

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение высшего образования
«Тольяттинский государственный университет»

Институт математики, физики и информационных технологий
(наименование института полностью)

Кафедра «Прикладная математика и информатика»
(наименование)

09.03.03 Прикладная информатика

(код и наименование направления подготовки, специальности)

«Бизнес-информатика»

(направленность (профиль) / специализация)

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА (БАКАЛАВРСКАЯ РАБОТА)

на тему Создание корпоративного хранилища данных на основе экосистемы
открытого программного обеспечения

Обучающийся

Д.А. Причина

(Инициалы Фамилия)

(личная подпись)

Руководитель

Н.Н. Казаченок

(ученая степень (при наличии), ученое звание (при наличии), Инициалы Фамилия)

Тольятти 2022

Аннотация

Бакалаврская работа выполнена на тему «Создание корпоративного хранилища данных на основе экосистемы открытого программного обеспечения».

Цель работы состоит в разработке на примере ООО «Айковер ПРО» корпоративного хранилища данных на основе экосистемы открытого программного обеспечения.

Во введении определены: актуальность рассматриваемой темы, объект и предмет исследования, цель работы и поставлены задачи, ведущие к её достижению. Выбрана и аргументирована методика проведения исследования.

В первой главе проведен анализ структуры предприятия и деятельности подразделений ООО «Айковер ПРО». Проведён анализ бизнес-процессов подразделений. Выявлены недостатки существующего бизнес-процесса, проведено его концептуальное моделирование. Определены и оформлены в виде проекта технического задания основные требования к информационной системе для решения выявленных недостатков, поставлена задача на её разработку. Проведен сравнительный анализ представленных на рынке коммерческих решений сходных задач. Смоделирован бизнес-процесс, лишенный найденных недостатков.

Во второй главе выполнено логическое и физическое моделирование информационной системы. Проведены работы по прототипированию ИС.

В третьей главе проводится выбор и аргументация методики определения экономической эффективности внедрения решения. В соответствии с методикой проведён анализ показателей экономической эффективности информационной системы.

В заключении приводятся оценки результатов исследования, выводы и рекомендации по практическому использованию материалов работы.

Бакалаврская работа включает: 70 страниц, 32 рисунок, 17 таблиц, 8 формул и 26 источников.

Оглавление

Введение.....	5
Глава 1 Функциональное моделирование информационной системы	8
1.1 Техничко-экономическая характеристика предприятия	8
1.2 Концептуальное моделирование деятельности отдела розничных продаж ООО «Айковер ПРО»	10
1.2.1 Выбор методологии концептуального моделирования бизнес-процессов.....	10
1.2.2 Разработка и анализ модели бизнес-процесса «Как есть»	11
1.2.3 Обоснование необходимости автоматизированного варианта решения и формирование требований к проектируемой информационной системе.....	13
1.3 Анализ существующих разработок на предмет соответствия сформулированным требованиям	16
1.3.1 Определение критериев анализа	16
1.3.2 Сравнительная характеристика существующих разработок.	17
1.4 Постановка задачи на разработку корпоративного хранилища данных.....	19
1.5 Разработка модели бизнес-процесса «Как должно быть»	21
Глава 2 Логическое проектирование информационной системы	25
2.1 Выбор технологии логического моделирования информационной системы	25
2.2 Логическая модель информационной системы и ее описание	26
2.2.1 Моделирование функционального аспекта логической модели системы.....	26
2.2.2 Моделирование элементного аспекта логической модели системы.....	27
2.2.3 Моделирование взаимодействия компонентов информационной системы в динамике.....	29

2.3 Информационное обеспечение информационной системы.....	30
2.3.1 Используемые классификаторы и системы кодирования	30
2.3.2 Характеристика нормативно-справочной и входной оперативной информации	31
2.3.3 Характеристика выходной информации.....	35
2.4 Проектирование базы данных информационной системы	36
2.4.1 Выбор технологии проектирования базы данных информационной системы	36
2.4.2 Разработка логической модели базы данных информационной системы	37
2.5 Требования к аппаратно-программному обеспечению информационной системы	38
Глава 3 Физическое проектирование информационной системы	42
3.1 Выбор архитектуры информационной системы.....	42
3.2 Выбор технологии разработки программного обеспечения информационной системы	44
3.3 Выбор системы управления базами данных.....	46
3.4 Разработка физической модели данных информационной системы	48
3.5 Разработка программного обеспечения информационной системы	53
3.6 Описание функциональности информационной системы.....	57
3.7 Оценка и обоснование экономической эффективности разработки информационной системы	62
Заключение.....	67
Список используемой литературы	68

Введение

Актуальность темы бакалаврской работы обусловлена ростом потребности в обработке и хранении корпоративных данных, связанный с глобализацией экономики посредством сети Интернет и вызванным этим ростом конкуренции на рынке. Высокий уровень развития современных экономики и маркетинга вынуждают всё больший объём организаций применять информационные технологии для сохранения своих позиций на высококонкурентных рынках, таких как, например, Москва [10].

Традиционные технологии управления компаниями всё чаще сталкиваются с проблемами масштабирования, возросших объёмов информации и скорости реакций на изменения среды – выход на международные рынки с миллионными аудиториями, кратное увеличение конкуренции как следствие глобализации, изменение платежеспособности населения, вызванные внешними и внутренними факторами (политическими, санитарными, монетарными), всё более широкий спектр и разнообразие маркетинговых данных клиентов необходимых для проведения сегментации рынка (возраст, пол, образование, интересы и прочие). Следовательно, для продолжения эффективной деятельности в условиях изменяющейся среды – организации вынуждены обращаться к инструментам, предоставляемым ИТ-индустрией, как наиболее целесообразному решению проблемы экспоненциально растущего объёма информации, нуждающегося в обработке.

Точность и скорость принятия управленческих решений становится всё более зависима от наличия и эффективности инструментов анализа и обработки корпоративного потока данных. Таким образом, внедрение информационной системы, производящей предварительную очистку и приведение оперативных данных к удобной для восприятия человеком форме, позволяет увеличить конкурентоспособность предприятия.

Коммерческие решения, доступные на рынке, способны закрыть потребности большинства организаций в инструментах хранения и обработки

информации, однако их внедрение сопряжено с определёнными рисками, наиболее значимыми из которых являются:

- ограничения на распространение (каждый экземпляр системы может требовать покупку дополнительной лицензии);
- ограниченная модифицируемость, затрудняющая подстройку решения под конкретные нужды предприятия;
- использование закрытых форматов данных, затрудняющее переход на альтернативные решения или делающее его невозможным;
- отказ в работе или обслуживании вызванный форсмажорными обстоятельствами на стороне производителя.

Экосистема открытого программного обеспечения (open-source) лишена подобных недостатков и гарантирует корпоративному субъекту владение (возможность изменять, обслуживать и использовать) системой при её размещении в рамках собственной инфраструктуры. Также, несомненным плюсом использования решений, основанных на программном обеспечении с открытым исходным кодом, является возможность проведения независимого аудита информационной безопасности, что является необходимым шагом для использования подобного рода информационных систем в деятельности государственных учреждений и госкорпораций. Таким образом, создание типовых решений хранилищ корпоративных данных на основе open-source программного обеспечения является – необходимым шагом для дальнейшего независимого развития отечественного сегмента IT-рынка и обеспечения устойчивого роста экономики региона, что подтверждает актуальность и востребованность темы исследования.

Целью работы является создание на основе компонентов, предоставляемых экосистемой программного обеспечения с открытым исходным кодом, корпоративного хранилища данных (Data Warehouse/DWH) – предметно-ориентированного, интегрированного, привязанного ко времени, неизменяемого набора данных, предназначенного для поддержки принятия решений [14, с. 15-17]. Объектом исследования является процесс разработки и

интеграции DWH в деятельность коммерческой организации, на примере открытых данных ООО «Айковер ПРО». Предметом исследования является методология построения архитектуры, подбора компонентов и разработки системы. Для достижения цели необходимо решение ряда задач:

- анализ структуры и технико-экономических характеристик предприятия, его бизнес-процессов;
- проведение моделирования бизнес-процесса «Как есть» с целью поиска путей внедрения ИС;
- анализ доступных альтернатив и необходимости проведения работ по созданию ИС;
- создание модели бизнес-процесса «Как должно быть»;
- разработка краткого технического задания (ТЗ) информационной системы, призванной решать найденные недостатки бизнес-процесса;
- логическое и физическое моделирование информационной системы (ИС) и её структур данных;
- разработка критериев выбора и анализ компонентов системы;
- создание физической модели ИС и описание её особенностей;
- оценка экономической эффективности предлагаемого решения.

Бакалаврская работа разделена на главы. В рамках первой главы рассматривается и анализируется структура предприятия и проводится моделирование предметной области. На основе проведённого моделирования составляется техническое задание на информационную систему (ИС). Используя информацию из первой главы, вторая глава развивает концепты до многоаспектной логической модели, посредством которой определяет параметры аппаратно-программного обеспечения необходимого для физической реализации системы. Третья глава является завершающей, она определяет физический облик, способ использования и экономическую эффективность предлагаемого решения. В заключении проводится анализ проведённой работы и потенциальные пути дальнейшего развития системы.

Глава 1 Функциональное моделирование информационной системы

1.1 Техничко-экономическая характеристика предприятия

ООО «Айковер ПРО» является компанией, ведущей деятельность в области электронной коммерции, связанной с продажей лайфстайл гаджетов и аксессуаров, и функционирующей на основе принципов современного маркетинга и экономики. Свою историю организация отсчитывает от 2007 года, когда бренд запустил свой первый специализированный магазин, реализующий гаджеты, аксессуары и технику марки Apple. Однако, выбранная бизнес-модель к 2015 году столкнулась с достижением пределов роста и необходимостью проведения реорганизации, изменения структуры закупок и введения категорийного менеджмента, что позволило менее чем за год увеличить ассортимент товаров, предлагаемых клиентам, в 10 раз – до более чем 20 тысяч наименований. Поступательное развитие бренда привело к пониманию его основной концепции – тщательный отбор товаров, новинок и инноваций, позволяющий предоставлять клиентам только лучшие экземпляры в категории. Начиная с 2016 года, компания самостоятельно ведёт внешнеэкономическую деятельность, что позволило зарекомендовать себя как надёжного партнёра и получить статус официального реселлера или единственного дистрибьютора многих нишевых брендов: JBL, Bose, Belkin, SGP, Wacom, Bang Olufsen, Logitech, Sennheiser, Sony, Anker, Aukey, HyperDrive [3][9].

В настоящий момент организация владеет двумя розничными магазинами в центре Москвы и сайтом iCover.ru с ежемесячной аудиторией более 350 тысяч посетителей, который входит в сотню крупнейших интернет-магазинов РФ. Также, компания представлена в B2B сегменте и занимается обслуживанием корпоративных клиентов и оптовыми продажами товаров со склада. Организация владеет собственной курьерской службой, но, в случае необходимости, способна провести доставку «до двери» посредством услуг

транспортных компаний на выбор клиента. Текущий штат сотрудников предприятия составляет более 100 человек [9].

Составим организационную структуру предприятия посредством анализа доступных вакансий, размещенных на сайте крупнейшей российской компании интернет-рекрутмента HeadHunter [3]. В качестве CASE-средства для создания схемы используется Microsoft Visio 2021. На рисунке 1 представлена схема организационной структуры ООО «Айковер ПРО».

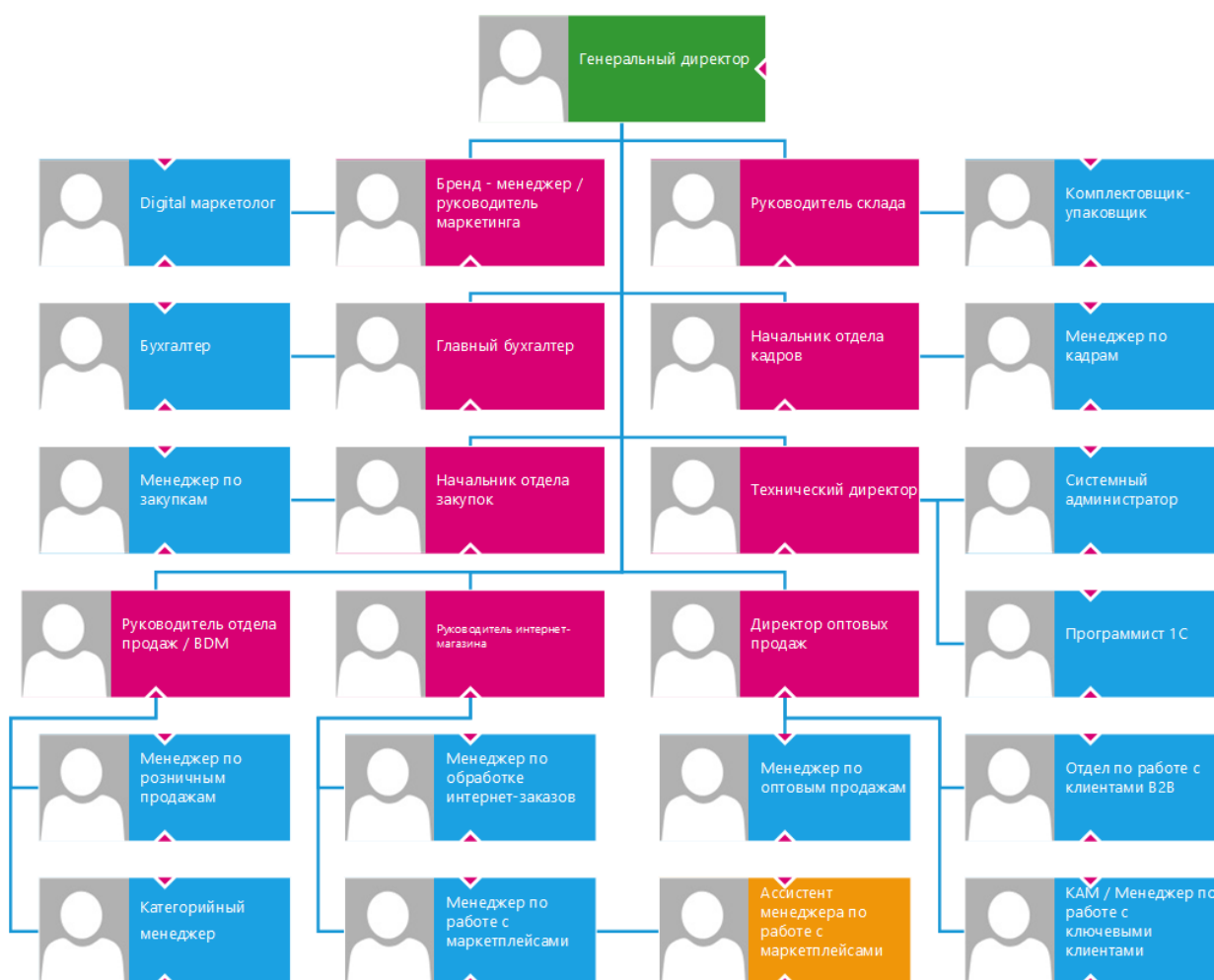


Рисунок 1 – Схема организационной структуры ООО «Айковер ПРО»

Проведём анализ полученной схемы и сделаем выводы о структуре управления организацией: компания использует линейно-функциональную структуру, где линейными подразделениями, выполняющими основную деятельность, являются отделы продаж, оптовых продаж и интернет-магазин.

Подразделениями, ведущими вспомогательную деятельность выступают отделы: маркетинга, кадров, закупок, технического (ИТ) обеспечения, а также склад и бухгалтерия.

В качестве примера рассмотрим деятельность отдела розничных продаж ООО «Айковер ПРО». В обязанности линейного сотрудника отдела розничных продаж (менеджера по розничным продажам) входят: необходимость оценки плана/факта продаж, ценообразование, контроль неликвидности товаров (overstock) и ситуация отсутствия товара (out of stock) [8]. В качестве информационной системы, контролирующей торговую деятельность в рамках компании, выступает решение 1С: Управление торговлей (УТ) версии 10.3/11 [12]. Однако умение пользоваться этим инструментом, в требованиях к занимаемой должности для менеджеров по розничным продажам – не значит. Отсюда можно сделать вывод о наличии бизнес-процесса требующего для своего осуществления непосредственного взаимодействия линейного менеджера и сотрудника ИТ-отдела, осуществляющего вспомогательную деятельность. Таким образом был выявлен неэффективный бизнес-процесс с высоким потенциалом для автоматизации.

