

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 «Разработка информационной системы учета и обработки заявок на
ремонт компьютерной техники»

Обучающийся

А.В. Евсеев

(Инициалы Фамилия)

(личная подпись)

Руководитель

В.Ф. Глазова

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

Тольятти 2022

Аннотация

Тема выпускной квалификационной работы – «Разработка информационной системы учета и обработки заявок на ремонт компьютерной техники».

Актуальность темы обусловлена все возрастающим значением внедрения информационных технологий в управление бизнес-процессами предприятий и организаций разных сфер деятельности.

Объектом исследования является деятельность IT-отдела торговой сети.

Предмет исследования – автоматизация учета и обработки заявок на ремонт компьютерной техники.

Целью работы является разработка информационной системы, позволяющей автоматизировать процесс учета и обработки заявок при выполнении ремонтных работ IT-отделом торговой сети.

Выпускная квалификационная работа представлена введением, тремя главами, заключением, списком используемой литературы и приложением.

В первой главе описана организационная структура торговой сети, выполнен структурный анализ бизнес-процессов с использованием методологий DFD и IDEF0, описана сущность задачи автоматизации и рассмотрены основные аналогичные системы.

Во второй главе выполнено логическое и физическое моделирование информационной системы, поведено обоснование выбора программного обеспечения для реализации ИС, построена физическая модель базы данных и функциональная схема проекта.

В третьей главе описана реализация разработанной системы и расчет экономической эффективности от внедрения разработки.

Работа состоит из 57 страниц, включая приложение. Содержит 32 рисунка, 7 таблиц, список из 22 источников.

Оглавление

Введение	4
Глава 1 Функциональное моделирование предметной области	6
1.1 Техничко-экономическая характеристика ТС «Яблоко»	6
1.2 Анализ существующих разработок и требования к разрабатываемой системе	18
Глава 2 Разработка и реализация проектных решений	21
2.1 Логическое моделирование информационной системы.....	21
2.2 Физическое моделирование информационной системы	29
Глава 3 Реализация разработанной системы и расчет экономической эффективности	34
3.1 Описание разработанной информационной системы.....	34
3.2 Оценка и обоснование экономической эффективности проекта .	42
Заключение	51
Список используемой литературы и используемых источников	53
Приложение А Программный код.....	55

Введение

На сегодняшний день сложно представить компанию или предприятие, которые бы не использовали в процессе работы какие-либо программные продукты для мониторинга процессов работы, составления отчетности и экономии времени при выполнении тех или иных задач.

Темой выпускной квалификационной работы: «Разработка информационной системы учета и обработки заявок на ремонт компьютерной техники».

Объект исследования – бизнес-процесс учета и обработки заявок на ремонт компьютерной техники IT-отделом торговой сети (ТС) «Яблоко».

Предметом исследования является автоматизация процесса учета и обработки заявок, поступающих из магазинов в IT-отдел на ремонт и замену компьютерной техники.

Разработка подобной системы является актуальной, так как внедрение такого проекта значительно упростит работу сотрудников, а также позволит руководству компании отслеживать ход и качество выполнения работ по ремонту компьютерной техники.

Использование информационной системы позволит упростить и упорядочить учёт техники, используемой внутри компании, что снизит количество ошибок, допускаемых при оформлении техники. Также экономический эффект от внедрения подобной системы позволит компании сократить расходы.

Для успешного выполнения поставленной цели необходимо будет решить следующие задачи:

- изучить работу IT-отдела ТС (построить организационную структуру компании, изучить бизнес-процессы и направления деятельности);
- разработать и построить концептуальную модель предметной области;
- провести анализ уже существующих разработок подобных систем;

- разработать модель базы данных и реализовать ее;
- разработать интерфейс информационной системы;
- разработать и реализовать информационную систему;
- провести анализ экономического эффекта по итогам внедрения информационной системы.

В первой главе рассматривается торговая сеть «Яблоко» и её IT-отдел, дается технико-экономическая характеристика торговой сети, создается концептуальная модель предметной области, определяются аспекты работы IT-отдела, которые будут автоматизированы, а также проводится анализ уже существующих на рынке разработок по данной теме.

Во второй главе разрабатывается логическая и физическая модели базы данных информационной системы, изучается технологическое обеспечение задачи, а также реализовывается контрольный пример проекта.

В третьей главе рассматривается методика расчета экономической эффективности и проводится расчёт показателей экономической эффективности разработанного проекта.

Глава 1 Функциональное моделирование предметной области

1.1 Техничко-экономическая характеристика ТС «Яблоко»

1.1.1 Характеристика ТС «Яблоко»

Полное название организации – Общество с ограниченной ответственностью (ООО) «Десерт», которое реализует товар через торговую сеть «Яблоко». Данная торговая сеть имеет 10 магазинов, расположенных в разных районах города Екатеринбург.

Основной деятельностью ООО «Десерт» является рознично-оптовая торговля продуктами питания.

Торговая сеть предоставляет комплекс торговых услуг с целью создания удобства для покупателей и повышения культуры обслуживания населения. Торговая сеть работает для повышения благосостояния своих клиентов, предлагая им качественные товары повседневного спроса по доступным ценам. Компания ориентирована на покупателей с различным уровнем доходов. Эффективный процесс доставки товаров в магазины возможен благодаря мощной логистической системе. Своевременную доставку продуктов во все магазины розничной сети позволяет осуществить собственный автопарк.

Магазины торговой сети могут предложить покупателям как широкий выбор основных ходовых товаров, так и продукцию собственного производства под частной торговой маркой «Яблоко», насчитывающей около 100 наименований. Также присутствуют отделы хозяйственных товаров и алкогольной продукции.

1.1.2 Организационная структура ТС «Яблоко»

Рассмотрим структуру организации, представленную на рисунке 1.

Торговая сеть имеет головной офис, разделенный на функциональные отделы, управляющие всей сетью: административный персонал; внутренний колл-центр; отдел маркетинга; отдел рекламы и PR; отдел кадровой службы;

транспортный отдел; отдел финансов и бухгалтерии; отдел розничных и корпоративных продаж; отдел информационных технологий.

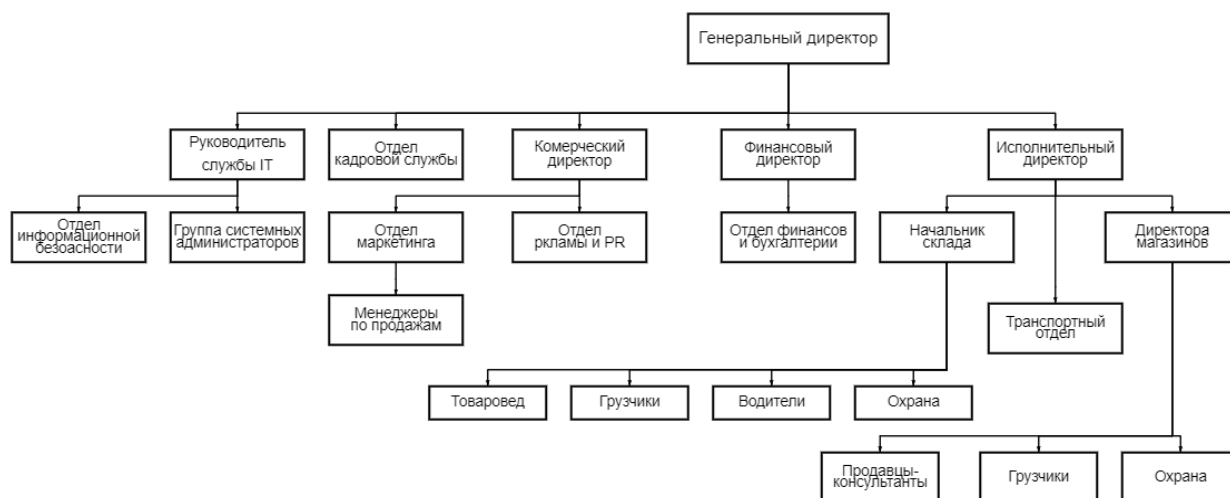


Рисунок 1 – Организационная структура ТС «Яблоко»

Отдел информационных технологий – это отдел, который обеспечивает всю торговую сеть необходимыми информационными технологиями, занимается поддержанием работоспособности этих систем. Данный отдел отслеживает работу сетей, эксплуатацию программного обеспечения (ПО) и безопасность. Также он отвечает за обеспечение оргтехникой, включая ремонт, другие отделы и установку ПО.

Отдел кадров – в основном занимается подбором персонала, планированием компенсаций, безопасностью на рабочих местах и следит за соблюдением трудового законодательства и подачей сведений в бухгалтерию.

Отдел маркетинга – отвечает за рекламу, продвижение, проведение опросов потребителей и разработку маркетинговой стратегии.

Отдел рекламы – занимается предоставлением средств, с помощью которых работа по созданию и выпуску новой рекламы может быть выполнена наиболее эффективным, экономичным и систематическим образом.

Отдел финансов – это отдел, который отвечает за финансовую часть компании.

Транспортный отдел – занимается организацией бесперебойной и безопасной транспортировкой товаров до магазинов торговой сети, созданием истории перевозок и проведением технического обслуживания и ремонта техники.

На основании полученной структуры можно сделать вывод, что в торговой сети в данный момент используется функциональная структура.

1.1.3 Краткая характеристика IT-отдела торговой сети

В соответствии с темой ВКР дальнейшей автоматизации будет подлежать деятельность IT-отдела. В основном отдел занимается обслуживанием кассового оборудования, оргтехники и настройкой необходимого ПО. Поскольку все магазины торговой сети в основном связываются с IT-отделом по телефону и путем отправки обычных сообщений, необходимо автоматизировать процесс обслуживания. На текущий момент, практически отсутствует статистика и отчетность по выполненным заявкам из-за отсутствия единой системы.

Рассмотрим более детально процесс обработки заявок на ремонт компьютерной техники в ТС «Яблоко». В данный момент в обслуживании и поддержке пользователей участвует два отдела торговой сети «Яблоко» – внутренний колл-центр и IT-отдел.

Колл-центр выполняют работу приемной инстанции, в которой происходит определение типа заявки или запроса и передача его специалисту IT-отдела. Данная процедура является очень затратной в плане использования ресурсов компании. Необходимо содержать целый штат сотрудников колл-центра и затрачивать много ресурсов на поддержание работоспособности этой системы [8].

1.1.4 Сущность задачи создания системы учета заявок пользователей на ремонт компьютерной техники

Актуальность задачи создания данной системы обусловлена тем, что в настоящее время на предприятии представлено огромное количество оргтехники, компьютерной техники и кассового оборудования. Вести правильный учет всех поступающих заявок с десяти магазинов без использования современных компьютерных технологий просто невозможно.

Главным плюсом внедрения проекта в работу является оптимизация рабочего времени сотрудников, что позволит сконцентрироваться на более важных и приоритетных задачах. У сотрудников появится больше времени на работу с ПО магазинов, обучение и планированию новых решений для торговой сети [9].

Такие системы сильно упрощают управление новыми задачами торговой сети, позволит легко организовать и вести отслеживание прогресса различных работ.

Системы такого плана широко используются во многих современных компаниях и офисах, позволяя легко распределять задачи между отделами и сотрудниками. При отсутствии жёсткого контроля появляется возможность точной настройки систем и позволяет добиться эффективной реализации необходимых бизнес-процессов [1].

Подводя итог, можно выделить основные результаты автоматизации:

- повышение эффективности IT-отдела;
- реализация необходимых бизнес-процессов;
- появление удобного инструмента для отслеживания заявок и техники в торговой сети;
- упрощение рабочего процесса при составлении отчетных документов;
- повысится качество обслуживания внутри торговой сети;
- увеличение «прозрачности» учета компьютерной техники;
- значительное ускорение времени проведения ремонтов;

– оперативное обеспечение руководства достоверной информацией.

Благодаря новой информационной системе IT-отдел перейдет на новый уровень, что позволит получить неоспоримый экономический эффект.

1.1.5 Концептуальное моделирование исследуемого бизнес-процесса

Произведем анализ работы отдела информационных технологий. Для этого создаем контекстные диаграммы для основных процессов работы IT-отдела (учет и обработка заявок). Для построения диаграмм используем методологию IDEF0. Построение данных диаграмм позволяет выявить недостатки в работе IT-отдела и учесть их при последующей автоматизации. Создадим контекстную диаграмму «Как есть» (рисунок 2).



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

В процессе анализа деятельности IT-отдела в основном рассматривается только основной процесс обслуживания ПК и оргтехники, поступающей из принадлежащих торговой сети магазинов и офисов.

При детальном рассмотрении разработанной диаграммы выделяем основные входные данные:

- заявка, поступающая от директора магазина путем связи с колл-центром торговой сети;
- данные о технике (название, инвентарный номер, описание проблемы);
- номер магазина или офиса откуда поступил запрос;
- создание накладной на перемещение в it-отел на ремонт (при необходимости).

Исполнителями данной работы являются сотрудники IT-отдела. Выходными данными является уведомление о выполненной работе путем обратной связи с колл-центром или напрямую с директором магазина.

Для проведения анализа существующего процесса произведем последовательную декомпозицию процесса до требуемого уровня детализации, представленную на рисунке 3.

