324.858 \text{ т. руб.}$$

По формуле 3.8 определим единовременные затраты на создание и внедрение ПС ВТ (К):

$$K = 30.00 + 70.00 = 100.00 \text{ т. руб.}$$

По формуле 3.9 определим коэффициент эффективности капитальных вложений (Е_р):

$$E_p = 324.858 / 100.00 = 3,25 > 1,5,$$

По формуле 4.10 определим срок окупаемости внедряемого программного средства ($T_{ок}$):

$$T_{ок} = 100.00 / 324.858 = 0,307 \text{ года.}$$

Произведенные расчеты показали, что внедрение программно-технического комплекса автоматизации сбора цен конкурентов является экономически выгодным и оправданным, поскольку расчетный коэффициент эффективности капитальных вложений больше выбранного граничного значения ($3,25 > 1,5$). Он позволит снизить затраты ручного труда и повысит качество и скорость получаемой информации, и ее последующую обработку.

ЗАКЛЮЧЕНИЕ

В рамках выпускной квалификационной работы была разработана система для анализа цен на товары бытовой техники и электроники и выполнена реализация программной части, представленная в сети Интернет по адресу <http://lifemiie.beget.tech>.

В проектной части отражены все этапы, начиная с предпроектных обследований, концепции системы и алгоритмов работы системы, также было проведено описание модулей, из которых состоит система.

В итоге был получен программно-информационный компонент, выполняющий все функциональные задачи. Также была разработана сопроводительная документация, включающая инструкцию по сопровождению АИС для специалиста и инструкцию по эксплуатации для пользователя, которым является работник ИТ-отдела.

В дальнейшем система может быть усовершенствована за счет добавления модуля регистрации, целью которого является многопользовательская работа в системе нескольких специалистов ИТ-отдела. А также модуль экспорта данных в формате .PDF для отправки отчетов по электронной почте в головной офис компании или другие филиалы.

На данный момент разработки программный компонент находится в опытной эксплуатации. Анализ результатов эксплуатации показывает, что значительно сократилось время на поиск цен конкурентов, расчет средних значений, анализ и принятие решений, что позитивно отразилось на компании. Сократились ошибки персонала при поиске цены конкурентов до нулевого значения. Оптимизирована ценовая политика компании для цен, расположенных в интернет-магазине. Таким образом, можно сделать вывод о том, что цели, поставленные в начале предпроектных работ, достигнуты.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Веллинг, Л. Разработка веб-приложений с помощью PHP и MySQL: /
2. Л. Веллинг, Л. Томсон. – Пер. с англ. Издательство Вильямс, 2011. – 447с.
3. Кевин, Я. PHP и MySQL от новичка к профессионалу: книга/ Я. Кевин. – Пер. с англ. Издательство Эксмо, 2013. – 384с.
4. Мазуркевич «PHP настольная книга программиста /Александр Мазуркевич, Дмитрий Еловой. — Мн.: Новое знание, 2013. — 480 с.: ил. »
5. Макконнелл, С. Совершенный код: книга/С. Макконнелл. - Издательство Питер, Русская Редакция, 2010. – 896с.
6. Хопкинс, К. PHP. Быстрый старт: / К. Хопкинс. – Пер. с англ. Михаил Райтман, 2014. – 160с.
7. ГОСТ 34.601-90 Автоматизированные Системы. Стадии создания. СССР – 1992. – 6с.
8. Шполянская И.Ю. Объектные методы моделирования процессов управления в бизнес-системах с использованием UML Ration Rose: учеб. Пособие/РГЭУ «РИНХ». – Ростов н/Д., - 2004.
9. ГОСТ 19.106-76 ЕСПД- Общие требования к программной документации.
10. РД 50 – 34.698 –90– «Методические указания. Информационная технология. Комплекс стандартов и руководящих документов на автоматизированные системы. Автоматизированные системы. Требования к содержанию документов».
11. PHP: Hypertext Preprocessor [Electronic resource] / Rasmus Lerdorf, 1995. – Mode of access: <http://php.net>. – Date of access: 19.05.2015
12. PHP.SU: Functions and classes [Electronic resource] / Rasmus Lerdorf, 1995. – Mode of access: <http://php.su>. – Date of access: 04.03.2015