1.2 Концептуальное моделирование деятельности отдела розничных продаж ООО «Айковер ПРО»

1.2.1 Выбор методологии концептуального моделирования бизнес-процессов

В качестве методологии следует использовать выразительные средства, предоставляющие возможность построения структур с точностью, достаточной для однозначного и непротиворечивого моделирования предметной области на основе доступных сведений.

Так как источником данных для текущей работы являются открытая информация доступная посредством сети Интернет, то построение подробной

схемы бизнес-процесса затруднительно. Однако, нам доступны данные отражающие входы и выходы процесса, а также механизмы и управление, используемые в ходе бизнес-процесса. Таким образом, наиболее эффективным решением, при существующих ограничениях, является использование методологии функционального моделирования IDEF0, для которой обозначенные данные являются минимальным необходимым набором информации [6].

В качестве CASE-средства, используемого для построения моделей рассматриваемого бизнес-процесса, используется Microsoft Visio 2021.

1.2.2 Разработка и анализ модели бизнес-процесса «Как есть»

Проведём анализ доступной информации и сформируем набор данных необходимых для построения модели «Как есть» верхнего уровня. В качестве входных ресурсов процесса выступают запрос данных для поддержки принятия решения сотрудником линейного отдела организации, коммерческие документы (счет-фактура) отражающие реальные факты продажи и движения материальных ресурсов компании. В качестве управления процесса служат законодательство Российской Федерации (как федеральное, так и местное), а также внутренние нормативные документы организации (устав, локальные нормативные акты и прочие). Механизмами, задействованными в рамках бизнес-процесса, являются менеджер по розничным продажам (менеджер), и сотрудник ИТ-отдела (программист), осуществляющий работу с информационной системой организации (ИС 1С:УТ). Выходом процесса является отчёт о движении товарно-материальных ценностей (отчёт), используемый менеджером для принятия оперативного решения. На основе приведённых данных проведём построение контекстной диаграммы функциональной модели существующего бизнес-процесса формирования отчёта.

На рисунке 2 отражена контекстная диаграмма бизнес-процесса формирования отчёта «Как есть».



Рисунок 2 – Контекстная диаграмма бизнес-процесса формирования отчёта «Как есть»

Полученная модель верхнего уровня не является достаточной для отражения недостатков модели – необходимо провести декомпозицию модели для поиска минимального необходимого уровня детализации бизнес-процесса, позволяющего выполнить обозначенную задачу. На рисунке 3 представлена декомпозиция бизнес-процесса формирования отчёта «Как есть».

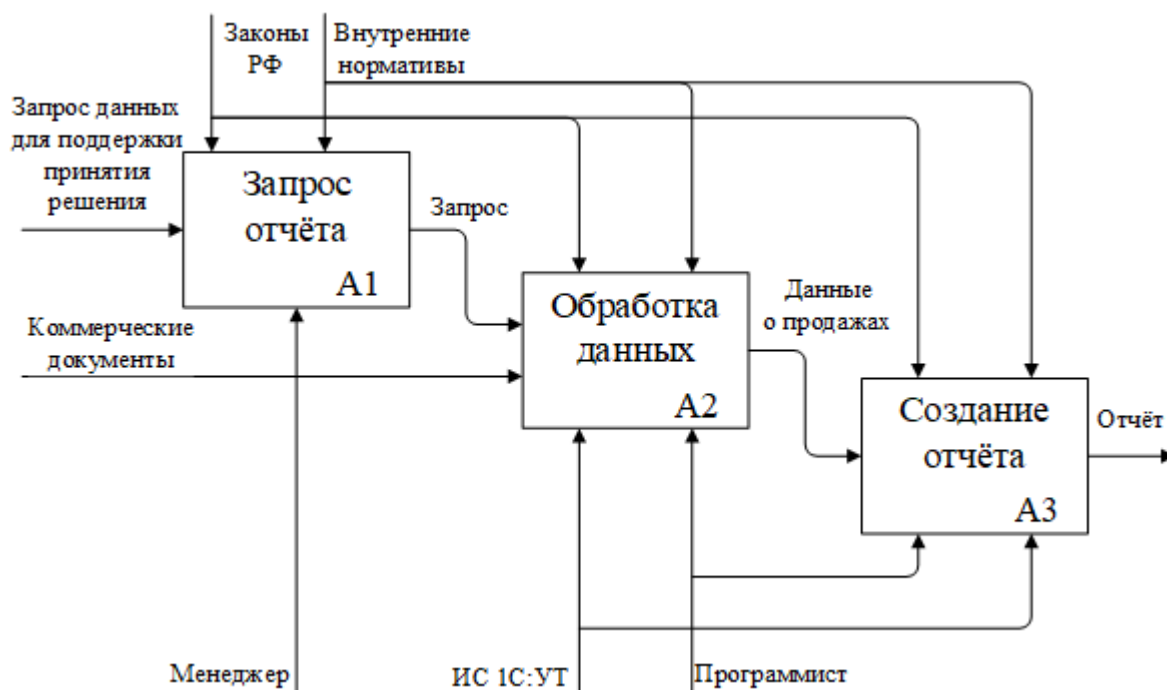


Рисунок 3 – Декомпозиция бизнес-процесса формирования отчёта «Как есть»

Целью реинжиниринга рассматриваемого бизнес-процесса является экономия человеко-часов посредством использования средств автоматизации, достигаемая освобождением механизма «Программист» или «Менеджер» – для повышения общей доступности ресурса для других процессов. Так как механизм «ИС 1С:УТ» является зависимым от механизма «Программист» в рассматриваемой модели бизнес-процесса, то дальнейшее проведение декомпозиции представляется нецелесообразным. Приступим к анализу недостатков существующего бизнес-процесса.

1.2.3 Обоснование необходимости автоматизированного варианта решения и формирование требований к проектируемой информационной системе

Как можно заметить на декомпозированной модели бизнес-процесса – необходимость в механизме «Программист» возникает из-за ограничений в функциональности (отсутствие сущности информационная панель) используемой информационной системы организации [16]. Таким образом, для решения поставленной задачи, необходима замена используемой в рамках рассматриваемого бизнес-процесса информационной системы на решение не предполагающее участие механизма «Программист».

Первым этапом работ по созданию информационной системы является формирование списка требований, предъявляемых к проектируемой системе [7]. Для этого воспользуемся классификацией требований предоставляемой методологией FURPS+ [2]:

- functionality (функциональность),
- usability (удобство использования),
- reliability (надежность),
- performance (производительность),
- supportability (сопровождаемость),
- implementation (ограничения разработки),
- interface (ограничения на интерфейсы),
- design (проектные ограничения),

– physical (физические ограничения).

В таблице 1 указаны атрибуты и требования к ним для разрабатываемой информационной системы.

Таблица 1 – Требования FURPS+ к информационной системе

Атрибут	Предъявляемые требования
Функциональность	Способность обрабатывать первичные коммерческие документы в формате УПД [11]
	Возможность автоматического формирования необходимых отчётов по запросу
	Возможность подключения к внешним источникам данных и интеграции с другими информационными системами
Удобство использования	Пользовательский интерфейс системы должен поддерживать возможность реализации как в виде веб-приложения, так и посредством исполняемого на ПК пользователя приложения
	Наличие и доступность справочной информации по использованию системы
	Функциональные компоненты системы имеют типовые графические интерфейсы, предоставляемые разработчиком
Надежность	Система должна обеспечивать бесперебойную работу компонента автоматической обработки данных на период обработки первичных коммерческих документов
	Система должна обеспечивать бесперебойный доступ пользователей к данным в рабочий период
	Система должна обеспечивать возможность резервного копирования обработанных данных
	Система должна обеспечивать возможность восстановления после сбоя в течение 24 часов после возникновения сбоя
Производительность	Система должна обеспечивать одновременную комфортную работу не менее 40 пользователей, подключенных к системе
	Система должна обеспечивать обработку первичных коммерческих документов не менее чем за 30-дневный период
	Система должна обеспечивать работу с объёмом данных не менее чем за 1 финансовый год
	Система должна обеспечивать обработку 30-дневного объёма первичных коммерческих документов в течение 3 часов

Продолжение таблицы 1

Атрибут	Предъявляемые требования
Сопровождаемость	Возможность удалённого администрирования
	Возможность удалённого восстановления после сбоя в течение 24 часов
	Возможность удалённого резервного копирования и восстановления из резервной копии
	Возможность масштабирования и расширения функциональности системы
	Возможность обновления компонентов системы без потери функциональности системы и возникновения сбоя
Ограничения разработки	В качестве компонентов системы могут быть использованы только программные средства с открытым исходным кодом
	Языком программирования, используемым для построения системы, является Python 3
	Обработка данных происходит с использованием библиотеки для работы с данными pandas версии 1.4 и выше
Ограничения на интерфейсы	Первичная коммерческая информация представлена в виде валидных XML документов, содержащих данные в формате УПД [11]
	Коммуникация между компонентами системы происходит в защищенном режиме посредством сети передачи данных
	Компонент визуализации данных должен поддерживать подключение к хранилищу по интерфейсу DB API 2.0
	Компонент обработки первичной коммерческой документации должен поддерживать подключение к хранилищу по интерфейсу DB API 2.0
Проектные ограничения	Архитектура «клиент-сервер»
	Использование реляционной модели базы данных
	Операционной системой сервера для размещения компонентов системы является Debian версии 11 и выше
	Клиент (визуализатор) использует SQL-запросы для формирования отчёта
Физические ограничения	Сервер информационной системы и базы данных должен иметь физическую защиту данных и посредством организационных мероприятий и совокупности прочих средств – препятствовать несанкционированному проникновению или доступу неуполномоченных лиц
	Сервер базы данных должен иметь достаточную для работы с информацией не менее чем 1 финансового года ёмкость накопителя данных

Ограничения, наложенные на проект, отражают современные тенденции рынка и призваны минимизировать риски, связанные с использованием непопулярных или нишевых решений. Так, например, использование Python 3 в качестве языка для построения системы обусловлено очень высокой

популярностью этого средства создания программных продуктов [25], что позволяет предположить высокую доступность на рынке специалистов для дальнейшей разработки и поддержки проекта. Выбор в качестве серверной операционной системы Debian 11 связан с доминированием этого дистрибутива (дистрибутив Ubuntu основан на Debian, и они совместимы друг с другом) в сегменте веб-серверов [26] и, соответственно, с наличием большого комьюнити пользователей и подробной документации – удешевляющих и упрощающих поддержку разрабатываемого решения.

Представленные требования послужат основой для проведения дальнейших работ по созданию информационной системы, решающей выявленные проблемы бизнес-процесса.

1.3 Анализ существующих разработок на предмет соответствия сформулированным требованиям

1.3.1 Определение критериев анализа

В качестве критериев анализа для существующих разработок воспользуемся полнотой (степенью) удовлетворения требований к информационной системе, указанных в пункте 1.2.3 данной работы. Где выполнение одного требования, указанного для атрибута методологии FURPS+, соответствует одному баллу полноты соответствия оцениваемого параметра. Таким образом возможно оценить степень соответствия анализируемого коммерческого решения выдвигаемым к разрабатываемой информационной системе требованиям. Так как существует разница в количестве требований к различным атрибутам системы, то необходимо провести нормализацию анализируемых баллов разделив на количество требований, тем самым приводя баллы анализируемого параметра к процентам.

Предполагая одинаковую важность каждого из атрибутов для реализации проекта – в качестве интегрального параметра для оценки общего

соответствия предлагаемого решения выдвигаемым FURPS+ требованиям к информационной системе воспользуемся средним арифметическим процентов соответствия программного комплекса каждому из атрибутов.

1.3.2 Сравнительная характеристика существующих разработок

Воспользуемся данными, предоставленными компанией Gartner, о лидерах рынка Business Intelligence платформ [22] для формирования списка лидирующих представителей зарубежных разработок информационных систем призванных решать проблемы, выявленные в анализируемом бизнес-процессе. На рисунке 4 представлен анализ рынка аналитических и BI-платформ от компании Gartner на март 2022 года.

Figure 1: Magic Quadrant for Analytics and Business Intelligence Platforms



Source: Gartner (March 2022)

Рисунок 4 – Анализ BI-платформ от компании Gartner

В сегменте лидеров рассматриваемого квадранта присутствуют следующие решения: Microsoft Power BI, QlikTech Qlik, Salesforce Tableau. Компания Gartner позиционирует эти решения как наиболее полные и функциональные BI-платформы, представленные на зарубежном рынке.

Также, в качестве анализируемой альтернативы, рассмотрим отечественное решение «1С:Аналитика» являющееся отдельно поставляемой частью платформы «1С:Предприятие 8». Таким образом, становится возможным проведение комплексного анализа рынка существующих разработок, как зарубежных, так и отечественных.

В таблице 2 представлен сравнительный анализ вышеозначенных разработок согласно критериям, указанным в пункте 1.3.1 данной работы.

Таблица 2 – Сравнительный анализ существующих разработок

Критерий	Power BI	Qlik	Tableau	1С:Аналитика
Функциональность	66%	66%	66%	100%
Удобство использования	66%	100%	100%	100%
Надежность	50%	75%	50%	50%
Производительность	100%	100%	100%	100%
Сопровождаемость	40%	40%	40%	60%
Ограничения разработки	100%	100%	100%	100%
Ограничения на интерфейсы	25%	25%	25%	50%
Проектные ограничения	25%	25%	25%	25%
Физические ограничения	50%	100%	50%	100%
Средний процент соответствия требованиям	58%	70%	62%	76%

Процент соответствия критерию ограничения разработки принимается равным 100%, так как на сторонние решения ограничения данного атрибута FURPS+ наложены быть не могут, следовательно, является необходимым

изъятие его влияния на общий процент соответствия выдвигаемым требованиям. Результаты проведенного анализа указывают на необходимость глубокой доработки представленных альтернатив для полного соответствия заявленным к информационной системе требованиям. Таким образом, по совокупности факторов – разработка собственного решения представляется более целесообразной чем доработка или внедрение существующих систем.

1.4 Постановка задачи на разработку корпоративного хранилища данных

Для постановки задачи на разработку проекта создания информационной системы для отдела розничных продаж ООО «Айковер ПРО» воспользуемся структурой для построения технического задания (ТЗ) предоставленной в ГОСТ 19.201-78 [5]:

а) введение:

1) наименование программы:

наименование программы – «Информационная панель iCover»;

2) краткая характеристика области применения:

система «Информационная панель iCover» предназначена для информационной поддержки принятия решений сотрудников ООО «Айковер ПРО»;

б) основания для разработки:

информационная система «Информационная панель iCover» разрабатывается по личной инициативе автора бакалаврской работы, наименование темы разработки – «Разработка информационной системы Информационная панель iCover», условное обозначение темы разработки (шифр темы) – «iCover-VI»;

в) назначение разработки:

программа будет использоваться в организации ООО «Айковер ПРО» двумя группами пользователей: сотрудниками линейных подразделений компании и руководящим составом:

1) функциональное назначение:

генерация консолидированных отчётов по продажам на основе коммерческих документов ООО «Айковер ПРО» по запросам менеджеров организации по розничным продажам;

2) эксплуатационное назначение:

серверные компоненты системы производят ETL-обработку и сохранение данных из первичных коммерческих документов, клиентская часть системы (визуализатор) производит формирование консолидированных отчётов на основе данных, предоставляемых хранилищем системы (базой данных);

г) требования к программе или программному изделию:

требования, выдвигаемые к разрабатываемой информационной системе указаны в пункте 1.2.3 данной работы;

д) требования к программной документации:

предварительный состав программной документации:

1) техническое задание,

2) пользовательская документация,

3) программа и методика испытаний;

е) технико-экономические показатели:

целью разработки информационной системы «Информационная панель iCover» является экономия трудочасов сотрудников ООО «Айковер ПРО» достигаемая за счёт внедрения ИС, экономический эффект от внедрения программного комплекса достигается посредством высвобождения рабочего времени работников и увеличения полноты информированности сотрудников организации при принятии оперативных и управленческих решений;

ж) стадии и этапы разработки:

проект реализуется в три стадии:

- 1) техническое задание,
- 2) технический (и рабочий) проекты,
- 3) внедрение;

на фазе «Техническое задание» выполняется разработка, согласование и утверждения настоящего технического задания, фаза «Технический (и рабочий) проекты» предполагает разработку проекта, составление необходимой пользовательской документации и проведение тестирования проекта с целью подтверждения заявленных характеристик, на стадии «Внедрение» происходит обучение пользователей работе с системой и передача проекта конечным пользователям;

- з) порядок контроля и приемки:

тестирование решения проводится согласно пожеланиям конечных пользователей системы и предполагает активное участие заинтересованных лиц в доведении программного продукта до минимально жизнеспособного уровня позволяющего удовлетворять рабочие потребности сотрудников организации.

