

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 проекта автоматизации контакт центра компании»

Обучающийся И. С. Садретдинов (И.О. Фамилия) \_\_\_\_\_ (личная подпись)

Руководитель канд. пед. наук, доцент, Т.А. Агошкова  
(ученая степень, звание, И.О. Фамилия)

Тольятти 2023

## Аннотация

Выпускная квалификационная работа состоит из 52 страниц, 25 рисунков, 4 таблиц, 26 источников литературы.

Темой выпускной квалификационной работы является разработка информационной системы автоматизации контакт центра на примере компании провайдера, предназначенная для повышения эффективности обработки обращений клиентов.

Автоматизация контакт центра компании провайдера является актуальной в связи с тем, что автоматизация позволяет предоставлять быстрое и точное обслуживание клиентов, выполнять повторяющиеся и рутинные задачи автоматически, освобождая время операторов для более сложных и важных задач.

Проектирование организационной структуры было выполнено с помощью программного продукта draw.io. Моделирование бизнес-процессов выполнялось с помощью инструментов программы Ramus Education. Проектирование диаграмм, основанное на объектно-ориентированном подходе, было выполнено с помощью программного продукта StarUML. Схемы моделей данных были смоделированы с помощью программного продукта AllFusion ERwin Data Modeler.

Автоматизация контакт центра компании обеспечивает хранение данных в базе данных, с помощью запросов к базе пользователь получает списки необходимых данных о заявках, сотрудниках, клиентах. Основной функцией системы является ведение учета заявок и их обслуживание в срок. Информационная система помогает снизить количество ошибок при получении и обработке заявок, также упрощает учет поступивших заявок от клиентов. На основе этих данных есть возможность вести статистику часто поступающих заявок и проработать меры по устранению проблем.

## Содержание

Введение.....	4
1 Исследование предметной области .....	6
1.1 Анализ деятельности компании провайдера.....	6
1.2 Выделение бизнес-процессов компании .....	8
1.3 Постановка задачи на разработку ИС .....	13
1.4 Анализ существующих разработок .....	16
2 Проектирование информационной системы .....	19
2.1 Выбор технического обеспечения и средств разработки.....	19
2.2 Характеристика входной и выходной информации .....	23
2.3 Описание структуры базы данных .....	24
3 Разработка информационной системы контакт центра .....	30
3.1 Функции системы и описание интерфейса информационной системы	30
3.2 Структурная схема разрабатываемой системы.....	32
3.3 Описание программных модулей .....	36
3.4 Схема работы информационной системы .....	39
3.5 Тестирование информационной системы.....	43
Заключение .....	49
Список используемой литературы .....	50

## Введение

Выпускная квалификационная работа посвящена разработке проекта автоматизации контакт центра на примере компании провайдера.

Услуги компании провайдера имеют высокую актуальность в современном мире коммуникаций и информационных технологий [1].

Провайдеры обеспечивают доступ к интернету, который является основным источником информации в нашей цифровой эпохе. Они предоставляют высокоскоростное и надежное соединение, обеспечивая доступность к информации для пользователей в любое время и в любом месте. Компании провайдеры также заботятся о безопасности данных и обеспечивают защиту от вредоносных атак и кибер угроз.

Провайдеры предоставляют услуги связи, такие как телефония, мобильная связь и электронная почта, которые играют важную роль в наших повседневных коммуникациях. Они обеспечивают возможность общения на расстоянии, деловых переговоров, проведения виртуальных встреч и других форм коммуникации, что становится все более важным в международном и удаленном рабочем окружении.

Они играют важную роль в нашей повседневной жизни, обеспечивая связность и поддерживая развитие цифровой экономики. С каждым днем увеличивается число клиентов, которых необходимо обслуживать и оказывать поддержку в случае прецедентов.

Повышение эффективности обработки обращений клиентов в контакт центре компании может быть достигнуто с помощью внедрения системы типа Help Desk.

Цель работы – разработка проекта автоматизации контакт центра на примере компании провайдера.

Задачи работы:

– изучить деятельность и основные бизнес-процессы компании провайдера, и выявление проблемных мест;

- провести проектирование программного продукта;
- реализовать программный продукт контакт центра, написать руководство пользователя.

Объектом исследования является компания провайдер.

Предметом исследования является разработка проекта автоматизации контакт центра.

В первом разделе проведено исследование деятельности и основных бизнес-процессов компании провайдера. Рассмотрены основные бизнес-процессы и взаимодействие с клиентами компании. Выполнено моделирование деятельности компании с помощью методологии IDEF0.