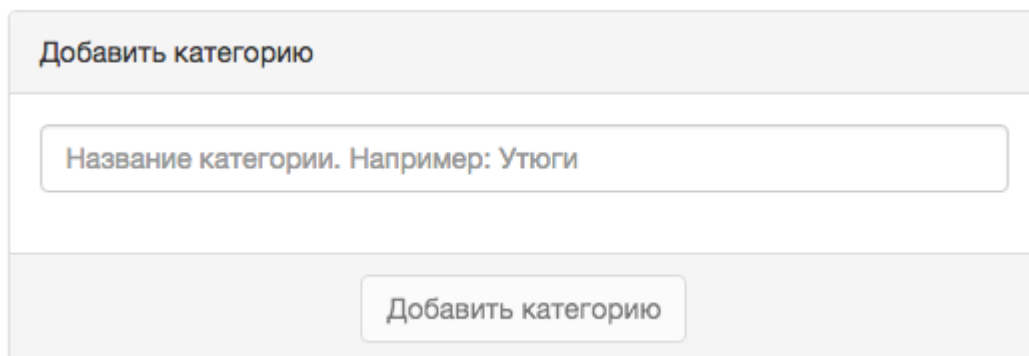
ПЕРЕЧЕНЬ СОКРАЩЕНИЙ И УСЛОВНЫХ ОБОЗНАЧЕНИЙ

- 1) АИС – Автоматизированная информационная система.
- 2) БД – база данных
- 3) ПС – программные средства
- 4) ИТ – информационные технологии
- 5) СУБД – система управления базами данных
- 6) ООО – общество с ограниченной ответственностью

Приложения

(с. 64 – 79)

ПРИЛОЖЕНИЕ А. ФОРМЫ ВХОДНЫХ ДАННЫХ



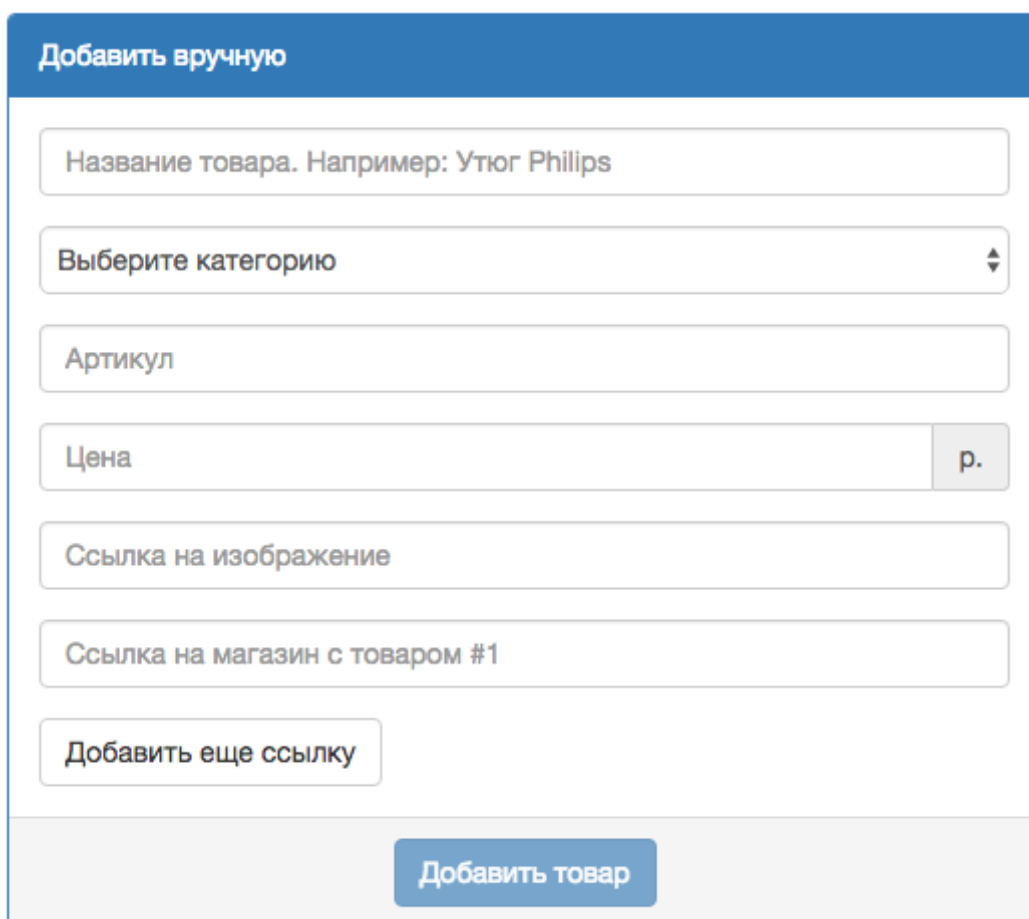
Добавить категорию

Название категории. Например: Утюги

Добавить категорию

The form is titled 'Добавить категорию' (Add category). It features a text input field with the placeholder text 'Название категории. Например: Утюги' (Category name. Example: Irons). Below the input field is a button labeled 'Добавить категорию' (Add category).