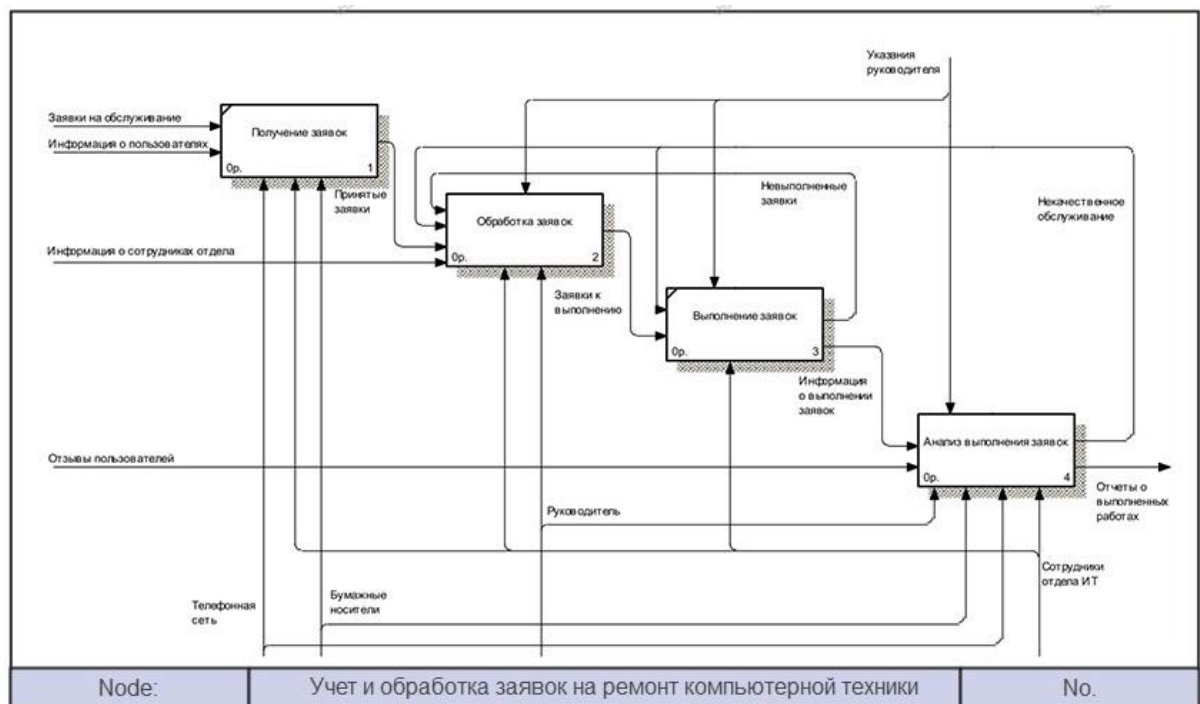


Рисунок 3 – Диаграмма IDEF0 декомпозиции основного процесса «Как есть»

Обслуживание ПК и оргтехники можно разделить на 4 основных процесса:

- получение заявок;
- обработка заявок;
- выполнение заявок;
- анализ выполненных заявок.

С помощью методологии DFD выполняем функциональную декомпозицию активности «Прием заявки» (рисунок 4).

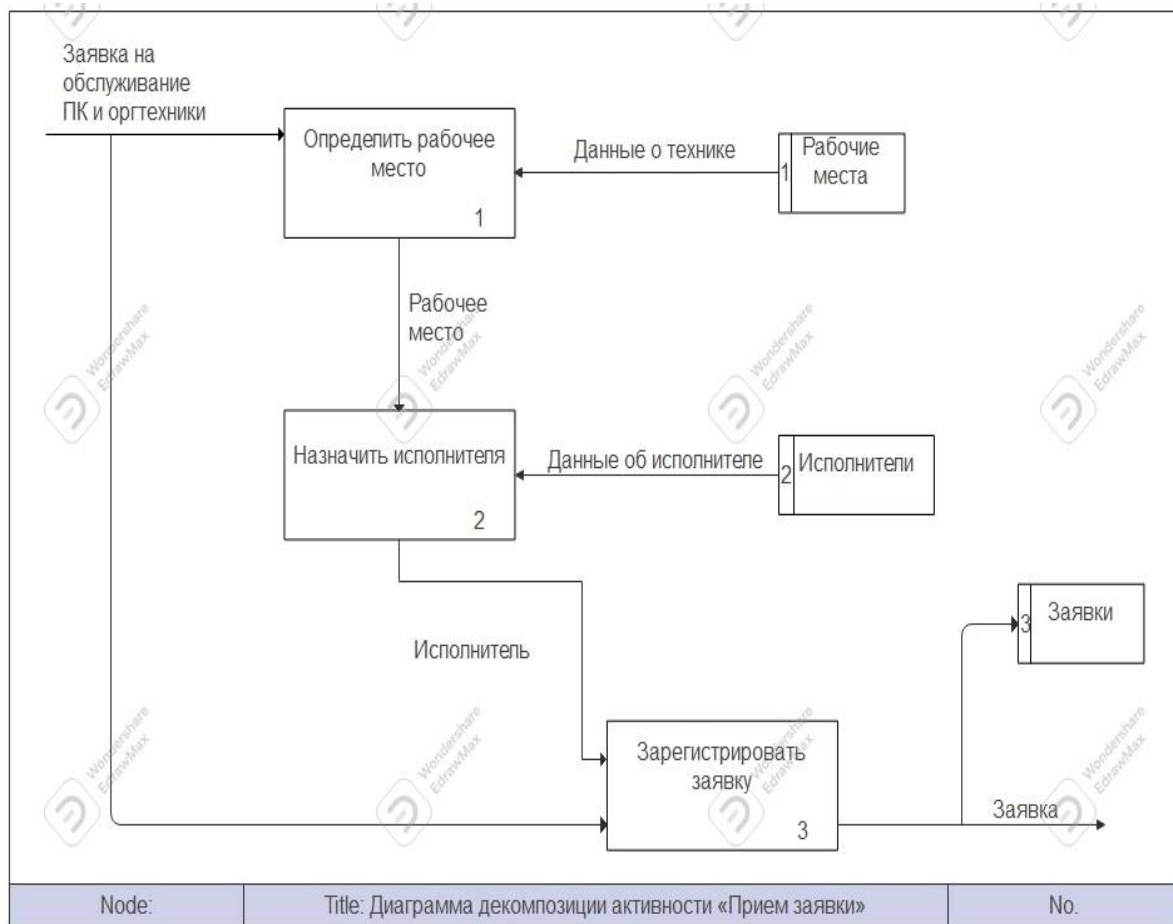


Рисунок 4 – Диаграмма декомпозиции активности «Прием заявки»

На основе методологии DFD выполняем функциональную декомпозицию активности «Выполнение заявки», которая представлена на рисунке 5.

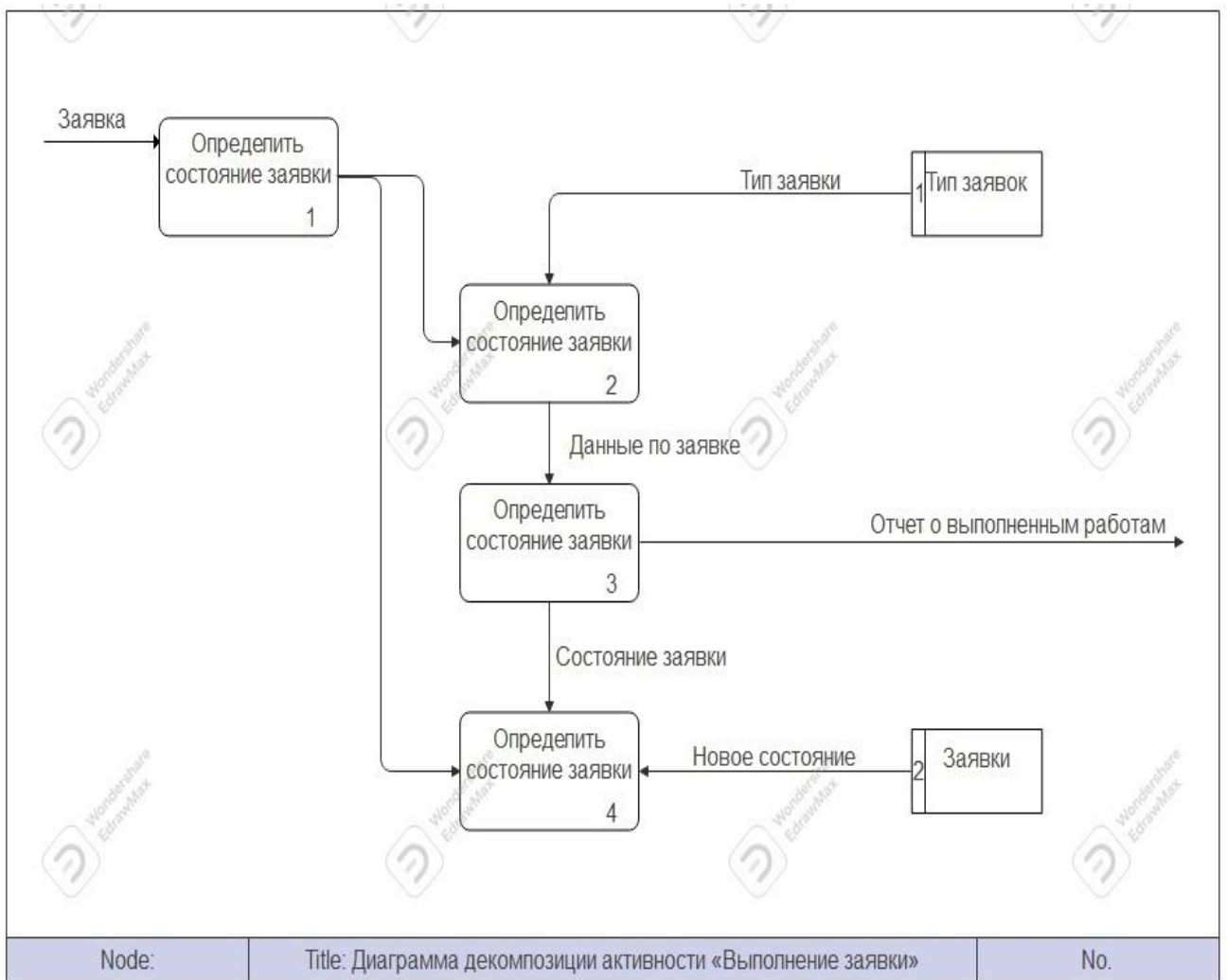


Рисунок 5 – Диаграмма декомпозиции активности «Выполнение заявки»

Также нам понадобится диаграмма последовательности управления заявками пользователей «Как есть», представленная на рисунке 6.

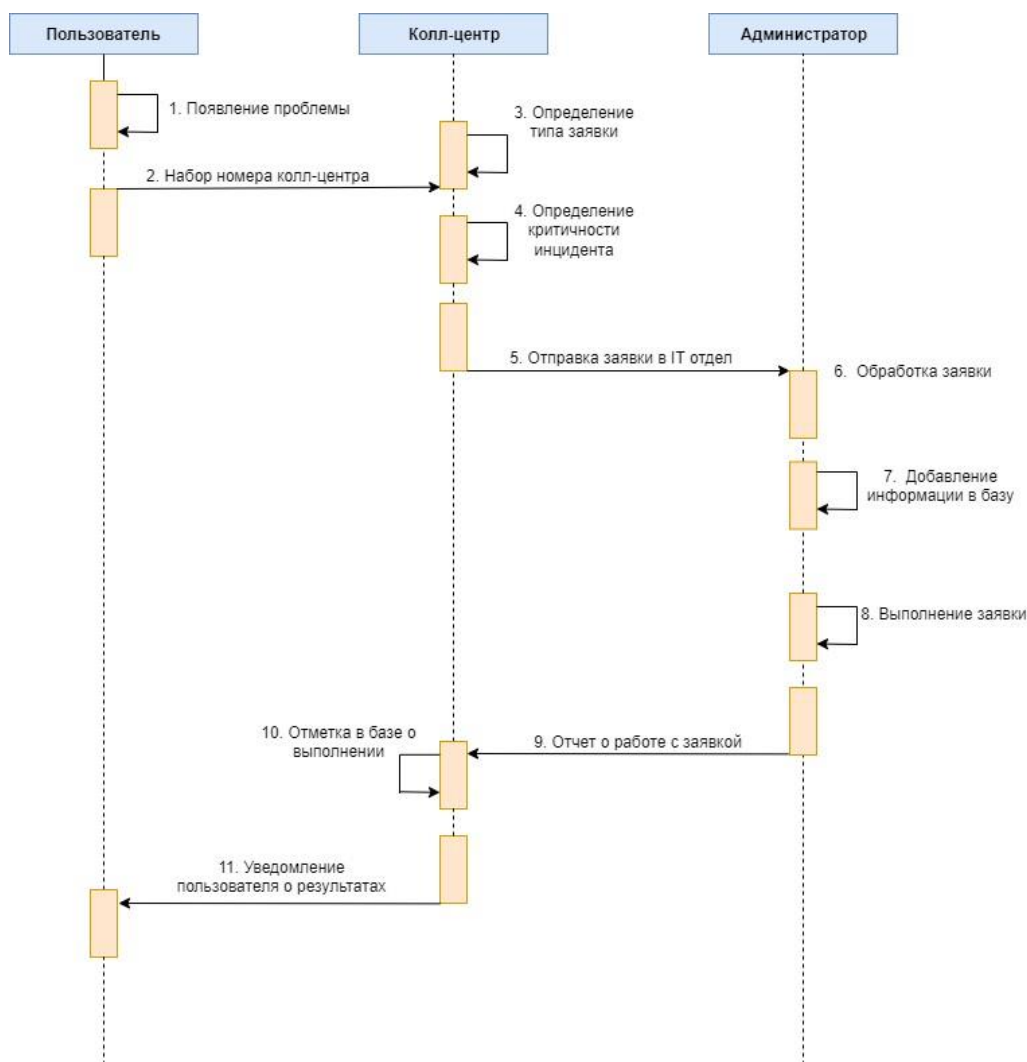


Рисунок 6 – Диаграмма последовательности управления заявками пользователей «Как есть»

На данных диаграммах присутствуют такие основные данные как заявки на обслуживание, получение заявок путем связи с колл-центром, информация о пользователях и их отзывы. Во всем процессе участвуют сотрудники колл-центра, отдел информационных технологий и непосредственный руководитель [5].