Во втором разделе работы проведено описание программного продукта, потоки входящей информации, основные документы с которыми работает компания при обращении клиента. Описаны структура программы и инструменты разработки.

В третьем разделе подробно выполнено описание реализации программного продукта, представлен интерфейс пользователя.

Разработанный программный продукт по возможности можно внедрить в компании провайдера для повышения эффективности обработки обращений клиентов.

# **1 Исследование предметной области**

## **1.1 Анализ деятельности компании провайдера**

Предметной областью настоящей выпускной квалификационной работы является деятельность компании провайдера

Компания провайдер – это организация, которая предоставляет определенные услуги связи или доступа к интернету. Они играют важную роль в обеспечении коммуникационных услуг для частных лиц, предприятий и организаций [2].

Провайдеры могут предлагать различные услуги, включая следующие:

- интернет-подключение;
- телефония;
- телевидение;
- облачные услуги;
- услуги связи для бизнеса.

Цель компании провайдера заключается в предоставлении коммуникационных услуг и доступа к информации для своих клиентов.

Основные функции компании провайдера:

- предоставление доступа к интернету;
- телефонные и мобильные услуги;
- телевизионные услуги;
- облачные услуги;
- услуги связи для бизнеса;
- техническая поддержка;
- инновации и развитие.

На рисунке 1 показана организационная структура компании провайдера.



Рисунок 1 – Организационная структура компании провайдера

Для повышения эффективности обслуживания клиентов компании задействовано сразу два отдела, это отдел клиентского обслуживания и техническая поддержка.

Функции отдела клиентского обслуживания заключаются в следующем [3]:

- проводить переговоры с клиентом на этапе информирования;
- оформление и продажа клиенту услуг или продуктов компании;
- ввод данных клиента и регистрация при первом обращении;
- изменением планов или тарифов;
- обработка платежей клиентов и ведение учета счетов;
- консультация по услугам компании [4].

Функции отдела технической поддержки:

- прием заявок от клиентов на устранение прецедентов;
- консультирование клиентов на первоначальном этапе;
- устранение неисправностей со связью или доступом;
- формирование отчетности по проделанной работе.

Оба отдела имеют приоритет при обслуживании клиентов, так как именно от эффективности работы этих отделов зависит прибыль и репутация компании.

## 1.2 Выделение бизнес-процессов компании

Для проведения анализа бизнес-процессов и выявления узких мест необходимо рассмотреть их более подробно. Проведем моделирование основных бизнес-процессов контакт центра (отдела) технической поддержки клиентов компании провайдера с помощью методологии IDEF0 [5].

Выделим основные бизнес-процессы отдела:

- прием и регистрация запросов;
- диагностика и анализ проблемы;
- решение проблемы и поддержка;
- закрытие заявки и обратная связь.

Моделирование будет осуществляться с помощью программного продукта Ramus Education.

Контекстная диаграмма представлена на рисунке 2.