Рисунок А.1 – Добавление категории товаров



Добавить вручную

Название товара. Например: Утюг Philips

Выберите категорию

Артикул

Цена р.

Ссылка на изображение

Ссылка на магазин с товаром #1

Добавить еще ссылку

Добавить товар

The form is titled 'Добавить вручную' (Add manually). It contains several input fields: 'Название товара. Например: Утюг Philips' (Product name. Example: Philips Iron), 'Выберите категорию' (Select category) with a dropdown arrow, 'Артикул' (SKU), 'Цена р.' (Price in rubles), 'Ссылка на изображение' (Image link), and 'Ссылка на магазин с товаром #1' (Store link with product #1). There is a button 'Добавить еще ссылку' (Add another link) and a final blue button 'Добавить товар' (Add product).

Рисунок А.2 – Добавление товаров для анализа

Магазины

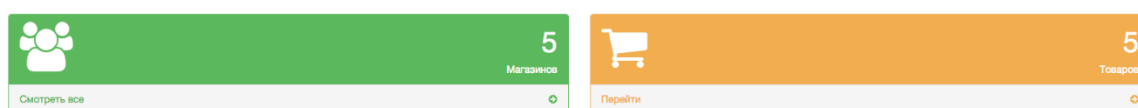
tehnosklad.com	
Количество товаров	
4	
Маска для поиска цены	
	<input type="text" value=".price-value h2"/>
<input type="button" value="Сохранить"/>	

region.grandtechno.ru	
Количество товаров	
3	
Маска для поиска цены	
	<input type="text" value=".price-current strong"/>
<input type="button" value="Сохранить"/>	