В процессе построения диаграмм были обнаружены недостатки, которые мы сможем исключить путем внедрения информационной системы:

- участие в процессе большого количества людей, использование для связи целого отдела «Колл-центр»;

- нет легкого доступа к архивам и статистике выполненных заявок, каждый раз приходится долго выяснять подробности по ранее проделанным работам;
- нет возможности оценить продуктивность работы IT-отдела, что не позволяет оптимизировать его работу;
- перегрузы телефонной линии в техническом отделе;
- низкая скорость получения и решения выполняемых работ;
- нет возможности отказаться от работы с бумажными носителями информации.

Теперь построим новые диаграммы «Как должно быть» с учетом внедрения ИС. На рисунке 7 показана контекстная диаграмма «Как должно быть» для бизнес-процесса «Учет и обработка заявок на ремонт компьютерной техники».



Рисунок 7 – Контекстная диаграмма основного процесса «Как должно быть»

На представленной диаграмме можно увидеть такие входные данные как заявки на обслуживание, поступающие от пользователей, информация о пользователях и их отзывы. Обслуживанием занимаются сотрудники и

руководитель IT-отдела, также при помощи информационной системы в этом участвуют и сами пользователи (рисунок 8).

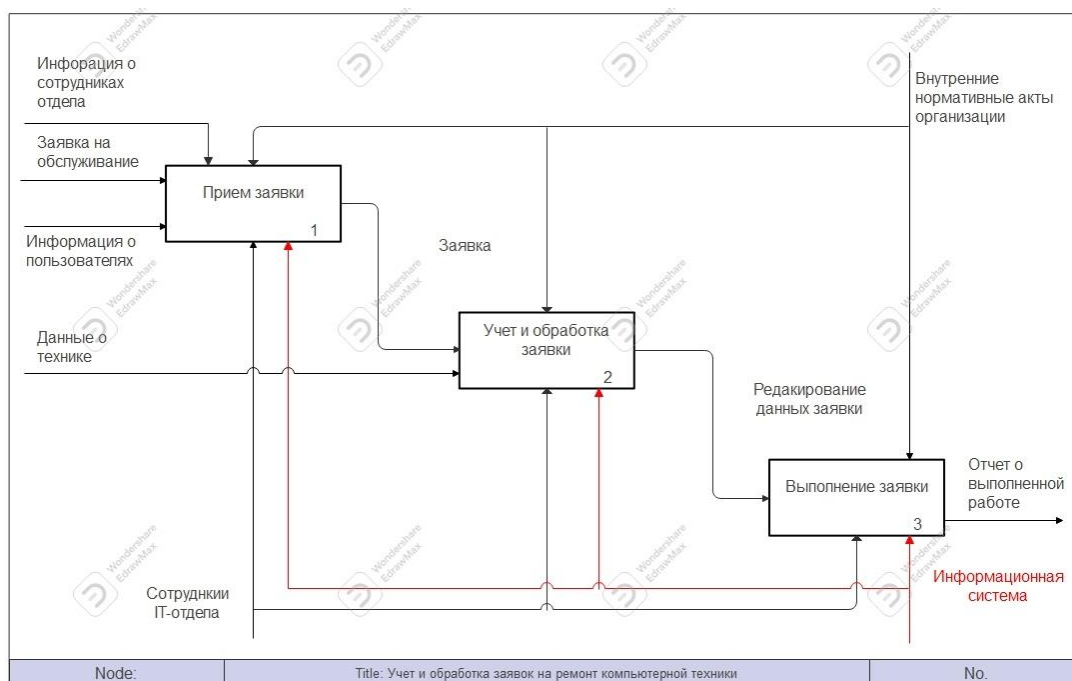


Рисунок 8 – Диаграмма декомпозиции «Как должно быть»

На диаграмме «Как должно быть» отсутствует блок «Обработка заявок». Все функции, принадлежащие к данному блоку, теперь выполняет информационная система, тем самым предоставляя сотрудникам IT-отдела больше свободного времени. Использование системы позволит отказаться от ненадежных бумажных носителей, сохраняя информацию в электронной базе данных.

Далее на рисунках 9 и 10 рассматриваются процессы приема заявки и её выполнения, но уже с использованием информационной системы. В этих процессах также отсутствует коммуникация с колл-центром, все необходимые действия по работе с заявками происходят непосредственно через информационную систему.

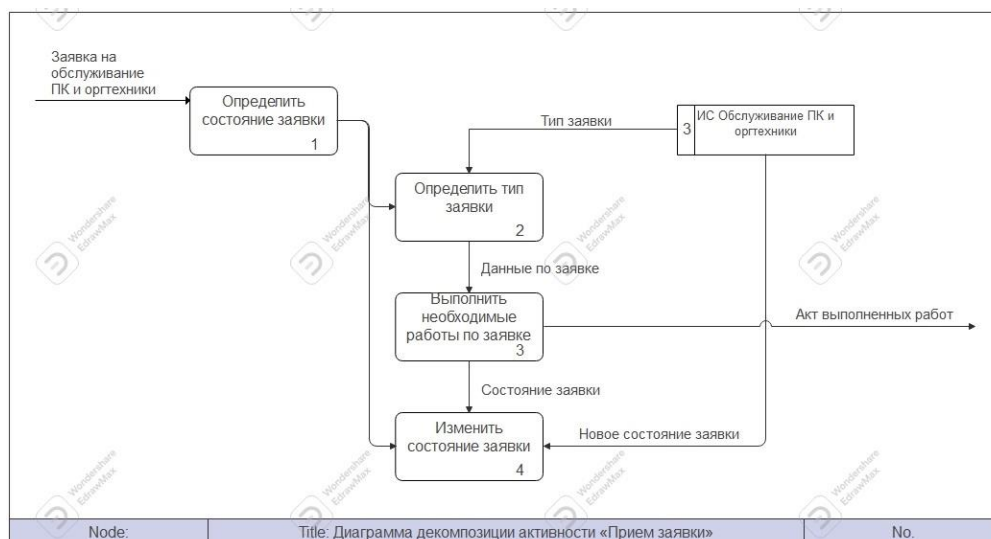


Рисунок 9 – Диаграмма декомпозиции активности «Прием заявки» «Как должно быть»

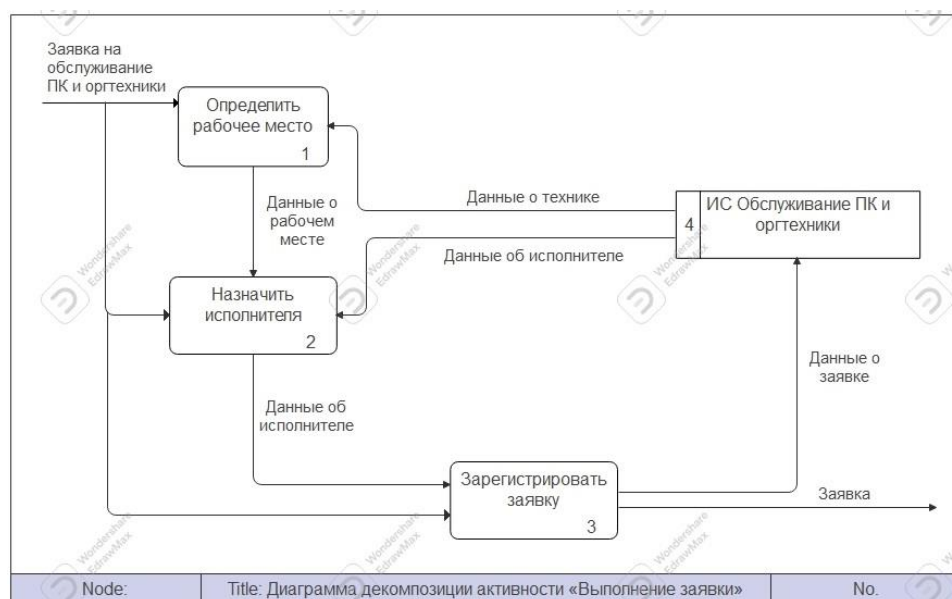


Рисунок 10 – Диаграмма декомпозиции активности «Выполнение заявки» «Как должно быть»

Можно отметить преимущества использования системы. Сильно упростился и процесс формирования отчетов, и их анализ занимает меньше времени у руководства. Все отчеты теперь составляются также в электронном виде [3].

1.2 Анализ существующих разработок и требования к разрабатываемой системе

1.2.1 Функциональные требования к системе

Предназначением данной информационной системы является сокращение затрат времени в процессе выполнения должностных обязанностей ИТ-специалистами при формировании и сопровождении заявок на обслуживание. Основной целью ставится повышение оперативности, уровня организованности и производительности труда, упрощения процессов подачи пользователями заявок на обслуживание, а также обработка данных и отчетов о заявках и их централизованное хранение.

Выделим функционал системы, которому она должна отвечать, для успешной реализации поставленных целей:

- автоматизированный процесс подачи заявок;
- легкость и удобство редактирования информации;
- централизованное хранение данных в электронной базе системы;
- контроль возможных ошибок и сбоев;
- взаимодействие интерфейсов с базой данных.

Рассмотрим существующие системы, в той или иной степени отвечающие сформулированным требованиям [6].

1.2.2 Существующие аналоги системы

В настоящее время прикладное программирование может обеспечить ИТ-компаниям автоматизацию большего числа внутренних рабочих процессов и значительно улучшить таким образом взаимодействие с клиентской базой.

CRM-системы могут обеспечивать следующий функционал и возможности:

- сбор всех данных о работе ИТ-компании в реальном времени;
- визуализация в формате графиков и диаграмм;
- сокращение издержек на фоне улучшения обслуживания клиентов;

- с помощью CRM создается единое онлайн-пространство для IT-персонала;
- контроль руководством поставленных задач и составление электронной отчетности.

Такие возможности CRM-систем в сегменте информационных технологий заметно повышают эффективность IT-отделов [18].

Изучив существующие системы можно получить общее представление о программных продуктах в сфере IT. В таблице 1 приведено сравнение актуальных на данный момент систем автоматизации учета и обработки заявок пользователей.

Таблица 1 – Сравнение аналогов

	Поддерживаемое ПО	Возможность интеграции	Функционал CRM	Стоимость продукта
АМОСРМ	MS SQL, Tomcat, DB2, Oracle WebLogic, HTTP Server, IBM	Возможность работы с 15 внешними приложениями, системами мониторинга, службами каталогов. Интеграция с телефонией и электронной.	Порядка 14 процессов ИТІЛ, интеграция с электронной почтой и телефонией.	Высокая стоимость и модель оплаты по годовой подписке.
БИТРИКС24	MS SQL, PostgreSQL, Oracle WebLogic, IBM WebSphere	Web сервисы, интерфейсы для работы с таблицами БД, XML.	Поддерживает 7 ИТІЛ процессов, рассылка SMS сообщений.	Средняя цена, возможность приобретения постоянной лицензии
5СRМ	MS SQL, PostgreSQL, HTTP Server, DB2	Использует свой собственный интеграционный модуль, и web-сервисы.	Поддерживает 9 ИТІЛ процессов, надежное облачное хранилище.	Средняя цена и модель оплаты по годовой подписке.

Изучив несколько актуальных на рынке CRM-систем, можно выделить их основные преимущества, возможности и функционал, что позволит

использовать эти данные при разработке ИС. Проанализируем три CRM-системы:

– АМОСМ – это система управления, оснащенная широким набором опций для работы с большой базой клиентов ИТ-компании. В системе присутствует интеграция с электронной почтой и телефонией. Также присутствует оценочная аналитика продаж;

– БИТРИКС24 – это CRM-система, имеющая большой перечень инструментов для работы с базой клиентов и маркетинга. Позволяет обеспечить сохранность всех коммуникаций. Использует сервисы для рассылки SMS и почтовых сообщений;

– 5СМ – это система, позволяющая легко и эффективно управлять рабочими процессами компании, задачами и сделками. Имеет надежное облачное хранилище и расширенную аналитику.

В итоге, можно сделать вывод, что подобные системы и их внедрение поднимают деятельность ИТ-отдела на новый уровень. Новый функционал и возможности систем оказывают положительный экономический эффект. Неоспоримым преимуществом является способность таких систем ликвидировать основные слабые места в процессе учета заявок и обработки отчетов компании.

Выводы по главе 1

В первой главе дана характеристика торговой сети «Яблоко», построена её организационная структура. С помощью диаграмм «Как есть» выявлены недостатки организации бизнес-процесса учета и обработки заявок на ремонт компьютерной техники, на диаграммах «Как должно быть» показаны преимущества внедрения информационной системы. Выполнен анализ существующих CRM-систем и выделены их основные преимущества и недостатки, которые планируется учитывать при разработке новой системы.

Глава 2 Разработка и реализация проектных решений

2.1 Логическое моделирование информационной системы

2.1.1 Логическая модель и ее описание

Была построена диаграмма вариантов использования для ИТ-отдела торговой сети (рисунок 11). Диаграмма создана для модели «Как должно быть». На диаграмме задействованы следующие актёры:

- директор магазина (пользователь);
- системный администратор ИТ-отдела.