1.5 Разработка модели бизнес-процесса «Как должно быть»

Внесём необходимые изменения в функциональную модель «Как есть», продемонстрированную в пункте 1.2.2 данной работы, и приведём её к форме «Как должно быть», лишенной найденных недостатков.

На рисунке 5 отражена контекстная диаграмма бизнес-процесса формирования отчёта «Как должно быть». Сущности существующей информационной системы организации («ИС 1С:УТ») и «Программист» были изъяты из бизнес-процесса с целью экономии ресурсов организации и достижения искомого экономического эффекта от внедрения решения.

Привнесённая сущность информационной системы корпоративного хранилища данных («ИС DWH») обозначена на модели красным цветом.



Рисунок 5 – Контекстная диаграмма бизнес-процесса формирования отчёта «Как должно быть»

Для раскрытия изменений, произошедших с моделью, проведём декомпозицию модели формирования отчёта «Как должно быть». Рисунок 6 отражает декомпозированную структуру модели бизнес-процесса формирования отчёта «Как должно быть».

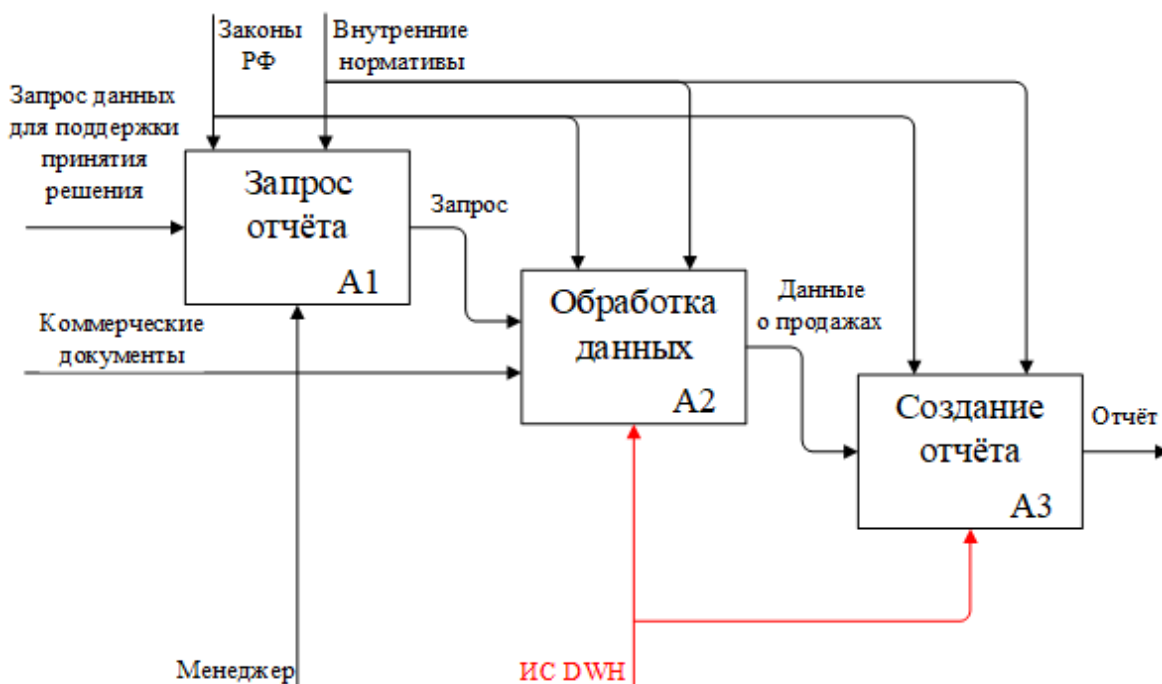


Рисунок 6 – Декомпозиция бизнес-процесса формирования отчёта «Как должно быть»

Изменению подверглись механизмы, используемые в бизнес-процессе, а именно – связка механизмов «ИС 1С:УТ» и «Программист» заменена на механизм «ИС DWH» (обозначен красным цветом), который выполняет в автоматическом режиме рутинные операции обработки данных и создания на их основе отчётов о коммерческих операциях, необходимых для исполнения должностных обязанностей менеджерами по розничным продажам.

Для построения корпоративного хранилища данных (ИС DWH) необходимо обеспечить формирование предметно-ориентированного, интегрированного, привязанного ко времени, неизменяемого набора данных на основе первичных коммерческих документов предприятия. Проведём анализ соответствия диаграмм потоков данных разрабатываемой системы представленному определению. На рисунке 7 представлена концептуальная DFD-модель движения потоков данных разрабатываемой информационной системы.

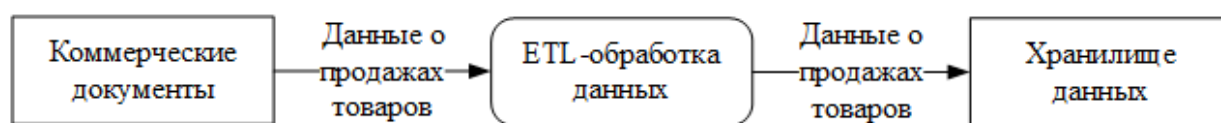


Рисунок 7 – DFD-диаграмма информационной системы DWH

Доступные первичные коммерческие документы – отражают факты продаж товаров в конкретные моменты времени и служат информационным источником для регулярного пополнения базы данных информационной системы, тем самым формируя структурированный и привязанный ко времени массив данных о предмете продаж товаров. Данные, содержащиеся в СУБД информационной системы, могут быть использованы для поддержки принятия решений сотрудников организации – менеджеров и их руководителей. Таким образом, можно заключить, что общая концепция предлагаемой информационной системы соответствует ранее обозначенному определению корпоративного хранилища данных.

Выводы по главе 1

В первой главе бакалаврской работы была проанализирована организационная структура ООО «Айковер ПРО». Выявлены и классифицированы основные и вспомогательные подразделения компании. Рассмотрена деятельность организации и определен неэффективный бизнес-процесс, пригодный для реинжиниринга. Выбрана методология моделирования и создана функциональная модель бизнес-процесса «Как есть», которая была отражена посредством контекстной диаграммы. Была проведена декомпозиция полученной модели с целью поиска путей решения выявленных проблем. Обоснована необходимость создания автоматизированного варианта решения и, посредством методологии FURPS+, сформированы требования, выдвигаемые к подобному решению. Также, были определены критерии и, на их основе, проведён сравнительный анализ существующих решений доступных на рынке. По результатам которого, аргументирована необходимость проведения дальнейших работ по созданию новой информационной системы. Была проведена работа по постановке задачи на разработку системы и создано техническое задание. Разработана модель бизнес-процесса «Как должно быть» лишенная найденных недостатков и проведена её декомпозиция с целью анализа изменений, привнесённых реинжинирингом. Приведена диаграмма потоков данных разрабатываемого корпоративного хранилища данных.

Глава 2 Логическое проектирование информационной системы

2.1 Выбор технологии логического моделирования информационной системы

Так как сущности, представленные в разрабатываемой информационной системе, отражают объекты реального мира и их взаимосвязи между собой, то для проведения логического моделирования наиболее предпочтительным выглядит использование методологии объектно-ориентированного анализа и построения моделей.

Для более полного описания информационной системы необходимо смоделировать следующие динамические и статические её аспекты:

- взаимодействие пользователей и системы;
- структуру, связи и элементный состав системы;
- взаимодействие элементов системы.

Учитывая всё вышесказанное, для сохранения унификации и единообразного стиля моделей, необходимо использовать методологию, выразительные средства которой – способны отразить все обозначенные аспекты системы. На текущий момент, наиболее популярным и доказавшим свою надёжность инструментом, способным отобразить весь указанный спектр граней информационной системы [1, с. 22-26], является Unified Modeling Language (UML) – унифицированный язык графического описания для объектного моделирования.

Смоделируем ранее указанные аспекты информационной системы посредством UML-диаграмм. В качестве CASE-средства для разработки UML-диаграмм выступает Microsoft Visio 2021.

2.2 Логическая модель информационной системы и ее описание

2.2.1 Моделирование функционального аспекта логической модели системы

Воспользуемся, как основой, контекстной диаграммой «Как должно быть» бизнес-процесса формирования отчёта для сотрудника организации указанной в параграфе 1.5 данной работы, расширим её и отразим посредством диаграммы прецедентов (use case diagram) функциональный аспект логической модели информационной системы [1, с. 27-31].

На рисунке 8 изображена UML-диаграмма прецедентов информационной системы.

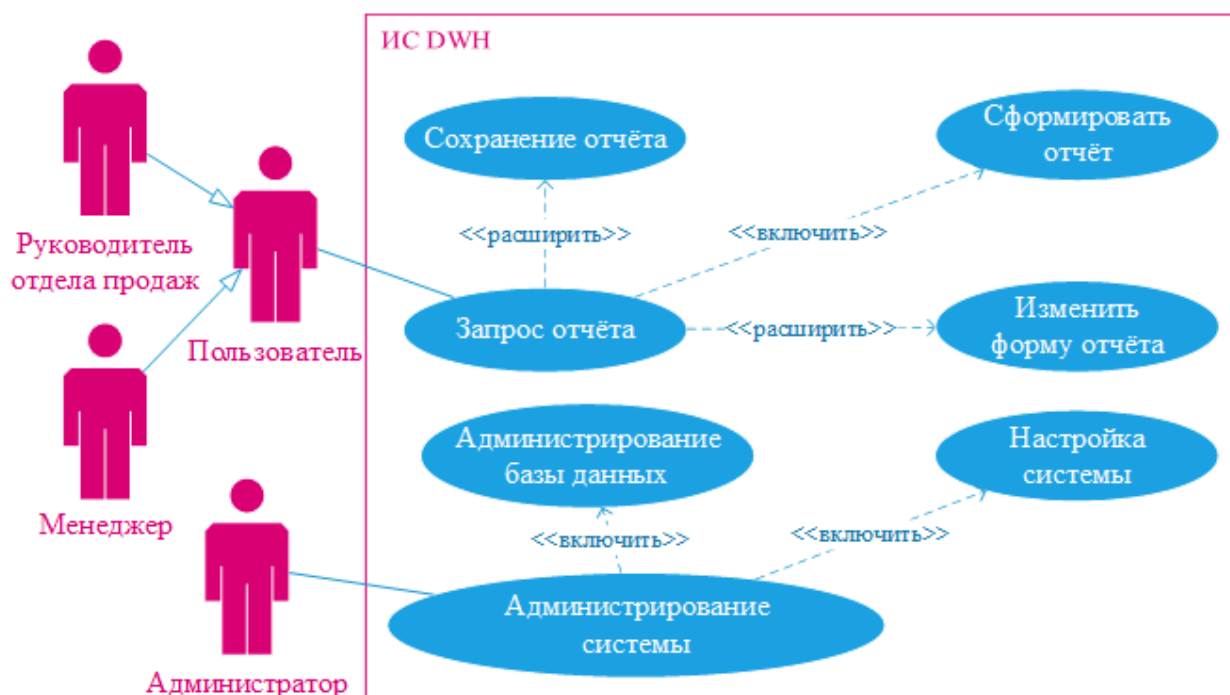


Рисунок 8 – UML-диаграмма прецедентов информационной системы

Проанализируем полученную модель: механизм «Менеджер» теперь представлен действующим лицом [1, с. 27] «Пользователь», которое является обобщением для всех непосредственных пользователей системы и включает как линейных сотрудников, так и руководящий состав. Пользователи могут

запрашивать у системы отчёты, в произвольной форме, содержащие информацию из коммерческих документов, доступную в базе данных, и сохранять их для дальнейшего использования. Так как доступ к ИС предполагается по защищенному каналу в рамках локальной сети передачи данных предприятия, то аутентификации пользователя в системе не требуется, и пользователь может свободно с ней взаимодействовать.

Также, с целью выполнения вспомогательных процессов в рамках информационной системы – было добавлено действующее лицо «Администратор», занимающееся поддержанием работоспособности системы – настройкой параметров и администрированием необходимых для её корректной работы справочников и классификаторов.

2.2.2 Моделирование элементного аспекта логической модели системы

Рассмотрим взаимодействующие элементы информационной системы и требования к ним. Для этого проанализируем информацию необходимую для проведения бизнес-процесса формирования отчёта для отдела розничных продаж. Линейному сотруднику отдела розничных продаж необходимо проводить оценку плана и факта продаж, анализировать категории продаваемых товаров, их цену, и факты наличия/отсутствия товара или его низких продаж. Таким образом, минимальным необходимым набором классов, для функционирования информационной системы формирования отчётов, является следующий список:

- информация о продаже товаров,
- сведения о доступности и цене товаров,
- информация о товаре,
- информация о единицах измерения товара,
- информация о категории товара.

На основе представленного списка сформируем UML-диаграмму классов [1, с.31-34]. На рисунке 9 представлена UML-диаграмма классов информационной системы.

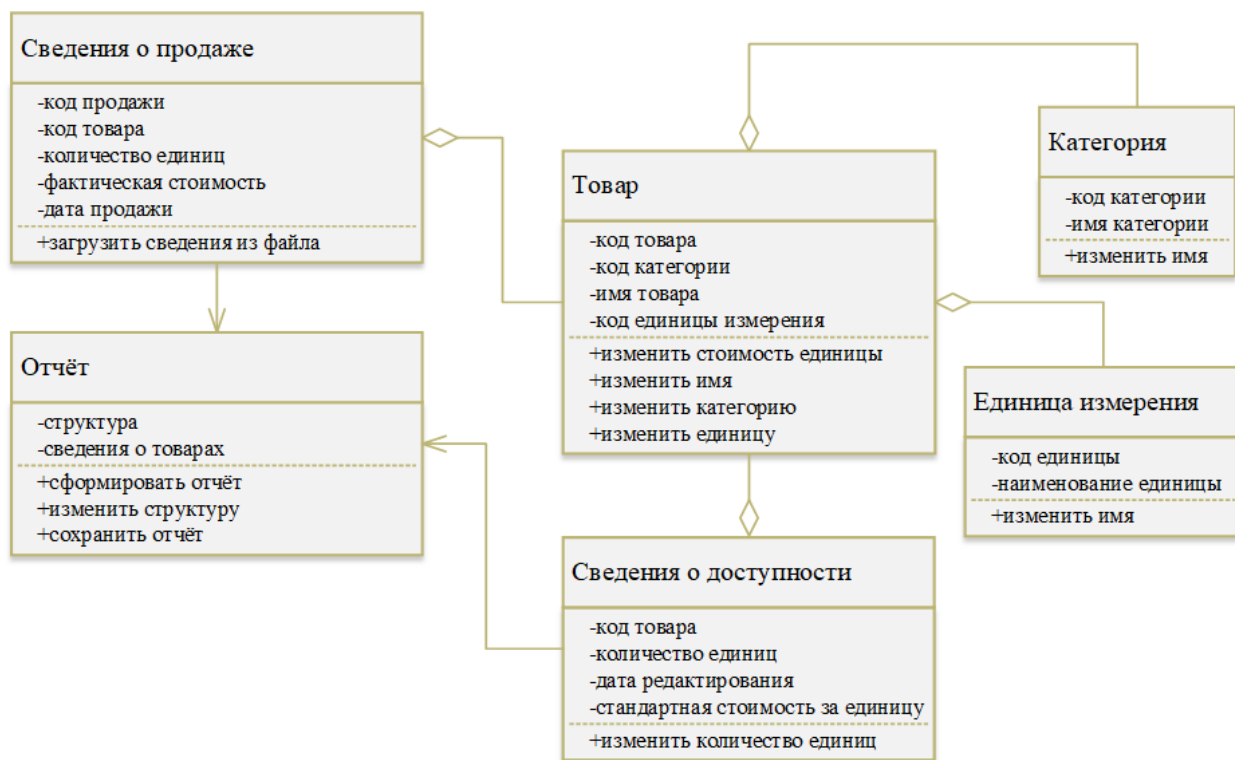


Рисунок 9 – UML-диаграмма классов информационной системы

Рассмотрим информационные сущности участвующие в процессе формирования отчёта. Для формирования отчёта необходимо в равной степени как наличие информации о продажах, так и наличие информации о доступности товаров. Обе сущности, как продажа, так и доступность, оперируют с объектами класса товаров. Однако, в зависимости от класса, рассматриваются либо стандартная, либо фактическая стоимость товара (так как необходимо учитывать скидки, акции и прочие маркетинговые процессы). Класс товара является ключевым и содержит информацию как о самом товаре, так и его классификации (категории), единицах измерения товара (штуки, метры, килограммы и так далее). Товары, их категории, единицы измерения, сведения о доступности каждого конкретного товара и стандартной стоимости за единицу вносятся в систему администратором в течение её эксплуатации. Сведения о продажах вносятся в систему из первичных документов в автоматическом режиме посредством компонентов системы.