Рисунок А.3 – Добавление настроек поиска на сайте конкурентов

ПРИЛОЖЕНИЕ Б. ФОРМЫ ВЫХОДНЫХ ДАННЫХ

Рабочий стол



Количество добавленных товаров в магазин

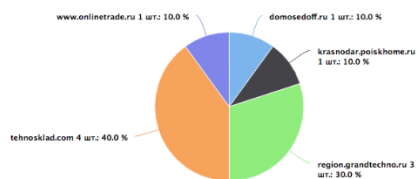


Рисунок Б.1 – Вывод данных по количеству товаров для каждого магазина



Рисунок Б.2 – Вывод данных по количеству товаров в категории

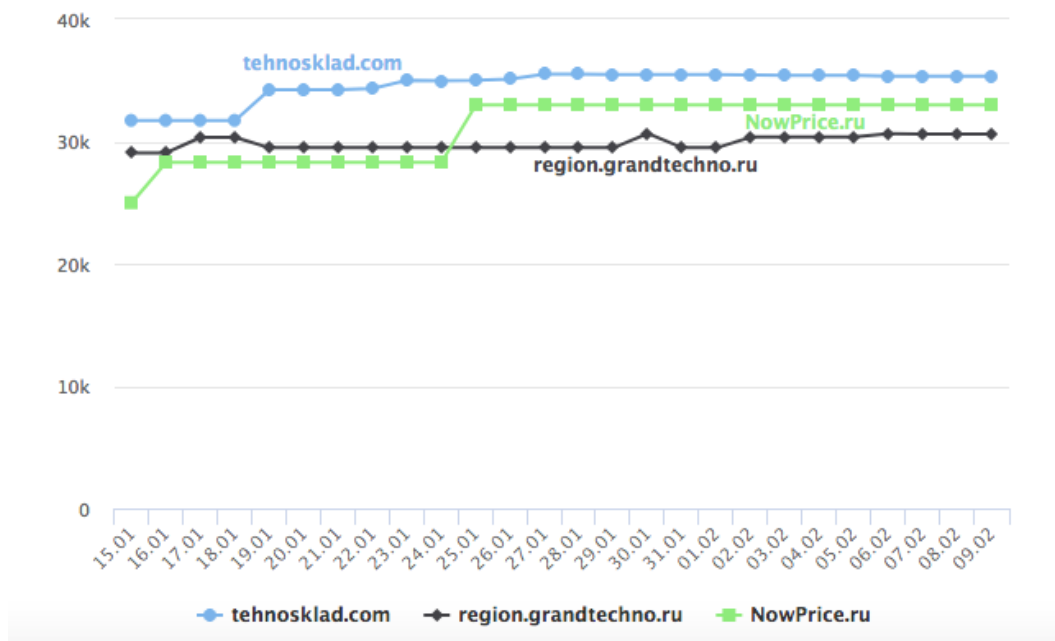


Рисунок Б.3 – Вывод данных по ценам для товара

Информация о товаре		Цена	Категория
Id	Артикул	33000	Холодильники
1	GA-B419		
Изменить			
Магазины для товара			
Название	URL	Действия	
tehnosklad.com	http://tehnosklad.com/108_Kholodilniki/84978_Kholodilnik_lg_ga_b419e j	x	
region.grandtechno.ru	https://region.grandtechno.ru/magazin/product/lg-ga-b-419-s j	x	

Рисунок Б.4 – Вывод данных определенного товара в карточке

ПРИЛОЖЕНИЕ В. ЛИСТИНГ ПРОГРАММЫ

```
<?php

class EntityManager extends CComponent {

    private $repositorys = array();
    private $map;

    public function init() {

    }

    private function getRepository($clazz) {
        if (!$clazz instanceof Reflections) {
            $clazz = call_user_func(array(Yii::import($clazz), 'refClass'));
        }

        $discriminator = $clazz->getAnnotation()->getParameter("discriminator");
        if ($discriminator != null) {
            $clazz = new Reflections($clazz->getParentClass()->getName());
        }

        $name = $clazz->getName();
        if (false === isset($this->repositorys[$name])) {
            $repository = $clazz->getAnnotation()->getParameter("repository");
            if ($repository != null) {
                $repositoryName = Yii::import($repository);
                $this->repositorys[$name] = new $repositoryName();
            } else {
                $this->repositorys[$name] = new SingleTableRepository($clazz);
            }
        }

        return $this->repositorys[$name];
    }

    public function count($clazz, $criteria) {
        return $this->getRepository($clazz)->count($criteria);
    }

    public function find($clazz, CDbCriteria $criteria) {
```

```

    $obj = $this->getRepository($clazz)->find($criteria);
    if ($obj === null) {
        throw new NoResultException("NoResultException");
    }
    return $obj;
}

public function findAll($clazz, $criteria) {
    return $this->getRepository($clazz)->findAll($criteria);
}

public function findByPk($clazz, $primaryKey) {
    if($primaryKey == null) {
        throw new NoResultException;
    }
    $obj = $this->getRepository($clazz)->findByPk($primaryKey);
    if ($obj === null) {
        throw new NoResultException();
    }
    return $obj;
}

    public function findByKey($uniqueKey) {
        list($key, $id) = explode(':', $uniqueKey);
        $config = $this->getConfigByKey($key);
        return $this->findByPk($config['class'], $id);
    }

public function persist(AppObject $obj) {
    return $this->getRepository($obj::refClass())->persist($obj);
}

public function insert(AppObject $obj) {
    return $this->getRepository($obj::refClass())->insert($obj);
}

public function update(AppObject $obj, $id) {
    return $this->getRepository($obj::refClass())->update($obj, $id);
}

public function remove(AppObject $obj) {
    $this->getRepository($obj::refClass())->remove($obj);
}

    protected function getConfigByKey($key) {

```

```

        foreach ($this->map AS $it) {
            if($it['key'] == $key) {
                return $it;
            }
        }
        throw new Exception("В карте сущности с таким ключем");
    }

    public function getUniqueKey(AppObject $obj) {
        foreach ($this->map AS $it) {
            if($it['className'] == $obj::refClass()->getName()) {
                return $it['key'].":".$obj->getId();
            }
        }
        throw new Exception("В карте сущностей объект
'{ $obj::refClass()->getName()}' не найден");
    }

    public function setMap($map) {
        foreach ($map AS &$it) {
            $it['className'] = substr($it['class'], strpos($it['class'],
".")+1);
        }
        $this->map = $map;
    }
}

```

<?php

```

class AppObject {

    public function getLabel() {
        throw new Exception("Этот метод должен быть перенаследован");
    }

    /** @var Reflection[] */
    private static $reflections = array();

    /** @return Reflection */
    public final static function refClass() {
        $class = get_called_class();
        if (!isset(self::$reflections[$class])) {
            self::$reflections[$class] = new Reflections($class);
        }
    }
}