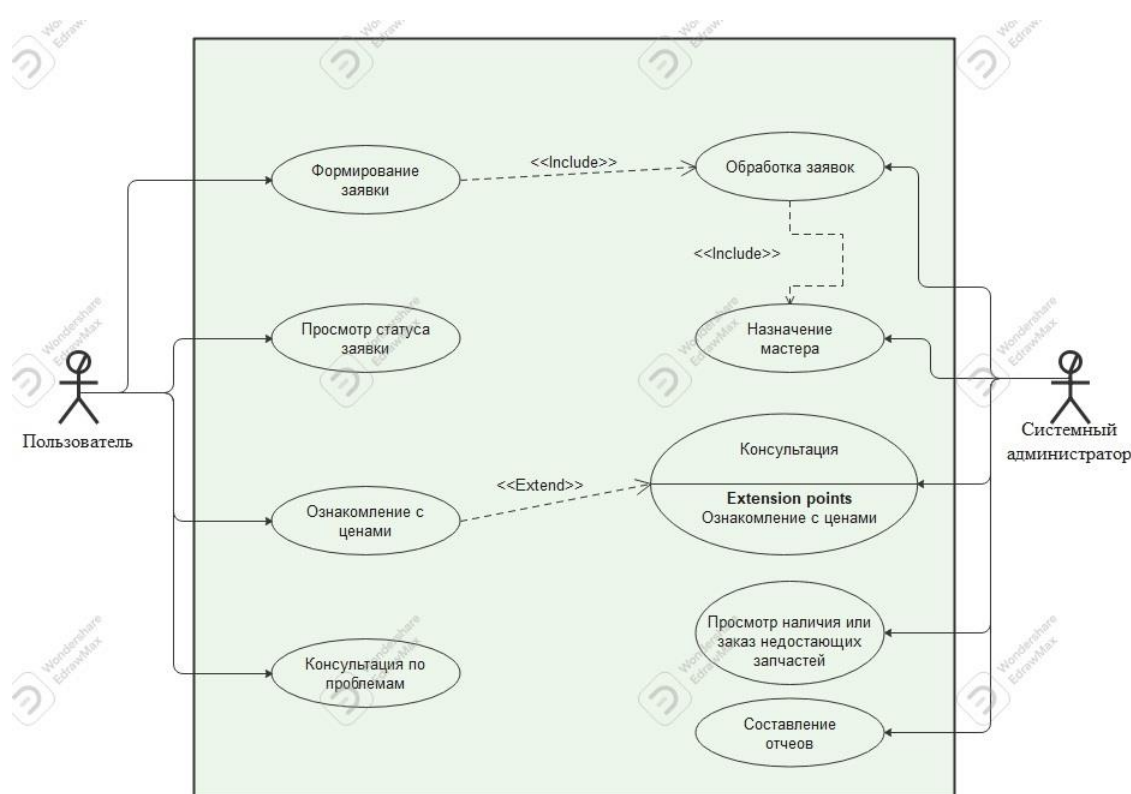


Рисунок 11 – Диаграмма вариантов использования

Для пользователя (директора магазина) возможны следующие варианты событий:

- создание новой заявки на ремонт (определение срока выполнения работ, места работы, объема работ, приоритетности заявки);

- мониторинг плана и состояния заявки;
- ознакомление с результатами ремонта.

Для системного администратора IT-отдела возможны следующие варианты событий:

- оформление новой заявки;
- оформление акта по итогам ремонта;
- закрытие выполненной заявки.

Далее рассмотрим диаграмму последовательностей и диаграмму классов основного бизнес-процесса «Как должно быть» (рисунки 12 и 13).

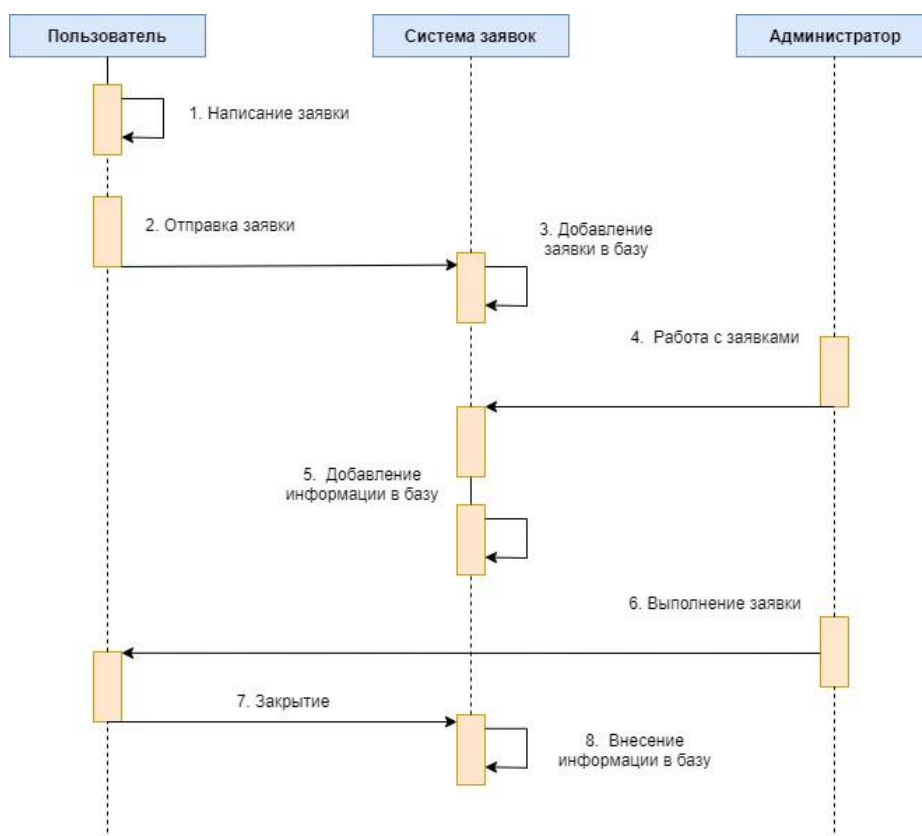


Рисунок 12 – Диаграмма последовательностей «Как должно быть»

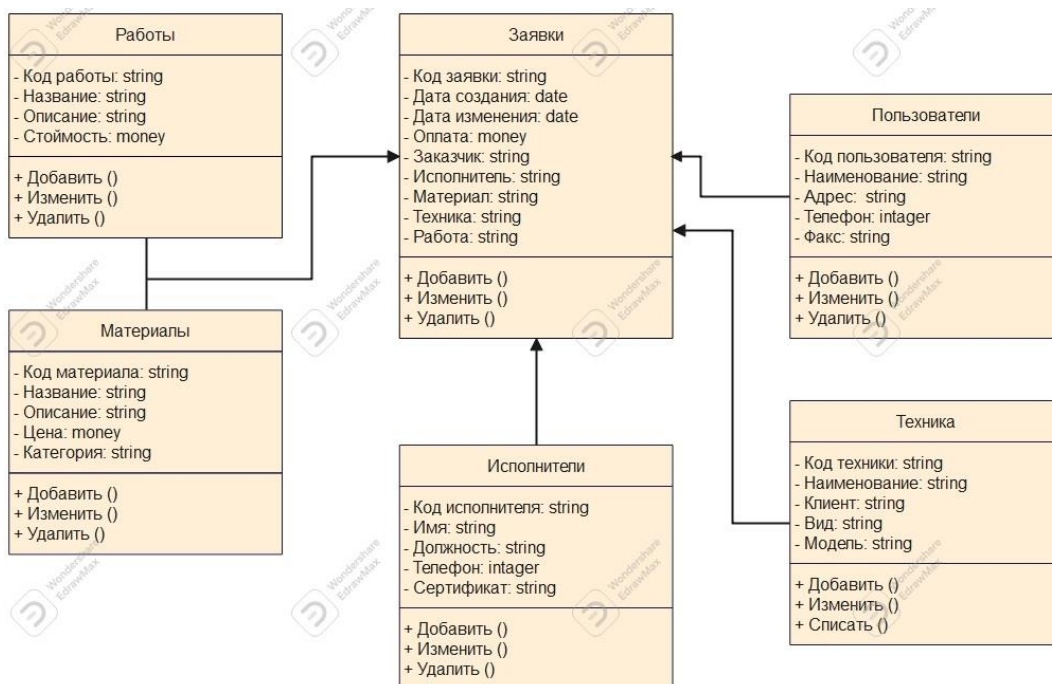


Рисунок 13 – Диаграмма классов «Как должно быть»

Теперь нам необходимо перейти к концептуальному проектированию, предварительно создать сущности с обязательным указанием атрибутов и типов данных, которые приведем в таблицах 2 – 7 [5].

Таблица 2 – Сущность «Работы»

Ключ	Атрибут	Тип данных
PK	КодРаботы	String
	Название	String
	Описание	String
	Стоимость	Money

Таблица 3 – Сущность «Материалы»

Ключ	Атрибут	Тип данных
PK	КодМатериала	String
	Название	String
	Описание	String
	Цена	Money
	Категория	String

Таблица 4 – Сущность «Заявки»

Ключ	Атрибут	Тип данных
PK	КодЗаявки	String
	ДатаСоздания	Date
	ДатаИзменения	Date
	Оплата	Money
	Заказчик	String
PK	Исполнитель	String
	Материал	String
	Техника	String
	Работа	String

Таблица 5 – Сущность «Исполнители»

Ключ	Атрибут	Тип данных
PK	КодИсполнителя	String
	Имя	String
	Должность	String
	Телефон	Intager
	Сертификат	String

Таблица 6 – Сущность «Пользователи»

Ключ	Атрибут	Тип данных
PK	КодПользователя	String
	Наименование	String
	Адрес	String
	Телефон	Intager
	Факс	String

Таблица 7 – Сущность «Техника»

Ключ	Атрибут	Тип данных
PK	КодТехники	String
	Наименование	String
	Клиент	String
	Вид	String
	Модель	String

Далее рассмотрим диаграмму логической модели данных (рисунок 14). Она содержит более подробную информацию о концепциях в

рассматриваемой области. В ней содержатся атрибуты данных и показываются отношения между сущностями [19].

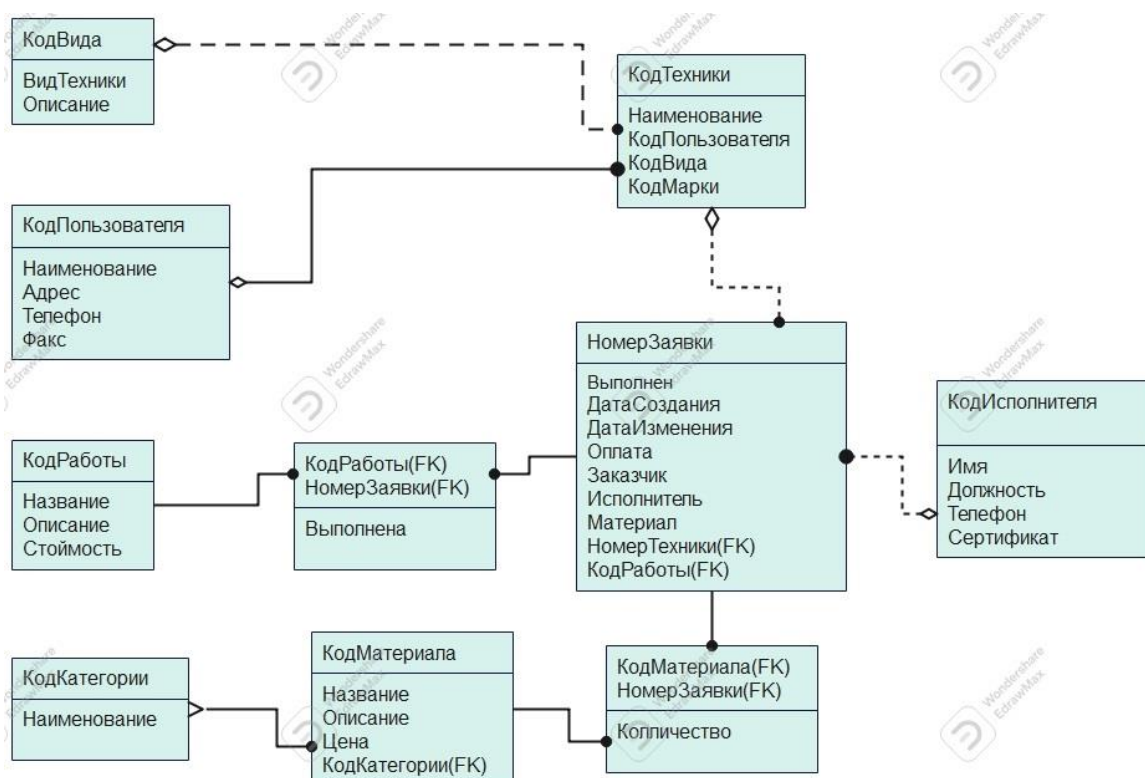


Рисунок 14 – Диаграмма логической модели данных

Заявка – это список проблем, составленный директором магазина или оператором, которые необходимо решить.

Исполнитель – это системные администраторы IT-отдела, занимающиеся обслуживанием и ремонтом техники.

Техника – это любая техника, используемая в магазинах торговой сети (ПК, кассовое оборудование, терминалы безналичной оплаты, программируемые весы и т.д.).

Ремонт – это выполнение необходимого комплекса работ для восстановления работоспособности техники или ПО.