2.2.3 Моделирование взаимодействия компонентов информационной системы в динамике

Статическая модель классов не отражает особенностей динамического изменения информационной системы и её взаимосвязей. Поэтому, для формирования представления о том каким образом взаимодействуют компоненты системы во времени необходимо провести моделирование динамического аспекта логической модели.

На рисунке 10 представлена UML-диаграмма последовательностей [1, с.38-41], отражающая динамический аспект логической модели информационной системы.

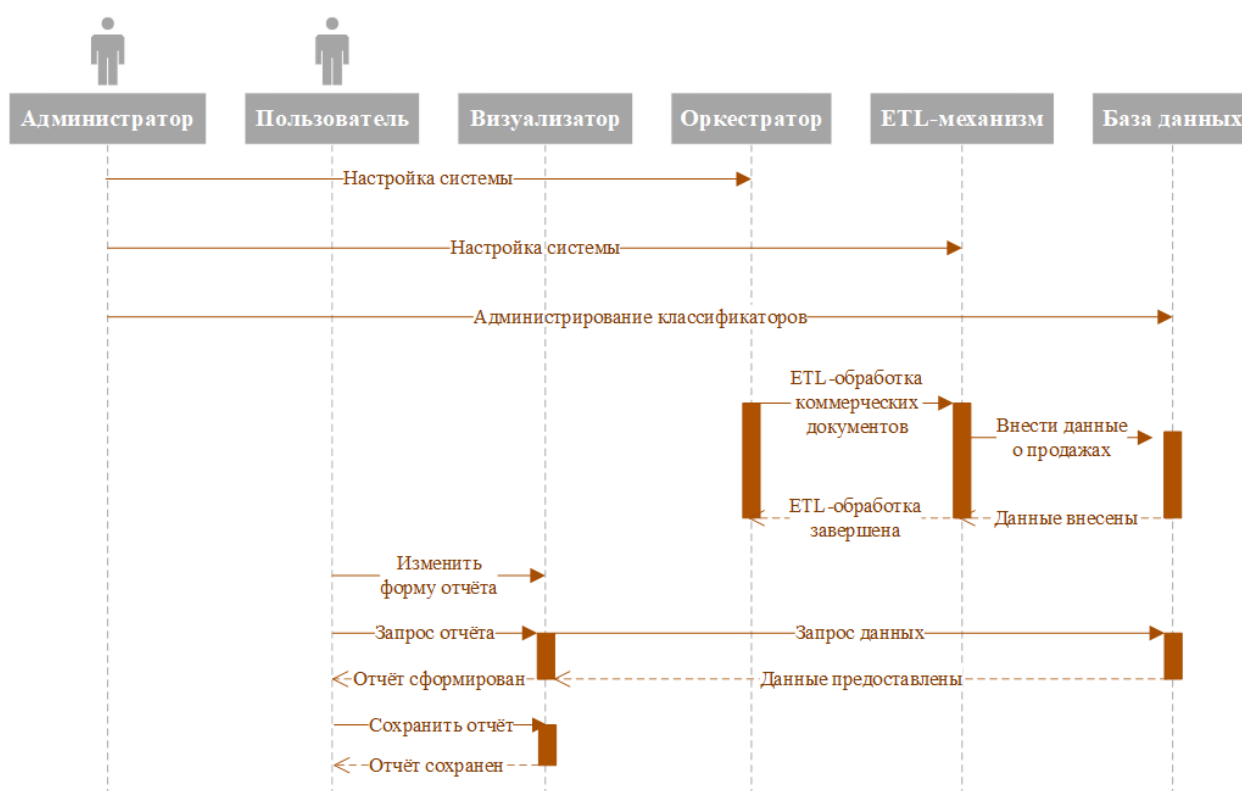


Рисунок 10 – UML-диаграмма последовательностей ИС

В качестве субъектов системы выступают пользователи и администратор. Объектами системы являются визуализатор, оркестратор, ETL-механизм и база данных.

Взаимодействия с администратором не имеет четко выраженного формализованного процесса и зависит от конкретной реализации компонента информационной системы, поэтому взаимодействие с администратором показано как простое сообщение компоненту системы. Оркестратор инициирует процесс ETL-обработки коммерческих документов по заранее настроенному расписанию. В процессе обработки данные извлекаются из первичных коммерческих документов и вносятся в базу данных. Конкретные особенности процесса ETL-обработки зависят от деталей реализации. Пользователь может запросить визуализатор сформировать отчет, тем самым инициируя процесс формирования отчета. Процесс формирования отчета зависит от конкретной реализации визуализатора, однако концептуально представляет собой запрос к базе данных информационной системы. Визуализатор поддерживает возможность изменения визуальной и информационной структуры отчета (представлено как простое сообщение компоненту системы) и сохранение сформированного отчета (представлено в виде процесса) для его дальнейшего использования.

2.3 Информационное обеспечение информационной системы

2.3.1 Используемые классификаторы и системы кодирования

Для формирования отчета необходимого менеджерам отдела розничных продаж, помимо данных о продажах, необходимо иметь данные о самих товарах и категориях, к которым эти товары относятся. Эти дополнительные данные представлены в базе данных в локальных классификаторах. Классификатор товара может использоваться в данных предоставляемых посредством универсальным передаточным документом (УПД) [11] для упрощения реализации информационной системы. В таблице 3 перечислены используемые классификаторы.

Таблица 3 – Используемые дополнительные классификаторы

Множество объектов классификации	Разрядность кода	Система кодирования	Система классификации	Вид классификатора
Товар	5	Порядковая	Отсутствует	Локальный
Единица измерения товара	3	Порядковая	Отсутствует	Локальный
Категория товара	3	Порядковая	Отсутствует	Локальный

Классификаторы единиц измерения и категорий товаров используются только в рамках информационной системы для формирования отчётов. Заполнение кодов классификаторов происходит автоматически – средствами базы данных.

2.3.2 Характеристика нормативно-справочной и входной оперативной информации

В качестве нормативно-справочной и входной информации выступают справочники и первичные коммерческие документы. Справочники содержат данные о товарах, их характеристиках, категории, продажах и доступности каждого конкретного товара. Справочники категорий, характеристики товаров и их доступности вносятся в систему администратором по мере внедрения и эксплуатации системы. Справочник о продажах обновляется в автоматическом режиме компонентами системы на основе первичных коммерческих документов. Рассмотрим детальную информацию по каждому из справочников, вносимых администратором.

Категория является абстрактным понятием и характеризуется именем категории. Данные о принадлежности конкретного товара категории хранятся вне справочника категорий, таким образом добавление новой категории не оказывает влияния на текущую классификацию товаров. В таблице 4 указаны параметры справочника категорий.

Таблица 4 – Справочник категорий

Идентификатор	Параметры информации
Тип данных	Текстовая
Частота обновления	По мере выделения новых категорий
Источник информации	Категорийный менеджер
Перечень атрибутов	Имя категории

Единица измерения является конкретным понятием и характеризуется именем единицы измерения. Данные о единицах измерения каждого конкретного товара размещены вне справочника единиц измерения. Таким образом, при добавлении новой единицы измерения – данные текущих отчётов не будут искажены. В таблице 5 представлены параметры атрибутов справочника единиц измерения.

Таблица 5 – Справочник единиц измерения

Идентификатор	Параметры информации
Тип данных	Текстовая
Частота обновления	По мере выделения новых единиц измерения
Источник информации	Начальник отдела закупок
Перечень атрибутов	Имя единицы измерения

Информация о товарах является конкретной и содержит как текстовую информацию, отражающую имя товара, так и числовую, посредством которой представлены классификация (по единицам измерения, по категориям).

В случае изменения классификации (добавления новых единиц измерения или категорий), для отражения изменений в формируемых системой отчётах – необходимо изменить атрибуты всех товаров, относящихся к новой классификации. В таблице 6 представлена информация об атрибутах справочника товаров.

Таблица 6 – Справочник товаров

Идентификатор	Параметры информации
Тип данных	Текстовая и числовая
Частота обновления	По мере появления новых товаров в продаже или редактирования атрибутов
Источник информации	Начальник отдела закупок
Перечень атрибутов	Имя товара, код категории товара, код единицы измерения

Информация о доступности товаров содержит числовую информацию о доступных на складе единицах товара и стандартной стоимости единицы, возможно редактирование атрибута стандартной стоимости товара при изменении отпускной цены. Данные о дате редактирования доступности вносятся системой автоматически при редактировании. В таблице 7 представлена информация об атрибутах справочника доступности товаров.

Таблица 7 – Справочник доступности товаров

Идентификатор	Параметры информации
Тип данных	Числовая
Частота обновления	Через заданные промежутки времени (месяц/неделя)
Источник информации	Руководитель склада
Перечень атрибутов	Код товара, единиц товара доступно, стандартная стоимость, дата обновления

Рассмотренные справочники редактируются администратором системы. Внесение новых данных в справочники, или их редактирование, происходят по мере необходимости, в случаях: справочников категории, единиц измерения и товаров, и на регулярной основе через заданные промежутки времени в случае справочника доступности товаров.

Приведём информацию об атрибутах справочника, автоматически обновляемого компонентами информационной системы. В таблице 8 представлены данные атрибутов справочника продаж.

Таблица 8 – Справочник продаж

Идентификатор	Параметры информации
Тип данных	Числовая
Частота обновления	Ежедневно
Источник информации	Первичные коммерческие документы
Перечень атрибутов	Код продажи, код товара, количество единиц, фактическая стоимость, дата продажи

В качестве входной оперативной информации выступают первичные коммерческие документы в формате универсальных передаточных документов (УПД) [11]. Документы представлены в формате расширяемого языка разметки (XML) и предполагают возможность автоматической обработки посредством программных средств на языке программирования Python 3. На рисунке 11 представлен фрагмент подобного УПД XML-файла с информацией о проведённой продаже.

```

<ТаблСчФакт>
<СведТов НомСтр="1" НаимТов="Вино Тамань" ОКЕИ_Тов="796" КолТов="25" ЦенаТов="450.43" СтТовБезНДС="11260.75" НалСт="20%" СтТовУчНал="13512.9">
  <Акции>
    <СумАкции>20</СумАкции>
  </Акции>
  <СумНал>
    <СумНал>2252.15</СумНал>
  </СумНал>
  <ДопСведТов ПрТовРаб="1" НаимЕдИзм="Штука" ХарактерТов="вино" СортТов="высший" АртикулТов="F0001" КодТов="F0001_1"/>
</СведТов>
<СведТов НомСтр="2" НаимТов="Услуги по упаковке" ОКЕИ_Тов="876" КолТов="1" ЦенаТов="500" СтТовБезНДС="500" НалСт="20%" СтТовУчНал="600">
  <Акции>
    <БезАкции>без акциза</БезАкции>
  </Акции>
  <СумНал>
    <СумНал>100</СумНал>
  </СумНал>
  <ДопСведТов ПрТовРаб="3" НаимЕдИзм="Условная единица"/>
</СведТов>
<СведТов НомСтр="3" НаимТов="Услуги по доставке" ОКЕИ_Тов="876" КолТов="2" ЦенаТов="1500" СтТовБезНДС="3000" НалСт="20%" СтТовУчНал="3600">
  <Акции>
    <БезАкции>без акциза</БезАкции>
  </Акции>
  <СумНал>
    <СумНал>600</СумНал>
  </СумНал>
  <ДопСведТов ПрТовРаб="3" НаимЕдИзм="Условная единица"/>
</СведТов>
<ВсегоОпл СтТовБезНДСВсего="14760.75" СтТовУчНалВсего="17712.9">
  <СумНалВсего>
    <СумНал>2952.15</СумНал>
  </СумНалВсего>
</ВсегоОпл>
</ТаблСчФакт>

```

Рисунок 11 – Фрагмент УПД XML-файла

Рассматриваемый формат является формализованным требованием федеральной налоговой службы и регулируется законодательством [11]. Так как эта форма представления данных уже используется компанией в рамках экосистемы конфигураций 1С [15], то возможно её использование и для построения информационной системы отдела розничных продаж.

2.3.3 Характеристика выходной информации

Основным документом, характеризующим выходную информацию системы, является отчёт для менеджеров отдела розничных продаж. Отчёт является консолидированной информацией о продажах, разбитой по категориям и единицам измерения товара. Форма отчёта не является фиксированной, однако, так как отчёт формируется на основе информации, описанной в пунктах 2.3.1 и 2.3.2 – существует его полная форма, оперирующая всеми доступными данными. Отчёт представляет собой информационную таблицу, содержащую столбцы с данными, извлечёнными из базы данных информационной системы. В таблице 9 представлены названия столбцов, типы данных, справочники участвующие в их формировании и описание информации размещённой в столбцах таблицы.

Таблица 9 – Описание характеристик столбцов таблицы отчёта

Название столбца	Тип данных	Справочник	Описание столбца
Категория	Текстовый	Категорий	Имя категории, к которому отнесён товар
Единица измерения	Текстовый	Единиц измерения	Имя единицы измерения товара
Имя товара	Текстовый	Товаров	Имя товара как оно внесено в справочник товаров
Период продаж	Текстовый	Продаж	Отчётный период продаж на основе данных о котором формируется отчёт
Всего единиц	Числовой	Продаж	Всего единиц проданного товара за отчётный период
Суммарная стоимость	Числовой	Продаж	Суммарная стоимость проданных товаров за отчётный период

Продолжение таблицы 9

Название столбца	Тип данных	Справочник	Описание столбца
Доступно единиц	Числовой	Доступности товаров	Всего доступно единиц товара на период формирования отчёта
Дата редактирования доступности	Текстовый	Доступности товаров	Дата последнего обновления информации о доступности товара

Пользователь системы может как воспользоваться полным отчётом, так и исключить некоторые столбцы из информационной таблицы для формирования индивидуальных отчётов по интересующей информации. Данные в информационную таблицу вносятся динамически во время процесса формирования отчёта в клиенте информационной системы (визуализаторе). Информация доступная в отчёте может быть сохранена посредством инструментов, предоставляемых визуализатором.

2.4 Проектирование базы данных информационной системы

2.4.1 Выбор технологии проектирования базы данных информационной системы

Методология проектирования базы данных может быть обобщена до следующих трёх этапов [13, с. 142]:

- концептуальное моделирование,
- логическое моделирование,
- физическое моделирование.

На этапе логического проектирования необходимо смоделировать базу данных в терминах реляционной модели, но без учёта физических особенностей представления данных [13, с. 142]. Для решения подобных задач существует несколько альтернативных методологий, самой развитой и современной из которых является нотация IDEF1X [13, с. 143]. Построение модели осуществляется с помощью CASE-средства Microsoft Visio 2021.

2.4.2 Разработка логической модели базы данных информационной системы

В качестве концептуальной модели базы данных воспользуемся UML-диаграммой классов информационной системы, показанной на рисунке 9. Приведём классы UML-диаграммы к сущностям IDEF1X и обозначим отношения между этими сущностями. На рисунке 12 изображена логическая модель базы данных информационной системы в нотации IDEF1X.



Рисунок 12 – Логическая модель базы данных информационной системы

Показанная модель отражает связи между сущностями системы и типы показанных связей. В качестве первичных ключей в сущностях используются локальные классификаторы, указанные в пункте 2.3.1. Образование связей между сущностями осуществляются посредством использования внешних ключей, ссылающихся на первичные ключи других сущностей. Сущности

информации о продажах и информации о доступности товара на складе не имеют собственных первичных ключей, однако должны поддерживать уникальность записей и выполнять роль агрегаторов данных о товарах, их численности и стоимости. Неидентифицирующей [13, с. 148] является связь категории и товара. Идентифицирующими [13 с. 147] связями являются:

- товара и единицы измерения,
- товара и продаж,
- товара и доступности товара.