```



```

    }

    return self::$reflections[$class];
}

public function getUniqueId() {
    return Yii::app()->entityManager->getUniqueKey($this);
}

public function setter(array $data) {
    foreach ($data as $key => $value) {
        $r = $this::refClass()->getProperty($key);
        $r->setAccessible(true);
        $r->setValue($this, $value);
    }
}

public function __toString() {
    return get_class($this);
}

}

<?php
/**
 * Парсер для запуска ночью и сканирования файлов.
 */
Yii::import('common.models.parser.*');
class Parser extends AbstractProcedure {

    private $products = [];

    private $logger = [];

    public function __construct() {
        $products = Yii::app()->db->createCommand("SELECT ps.*, s.mask,
s.user_id from products_shop as ps JOIN shops as s on s.id = ps.shop_id")-
>queryAll();
        $this->products = array_map(function ($item) { return (object) $item;},
$products);
    }

    protected function execute() {
        # Начинаем пробегать по каждому товару.
        foreach($this->products as $product) {

```

```

        #mask
        if(!$product->mask) {
            $this->logger("id" . $product->id. " нет маски. Пропущен");
            continue;
        }
        # curl
        $curl = curl_init($product->url);
        curl_setopt($curl, CURLOPT_RETURNTRANSFER, 1);
        curl_setopt($curl, CURLOPT_FOLLOWLOCATION, 1);
//следование 302 redirect
        $html = curl_exec($curl);
        curl_close($curl);

        phpQuery::newDocument($html);
        $get = pq($product->mask)->text();
        $price = (float) trim(preg_replace('/[^0-9,\.\.]/', '', $get));
        Yii::app()->db->createCommand("INSERT INTO report (`moment`,
products_shop_id, `price`) VALUES (NOW(), {$product->id}, '$price')")->execute();
        $this->logger("id" . $product->id. " получена цена.");
        phpQuery::unloadDocuments();
    }
    Yii::consolelog(implode("", $this->logger));
}

private function logger($message) {
    array_push($this->logger, $message . "\n");
}

}

<?php

Yii::import('common.models.products.*');
class ReportService extends AbstractService {

    /**
     * @removeCall
     */
    public function getByProduct($id) {
        $data = Yii::app()->db->createCommand("(SELECT DATE(moment) as
`date`, price, name, s.id FROM report as r
JOIN products_shop as ps on r.products_shop_id = ps.id
JOIN shops as s ON s.id = ps.shop_id
WHERE ps.product_id = :id ORDER BY 1 ASC)
UNION ALL

```

```

        SELECT DATE(`datetime`) as `date`, price, "" . Yii::app()->params['name']
        . "" as name, 0 as id FROM price_history WHERE product_id = :id and `datetime` in
        (select max(`datetime`) FROM price_history WHERE product_id = :id GROUP BY
        date(`datetime`))"->bindValue(":id", $id, PDO::PARAM_INT)->queryAll();
        if(!$data) {
            throw new Exception("Нет данных для визуализации");
        }
        $minDate = $maxDate = Date::create("now")->format('Y-m-d');
        foreach($data as $row) {
            if($minDate > $row['date']) $minDate = $row['date'];
            if($maxDate < $row['date']) $maxDate = $row['date'];
        }
        $period = new DateTimeRange($minDate, $maxDate);
        $shops = [];
        foreach($data as $v) {
            $shops[] = $v['name'];
        }
        $shops = array_unique($shops);

        $result = [];
        foreach($shops as $shop) {
            $lastPrice = 0;
            $prices = ArrayHelper::filterColumn($data, ['name' => $shop]);
            $result[$shop]['name'] = $shop;
            $result[$shop]['tooltip'] = ['pointFormat' => " {series.name}: {point.y}
рублей"];
            for($cur = $period->getDateStart()->cloneable(); $cur->getTimestamp()
<= $period->getDateEnd()->getTimestamp(); $cur->modify("+1 day")) {
                $price = $lastPrice;
                $today = ArrayHelper::filterColumn($prices, ['date' => $cur-
>format('Y-m-d')]);
                if(!empty($today)) {
                    $price = $lastPrice = $today[0]['price'];
                }
                $result[$shop]['data'][$cur->format('Y-m-d')] = ["name" => $cur-
>format('d.m'), "y" => (float) $price];
            }
            ksort($result[$shop]['data']);
            $result[$shop]['data'] = array_values($result[$shop]['data']);
        }
        return array_values($result);
    }

/**
 * @removeCall