Расход – это использование необходимых расходных материалов для замены вышедших из строя компонентов оборудования.

Категория – это разные виды компонентов и материалов.

Материал – это расходные материалы, используемые для ремонта и обслуживания оборудования.

Работа – это список действий по обслуживанию и ремонту техники.

Вид – это разная техника, разделенная на определенные группы, по своей принадлежности и функциям.

Производитель – марки и модели компьютерной техники.

Пользователь – это ответственное лицо, отвечающие за создание заявки на ремонт и постановки задачи IT-отделу.

2.1.2 Характеристика входной и справочной информации

Входная информация – это вся информация, используемая для оценки последующего решения поставленных задач. Перед началом проектирования составляется перечень всей используемой входной информации и всех необходимых реквизитов [12].

Также существует и постоянная информация, которая храниться в виде справочников.

Работа разрабатываемой ИС основывается на шести типах справочников. В каждом из них имеются кнопки для редактирования содержащейся в них информации и поиска по ней.

Справочник «Сотрудники» содержит информацию обо всех сотрудниках, работающих в IT отделе торговой сети. Можно узнать имя, специализацию, или телефон для связи с сотрудником.

Справочник «Заказчики» содержит информацию об обратившемся клиенте, его имя, адрес и телефон.

Справочник «Техника» используется для хранения информации обо всей технике, находящейся на ремонте или хранении в IT-отделе.

Справочник «Материалы» содержит информацию обо всех комплектующих, инструментах и материалах, используемых в процессе ремонта.

Справочник «Работы» хранит информацию обо всех работах, выполняемых ИТ отделом. Содержит цену, тип и описание работы.

Справочник «Заявки», является основным и содержит всю информацию о существующих заявках. Использует информацию из всех справочников.

2.1.3 Характеристика документов и отчетов

Для контроля руководством работы ИТ-отдела в системе присутствует возможность формирования отчета по текущему состоянию справочника заявок.

В отчете отражена вся основная информация необходимая для оценки работы сотрудников:

- дата формирования отчета;
- статус заявок;
- дата создания и изменения;
- цена выполненных работ;
- заказчик и исполнитель;
- используемые материалы;
- техника;
- вид выполненных работ.

Пример отчета представлен на рисунке 15.

Выполнено	Код	Дата создания	Дата изменения	Оплата	Заказчик	Исполнитель	Материал	Техника	Работа
Да	1	17.05.2022 23:40	18.05.2022 00:09	1500	Директор Яблоко-5	Евсеев Антон Викторович	Материнская плата	ПК	Ремонт ПК
Нет	2	19.05.2022 21:30	20.05.2022 14:10	2000	Директор Яблоко-3	Подолым Никита Андреевич	Процессор Intel i5	ПК	Замена комплектующих
Да	3	21.05.2022 11:44	23.05.2022 12:45	500	Директор Яблоко-1	Циулин Никита Андреевич	Модуль печати	Весы	Замена печатающей головы
Да	4	23.05.2022 13:54	23.05.2022 21:41	100	Директор Яблоко-9	Евсеев Антон Викторович	Коннектор RJ45	Денежный ящик	Ремонт питания
Да	5	29.05.2022 09:51	30.05.2022 15:22	1200	Директор Яблоко 6	Борухов Владислав Олегович	Фискальный накопитель	Фискальный регистратор	Замена ФН

Рисунок 15 – Пример отчета

Отчет формируется в формате электронной таблицы Microsoft Excel. Для получения отчета в системе во вкладке «Справочник заявок» присутствует кнопка «Построить отчет».

2.1.4 Выбор подходящих для реализации вариантов ПО

Для реализации базы данных будет использован MS SQL Server 2019. «Стоит выделить основные преимущества MS SQL Server 2019:

- решение для обработки больших данных на основе Kubernetes, встроенное в SQL Server, позволяет легко развернуть кластер больших данных и работать с ним. Доступность для большинства распространенных компьютерных платформ;
- данные накапливаются в своем естественном формате, не обязательно реляционном, но могут быть представлены в виде виртуальной таблицы. Виртуализация позволяет интегрировать данные разного формата, из разнородных источников и мест хранения без их репликации и перемещения, создавая единую виртуальную матрицу данных;
- не первый год Microsoft подтверждает высокую производительность SQL Server транзакционными тестами и тестами производительности хранилищ данных. Версия 2019 отмечена отличными результатами;
- любой файл SQL Server, помещенный на устройство PMM, теперь доступен напрямую, минуя стек хранения операционной системы, используя эффективные операции memsru;
- модель HTAP позволяет одновременно осуществлять операционные транзакции и аналитику на одних и тех же данных в одной и той же памяти, также реализуя подход in memory;
- SQL Server 2019 отличается гибкостью в отношении выбора платформы, языка программирования и средства доставки;
- возможности аналитики и визуализации данных. Правильный анализ и эффективное представление результатов напрямую влияет на

эффективность анализа данных и возможность принимать на их основе управленческие решения.» [22]

В качестве языка программирования выберем C#. Потому что в нем есть лучшее из многих парадигм программирования, статическое и динамическое программирование, объектно-ориентированное и функциональное программирование. Кроме того, Microsoft активно поддерживает его, а Visual Studio – отличная IDE с множеством возможностей. Частичные классы вместе с текстовыми шаблонами – лучший способ создавать и контролировать большую базу кода.

Можно выделить несколько преимуществ C#:

- лучшая интегрированная среда разработки;
- достаточно быстрый прогресс изучения;
- разнообразие применений;
- бесплатность;
- библиотека MSDN;
- LINQ (Language-Integrated Query) представляет простой и удобный язык запросов к источнику данных.

2.2 Физическое моделирование информационной системы

2.2.1 Выбор архитектуры информационной системы

Архитектура ИС определена основными требованиями к составу модулей ИС. Данная ИС планируется к установке у сотрудников IT-отдела и директоров магазинов торговой сети (рисунок 16).

В качестве архитектуры будем использовать клиент-серверную архитектуру. В данной архитектуре существует основная служебная машина, называемая – сервер, также ее часто называют хост-система, остальные же компьютеры в данной архитектуре только отправляют и получают услуги от хост-системы [7].

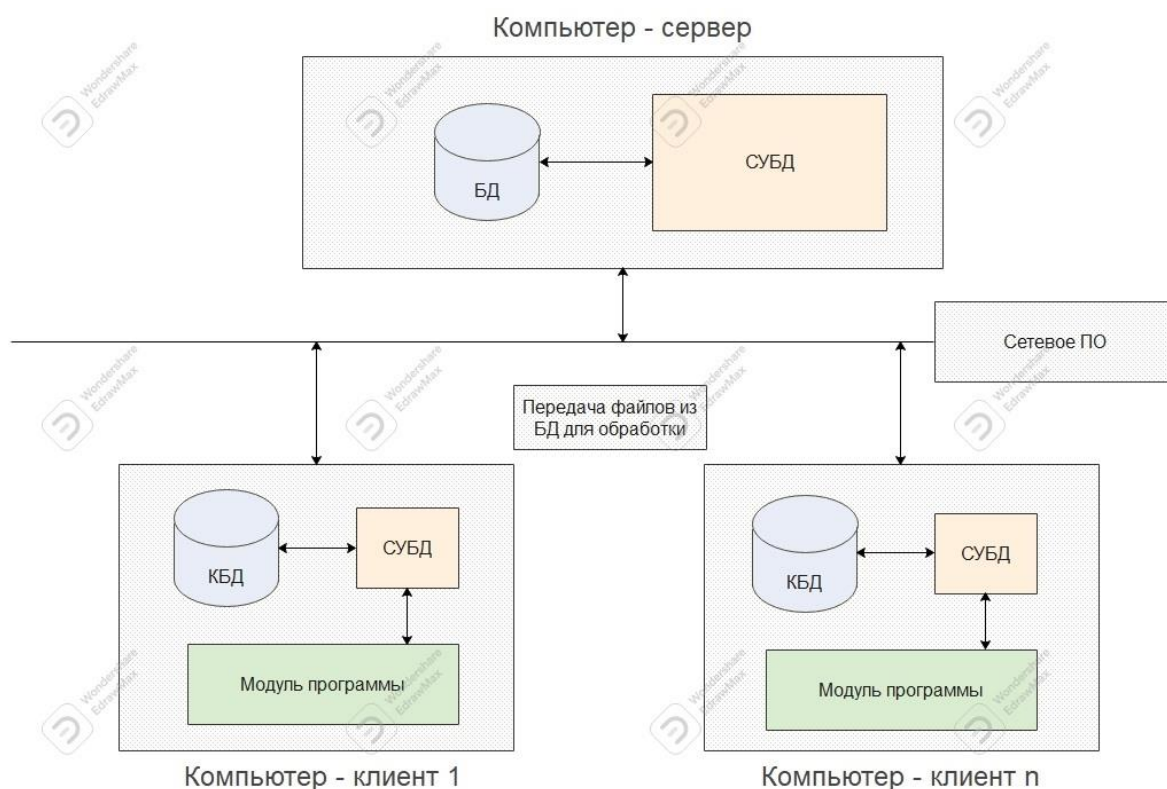


Рисунок 16 – Архитектура разрабатываемой ИС

Клиентская машина предоставляет пользователю так называемый «дружественный интерфейс», чтобы облегчить его взаимодействие с сервером.

Клиент-серверная архитектура может быть разноуровневой, нам подходит двухуровневый вариант такой архитектуры. «К двухуровневой архитектуре «клиент-сервер» следует относить такую, в которой прикладные программы сосредоточены на сервере приложений, а в рабочих станциях находятся программы-клиенты, которые предоставляют для пользователей интерфейс для работы с приложениями на общем сервере». [4]

Преимущества двухуровневой архитектуры:

- легко конфигурировать и модифицировать приложения;
- пользователю обычно легко работать в такой среде;
- хорошая производительность и масштабируемость.

2.2.2 Физическая модель базы данных

В качестве технологии доступа к данным была выбрана ORM система Entity Framework Core версии 3.1 и СУБД Microsoft SQL Server 2019.

«Entity Framework Core – это технология для доступа к базам данных от Microsoft. Оно позволяет взаимодействовать с СУБД с помощью сущностей (entity), то есть классов и объектов NET, а не таблиц базы данных. Это самый известный и функциональный ORM – инструмент в C#. ORM – это object-relational mapping – отображение данных на реальные объекты.» [21]

Существует несколько вариантов работы с ORM, мы используем первый подход (Code First). Выбранный подход подразумевает, что сначала пишется код на C#, а потом по этому коду создается база данных. Для этого подхода очень важно определить классы модели или entity, которая будет храниться в базе данных, описать ее в классах C# в виде модели, и написать класс контекста, который и будет работать с используемой базой данных [17].

В процессе работы были спроектированы классы моделей данных, установлены зависимости между ними. Далее с помощью подхода code-first классы моделей были сконфигурированы.

С помощью инструмента миграции была создана база данных на основе спроектированных сущностей и их зависимостей и спроектирована ее физическая модель (рисунок 17). Также данный инструмент легко синхронизирует с моделью данных все внесенные изменения в базах данных [13].

При работе с инструментом миграции можно использовать операции, позволяющие изменять данные, вносить правки в таблицы и внешние ключи, удалять и редактировать настройки. В процессе создания миграции автоматически генерируется класс, выполняющий все необходимые операции для применения миграции [2].

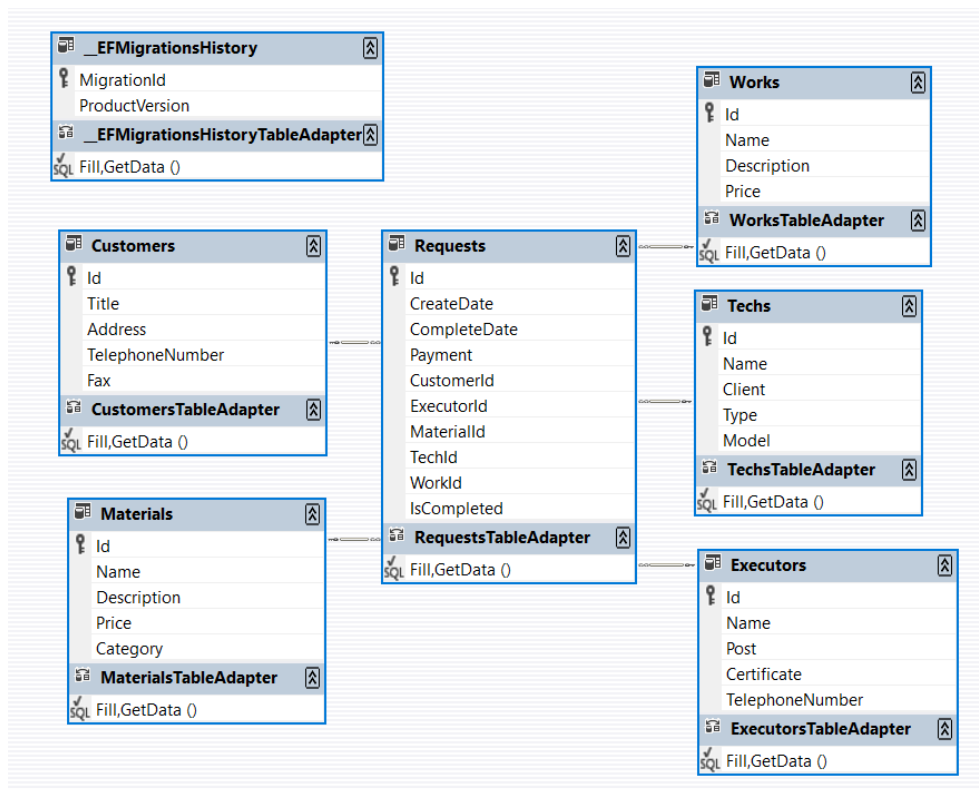


Рисунок 17 – Диаграмма физической модели базы данных

Для корректной работы с базой данных в проекте используется технология LINQ. «LINQ использует похожие на SQL выражения языка C# для получения данных из базы данных. Любая реляционная база данных работает через запросы на языке SQL, и Entity Framework Core выражения LINQ to Entities транслирует в запросы SQL, которые понятны для используемой базы данных.» [22]

2.2.3 Функциональная схема проекта

Основным назначением разрабатываемой системы является обслуживание ПК и оргтехники торговой сети, а также коммуникации между директорами магазинов и сотрудниками IT-отдела. Для лучшей адаптации сотрудников к новой ИС необходимо создать интуитивно понятный интерфейс. Основой данного интерфейса будет являться главное окно программы. Также будут присутствовать формы для ввода информации. Функциональная схема ИС представлена на рисунке 18.