Возможно классифицировать продемонстрированную логическую модель как относящуюся к схеме «Звезда» [14, с. 20-21]. Так как основной анализируемой сущностью разрабатываемой информационной системы являются товар и его атрибуты, то справочник товаров выполняет в этой схеме роль таблицы фактов и обеспечивает, с помощью использования внешних ключей, связи между таблицами размерностей (измерений), роль которых выполняют справочники категорий, единиц измерения, продаж и доступности товаров. Подобные схемы хорошо себя зарекомендовали как структуры данных, обеспечивающие высокую скорость выполнения запросов (что критически важно для обеспечения комфортной одновременной работы множества пользователей) посредством денормализации данных [14, с. 20].

2.5 Требования к аппаратно-программному обеспечению информационной системы

Рассмотрим требования, выдвигаемые в пункте 1.2.3 к производительности информационной системы и наложенные проектные ограничения. Требования к производительности предполагают размещение компонентов системы на сервере для обеспечения одновременного доступа не менее 40 пользователей системы. Определим архитектуру, на которой необходимо разместить такой сервер. Так как использование малораспространённой архитектуры набора команд повышает стоимость

поддержки и эксплуатации информационной системы, то необходимо выбрать наиболее популярный тип процессора на серверном рынке. На рисунке 13 изображена динамика изменения доли основных игроков рынка серверных процессоров по кварталам начиная с 2000 года.

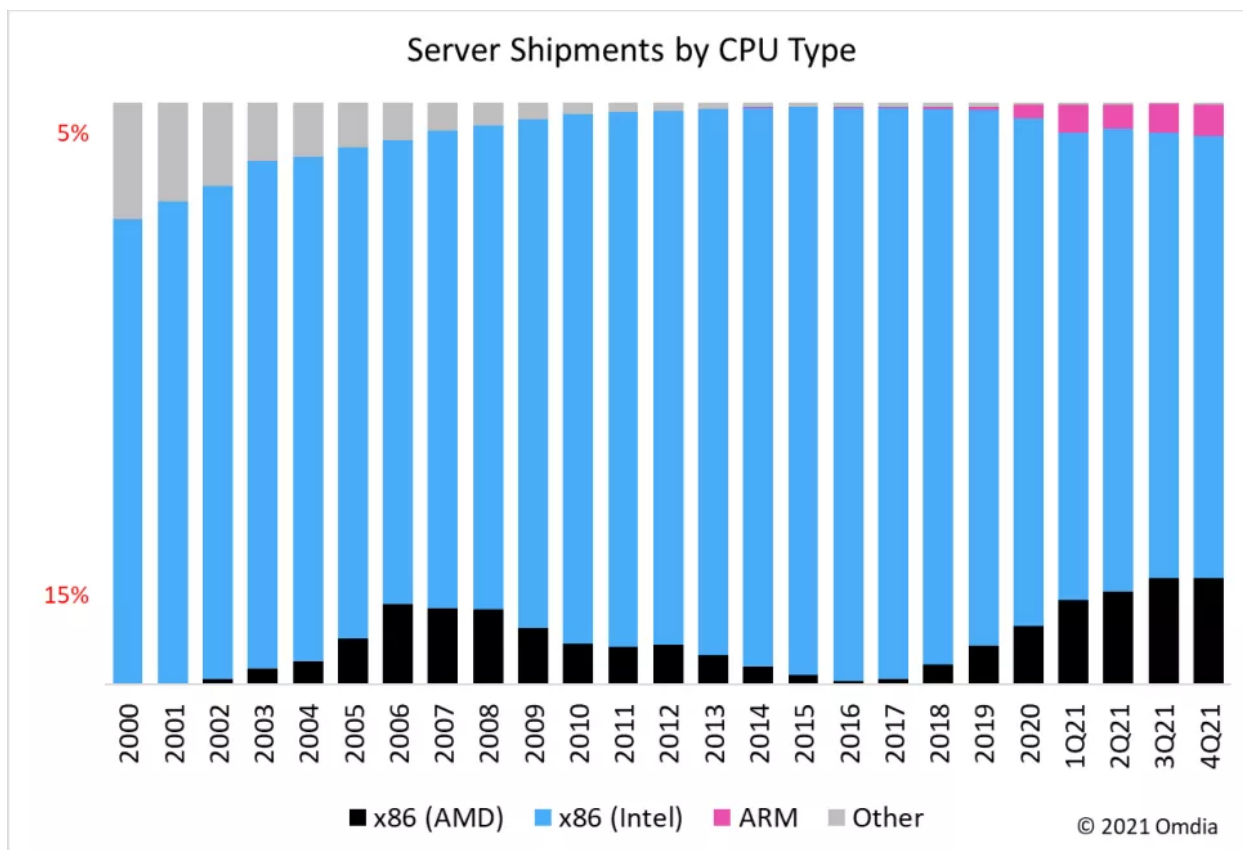


Рисунок 13 – Динамика изменения доли рынка игроков серверного сегмента

Как показывает анализ – наиболее широко в современном серверном сегменте представлена архитектура x86 с 94,3% от общего объёма рынка [18]. Следовательно, необходимо определить оптимальные системные требования, предъявляемые к серверу на x86 архитектуре.

Для определения оптимальных параметров аппаратной конфигурации сервера необходимо провести анализ используемых программных средств и ограничений, вызванных функциональным назначением серверных компонентов системы. Так как, фактически, система представляет собой набор программных решений, работающих поверх слоя абстракций,

предоставляемых операционной системой, но необходимо определить минимальную аппаратную конфигурацию сервера, поддерживающую установку и запуск операционной системы.

Одним из проектных ограничений является использование операционной системы Debian версии 11 и выше. Определим минимальные требования к аппаратному обеспечению, которые позволяют установить и использовать указанную операционную систему. В таблице 10 указаны минимальные системные требования для установки [21] и запуска Debian 11.

Таблица 10 – Минимальные системные требования к серверу x86

Частота процессора	Число ядер процессора	Объём оперативной памяти	Объём запоминающего устройства	Число запоминающих устройств	Тип RAID
от 1 ГГц	1	от 512 Мбайт	от 10 Гбайт	1	Нет

Однако, подобная конфигурация не является достаточной для обеспечения полноценного функционирования информационной системы. Поэтому, необходимо провести анализ аппаратных потребностей системы, основанный на функциональном предназначении её компонентов. Так как размеры предприятия позволяют предположить отсутствие необходимости в обработке огромных объёмов оперативных данных, классифицируемых как BigData [14, с. 36], то основным лимитирующим фактором для построения корпоративного хранилища данных является производительность компонента базы данных информационной системы. Определим оптимальные характеристики для размещения сервера базы данных на основе операционной системы Debian 11. В таблице 11 отражена оптимальная конфигурация [4] для обеспечения одновременной работы 40 пользователей (для обеспечения комфортной работы необходимо обеспечить 20% запас по производительности в сравнении с максимальной нагрузкой).

Таблица 11 – Оптимальные системные требования к серверу x86

Частота процессора	Число ядер процессора	Объём оперативной памяти	Объём запоминающего устройства	Число запоминающих устройств	Тип RAID
от 2,5 ГГц	8	от 16 Гбайт	от 100 Гбайт	6	RAID 5

Оркестратор и механизм, обеспечивающий ETL-процесс обработки первичных коммерческих документов, размещается на том же сервере что и база данных. Так как обработка данных происходит вне рабочего периода, то основная нагрузка на сервер распределена во времени и возможно использование одной аппаратной конфигурации сервера для обеспечения работы всех серверных компонентов системы без падения её производительности.

Выводы по главе 2

Во второй главе были выделены аспекты логической модели информационной системы, необходимые для более полного отражения поведения системы. Выбрана и аргументирована методология построения логической модели. Используя материалы, предоставленные в первой главе бакалаврской работы в качестве базы, посредством UML-диаграмм, смоделированы функциональный, элементный и динамический аспекты системы. Рассмотрено информационное обеспечение системы: классификаторы и системы кодирования, нормативно-справочная и входная оперативная информация, характеристика выходной информации. Выбрана технология проектирования базы данных (БД) и посредством нотации IDEF1X отражена логическая модель БД. Основываясь на требованиях к производительности и проектных ограничениях, проведен анализ и определены минимальная (для запуска ОС) и оптимальная (для обеспечения комфортного функционирования) аппаратные конфигурации информационной системы.

Глава 3 Физическое проектирование информационной системы

3.1 Выбор архитектуры информационной системы

Корректное функционирование информационной системы предприятия обусловлено факторами корректности и достаточной эффективности функционирования её аспектов:

- программного (функциональность средств программного обеспечения компонентов системы);
- информационного (качество информационных массивов баз данных и их актуальность);
- пользовательского (навыки, осведомлённость пользователей и совокупность сопроводительной документации).

Таким образом, для обеспечения эффективного функционирования и эксплуатации информационной системы предприятия, необходимо удостовериться, что программные компоненты и архитектура ИС полностью выполняют все требования, предъявляемые к разрабатываемой системе. Проведём анализ соответствия требованиям наиболее распространённых архитектурных решений: файл-серверной, двух- и трехзвенной клиент-серверной архитектуры. Для анализа воспользуемся проверкой на соответствие всем пунктам FURPS-части требований их пункта 1.2.3, без учёта дополнительных ограничений, налагаемых на проект, так как они зависят от выбора конкретной реализации, а не архитектуры информационной системы. В таблице 12 указаны атрибуты методологии FURPS. Случай полного соответствия предъявляемым к разрабатываемой информационной системе требованиям обозначается знаком «+», в то время как несоответствие хотя бы одному из пунктов обозначается знаком «-». Предпочтительной считается наиболее простая архитектура, полностью удовлетворяющая всем требованиям.

Таблица 12 – Анализ архитектур на соответствие требованиям FURPS

Атрибут	Файл-сервер	Двухзвенный клиент-сервер	Трёхзвенный клиент-сервер
Functionality (функциональность)	+	+	+
Usability (удобство использования)	-	+	+
Reliability (надежность)	+	+	+
Performance (производительность)	-	+	+
Supportability (сопровождаемость)	-	+	+

Анализ показал, что архитектура файл-сервера имеет серьёзные изъяны и не обеспечивает должного выполнения требований. Двухзвенная и трёхзвенная архитектуры удовлетворяют предъявляемым требованиям. Таким образом, предпочтительным выглядит использование двухзвенной клиент-серверной архитектуры как наиболее простого решения, полностью соответствующего требованиям FURPS описанным в пункте 1.2.3.

Воспользуемся ранее использованной методологией UML и отразим посредством диаграммы компонентов архитектуру разрабатываемой системы. В качестве CASE-средства используется Microsoft Visio 2021. На рисунке 14 показана UML-диаграмма компонентов информационной системы.

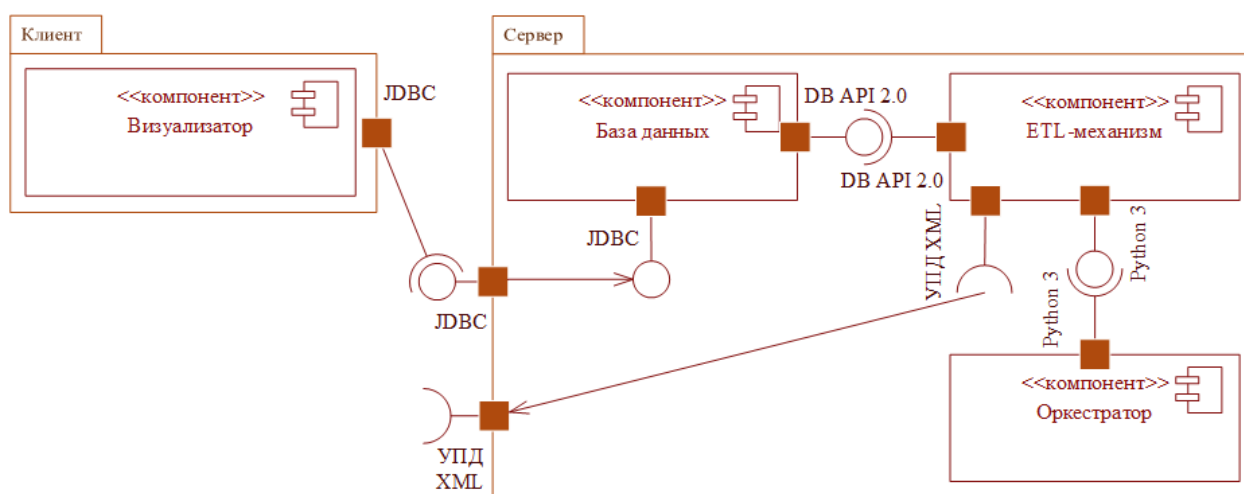


Рисунок 14 – UML-диаграмма компонентов информационной системы

Интерфейсом связи клиента (визуализатора) и сервера (базы данных) является JDBC-коннектор, обеспечивающий платформенно независимый промышленный стандарт взаимодействия между компонентами системы. Как было указано ранее в пункте 2.3.2 – входным интерфейсом серверных компонент системы (ETL-механизм) являются УПД XML-файлы. В качестве интерфейса между оркестратором и ETL-механизмом выступает язык Python 3, что является одним из ограничений разработки указанным в пункте 1.2.3. Так как, стандартным интерфейсом взаимодействия между языком программирования Python 3 и реляционными базами данных, является спецификация DB API 2.0 [24], то интерфейсом взаимодействия ETL-механизма и базы данных была выбрана именно она. Компоненты системы условно разделены на пакеты «клиент» и «сервер», которые отражают физическое размещение компонентов системы при двухзвенной клиент-серверной архитектуре, однако каждый из компонентов при необходимости изменения архитектуры системы может быть размещён на индивидуальной физической или виртуальной машине без потери функциональности.

3.2 Выбор технологии разработки программного обеспечения информационной системы

Для создания информационной системы отдела розничных продаж ООО «Айковер ПРО» необходимо пройти весь цикл разработки. С целью обеспечения наибольшей полезности системы при наименьших затратах на её создание – необходимо определить оптимальную технологию разработки программных компонентов ИС. Выбор технологии должен учитывать потребности конечного пользователя и функциональность инструментов разработки.

Так как конечные потребности пользователя, которые должна удовлетворить разрабатываемая система, являются внешними по отношению к разработчикам системы, то необходимо обеспечить наиболее оперативное

реагирование проекта на обозначенные потребности конечных пользователей. Таким образом, наиболее подходящей технологией разработки проекта информационной системы является гибкая методология разработки, позволяющая за счёт сведения разработки к серии коротких циклов (итераций) минимизировать риски, связанные с изменением требований к разрабатываемой системе.

Одним из компонентов разрабатываемой системы является ETL-механизм, который необходимо реализовать посредством кода на языке программирования. В пункте 1.2.3 одним из ограничений разработки было названо использование языка программирования Python 3, для построения системы. Таким образом, необходимо рассмотреть наиболее популярные открытые интегрированные среды разработки (IDE) и выбрать подходящую. В таблице 13 проанализированы популярные открытые решения IDE, поддерживающие разработку на Python 3 по параметрам: порог вхождения (уровень затрат времени необходимых для изучения базовой функциональности инструмента и начала самостоятельной продуктивной работы), поддержка систем контроля версий, доступные лицензии и поддерживаемые операционные системы.

Таблица 13 – Анализ популярных открытых IDE для Python 3

Название IDE	Порог вхождения	Системы контроля версий	Доступные лицензии	Операционные системы
Eclipse	Высокий	SVN, Git	Eclipse Public License (EPL)	Windows, Linux, Mac OS X
Visual Studio Code	Низкий	SVN, Git	MIT License	Windows, Linux, Mac OS X
PyCharm	Низкий	SVN, Git	GNU General Public License, коммерческая лицензия	Windows, Linux, Mac OS X

Из данных, представленных в анализе, можно заключить, что все представленные IDE обладают сходной поддержкой систем контроля версий

и операционных систем. Таким образом, критериями отбора становится порог вхождения и доступные лицензии, что не оказывает влияния на конечное качество реализации проекта, поэтому среда разработки может быть выбрана на основании личных предпочтений конкретного программиста.

3.3 Выбор системы управления базами данных

Определим, наиболее соответствующий логической модели (изображена на рисунке 12) способ хранения данных СУБД информационной системы. По способу хранения данных [17] системы управления базами данных делятся на:

- строковые,
- столбцовые (колоночные).