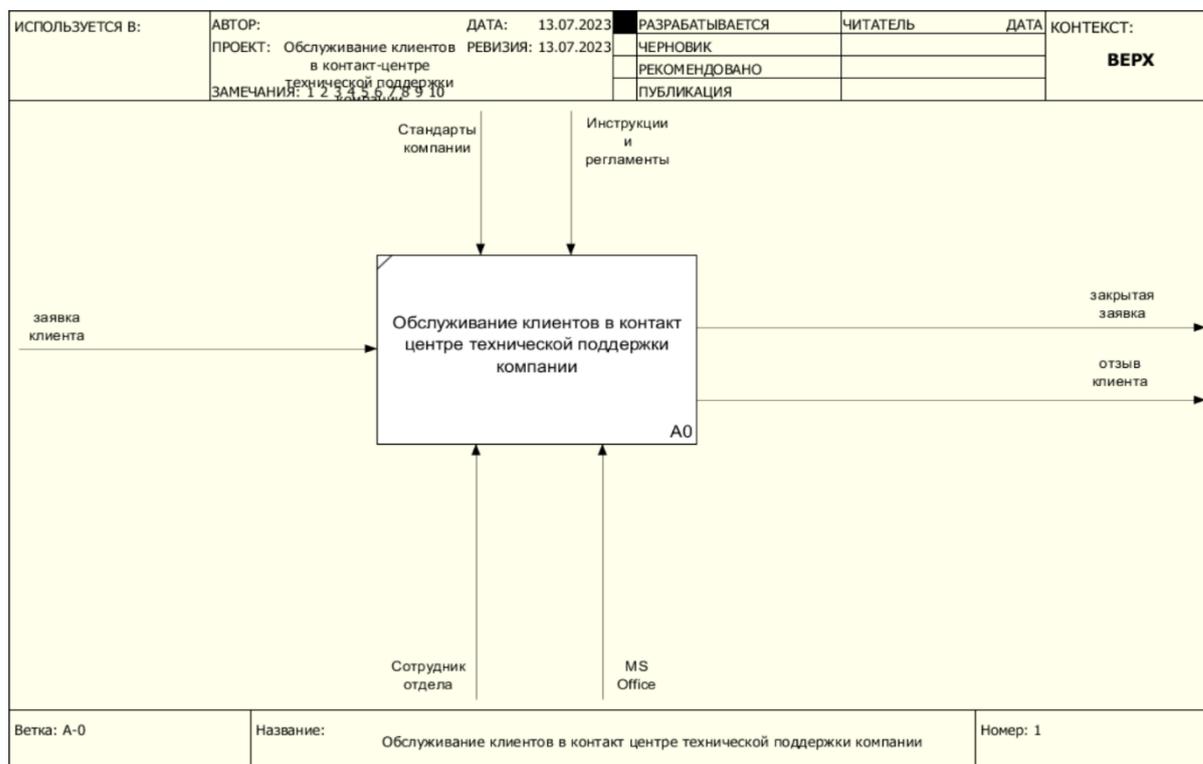


Рисунок 2 – Контекстная диаграмма «как есть» процесса обслуживания клиентов

Обслуживание клиентов в контакт центре разбивается на несколько подпроцессов.

Прием и регистрация заявок. В этом процессе сотрудники отдела технической поддержки принимают заявки и запросы от клиентов, связанные с техническими вопросами, неисправностями или проблемами. Заявки могут поступать через различные каналы коммуникации, такие как телефон, электронная почта или онлайн-чат. Сотрудники регистрируют запросы и получают необходимую информацию для дальнейшего анализа и решения проблемы. [26]

Диагностика и анализ проблемы. После регистрации запроса сотрудники технической поддержки производят диагностику проблемы, задают клиенту уточняющие вопросы и анализируют возможные причины неисправности. Это может включать удаленное подключение к компьютеру клиента, анализ журналов ошибок или другие методы для определения проблемы.

Решение проблемы и поддержка. После анализа проблемы сотрудники технической поддержки предоставляют клиенту решение или поддержку по устранению неисправности. Это может включать предоставление «инструкций по решению проблемы, советы по настройке или конфигурации оборудования, или же в случае необходимости назначение визита специалиста для ремонта или замены оборудования». [18]

Закрытие заявки и обратная связь. По окончании процесса решения проблемы сотрудники технической поддержки закрывают заявку, отмечая, что проблема решена. Важным шагом является сбор обратной связи от клиента, чтобы оценить уровень удовлетворенности предоставленным решением и качеством обслуживания. Это может включать запрос обратной связи по телефону, отправку электронного опроса или другие способы получения мнения клиента. [25]

Сотрудник отдела фиксирует данные на текущий момент в электронные документы Word и Excel, специализированной программы для отслеживания и регистрации заявок от клиентов нет.

Декомпозиция процесса представлена на рисунке 3.