Рисунок 18 – Функциональная схема ИС

Информация по итогам выполненных работ будет формироваться при использовании блока формирования отчетов. Для удобства пользователей в системе будет присутствовать справочная система.

Выводы по главе 2

Построены диаграммы вариантов использования и последовательностей с использованием методологии UML. Построена логическая модель будущей системы. При построении данной модели была определена входная информация, а также справочники, которые хранят постоянную информацию, определена результатная информация системы.

Сформированы характеристики документов и отчетов. Осуществлен подбор программного обеспечения и архитектуры информационной системы. Также была спроектированная физическая модель базы данных. На основе проведенного моделирования и анализа была сформирована функциональная схема проекта.

Глава 3 Реализация разработанной системы и расчет экономической эффективности

3.1 Описание разработанной информационной системы

3.1.1 Описание реализации разработанной системы

В качестве целевой среды для реализации программы была выбрана .NET Framework версии 4.7.2 и технология Windows Forms.

«.NET Framework – это зрелая и полнофункциональная платформа, которая поставляется вместе с Windows. Экосистему .NET Framework разработчики используют уже много лет. Она поддерживает разработку современных приложений и обладает высочайшей совместимостью с существующими приложениями и библиотеками.» [16]

Существуют различные версии .NET. Каждая позволяет коду .NET выполняться в разных системах – Linux, macOS, Windows, iOS, Android и др. Также платформа является цельным компонентом с длительным циклом выпуска и множеством API.

Для разработки графических интерфейсов используют технологии – WPF, Window Forms, UWP, они создаются при помощи платформы .NET. Благодаря удобству и простоте использования предпочтение отдается технологии Window Forms.

В Window Forms используется множество различных классов, которые содержат визуальные стили, позволяющие создавать собственные меню и панели задач всевозможных типов, также присутствуют богатые возможности для связи источников данных – базами данных, с используемыми компонентами [11].

Были спроектированы классы моделей данных, установлены зависимости между ними. Далее с помощью подхода code-first классы моделей были сконфигурированы, созданы миграции, на основании которых сгенерирована база данных (рисунок 19) [15].

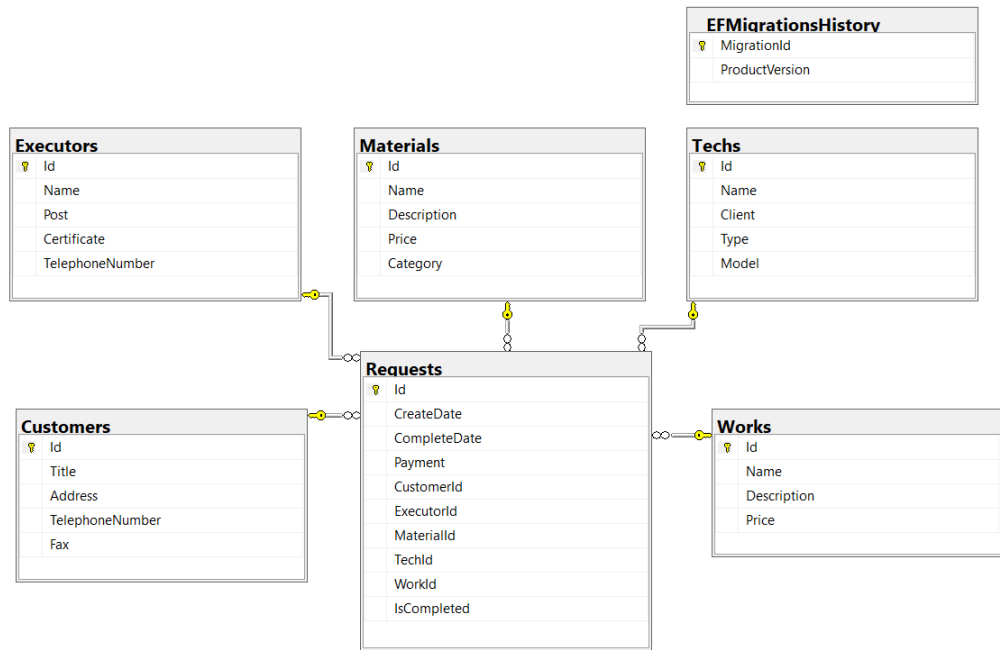


Рисунок 19 – Диаграмма базы данных

Чтобы инкапсулировать технологию доступа к хранилищу данных, был реализован Репозиторий.

Для четкого разделения ответственности решение содержит несколько проектов (рисунок 20):

- Domain – слой, содержащий классы моделей данных;
- DatabaseLayer – слой непосредственного доступа к данным, содержит контекст Entity Framework Core, а также конфигурации классов модели;
- DbRepository – слой, в котором реализован паттерн Репозиторий;
- Infrastructure – набор классов помощников и методов расширений;
- ApplicationProcessingSystem – пользовательский интерфейс приложения.

Часть программного кода описанных проектов представлена в приложении А.

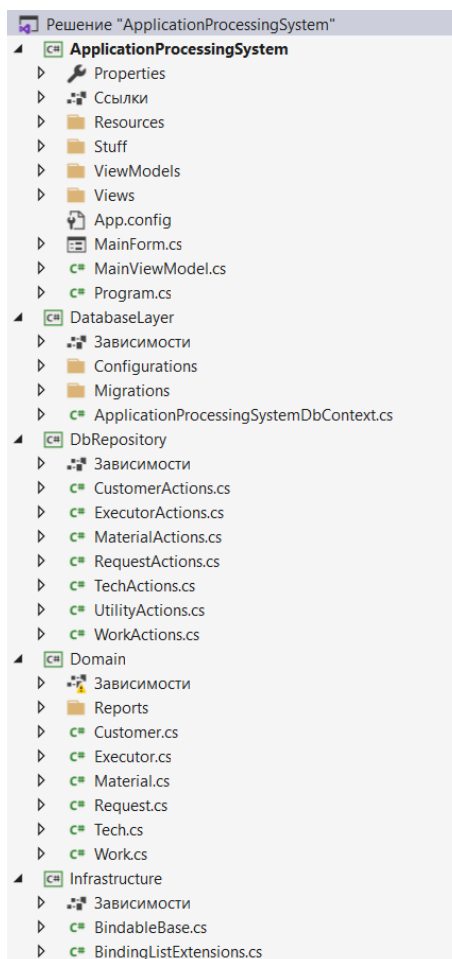


Рисунок 20 – Список проектов

При проектировании пользовательского интерфейса приложения был использован паттерн Model-View-ViewModel. Данный паттерн состоит из трех компонентов:

- модели (Model) – описывает используемые в приложении данные. Модели могут содержать логику, непосредственно связанную этими данными, например, логику валидации свойств модели. В то же время модель не должна содержать никакой логики, связанной с отображением данных и взаимодействием с визуальными элементами управления;
- представления (View) – определяет визуальный интерфейс, через который пользователь взаимодействует с приложением;

– модели представления (ViewModel) – связывает модель и представление через механизм привязки данных. Если в модели изменяются значения свойств, при реализации моделью интерфейса `INotifyPropertyChanged` автоматически идет изменение отображаемых данных в представлении, хотя напрямую модель и представление не связаны.

Паттерн позволяет отделить логику приложения от визуальной части (представления). Данный паттерн является архитектурным, то есть он задает общую архитектуру приложения [10].

3.1.2 Описание пользовательского интерфейса

Основная форма разработанной АИС используется для создания, редактирования, просмотра статуса заявок от работников магазинов.

Как уже отмечалось ранее, существует несколько типов пользователей данной системы – это работники магазинов торговой сети и сотрудники IT-отдела, которые выполняют работы по ремонту компьютерной и кассовой техники, а также по установке программного обеспечения.

Основное окно программы, с которым будут взаимодействовать пользователи, представлено на рисунке 21.

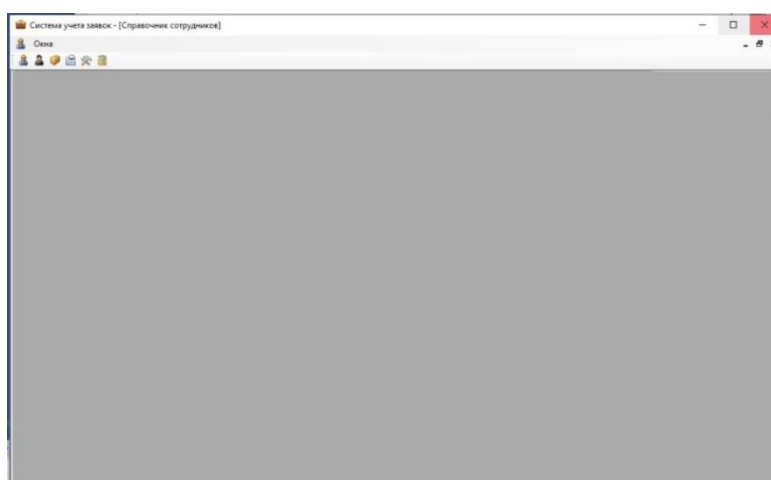


Рисунок 21 – Основное окно программы

На панели сверху, располагаются шесть иконок, каждая из которых отвечает за свой раздел программы (рисунки 21-27):

- Справочник сотрудников;
- Справочник заявок;
- Справочник заказчиков;
- Справочник материалов;
- Справочник техники;
- Справочник работ.

Код	Имя	Должность	Телефон	Сертификат
1	Подольян Никита Андреевич	Старший Системный администратор	358-45-66	Есть
2	Евсеев Антон Викторович	Системный Администратор	358-84-47	Есть
3	Борухов Владислав Олегович	Программист	358-55-71	Отсутствует
4	Циулин Никита Андреевич	Системный администратор	358-59-98	Есть

Рисунок 22 – Справочник сотрудников

Выполнено	Код	Дата создания	Дата изменения	Оплата	Заказчик	Исполнитель	Материал	Техника	Работа
<input checked="" type="checkbox"/>	1	17.05.2022 23:40	18.05.2022 00:09	1500,00	Директор Яблоко-5	Евсеев Антон Вик...	Материнская пл...	ПК	Ремонт ПК
<input type="checkbox"/>	2	31.05.2022 12:20	31.05.2022 12:20	200,00	Директор Яблоко-1	Подольян Никита...	Чековая Лента	Фискальный реи...	Мелкий ремонт (...)
<input type="checkbox"/>	3	31.05.2022 12:21	31.05.2022 12:21	200,00	Директор Яблоко-9	Циулин Никита А...	Роутер D-Link	ПК (комплект)	Мелкий ремонт (...)
<input checked="" type="checkbox"/>	5	31.05.2022 12:23	31.05.2022 12:24	2500	Заведующая Ябл...	Борухов Владисл...	Компонент терм...	Весы	Ремонт весов

Рисунок 23 – Справочник заявок

Код	Наименование	Адрес	Телефон	Факс
1	Директор Яблоко-1	Техническая 37	359-84-98	-
2	Директор Яблоко-5	Мичурина 99	359-78-98	-
3	Заведующая Яблоко-7	Библиотечная 87	359-87-44	-

Рисунок 24 – Справочник заказчиков

Код	Название	Описание	Цена	Категория
1	Чековая лента	Для фискальных аппаратов	30,00	Кассовое оборудование
2	Материнская плата ASUS	-	5990	Комплектующие ПК
3	Роутер D-Link	-	2598	Сетевое оборудование
4	Бухта интернет кабеля	1000 м	5000	Сетевое оборудование

Рисунок 25 – Справочник материалов

Код	Наименование	Клиент	Вид	Модель
1	Фискальный регистратор	-	-	M54-KX
2	Персональный компьютер (комплект)	Я-6	Intel i-3, 4gb ddr4	IN WIN
3	Клавиатура Logitech	Я-7	-	BX-93dse

Рисунок 26 – Справочник техники

Код	Название	Описание	Стоимость
1	Замена кассовой ленты	Произвести замену отрыв кассы аппарата, пр...	50,00
2	Ремонт фискального регистратора	Передать аналитику для выяснения причин сбоев	900
3	Ремонт ПК		1500

Рисунок 27 – Справочник работ

Во всех справочниках присутствует строка для быстрого поиска с кнопкой «Найти» и кнопки для редактирования: «Добавить», «Удалить».

Информацию в любом выбранном справочнике можно обновить по нажатию кнопки «Обновить список». В справочнике заявок присутствует специальная кнопка «Построить отчет», с ее помощью создается электронный отчет в Microsoft Excel.