Так как представленная модель находится в первой нормальной форме [13, с. 160] и исполнение основной функции информационной системы (формирования отчёта) требует извлечение всего массива данных, то при использовании аналогичных дисковых массивов и подсистем ввода-вывода – изменение способа хранения данных СУБД не окажет существенного влияния на производительность решения. Следовательно, проведение сравнительного анализа строковых и столбцовых решений корректно.

Воспользуемся данными о наиболее популярных СУБД, представленных на портале DB-Engines [20], и проведём сравнительный анализ трёх наиболее популярных реляционных решений с открытым исходным кодом доступных в дистрибутиве Debian: MySQL, PostgreSQL, MariaDB. Критериями для сравнения альтернатив являются: соответствие требованиям интерфейсов, указанных в параграфе 3.1, диалект SQL, поддержка многопользовательского режима, поддержка удалённого доступа, способ хранения данных, доступные лицензии и ранг СУБД на портале DB-Engines на май 2022. Предпочтительным вариантом является наиболее популярная альтернатива, не предполагающая увеличения общей стоимости

реализации проекта, полностью соответствующая требованиям, предъявляемым к интерфейсам СУБД информационной системой. В таблице 14 представлен сравнительный анализ указанных решений по обозначенным критериям.

Таблица 14 – Сравнительный анализ СУБД

Критерий	MySQL	PostgreSQL	MariaDB
Поддержка JDBC	Поддерживает	Поддерживает	Поддерживает
Поддержка DB API 2.0/ Имя библиотеки Python 3	Поддерживает/ mysql-connector- python	Поддерживает/ psycopg2	Поддерживает/ mariadb
Поддержка многопользовательского режима	Поддерживает	Поддерживает	Поддерживает
Поддержка удалённого доступа	Поддерживает	Поддерживает	Поддерживает
Диалект SQL	MySQL	PostgreSQL	MySQL
Способ хранения данных	Строковый и столбцовый	Строковый и столбцовый	Строковый и столбцовый
Доступные лицензии	GNU GPL 2, коммерческая	PostgreSQL Licence	GNU GPL, GNU LGPL
Ранг СУБД на портале DB- Engines на май 2022	2	4	12

Из данных представленного анализа можно заключить, что анализируемые характеристики рассмотренных СУБД имеют сходные параметры. Таким образом, выбор конкретного решения зависит от предпочтительности конкретной лицензии и диалекта SQL. Предполагая необходимость использования коммерческой лицензии MySQL (в случае её выбора) и приоритетность сокращения общей стоимости реализации и внедрения проекта – наиболее предпочтительным выглядит разработка решения, построенного на СУБД PostgreSQL. Это наиболее популярное [20] решение, удовлетворяющее всем требованиям к интерфейсам взаимодействия с базой данных и не предполагающее покупки коммерческой лицензии для использования.

3.4 Разработка физической модели данных информационной системы

Воспользуемся логической моделью базы данных информационной системы, показанной на рисунке 12, и реализуем структуру в терминах СУБД PostgreSQL версии 14.3, с учётом конкретных физических особенностей хранения [13, с. 142].

На рисунке 15 показан SQL-запрос на создание структуры таблицы категорий.

```
CREATE TABLE IF NOT EXISTS public.category
(
    id integer NOT NULL DEFAULT nextval('category_id_seq'::regclass),
    name text COLLATE pg_catalog."default",
    CONSTRAINT category_pkey PRIMARY KEY (id)
)
```

Рисунок 15 – SQL-запрос создания таблицы категорий

Поле `id` выполняет роль первичного ключа и отражает код категории. Поля `name` является текстовым и содержит имя категории. На рисунке 16 показан пример внесения тестовой категории в таблицу.

```
INSERT INTO public.category (name) VALUES ('Тестовая категория');
```

Рисунок 16 – Пример SQL-запроса внесения категории в таблицу

Как можно заметить, генерация кода категории происходит автоматически средствами СУБД и является последовательно увеличивающимся целым числом.

На рисунке 17 отражен SQL-запрос формирования структуры таблицы единиц измерения.


```
CREATE TABLE IF NOT EXISTS public.units
(
    id integer NOT NULL DEFAULT nextval('units_id_seq'::regclass),
    name text COLLATE pg_catalog."default",
    CONSTRAINT units_pkey PRIMARY KEY (id)
)
```

Рисунок 17 – SQL-запрос создания таблицы единиц измерения

Поле `id` выполняет роль первичного ключа и отражает код единицы измерения. Поля `name` является текстовым и содержит имя единицы измерения. На рисунке 18 показан пример внесения тестовой единицы измерения в таблицу.

```
INSERT INTO public.units (name) VALUES ('штуки');
```

Рисунок 18 – Пример SQL-запроса внесения единицы измерения в таблицу

Генерация кода единицы измерения происходит автоматически средствами СУБД и является последовательно увеличивающимся целым числом.

Коды категории товаров и коды единиц измерения используются как внешние ключи в таблице товаров и отражают дополнительные измерения данных о товарах. Таким образом, наименее трудозатратным с точки зрения внесения данных в системы является первоначальное внесение всех необходимых категорий и единиц измерения в системы с последующим использованием уже готовых ключей при внесении данных в таблицу товаров.

Рассмотрим процесс создания центральной таблицы фактов. На рисунке 19 изображен SQL-запрос на создание структуры таблицы товаров.

```

CREATE TABLE IF NOT EXISTS public.goods
(
    id integer NOT NULL DEFAULT nextval('goods_id_seq'::regclass),
    name text COLLATE pg_catalog."default",
    category_id integer NOT NULL DEFAULT nextval('goods_category_id_seq'::regclass),
    units_id integer NOT NULL DEFAULT nextval('goods_units_id_seq'::regclass),
    CONSTRAINT goods_pkey PRIMARY KEY (id),
    CONSTRAINT goods_category_id_fkey FOREIGN KEY (category_id)
        REFERENCES public.category (id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION,
    CONSTRAINT goods_units_id_fkey FOREIGN KEY (units_id)
        REFERENCES public.units (id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
)

```

Рисунок 19 – SQL-запрос создания таблицы товаров

Поле `id` выполняет роль первичного ключа и отражает код единицы измерения. Поля `name` является текстовым и содержит имя единицы измерения. Поля `category_id` и `units_id` являются внешними ключами и отражают первичные ключи таблиц `category` и `units`. Эти поля связывают конкретные товары с конкретными категориями и конкретными единицами измерения. На рисунке 20 показан пример внесения тестового товара в таблицу.

```

INSERT INTO public.goods (name, category_id, units_id) VALUES ('Тест', 1, 1);

```

Рисунок 20 – Пример SQL-запроса внесения товара в таблицу

Генерация кода товара происходит автоматически средствами СУБД и является последовательно увеличивающимся целым числом.

Код товара используется как внешний ключ в таблицах продаж и доступности товаров. Таким образом, наиболее эффективным способом внесения данных о продажах и/или доступности является начальное внесение

в базу данных всей информации о товарах с последующим использованием готовых ключей товаров для формирования консолидирующих таблиц.

На рисунке 21 приведён пример SQL-запроса формирования структуры таблицы продаж.

```
CREATE TABLE IF NOT EXISTS public.sales
(
    sale_id text COLLATE pg_catalog."default" NOT NULL,
    goods_id integer NOT NULL DEFAULT nextval('sales_goods_id_seq'::regclass),
    quantity bigint NOT NULL,
    price double precision NOT NULL,
    sale_date date NOT NULL,
    CONSTRAINT sales_sale_id_goods_id_key UNIQUE (sale_id, goods_id),
    CONSTRAINT sales_goods_id_fkey FOREIGN KEY (goods_id)
        REFERENCES public.goods (id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
)
```

Рисунок 21 – SQL-запрос создания таблицы продаж

Поле `sale_id` является текстовым и служит для идентификации факта принадлежности товара конкретной продаже. Поле `goods_id` является внешним ключом и отражает первичный ключ таблицы `goods`. Это поле связывает продажу с конкретным товаром. Поля `quantity` и `price` являются целым и дробным числом соответственно, отражая количество проданного товара и его фактическую общую стоимость. Поле `sale_date` представляет собой данные о дате продажи. Сочетание колонок `sale_id` и `goods_id` уникально в рамках таблицы чем обеспечивается неизменность данных.

На рисунке 22 представлен пример SQL-запроса для внесения тестовых данных в таблицу продаж.

```
INSERT INTO public.sales (sale_id, goods_id, quantity, price, sale_date)
VALUES ('test-sales-id', 1, 3, 5000.30, '2022-05-09');
```

Рисунок 22 – Пример SQL-запроса внесения продажи в таблицу

Как можно увидеть на тестовом запросе – все поля данных продажи являются обязательными. Источником данных для таблицы продаж служат первичные коммерческие документы в УПД XML-формате автоматически обрабатываемые ETL-механизмом системы.

На рисунке 23 показан пример SQL-запроса для формирования структуры таблицы доступности товаров.

```
CREATE TABLE IF NOT EXISTS public.availability
(
  goods_id integer NOT NULL DEFAULT nextval('availability_goods_id_seq'::regclass),
  quantity bigint NOT NULL,
  last_update date NOT NULL DEFAULT CURRENT_DATE,
  price double precision NOT NULL,
  CONSTRAINT availability_goods_id_key UNIQUE (goods_id),
  CONSTRAINT availability_goods_id_fkey FOREIGN KEY (goods_id)
    REFERENCES public.goods (id) MATCH SIMPLE
    ON UPDATE NO ACTION
    ON DELETE NO ACTION
)
```

Рисунок 23 – SQL-запрос создания таблицы доступности товаров

Поле `goods_id` – внешний ключом и является первичным ключом таблицы `goods`, является уникальным полем. Это поле отражает связь с конкретным товаром. Поля `quantity` (целое число) и `price` (дробное число) отражают количество доступного на складе товара и стандартную стоимость единицы товара. Поле `last_update` показывает дату последнего обновления данных о доступности конкретного товара. На рисунке 24 показан пример SQL-запроса внесения данных о доступности товара.

```
INSERT INTO public.availability (goods_id, quantity, price) VALUES (1, 5, 200.20);
```

Рисунок 24 – Пример SQL-запроса внесения данных о доступности товара

Как можно заметить, дата последнего обновления данных о доступности товара на складе вносится автоматически, средствами СУБД и является датой исполнения запроса к базе данных о внесении товара.

3.5 Разработка программного обеспечения информационной системы

Информационная система отдела розничных продаж ООО «Айковер ПРО» является проектом адаптации инструментов экосистемы программного обеспечения с открытым кодом к особенностям ведения коммерческой деятельности предприятия. Таким образом, для реализации информационной системы необходимо выбрать и сконфигурировать доступное open-source программное обеспечение для достижения необходимой функциональности в соответствии с ранее обозначенными требованиями и ограничениями. Так как используемая база данных определена в параграфе 3.3, а реализация ETL-механизма зависит от конкретного оркестратора, то необходимо произвести выбор оставшихся компонентов системы.

Определим критерии анализа визуализатора системы. Воспользуемся данными о наиболее популярных решениях реализации отчётности [23] и проведём анализ по критериям: поддержка JDBC, поддерживаемые платформы, доступные лицензии, формат представления отчёта, дата последнего обновления программы и порог вхождения. В таблице 15 представлен сравнительный анализ альтернатив.

Таблица 15 – Сравнительный анализ решений реализации отчётности

Критерий	BIRT	JasperReports	Pentaho
Поддержка JDBC	Поддерживает	Поддерживает	Поддерживает
Поддерживаемые платформы	Windows, Linux, Mac OS X	Windows, Linux, Mac OS X	Windows, Linux, Mac OS X

Продолжение таблицы 15

Критерий	BIRT	JasperReports	Pentaho
Доступные лицензии	Eclipse Public License (EPL)	GNU Lesser General Public License (LGPLv3), EPL	GNU Lesser General Public License (LGPLv2.1)
Формат представления отчёта	XML (.RPTDESIGN)	Java Byte Code (.JASPER)	XML (.PRPT)
Дата последнего обновления	16.05.2021	29.03.2022	06.05.2022
Порог вхождения	Низкий	Высокий	Средний

Анализ показал, что все продукты имеют поддержку JDBC и сходную функциональность, таким образом выбор конкретного визуализатора зависит от предпочтений пользователя и не влияет на производительность системы. Используя параметр порога вхождения, который влияет на длительность периода начального обучения пользователя информационной системы, в качестве критерия отбора, предпочтительным выглядит выбор решения BIRT от Eclipse Foundation, как наиболее простого в освоении.

Важным компонентом информационной системы является оркестратор – Workflow Management System (WMS), он автоматизирует наиболее затратную по времени часть работы – внесение оперативных данных в систему. Оркестратор управляет процессом работы ETL-механизма и обеспечивает его всеми необходимыми ресурсами. Одним из ограничений разработки, оговоренных в пункте 1.2.3, является использование языка программирования Python 3 для построения системы. Таким образом, проведём сравнение двух наиболее популярных WMS-решений с открытым исходным кодом, использующих Python 3 как язык разработки: Airflow и Luigi [19]. Критериями сравнительного анализа являются: разработчик решения, структура исполнения работ, возможность распределённого исполнения, наличие планировщика задач, порог вхождения, наличие веб-интерфейса, возможность взаимодействовать с процессами работ. В таблице 16 показан результат сравнительного анализа WMS-решений.

Таблица 16 – Анализ WMS-решений на базе языка Python 3

Критерий	Airflow	Luigi
Разработчик решения	AirBnB	Spotify
Структура исполнения работ	Ориентированный ациклический граф	Конвейер
Возможность распределённого исполнения	Присутствует	Отсутствует
Наличие планировщика задач	Присутствует	Отсутствует
Порог вхождения	Высокий	Низкий
Наличие веб-интерфейса	Присутствует	Присутствует
Возможность взаимодействия с исполняющимися работами	Присутствует	Отсутствует

Возможности оркестратора Luigi удовлетворяют требованиям, предъявляемым к оркестратору в пункте 1.2.3, однако отсутствие планировщика задач и возможности распределённого исполнения ограничивают потенциал дальнейшего развития информационной системы, что вносит дополнительные риски в процесс её эксплуатации. Таким образом, более предпочтительным выглядит использование WMS-решения Airflow, как обладающего более обширными возможностями.

Компонентом, непосредственно отвечающим за трансформацию оперативных входных данных в записи базы данных, является ETL-механизм. Процесс ETL-трансформации состоит из двух шагов: интеграции с источником данных и синтаксического анализатора УПД-файлов, трансформирующего XML структуры в записи базы данных.

Воспользуемся средствами предоставляемыми программный интерфейс приложения Airflow и опишем базовую структуру процесса ETL-трансформации. В рамках выбранного оркестратора процесс трансформации и обработки данных представляется в виде ориентированного ациклического графа (DAG), сформированного посредством кода на языке Python 3. WMS-платформа контролирует процесс организованного, посредством структуры рёбер графа, исполнения команд (узлов графа). На рисунке 25 представлен фрагмент кода ETL-механизма в виде DAG для программной платформы Airflow.

```

from airflow import DAG
from datetime import datetime, timedelta
from airflow.operators.bash_operator import BashOperator
from airflow.operators.python_operator import PythonOperator

INPUT = '~/xml-data/*.xml'

DEFAULT = {'owner': 'Dzmitry Prychyna',
           'depends_on_past': False,
           'email': ['evil.java.deus.ex@yandex.by'],
           'retries': 3,
           'sla': timedelta(minutes=60),
           'retry_delay': timedelta(minutes=3),
           'start_date': datetime(2022, 5, 19)}

def parse_file(dbname, user, password, host, port):
    import os
    import psycopg2 as psy
    connector = psy.connect(dbname=dbname, user=user, password=password, host=host, port=port)
    [parse_xml(os.path.join('.', item), connector) for item in os.listdir('.')]
    if os.path.isfile(os.path.join('.', item)) and item.lower().endswith('.xml')]

with DAG('UPD_DAG', catchup=False, default_args=DEFAULT, schedule_interval="0 12 * * *") as dag:
    integrate = BashOperator(task_id="integrate_data", bash_command=f'cp {INPUT} .')
    parse = PythonOperator(task_id="parse_data", python_callable=parse_file,
                          op_kwargs={'dbname': 'postgres', 'user': 'postgres', 'password': '123456',
                                      'host': '172.23.176.1', 'port': '5432'})
    integrate >> parse

```

Рисунок 25 – Фрагмент кода ETL-механизма

Шаг интеграции с источником данных представлен узлом оператора командной оболочки, производящим перенос оперативных данных из архивного каталога в рабочую папку процесса. Также, возможен вариант исполнения процесса и без переноса данных в локальную папку, однако в случае распределённого выполнения кода создание локальной копии данных будет являться необходимым шагом для корректного и эффективного функционирования компонента. Шаг синтаксического анализа УПД XML-файлов представлен узлом оператора Python кода. Концептуально код представляет собой итеративный синтаксический анализ файлов с расширением «.xml» доступных в рабочем каталоге, это позволяет, при необходимости, перевести процесс на распределённую параллельную

обработку первичных коммерческих документов без необходимости глубокой модификации программного кода.