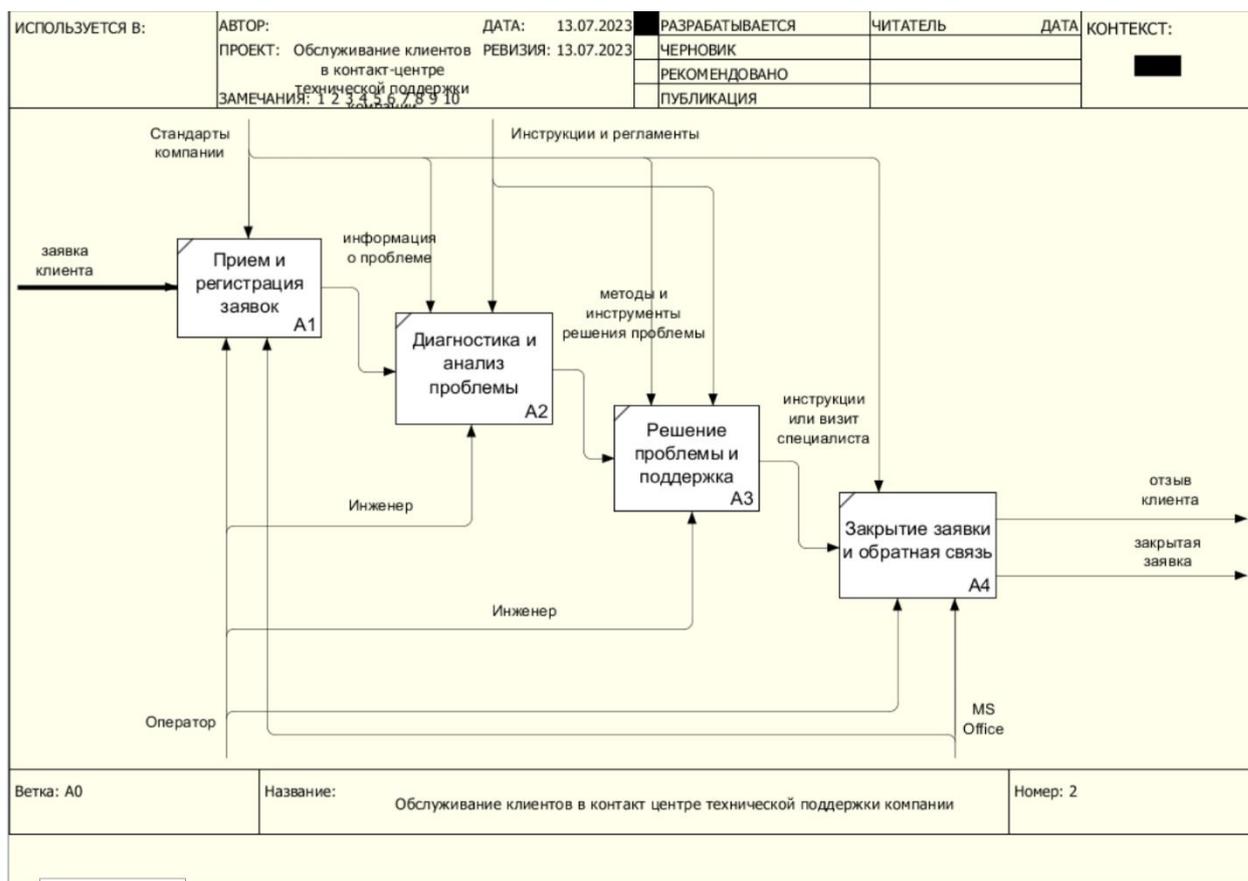


Рисунок 3 – Декомпозиция модели «как есть»

Автоматизация позволяет обрабатывать большой объем запросов и обращений клиентов за более короткое время. Автоматизированные системы могут быстро распознавать, классифицировать и направлять запросы клиентов, что способствует более оперативному решению проблем и сокращению времени ожидания для клиентов.

Автоматизированные системы могут быть настроены для предоставления точной и актуальной информации клиентам. Они могут предлагать решения для распространенных проблем, предоставлять инструкции по устранению неисправностей или давать советы по использованию продуктов и услуг.

Также позволяет предоставлять услуги и поддержку клиентам в любое время суток. Системы автоматического ответа на запросы клиентов могут работать круглосуточно, обеспечивая возможность получения поддержки в любое удобное для клиента время. Это особенно полезно для клиентов, которые нуждаются в помощи вне рабочего времени или в экстренных ситуациях.

Автоматизация контакт центра технической поддержки может помочь снизить затраты на персонал и операционные расходы. Уменьшение необходимости в большом количестве сотрудников и более эффективное использование ресурсов позволяют компании провайдеру сократить расходы и оптимизировать бюджет.

На основе выделенных диаграмм процесса «как есть» можно провести реинжиниринг процесса путем внедрения информационной системы по типу Help Desk.

На рисунке 4 показан процесс «как должно быть».