Также по нажатию кнопки «Окна» открывается список со всеми активными в данный момент справочниками (рисунок 28).

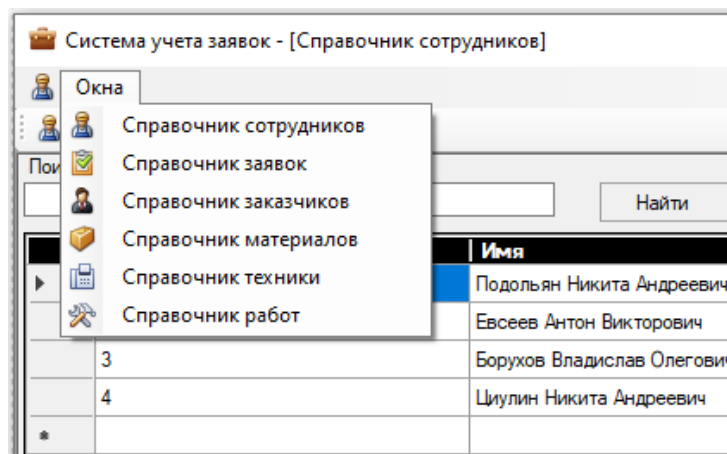


Рисунок 28 – Список активных справочников

Для регистрации новой заявки необходимо открыть «справочник заявок», представленный на рисунке 23. Для регистрации новой заявки необходимо нажать кнопку «Добавить», после чего пользователю выводится диалоговое окно в соответствии с рисунком 29.

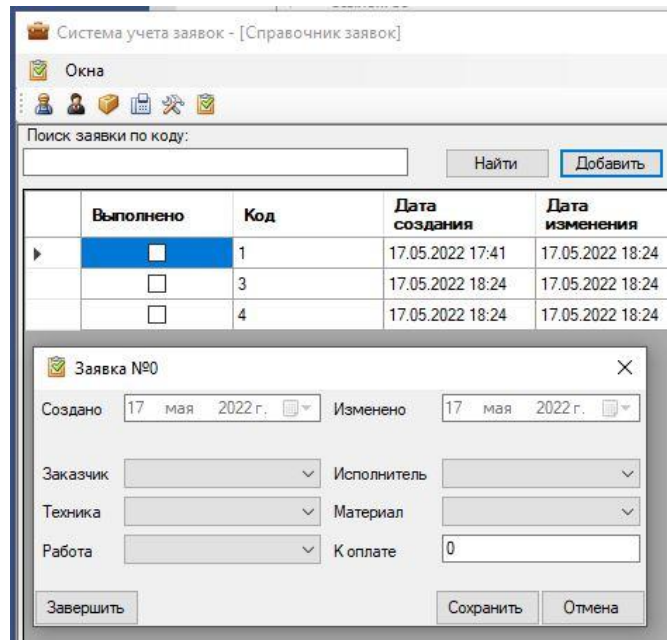


Рисунок 29 – Диалоговое окно

Все пункты в этом окне выполнены в виде выпадающего списка, на основе справочников, представленных на рисунках 21-27. После введения программы в эксплуатацию системные администраторы заполняют справочники необходимой информацией и это позволит сильно сократить время при регистрации новых заявок [14].

На рисунке 30 можно увидеть пример заполнения заявки с использованием информации из справочников.

The screenshot shows a window titled "Заявка №0" with a close button in the top right corner. The form contains the following fields and values:

- Создано: 17 мая 2022 г.
- Изменено: 17 мая 2022 г.
- Заказчик: Я-2 Мерцает монитор
- Исполнитель: Евсеев Антон Викторов
- Техника: Фискальный регистр
- Материал: Роутер D-Link
- Работа: (dropdown menu open with options: Замена кассовой ленты, Ремонт фискального регистр, Ремонт ПК)
- К оплате: 0

Buttons: Сохранить, Отмена

Рисунок 30 – Пример создания новой заявки

В форме создания и редактирования заявки присутствует поле «К оплате», оно автоматически формирует цену на основании выбранной работы и цены на указанный материал, но пользователь может ввести сумму и собственноручно.

Также при нажатии двойным щелчком на уже активную заявку, её можно редактировать, изменяя необходимые параметры или полностью отметить её как завершённую путем нажатия кнопки «Завершить».

После закрытия заявки она отмечается в списке галочкой и поля при редактировании становятся неактивными (рисунок 31).

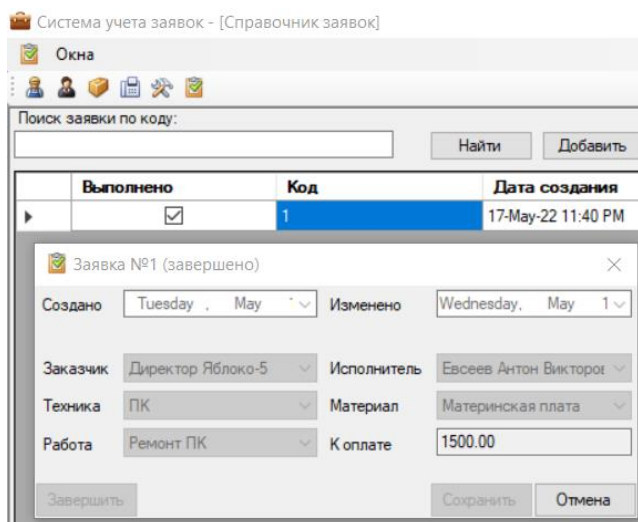


Рисунок 31 – Завершенная заявка

Данный интерфейс программы включает в себя весь функционал, необходимый сотрудникам торговой сети для быстрого отслеживания текущих работ и также он достаточно прост в освоении для новых пользователей.

3.2 Оценка и обоснование экономической эффективности проекта

3.2.1 Выбор и обоснование методики расчета экономической эффективности

Внедрение новой технологии будет оправдано лишь тогда, когда оно ведет к повышению производительности труда и улучшению условий труда.

Предметная область проекта содержит несколько недостатков, которые обусловлены использованием колл-центра при обработке заявок пользователей и формированием всей отчетности на бумажных носителях. Результатом является трата большого количества ресурсов и рабочего времени специалистов на эти процессы, имеются неточности и ошибки.

Для расчета экономической эффективности был выбран вариант сравнения трудозатрат базового и автоматизированного варианта выполнения основного процесса торговой сети.

Обоснование экономической эффективности обработки информации с использованием ИС позволяет:

- выявить целесообразность затрат на внедрение информационной системы для обработки информации внутри компании;
- рассчитать сроки окупаемости затрат на создание информационной системы;
- определить влияние новых технологических средств на экономические показатели деятельности компании.

Для расчета прямого эффекта были рассмотрены показатели трудовых и стоимостных затрат.

Трудовые показатели:

Абсолютное снижение трудовых затрат (ΔT) на обработку заявок может быть рассчитано по формуле:

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

где T_0 – трудозатраты по базисному варианту;

T_1 – трудозатраты по проектному варианту.

Коэффициент относительного снижения трудовых затрат (K_T):

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

Индекс повышения производительности труда (Y_T):

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

К стоимостным показателям относятся:

Абсолютное снижение стоимостных затрат (ΔC):

$$\Delta C = C_0 - C_1 \quad (4)$$

где C_0 – затраты по базовому варианту;

C_1 – затраты по проектному варианту.

Коэффициент относительного снижения стоимостных затрат (K_c):

$$K_c = \frac{\Delta C}{C_0} * 100\% \quad (5)$$

Индекс снижения стоимостных затрат (Y_c):

$$Y_c = \frac{C_0}{C_1} \quad (6)$$

Показатель стоимостных затрат на j -ой технологический процесс C_j представляет собой сумму затрат на j -ый технологический процесс по следующим статьям затрат: на заработную плату; на амортизацию; на ведение информационной базы; накладные расходы.

Этот показатель рассчитывается по формуле:

$$O_j = \sum_{i=1}^n C_{ij} \quad (7)$$

где C_{ij} – показатель стоимостных затрат на i -тую операцию j -того технологического процесса, включающий следующие составляющие:

$$C_{ij} = C_{з/п} + C_{нр} + C_a + C_{иб} \quad (8)$$

где $C_{з/п}$ – затраты на заработную плату пользователя, рассчитываемые по формуле:

$$C_{з/п} = t_{ij} * r_i \quad (9)$$

где t_{ij} – трудоемкость выполнения i -ой операции j -го технологического процесса;

r_i – тарифная ставка i -ой операции.

$C_{нр}$ – затраты на накладные расходы, которые рассчитывают, как производную величину от затрат на заработную плату:

$$C_{нр} = C_{з/п} * K_{нр} \quad (10)$$

где $K_{нр}$ – коэффициент накладных расходов, принимаемый, как правило, в размере 0,6 - 0,7 от затрат за заработную плату;

C_a – амортизационные отчисления на используемую технику, рассчитываемые по формуле:

$$C_a = t_{ij} * a_i \quad (11)$$

где a_i – норма амортизационных отчислений;

$C_{иб}$ – годовые затраты на ведение информационной базы.

Коэффициенты K_j и K_0 характеризуют рост производительности труда за счет внедрения более экономичного варианта проектных решений.

При оценке эффективности используются обобщающие и частные показатели. К основным обобщающим показателям экономической эффективности относятся:

- годовой экономический эффект;
- расчетный коэффициент эффективности капитальных вложений;
- срок окупаемости системы.

Годовой экономический эффект от внедрения проекта (\mathcal{E}) равен разности между годовой экономией и нормативной прибылью (руб./год):

$$\mathcal{E} = (C_0 + E_H * K_0) - (C_j + E_H * K_j) \quad (12)$$

где K_j и K_0 - капитальные затраты, которые включают следующие виды затрат:

- затраты на приобретение вычислительной техники в базовом и предлагаемом вариантах;
- затраты на покупку ПО;
- освоение программного обеспечения;
- проектирование и отладка проекта [20].

Помимо рассмотренных показателей эффективности также необходимо оценить годовой экономический эффект ($\mathcal{E}_{\text{год}}$) по формуле:

$$\mathcal{E}_{\text{год}} = C_1 - C_2 \quad (13)$$

где C_1 – себестоимость до внедрения программного комплекса;

C_2 – себестоимость после внедрения программного комплекса.

Коэффициент экономической эффективности (E):

$$E = \frac{\mathcal{E}_{\text{год}}}{K_{\text{вл}}} \quad (14)$$

где $\mathcal{E}_{\text{год}}$ – годового экономического эффекта;

$K_{\text{вл}}$ – капитальные вложения.

Срок окупаемости затрат на внедрение программного продукта ($T_{\text{ок}}$) вычисляется по формуле:

$$T_{\text{ок}} = \frac{1}{E} \quad (15)$$

Среднечасовая норма оплаты труда ($Z_{\text{ч}}$) рассчитывается по формуле:

$$Z_{\text{ч}} = \frac{\Phi_{\text{ч}}}{T_{\text{ч}}} \quad (16)$$

где $\Phi_{\text{ч}}$ – фонд заработной платы за изучаемый период;

$T_{\text{ч}}$ – отработанные человеко-часы.

Стоимостные затраты на бизнес-процесс учета и обработки заявок по базовому варианту ($Z_{\text{бв}}$) рассчитываются по формуле:

$$Z_{\text{бв}} = K_3 * Z_{\text{ч}} \quad (17)$$

где K_3 – среднее число заявок, поступающих в IT-отдел;

$Z_{\text{ч}}$ – среднечасовая норма оплаты труда.

Стоимостные затраты на бизнес-процесс учета и обработки заявок по проектному варианту ($Z_{\text{пв}}$) рассчитываются по формуле:

$$Z_{\text{пв}} = (Z_{\text{ч}} + C_a + T_{ij}) * T_{ij} \quad (18)$$

где C_a – часовая норма амортизации;

T_{ij} – трудоемкость на выполнение операций.

Используя приведенные показатели, можно определить экономическую эффективность от внедрения проекта путем сравнения трудоемкости базового и предлагаемого варианта учета и обработки заявок.

3.2.2 Расчет показателей экономической эффективности проекта

Чтобы узнать экономический эффект от внедрения проекта необходимо произвести расчет основных входных данных. Рассчитываем среднечасовую норму оплаты труда для сотрудника торговой сети. За основу берем стандартную 40-часовую рабочую неделю и месячную зарплату сотрудника в 30000 рублей, производим расчет по формуле (16):

$$z_{\text{ч}} = \frac{30000}{160} = 187,5$$

Средняя почасовая оплата составляет 187,5 руб./час.

Далее произведем расчет стоимостных затрат на бизнес-процесс учета и обработки заявок по базовому варианту. В базовом варианте на обработку и учет одной заявки уходит в среднем 60 минут, а среднее число заявок, поступающих в IT-отдел торговой сети 1874 заявки в год.

С использованием этих данных производим расчет по формуле (17):

$$Z_{\text{бв}} = 1874 * 187,5 = 351\ 375$$

По результатам получаем, что стоимостные затраты по базовому варианту равняются 351 375 рублей в год.

Для проведения расчета стоимостных затрат на бизнес-процесс учета и обработки заявок по проектному варианту, необходимо узнать время на выполнение этих операций с использованием ИС.