```

```

        */
        public function byShops() {
            $data = Yii::app()->db->createCommand("select SUM(IF(shop_id IS
NULL, 0,1)) as `count`, name from shops as t1
            left join products_shop as t2 on t1.id = t2.shop_id group by 2")-
>queryAll();
            $result = ["name" => "Количество добавленных товаров в магазин"];
            foreach($data as $shop) {
                $result['data'][] = ["name" => $shop['name'], "y" => (int)$shop['count']];
            }
            return $result;
        }

        /**
        * @removeCall
        */
        public function byCategory() {
            $data = Yii::app()->db->createCommand("select SUM(IF(category_id IS
NULL, 0,1)) as `count`, t1.name from categories as t1
            left join products as t2 on t1.id = t2.category_id group by 2")->queryAll();
            $result = ["name" => "Количество товаров в категории"];
            foreach($data as $shop) {
                $result['data'][] = ["name" => $shop['name'], "y" => (int)$shop['count']];
            }
            return $result;
        }
    }

<?php

/**
 * @tableName {{categories}}
 * @primaryKey id
 */
class Category extends AbstractEntity {

    private $name;
    /**
     * @column user_id
     */
    private $user;

```

```

/**
 * @removeParam name
 */
public function getName() {
    return $this->name;
}

public function create($name) {
    $this->name = trim(mb_ucfirst($name));
    $this->user = Yii::app()->user->id;
    return $this->modify();
}
}

<?php
Yii::import('common.models.shops.*');
class ShopHelper {

    public static function getOrCreate($url) {
        $url = strtolower($url);
        $parse = parse_url($url);
        $name = trim($parse['host']);

        # ищем существующий магазин
        $scr = new CDbCriteria();
        $scr->compare('t.name', $name);
        try {
            return Yii::app()->entityManager->find(Shop::refClass(), $scr);
        } catch (NoResultException $th) {
            return (new Shop()->create($name, $url));
        }
    }

    public static function createShopProduct($shop_id, $product_id, $url) {
        return Yii::app()->db->createCommand("INSERT IGNORE INTO
products_shop (`product_id`, `shop_id`, `url`) VALUES ($product_id, $shop_id,
'$url')")->execute();
    }
}

<?php

/**
 * @tableName {{shops}}

```

```

* @primaryKey id
*/
class Shop extends AbstractEntity {

    private $name;

    private $mask;

    /**
     * @column user_id
     */
    private $user;

    public function create($name) {
        $this->name = $name;
        $this->user = Yii::app()->user->id;
        return $this->modify();
    }

    /**
     * @removeParam countProduct
     */
    public function getCountProduct() {
        return (int) Yii::app()->db->createCommand("SELECT COUNT(*)
FROM products_shop WHERE shop_id = {$this->id}")->queryScalar();
    }

    /**
     * @removeParam name
     */
    public function getName() {
        return $this->name;
    }

    /**
     * @removeParam mask
     */
    public function getMask() {
        return $this->mask;
    }

    /**
     * @removeCall mask
     */

```

```

public function setMask($mask) {
    $this->mask = empty($mask) ? NULL : $mask;
    return $this;
}

}

<?php

Yii::import('common.models.products.*');
Yii::import('common.models.shops.*');
class ProductsService extends AbstractService {

/**
 * @removeCall
 */
public function create($name, $category, $articul, $image, $price, $urls) {
    $urls = (array) $urls;
    $urls = array_filter($urls, function($value) { return $value !== ""; });
    if(empty($urls)) {
        throw new Exception("Не одна ссылка на магазин не найдена!");
    }
    $category_id = $category->id;
    $product      =      $this->createEntity("Products")->create(trim($name),
$category_id, $articul, $image, floatval($price));
    foreach ($urls as $url) {
        $this->createShopForProduct($url, $product->getId());
    }
    return $product;
}

/**
 * @removeCall
 */
public function get() {
    return $this->getEntityProvider('Products');
}

/**
 * @removeCall
 */
public function removeProductShop($product, $shop) {
    if(!$product || !$shop) {
        throw new Exception("Не идентификатора товара или магазина");
    }
}

```

```
        return Yii::app()->db->createCommand("DELETE FROM products_shop
WHERE product_id = :product AND shop_id = :shop")->bindValue(':product',
$product, PDO::PARAM_INT)->bindValue(':shop', $shop, PDO::PARAM_INT)-
>execute();
    }

    /**
     * @removeCall
     */
    public function createShopForProduct($url, $id) {
        $shop = ShopHelper::getOrCreate($url);
        ShopHelper::createShopProduct($shop->getId(), (int) $id, $url);
    }

}
```