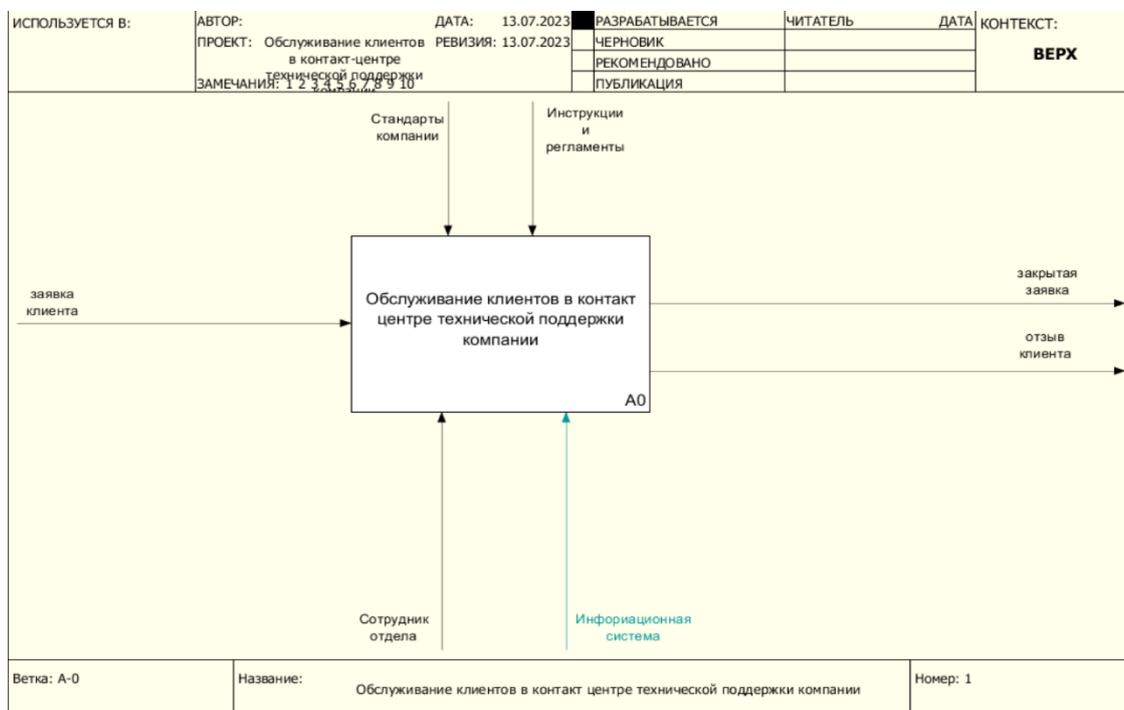


Рисунок 4 – Контекстная диаграмма «как должно быть» обслуживания клиентов

На рисунке 5 показана декомпозиция процесса «как должно быть».