Расчеты трудоемкости на выполнение этих операций приведены на рисунке 32 и составляют 151,02 часов в год. Исходя из полученных данных производим расчет для проектного варианта по формуле (18):

$$Z_{\text{пв}} = (187,5 + 28,76 + 151,02) * 151,02 = 52482,12 \text{ рублей в год.}$$

Операция технологического процесса	Объем работы в год (операций)	Временные затраты на одну операцию, мин.	Трудоёмкость (час в год)	Часовая зарплата специалиста	Часовая норма амортизации (руб/час)	Часовая стоимость накладных расходов	Стоимость руб/год
1	2	3	$4=2*3/60$	5	6	$7=5*0,7$	$8=(5+6+7)*4$
Вызов главного меню	1456	0,3	7,28	187,5	28,76	131,25	2529,87
Авторизация	1456	1	24,27	187,5	28,76	131,25	8432,91
Заполнение заявки	1456	3	72,80	187,5	28,76	131,25	25298,73
Регистрация выполнения заявки	1456	0,5	12,13	187,5	28,76	131,25	4216,45
Просмотр журнала заявок	456	2	15,20	187,5	28,76	131,25	5282,15
Формирование плана работ	456	2	15,20	187,5	28,76	131,25	5282,15
Печать фрагмента таблицы	456	0,5	3,80	187,5	28,76	131,25	1320,54
Импорт файла в Excel	206	0,1	0,34	187,5	28,76	131,25	119,31
Итого:	-	-	151,02	-	-	-	52482,12

Рисунок 32 – Оценка стоимости выполнения операции в информационной системе

Производим расчет годовой экономической эффективности от внедрения ИС по формуле (13):

$$Э_{\text{год}} = 351375 - 52482 = 298\,893 \text{ рублей в год}$$

Далее вычислим коэффициент экономической эффективности проекта по формуле (14):

$$E = \frac{351375}{52482} = 6,69$$

Срок окупаемости затрат на внедрение информационной системы по формуле (15):

$$T_{\text{ок}} = \frac{1}{6,69*12} = 1,8 \text{ месяца}$$

По итогам расчетов можно сделать вывод, что в современном мире, где стремительно развиваются информационные технологии необходимо внедрять и использовать программные продукты, позволяющие сильно сократить время и расходы на выполнение операций внутри компании. Такие системы снижают количество ошибок и ненужных затрат в процессе работы.

Проект окупится примерно за два месяца, с учетом затрат на его введение в эксплуатацию, а годовой экономический эффект составляет 298 893 рубля.

Выводы по главе 3

По итогам разработки была реализована информационная система с простым и интуитивно понятным пользовательским интерфейсом, включающая в себя все необходимые справочники и отчеты для отслеживания работы IT-отдела торговой сети.

В заключении можно сделать вывод, что получение прибыли после внедрения информационной системы ожидается с третьего месяца после начала её эксплуатации, а точные показатели эффективности от внедрения программы можно будет проанализировать только после разработки и эксплуатации внутри торговой сети.

Заключение

За время работы над выпускной квалификационной работой были решены следующие задачи:

- изучена структура торговой сети «Яблоко», ее организационная структура, основные бизнес-процессы и направления деятельности;
- разработаны и построены концептуальные модели бизнес-процессов IT-отдела торговой сети;
- произведен анализ уже существующих аналогов подобных систем;
- разработана необходимая модель базы данных и успешно реализована;
- выбрана подходящая архитектура информационной системы и разработано программное средство;
- разработан простой и интуитивно понятный интерфейс информационной системы;
- описан процесс обработки, передачи и сбора необходимой информации внутри системы;
- проведен анализ экономической выгоды по итогам внедрения проекта.

Благодаря внедрению ИС бизнес-процесс учета и обработки заявок на ремонт компьютерной техники сильно упрощается. Система позволяет достичь высокого уровня автоматизации внутри IT-отдела торговой сети. Проект упрощает работу сотрудников и позволяет руководству легко следить за работой и качеством ее выполнения.

Смоделированная и реализованная для информационной системы база данных позволила улучшить и упорядочить учёт техники, используемой внутри торговой сети, что снизило количество ошибок, допускаемых при оформлении техники, и сократило время, затрачиваемое на этот процесс.

После внедрения системы, торговая сеть может отказаться от использования колл-центра для обработки поступающих заявок, что сильно снизит уровень трудовых и экономических затрат.

Так как система имеет интуитивно понятный интерфейс, работе с программой может легко обучиться любой, даже не опытный пользователь. Графическая оболочка представляет собой легкую для восприятия структуру и делает процесс обработки информации очень простым.

Также по итогам проведённого анализа экономической эффективности проекта и выполненным сравнениям расчетов стоимостных затрат на бизнес-процесс учета и обработки заявок по базовому и проектному варианту, был сделан вывод, что внедрение информационной системы окупится примерно за два месяца с учетом затрат на его введение в эксплуатацию и будет иметь большой годовой экономический эффект для торговой сети.

Таким образом, разработанная информационная система подойдет для внедрения в компаниях и учреждениях, где присутствует необходимость мониторинга работы, учета и контроля за процессами обслуживания компьютерной техники, благодаря обширному набору функций для хранения информации.

Список используемой литературы и используемых источников

1. Автоматизированные информационные технологии в экономике / Под ред. Г.А. Титоренко. – М.: ЮНИТИ, 2011.
2. Агальцов В.П. Базы данных: Учебное пособие. – М.: Мир, 2012.
3. Александров Д. Инструментальные средства информационного менеджмента. М.: Финансы и статистика, 2012. – 224 с.
4. Архитектура «Клиент-Сервер» [Электронный ресурс]. URL: <https://itelon.ru/blog/arkhitektura-klient-server/> (дата обращения: 18.05.2022).
5. Боковой Ю. Особенности методологии проектирования информационных систем для малого и среднего бизнеса. М.: Изд. дом «Синергия», 2016. – 145 с.
6. Вендров А.М. Практикум по проектированию программного обеспечения экономических информационных систем. Учеб. пособие. – М.: Финансы и статистика, 2014.
7. Виейра Р. Программирование баз данных Microsoft SQL Server. Базовый курс. М.: Диалектика, 2009 – 812 с.
8. Виссема Х. Менеджмент в подразделениях фирмы (предпринимательство и координация в децентрализованной компании): Пер. с англ. – М.: ИНФРА-М, 2010. – 239 с.
9. Виханский О.С., Наумов А.И. Менеджмент: человек, стратегия, организация, процесс: учебник. – М., 2007. – 416 с.
10. Гарсиа-Молина Г., Ульман Дж. Системы баз данных. – М.: Изд. Дом «Вильямс», 2013. – 1088 с.
11. Грошев, А. С. Основы работы с базами данных: учебное пособие Ай Пи Ар Медиа, 2021. – 255 с.: [Электронный ресурс]. URL: <https://www.iprbookshop.ru/102038.html> (дата обращения: 01.06.2022).
12. Долженко, А. И. Управление информационными системами: Ай Пи Ар Медиа, 2021. – 180 с.: [Электронный ресурс]. URL: <https://www.iprbookshop.ru/102074.html> (дата обращения: 17.04.2022).

13. Исаев Г. Проектирование информационных систем. М.: Омега-Л, 2012. – 431 с.
14. Котляров, В. П. Основы тестирования программного обеспечения: Профобразование, 2019. – 335 с.: [Электронный ресурс]. URL: <https://www.iprbookshop.ru/86202.html> (дата обращения: 22.05.2022).
15. Марков А.С., Лисовский К.Ю. Базы данных. Введение в теорию и методологию. Учебник. – М.: Финансы и статистика, 2004. – 442 с.
16. Немного про .NET Framework и .NET Core [Электронный ресурс]. URL: [https://pcnews.ru/top/blogs/day/\[perevod\]_nemnogo_pro_net_framework_i_net_core_\[plus_poleznye_ssyunki\]-807544.html](https://pcnews.ru/top/blogs/day/[perevod]_nemnogo_pro_net_framework_i_net_core_[plus_poleznye_ssyunki]-807544.html) (дата обращения: 12.05.2022).
17. Переход на Microsoft SQL Server 2019 [Электронный ресурс]. URL: <https://softline.ru/about/blog/10-prichin-pereyti-na-microsoft-sql-server-2019> (дата обращения: 17.05.2022).
18. Родионов И. И. и др. Рынок информационных услуг и продуктов. – М.: МК-Периодика, 2012.
19. Федоров А.Г., Елманова Н.З. Базы данных для всех. КомпьютерПресс, М., 2011. – 256 с.
20. Экономическое обоснование эффективности дипломного проекта [Электронный ресурс]. URL: <https://pandia.ru/text/80/171/56140.php> (дата обращения: 20.05.2022)
21. Entity Framework Core [Электронный ресурс]. URL: <https://habr.com/ru/company/otus/blog/500012/> (дата обращения: 10.05.2022).
22. Visual Studio 2008. Краткий обзор [Электронный ресурс]. URL: <https://compress.ru/article.aspx?id=19604> (дата обращения: 11.05.2022).

Приложение А

Программный код

ViewModels

```
using DbRepository;
using Domain;
using Infrastructure;
using System.Collections.Generic;
using System.ComponentModel;
using System.Linq;

namespace ApplicationProcessingSystem.ViewModels.Directories
{
    public class CustomersDictionaryViewModel : BindableBase
    {
        private string _titleToSearch;
        public List<Customer> AllCustomers { get; } = new List<Customer>();
        public BindingList<Customer> Customers { get; } = new BindingList<Customer>();
        public string TitleToSearch
        {
            get => _titleToSearch ?? string.Empty;
            set => SetProperty(ref _titleToSearch, value);
        }
        public void Refresh()
        {
            var repo = new CustomerActions();
            var customers = repo.GetAllCustomers();
            AllCustomers.Clear();
            AllCustomers.AddRange(customers);
            FilterByName();
        }

        public void FilterByName()
        {
            var filteredCustomers = AllCustomers.Where(c =>
c.Title.Contains(TitleToSearch));
            Customers.ClearAndAddRange(filteredCustomers);
        }
        public void DeleteCustomer(Customer customer)
        {
            var repo = new CustomerActions();
            repo.DeleteCustomer(customer);
            AllCustomers.Remove(customer);
            Customers.Remove(customer);
        }
        public void AddCustomer(Customer customer)
        {
            AllCustomers.Add(customer);
            FilterByName();
        }
        public void UpdateCustomer(Customer customer)
        {
            var repo = new CustomerActions();
            repo.ReloadCustomer(customer);
        }
    }
}
```

Продолжение Приложения А

DataBaseConfiguration

```
using Domain;
using Microsoft.EntityFrameworkCore;
using Microsoft.EntityFrameworkCore.Metadata.Builders;

namespace DatabaseLayer.Configurations
{
    /// <summary>Конфигурация Заказчика</summary>
    public class CustomerConfiguration : IEntityTypeConfiguration<Customer>
    {
        /// <inheritdoc />
        public void Configure(EntityTypeBuilder<Customer> builder)
        {
            builder.ToTable("Customers").HasKey(p => p.Id);

            builder.Property(p => p.Id).HasColumnName("Id");
            builder.Property(p => p.Title).HasColumnName("Title");
            builder.Property(p => p.Address).HasColumnName("Address");
            builder.Property(p => p.TelephoneNumber).HasColumnName("TelephoneNumber");
            builder.Property(p => p.Fax).HasColumnName("Fax");
        }
    }
}
```

DataBaseRepository

```
using System.Collections.Generic;
using Domain;
using DatabaseLayer;
using System.Linq;
using Microsoft.EntityFrameworkCore;

namespace DbRepository
{
    public class CustomerActions
    {
        public ICollection<Customer> GetAllCustomers()
        {
            using var context = new ApplicationProcessingSystemDbContext();
            var customers = context.Customers.ToList();
            return customers;
        }

        public Customer GetCustomerById(int id)
        {
            using var context = new ApplicationProcessingSystemDbContext();
            var customer = context.Customers.SingleOrDefault(c => c.Id == id);
            return customer;
        }

        public void SaveCustomer(Customer customer)
        {
            using var context = new ApplicationProcessingSystemDbContext();
            var entry = context.Attach(customer);
            entry.State = entry.IsKeySet ? EntityState.Modified : EntityState.Added;
            context.SaveChanges();
        }
    }
}
```


Продолжение Приложения А

```
public void DeleteCustomer(Customer customer)
{
    using var context = new ApplicationProcessingSystemDbContext();
    context.Remove(customer);
    context.SaveChanges();
}

public void ReloadCustomer(Customer customer)
{
    using var context = new ApplicationProcessingSystemDbContext();
    var entry = context.Customers.Attach(customer);
    entry.Reload();
}
}
```

ExecutorActions

```
using System.Collections.Generic;
using Domain;
using DatabaseLayer;
using System.Linq;
using Microsoft.EntityFrameworkCore;

namespace DbRepository
{
    public class ExecutorActions
    {
        public ICollection<Executor> GetAllExecutors()
        {
            using var context = new ApplicationProcessingSystemDbContext();
            var executors = context.Executors.ToList();
            return executors;
        }

        public Executor GetExecutorById(int id)
        {
            using var context = new ApplicationProcessingSystemDbContext();
            var executor = context.Executors.SingleOrDefault(e => e.Id == id);
            return executor;
        }

        public void SaveExecutor(Executor executor)
        {
            using var context = new ApplicationProcessingSystemDbContext();
            var entry = context.Attach(executor);
            entry.State = entry.IsKeySet ? EntityState.Modified : EntityState.Added;
            context.SaveChanges();
        }

        public void DeleteExecutor(Executor executor)
        {
            using var context = new ApplicationProcessingSystemDbContext();
            context.Remove(executor);
            context.SaveChanges();
        }

        public void ReloadExecutor(Executor executor)
        {
            using var context = new ApplicationProcessingSystemDbContext();
            var entry = context.Executors.Attach(executor);
            entry.Reload();
        }
    }
}
```