3.6 Описание функциональности информационной системы

Одним из требований к удобству пользования, предъявляемых к системе в пункте 1.2.3, является возможность использование стандартных графических интерфейсов предоставляемых разработчиками компонентов системы. Воспользуемся UML-диаграммой прецедентов (рисунок 7) и проанализируем доступность графических интерфейсов, соответствующих указанным вариантам использования.

Рассмотрим примеры интерфейсов, используемых администратором системы для выполнения своих обязанностей. Веб-интерфейс приложения Airflow служит для настройки функционирования оркестратора и контроля исполнения ETL-процессов. На рисунке 26 представлен пример графического интерфейса WMS-средства, демонстрирующий список ETL-процессов информационной системы, ход их исполнения и параметры запуска работ.

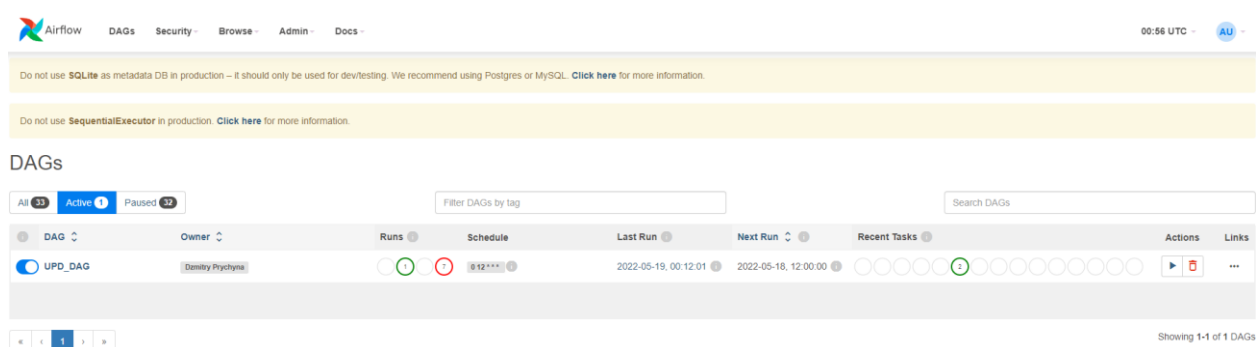


Рисунок 26 – Пример веб-интерфейса приложения Airflow

На рисунке 27 представлен фрагмент графического интерфейса программы pgAdmin – стандартного компонента поставки СУБД PostgreSQL, иллюстрирующий пример администрирования базы данных с целью добавления новой категории в справочник категорий.

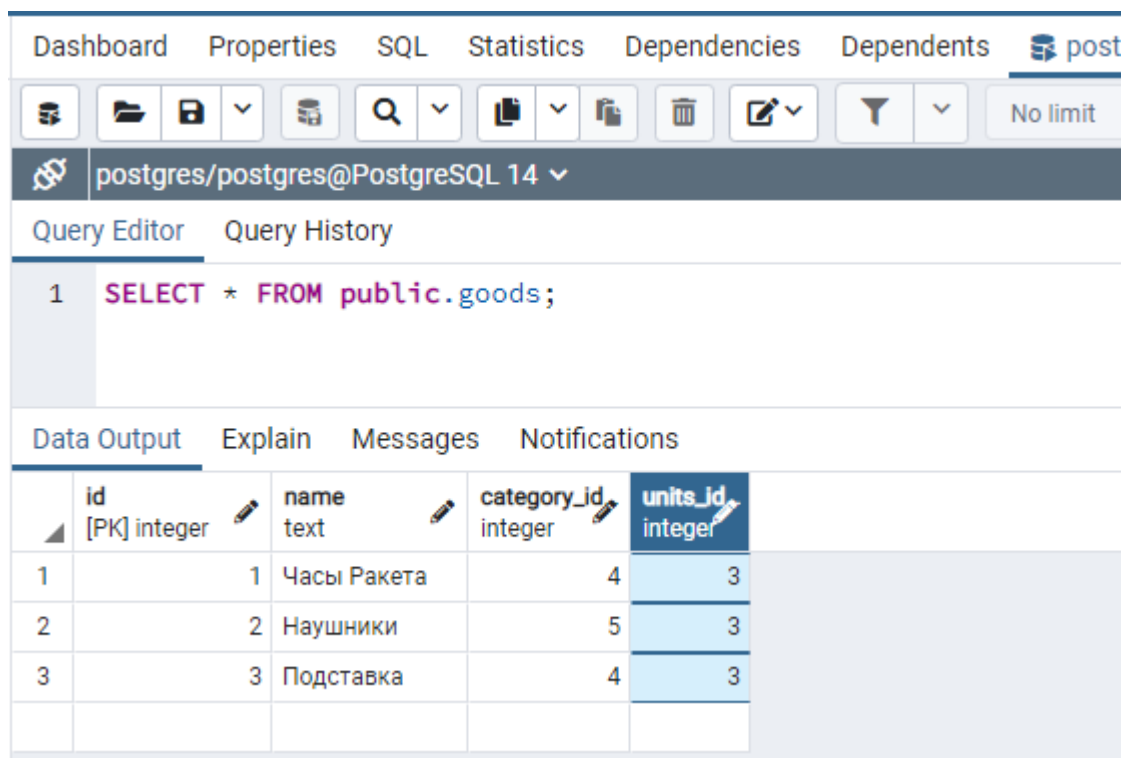


Рисунок 27 – Фрагмент графического интерфейса программы pgAdmin

Таким образом, можно заключить, что администратор системы имеет все необходимые инструменты для корректирования и поддержания работоспособности ИС.

Рассмотрим инструменты доступные пользователю – взаимодействие между пользователем и системой осуществляется посредством средства реализации отчётности BIRT от Eclipse Foundation (согласно параграфу 3.5). Пользователь системы имеет возможность создать отчёт на основе данных доступных в базе данных системы. Отчёт может быть сформирован как на основе стандартных, так и индивидуальных шаблонов, созданных пользователем. На рисунке 28 показан пример окна создания нового отчёта на основе стандартного пустого шаблона.

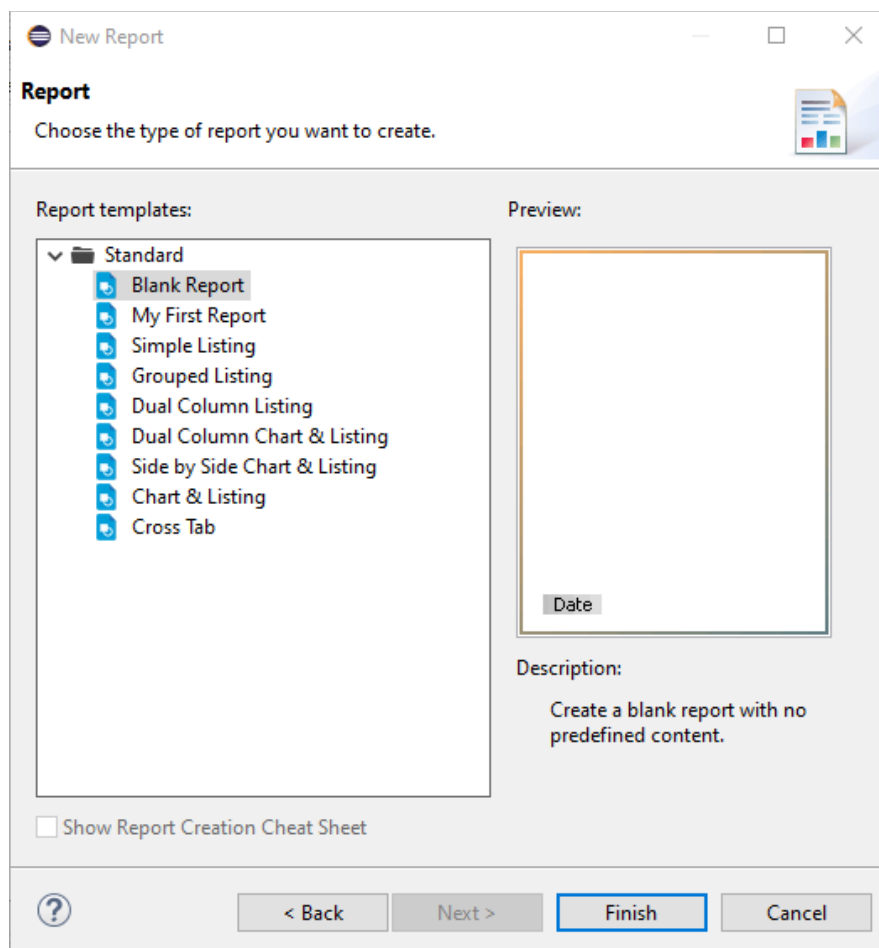


Рисунок 28 – Примеры окон создания шаблона отчёта и отчёта

Для автоматического формирования отчёта необходимо провести подключение JDBC интерфейса (параграф 3.1) базы данных в качестве источника данных. Для этого необходимо скачать необходимую версию JDBC-драйвера из доступных на интернет портале производителя СУБД (<https://jdbc.postgresql.org/download.html>) и с его помощью подключиться к базе данных информационной системы.

На рисунке 29 представлено окно настройки подключения к базе данных с параметрами этого подключения, отражающими: используемый драйвер СУБД и его версию, URL для подключения к JDBC-интерфейсу базы данных, логин пользователя и пароль от базы данных.

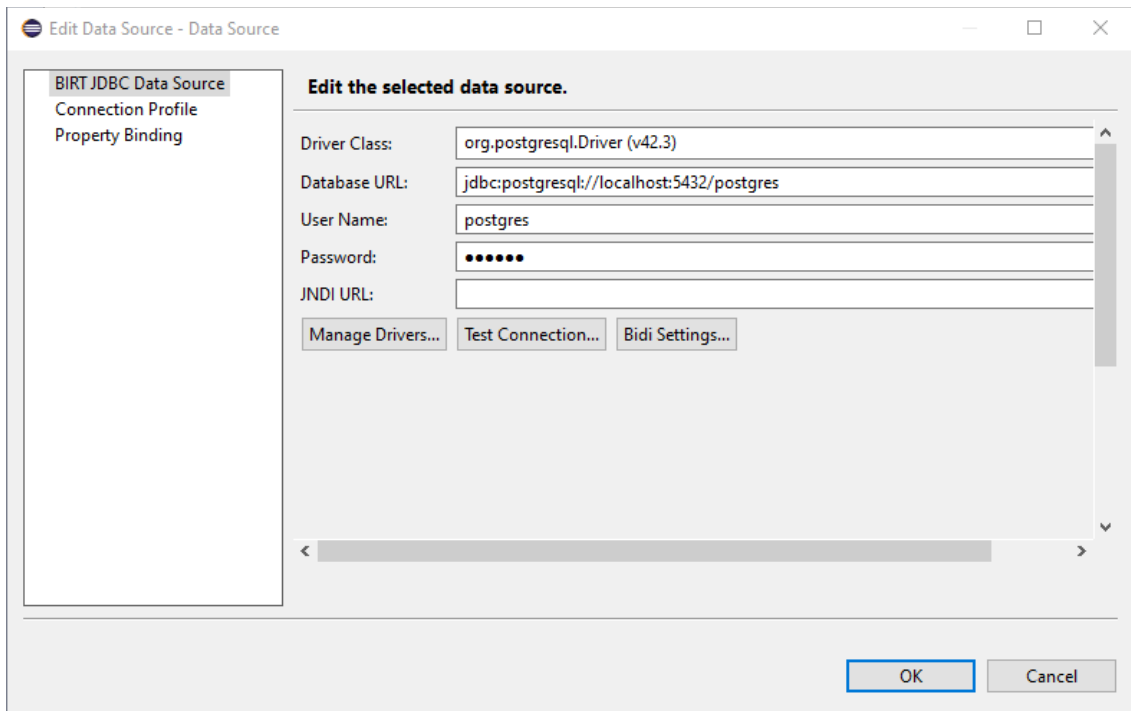


Рисунок 29 – Окно настройки подключения к базе данных

Далее, используя полученное подключение к базе данных пользователь формирует запрос, на основе которого будет формироваться отчёт. На рисунке 30 показано окно формирования SQL-запроса к базе данных.

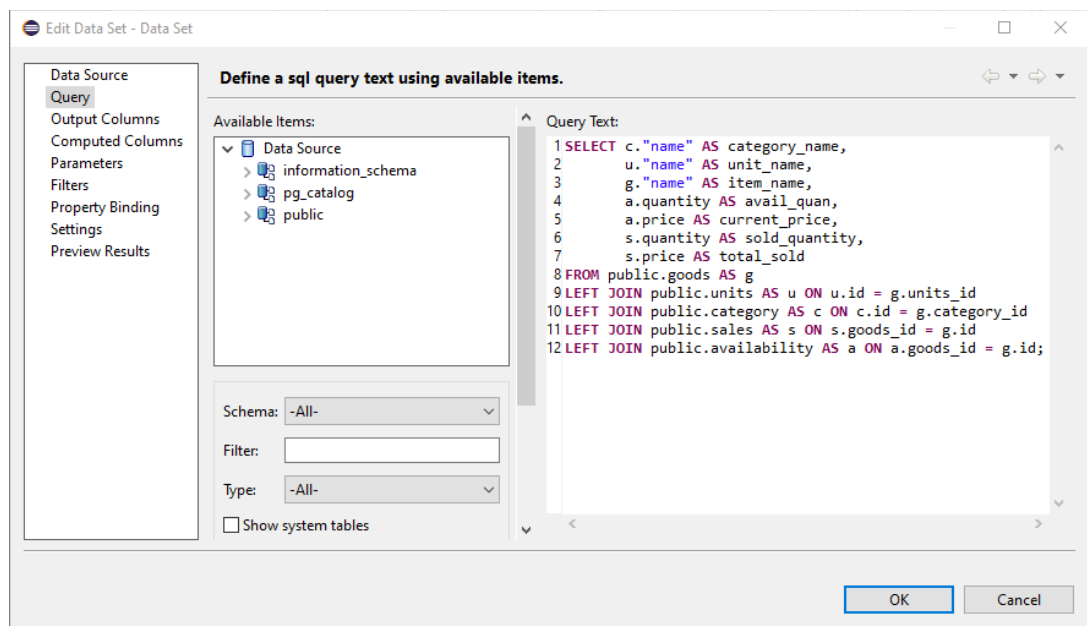


Рисунок 30 – Формирование запроса к базе данных

После этого, данные выборки используются для наполнения сформированной структуры отчёта. На рисунке 31 показан фрагмент диалога запроса формирования и сохранения отчёта в выбранном формате.

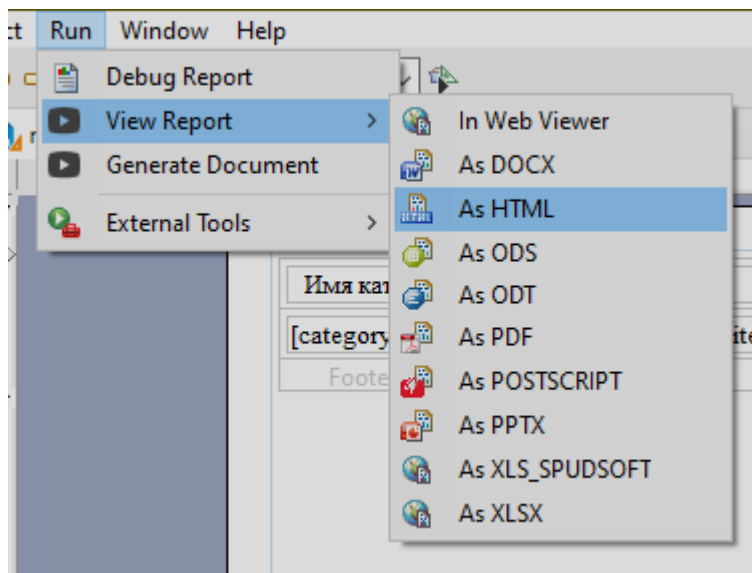


Рисунок 31 – Фрагмент диалога формирования отчёта

На рисунке 32 представлен пример фрагмента структуры отчёта (сверху) и результат заполнения структуры отчёта данными выборки (снизу).

Имя категории	Единица	Имя товара	Доступно	Цена	Продано	Всего
[category_name]	[unit_name]	[item_name]	[avail_quan]	[current_price]	[sold_quantity]	[total_sold]
Аудиотехника	штуки	Наушники			1	500
Аксессуары	штуки	Часы Ракета			25	11260,75
Аксессуары	штуки	Подставка			2	3000

18 мая 2022 г., 21:12

Рисунок 32 – Структура отчёта и результат её заполнения данными

На основе вышеприведённых экранных форм возможно сделать вывод, что пользовательское программное обеспечение информационной системы

позволяет формировать отчёт произвольной формы на основе информации из базы данных ИС. Таким образом, все компоненты системы полностью отвечают обозначенным требованиям к их функциональности.

3.7 Оценка и обоснование экономической эффективности разработки информационной системы

Оценим прямой экономической эффект от разработки и внедрения проекта в отделе розничных продаж ООО «Айковер ПРО». Определим месячные трудовые затраты на базовый вариант бизнес-процесса.

Воспользуемся требованием о производительности из пункта 1.2.3 и обозначим базовую потребность в 40 индивидуализированных отчётах за один день. Исходя из пяти рабочих дней и четырёхнедельного рабочего месяца, при базовой потребности в 40 отчётов, оценим трудовые затраты 1С-программиста на формирование этого объёма документов, предполагая, что на создание, сохранение и передачу одного отчёта уходит 5 минут рабочего времени. Таким образом, на обеспечение базового варианта бизнес-процесса в месяц приходится тратить 4000 минут рабочего времени. Предполагая, что рабочий час администратора эквивалентен рабочему часу программиста – рассчитаем трудозатраты на поддержку автоматизированного решения за тот же период. Предположим, что трудозатраты на администрирование базы данных и обеспечение работоспособности информационной системы составляют один час рабочего времени в день. Тогда, на поддержание проектного варианта за один месяц необходимо потратить 1200 минут времени администратора.

Рассчитаем показатели трудовых затрат. Формула расчёта абсолютного снижения трудовых затрат ΔT имеет следующий вид:

$$\Delta T = T_0 - T_1, \quad (1)$$

где T_0 – время, затрачиваемое на базовый вариант бизнес-процесса;

T_1 – время, затрачиваемое на проектный вариант бизнес-процесса.

Абсолютное снижение трудовых затрат равно 2800 минутам.

Формула коэффициента относительного снижения трудовых затрат K_T (в процентах), может быть представлена как:

$$K_T = \frac{\Delta T}{T_0} \times 100\%. \quad (2)$$

Коэффициента относительного снижения трудовых затрат для проектного решения достигает значения 70%.

Рассчитать индекс снижения трудовых затрат Y_T возможно посредством уравнения:

$$Y_T = \frac{T_0}{T_1}. \quad (3)$$

Индекс снижения трудовых затрат равен 3,34.

Рассмотрим стоимостные показатели затрат бизнес-процесса. Для определения стоимостных характеристик воспользуемся предположением что ценность единица времени администратора соответствует ценности равной единицы времени 1С-программиста и находится на уровне 200000 рублей за 160 часов работы [12]. Тогда стоимость одной минуты труда сотрудника составляет 20,48 рубля. Помимо компонентов информационной системы для реализации проекта также необходимо приобретение дополнительного оборудования. Таким образом, в стоимостные затраты на проектный вариант должна быть включена амортизация оборудования. Ежемесячные отчисления на амортизацию оборудования Σ_M могут быть рассчитаны по формуле:

$$\Sigma_M = \frac{C_{\Pi}}{n}, \quad (4)$$

где C_{Π} – первоначальная стоимость оборудования;

n – срок полезного действия, выражен в месяцах.

Таким образом, приняв СПИ в 36 месяцев, а первоначальную стоимость оборудования (сервера) в бухгалтерском учёте равной 55000 рублей. Получим ежемесячную норму амортизации в 1527,78 рублей. Величины затрат на обеспечение месяца работы по базовому варианту C_0 равна стоимости одной минуты труда, умноженной на время T_0 затрат на базовый вариант, что составляет 81920 рублей в месяц. Величина затрат на проектный вариант C_1 равна сумме ежемесячной амортизации оборудования и произведению времени T_1 на стоимость минуты труда. Таким образом стоимость поддержания работоспособности проектного варианта C_1 составляет 26103,78 рублей в месяц. Рассчитаем абсолютное снижение стоимостных затрат ΔC , выраженное уравнением:

$$\Delta C = C_0 - C_1, \quad (5)$$

где C_0 – величина затрат на базовый вариант;

C_1 – величина затрат на проектный вариант.

Абсолютное снижение стоимостных затрат составляет 55816,22 рублей в месяц.

Коэффициент относительного снижения стоимостных затрат K_C (в процентах) может быть представлен как:

$$K_C = \frac{\Delta C}{C_0} \times 100\%. \quad (6)$$

Тогда, для рассматриваемого проекта, коэффициент относительного снижения стоимостных затрат равен 0,68.

Индекса снижения стоимостных затрат Y_C выражен равенством:

$$Y_C = \frac{C_0}{C_1}. \quad (7)$$

Его значение для рассматриваемого случая составляет 3,14.

В таблице 17 представлены показатели экономической эффективности внедрения проекта информационной системы.

Таблица 17 – Показатели экономической эффективности внедрения проекта информационной системы

Трудоёмкость	Затраты		ΔT (1)	K_T (2)	Y_T (3)
	Базовый вариант	Проектный вариант			
	T_0 (час)	T_1 (час)			
	66,67	20	46,67	70%	3,34
Стоимость	C_0 (руб.)	C_1 (руб.)	ΔC (5)	K_C (6)	Y_C (7)
	81920	26103,78	55816,22	0,68	3,14

Как можно заметить значение индексов Y_T и Y_C незначительно отличаются, это связано с наличием в стоимостных показателях дополнительных трат ресурсов в виде амортизационных отчислений из-за использования дополнительного оборудования.

Капитальные затраты на создание проекта равны стоимости оборудования необходимого для развёртывания системы и расходами на её создание и внедрение. Проект может быть реализован одним программистом за два полных месяца работы. Таким образом, при месячной заработной плате в 200000 рублей, затраты на разработку проекта будут составлять 400000 рублей. В капитальных затратах необходимо учесть стоимость разового часового обучения пользователей (внедрения). При заработной плате менеджера по розничным продажам в 150000 рублей в месяц, час его обучения за счёт компании стоит 937,5 рублей. Также необходимо выделить по часу времени администратора на обучение каждой группы пользователей. Таким образом, предположив оптимальную численность группы в 20 человек, а общую численность пользователей системы в 40 человек, можно рассчитать стоимость внедрения проектного решения на предприятии. Она составит 39957,60 рублей. Таким образом общие капитальные затраты на проект составят 494957,60 рублей.

Рассмотрим срок окупаемости затрат на внедрение проекта T_{OK} равный:

$$T_{OK} = \frac{K_{\Pi}}{\Delta C} \quad (8)$$

где K_{Π} – капитальные затраты на создание проекта.

Таким образом, срок окупаемости затрат на внедрение проекта должен составить 8,87 месяцев, из этого можно сделать вывод, что решение о внедрении проекта является целесообразным и экономически обоснованным.

Выводы по главе 3

В третьей главе, на основании данных предыдущих глав, был произведён и аргументирован выбор архитектуры информационной системы и посредством UML-диаграммы компонентов продемонстрированы интерфейсы взаимодействия между её элементами. Был аргументирован выбор технологии разработки системы и рассмотрены популярные интегрированные среды разработки для языка Python 3. Приведен сравнительный анализ альтернативных решений СУБД и на его основе выбрано решение, наиболее удовлетворяющее системным требованиям. Разработана физическая модель базы данных системы. Рассмотрены особенности внесения данных в таблицы базы данных информационной системы. Были выбраны критерии для сравнения альтернативных компонентов системы и на их основе проведён анализ с целью поиска наиболее оптимальных решений. Представлен программный код ETL-механизма и описана его структура. В рамках описания функциональности информационной системы были показаны экранные формы и фрагменты, демонстрирующие визуальный интерфейс компонентов. Проведена оценка прямого экономического эффекта от разработки и внедрения проекта в отделе розничных продаж ООО «Айковер ПРО». Рассчитаны показатели трудовых и стоимостных затрат, оценены капитальные затраты на создание проекта и срок окупаемости затрат на внедрение проектного решения.

Заключение

Бакалаврская работа посвящена актуальной проблеме развития сегмента информационных систем на основе программного обеспечения с открытым исходным кодом. Это обусловлено необходимостью снижения издержек, обеспечения контроля, безопасности и защиты корпоративных данных.

Целью бакалаврской работы являлась разработка примера типовой архитектуры корпоративного хранилища данных на основе open-source решений. Для достижения цели был выполнен ряд задач:

- представлены технико-экономические характеристики предприятия, его организационная структура и рассмотрены бизнес-процессы;
- найден неэффективный бизнес-процесс и представлена его концептуальная модель «Как есть» в нотации IDEF0;
- проанализированы альтернативные проекты и составлено ТЗ на ИС;
- создана IDEF0-модель бизнес-процесса «Как должно быть»;
- проведено моделирование различных аспектов логической модели ИС и её базы данных, определены требования к аппаратно-программному обеспечению;
- проведено физическое моделирование, анализ компонентов и описание функциональных особенностей информационной системы;
- проведена оценка и обоснование экономической эффективности разработки проектного решения.

Таким образом, можно заключить, что все задачи, поставленные перед бакалаврской работой, были успешно выполнены, а цель работы – достигнута.

Потенциальными шагами дальнейшего развития системы являются: увеличение количества источников первичных коммерческих документов, контейнеризация вычислений и создание распределённого кластера обработки данных, расширение использования средств визуализации оперативных данных и совершенствование бизнес-процессов предприятия посредством увеличения степени автоматизации и информатизации труда.

Список используемой литературы и используемых источников

1. Бабич, А. В. Введение в UML : учебное пособие / А. В. Бабич. – 4-е изд. – Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2022. – 198 с. – ISBN 978-5-4497-1637-8.
2. Блинов, А.О. Реинжиниринг бизнес-процессов [Электронный ресурс]: учеб. пособие / А.О. Блинов [и др.]; под ред. А.О. Блинова. – М.: ЮНИТИ-ДАНА, 2015. – 343 с.
3. Вакансии компании Айковер ПРО. — Текст : электронный // HeadHunter : [сайт]. — URL: <https://hh.ru/employer/933066> (дата обращения: 29.04.2022).
4. ВЫБИРАЕМ СЕРВЕР ПОД БАЗЫ ДАННЫХ И ЭКОНОМИМ ДО 80% НА ПОКУПКЕ. — Текст : электронный // Server Gate : [сайт]. — URL: <https://servergate.ru/articles/vybiraem-server-pod-bazy-dannykh-i-ekonomim-do-80-na-rokupke/> (дата обращения: 12.05.2022).
5. ГОСТ 19.201-78 Единая система программной документации. Техническое задание. Требования к содержанию и оформлению. 1978. Режим доступа: <http://protect.gost.ru/document.aspx?control=7&id=155153>
6. Ипатова, Э. Р. Методологии и технологии системного проектирования информационных систем [Электронный ресурс]: учебник / Э. Р. Ипатова, Ю. В. Ипатов. - 3-е изд., стер. - Москва : ФЛИНТА, 2021. - 256 с.
7. Корнипаев Илья. Требования для программного обеспечения: рекомендации по сбору и документированию — М.: Издательство "Книга по требованию", 2013 — 118 с.
8. Менеджер по розничным продажам. — Текст : электронный // HeadHunter : [сайт]. — URL: <https://hh.ru/vacancy/54828123> (дата обращения: 30.04.2022).
9. О компании. — Текст : электронный // iCover.ru : [сайт]. — URL: <https://www.icover.ru/about/> (дата обращения: 29.04.2022).

10. Половина российских компаний работают без сайта. — Текст : электронный // Статистика 2ГИС : [сайт]. — URL: <https://stat.2gis.ru/research/50-rossiyskih-kompaniy-ne-imeyut-sayta/> (дата обращения: 29.04.2022).

11. Приказ ФНС России от 19.12.2018 № ММВ-7-15/820@. — Текст : электронный // Федеральная налоговая служба : [сайт]. — URL: https://www.nalog.gov.ru/rn77/about_fts/docs/8335278/ (дата обращения: 30.04.2022).

12. Программист 1С. — Текст : электронный // HeadHunter : [сайт]. — URL: <https://hh.ru/vacancy/54823735> (дата обращения: 30.04.2022).

13. Радыгин, В. Ю. Базы данных: основы, проектирование, разработка информационных систем, проекты : курс лекций. Учебное пособие / В. Ю. Радыгин, Д. Ю. Куприянов. — Москва : НИЯУ МИФИ, 2020. — 244 с. — ISBN 978-5-7262-2680-4.

14. Точилкина, Т. Е. Хранилища данных и средства бизнес-аналитики : учебное пособие / Т. Е. Точилкина, А. А. Громова. — Москва : Финансовый университет, 2017. — 161 с. — ISBN 978-5-7942-1387-4.

15. Универсальный передаточный документ (УПД). — Текст : электронный // 1С. Информационно-технологическое сопровождение : [сайт]. — URL: <https://its.1c.ru/db/updoc> (дата обращения: 11.05.2022).

16. Управление торговлей (1С:УТ) — что это такое и для чего она нужна, описание возможностей программы, обзор функций и конфигураций. — Текст : электронный // Клеверенс : [сайт]. — URL: <https://www.cleverence.ru/articles/elektronnaya-kommertsiya/upravlenie-torgovley-ut-cto-eto-takoe-i-dlya-chego-ona-nuzhna-opisanie-vozmozhnostey-programmy-obzo/> (дата обращения: 30.04.2022).

17. Что такое столбцовая (колоночная) база данных?. — Текст : электронный // ClickHouse : [сайт]. — URL: <https://clickhouse.com/docs/ru/faq/general/columnar-database/> (дата обращения: 16.05.2022).

18. AMD Sets All-Time CPU Market Share Record as Intel Gains in Desktop and Notebook PCs. — Текст : электронный // Tom's Hardware : [сайт]. — URL: <https://www.tomshardware.com/news/intel-amd-4q-2021-2022-market-share-desktop-notebook-server-x86> (дата обращения: 12.05.2022).

19. Data pipelines, Luigi, Airflow: everything you need to know. — Текст : электронный // Towards Data Science : [сайт]. — URL: <https://towardsdatascience.com/data-pipelines-luigi-airflow-everything-you-need-to-know-18dc741449b7> (дата обращения: 17.05.2022).

20. DB-Engines Ranking. — Текст : электронный // DB-Engines : [сайт]. — URL: <https://db-engines.com/en/ranking> (дата обращения: 16.05.2022).

21. How to Install Debian 11 (Bullseye) Server Using Net Install. — Текст : электронный // Tecmint : [сайт]. — URL: <https://www.tecmint.com/install-debian-11-minimal-server/> (дата обращения: 17.05.2022).

22. Microsoft named a Leader in the 2022 Gartner® Magic Quadrant™ for Analytics and BI Platforms. — Текст : электронный // Microsoft : [сайт]. — URL: <https://powerbi.microsoft.com/en-us/blog/microsoft-named-a-leader-in-the-2022-gartner-magic-quadrant-for-analytics-and-bi-platforms/> (дата обращения: 01.05.2022).

23. Open-Source Reporting Review - BIRT, Jaspersoft, Pentaho. — Текст : электронный // Innovent Solutions : [сайт]. — URL: <https://www.innoventsolutions.com/open-source-reporting-review-birt-jasper-pentaho.html> (дата обращения: 17.05.2022).

24. PEP 249 – Python Database API Specification v2.0. — Текст : электронный // Python Enhancement Proposals : [сайт]. — URL: <https://peps.python.org/pep-0249/> (дата обращения: 13.05.2022).

25. TIobe Index for April 2022. — Текст : электронный // TIobe : [сайт]. — URL: <https://www.tiobe.com/tiobe-index/> (дата обращения: 01.05.2022).

26. Usage statistics of Linux for websites. — Текст : электронный // W3Techs : [сайт]. — URL: <https://w3techs.com/technologies/details/os-linux> (дата обращения: 01.05.2022).