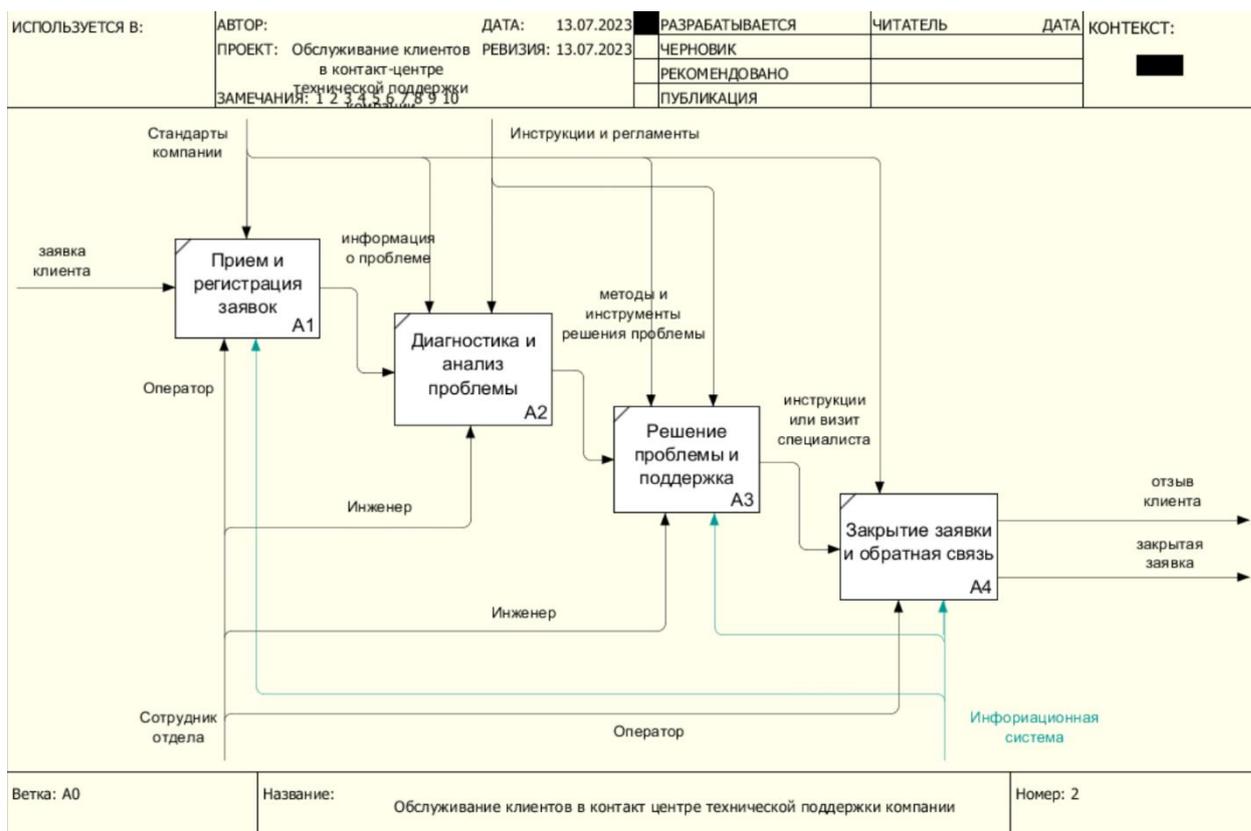


Рисунок 5 – Декомпозиция модели «как должно быть»

Опишем подробно каждый процесс как он будет осуществляться после автоматизации контакт центра.

При использовании системы Help Desk, клиент может использовать специальный портал или интерфейс для отправки заявки. Система Help Desk автоматически регистрирует поступающую заявку, присваивая ей уникальный идентификатор или номер. Заявка становится видимой в системе для дальнейшей обработки.

В системе Help Desk заявка может быть классифицирована в соответствии с ее типом, категорией или приоритетом. Это может помочь определить, какой отдел или специалист должен заниматься решением данной заявки, а также установить сроки выполнения. Система отправляет уведомления соответствующим сотрудникам о новой заявке, включая

информацию о ее содержании, приоритете и других важных деталях. Это помогает сотрудникам быть в курсе новых заявок и начать работу над ними.

При диагностике и решении проблемы сотрудник, получивший заявку, анализирует ее содержание, проводит диагностику проблемы и предпринимает необходимые шаги для ее решения. Это может включать коммуникацию с клиентом для уточнения информации или запроса дополнительных данных.

В ходе работы над заявкой сотрудник системы Help Desk обновляет ее статус, отражая текущий прогресс и состояние работы. Он также может добавлять комментарии, описывающие выполненные действия, результаты диагностики или предпринятые меры.

По завершении работы над заявкой сотрудник системы Help Desk закрывает ее, отмечая, что проблема решена. Клиенту может быть отправлено уведомление о решении заявки и запрошена обратная связь о качестве обслуживания или удовлетворенности полученным решением.

Таким образом система обеспечит структурированный и отслеживаемый процесс приема и регистрации заявок, что способствует более эффективной работе отдела технической поддержки клиентов и повышению качества обслуживания.

### **1.3 Постановка задачи на разработку ИС**

Цель: разработать и внедрить систему Help Desk для улучшения процесса приема, регистрации и обработки заявок клиентов в отделе технической поддержки компании провайдера.

Требования:

- разработать механизм для приема заявок от клиентов через различные каналы связи, такие как электронная почта, веб-форма, онлайн-чат или телефон;
- создать уникальный идентификатор или номер для каждой заявки и

автоматически регистрировать ее в системе;

- предусмотреть возможность классификации и приоритизации заявок на основе их типа, категории и важности;

- разработать механизм автоматической маршрутизации заявок к соответствующим отделам или специалистам, исходя из их классификации и приоритета;

- обеспечить возможность назначения конкретного сотрудника или группы для работы над каждой заявкой;

- предусмотреть механизм назначения сроков выполнения заявок и отслеживания их выполнения;

- создать механизм для коммуникации между сотрудниками и клиентами в рамках заявок;

- обеспечить возможность добавления комментариев и обновления статуса заявок сотрудниками системы Help Desk;

- учесть возможность будущего расширения и изменения функциональности системы в соответствии с потребностями отдела технической поддержки и развитием бизнеса.

Требования к ИС должны быть сформулированы таким образом, чтобы обеспечить эффективное и качественное обслуживание клиентов.

Некоторые из них:

- масштабируемость;
- интеграция с CRM;
- многоформатная поддержка;
- аналитика и отчетность;
- автоматическое распределение запросов;
- резервное копирование и отказоустойчивость.

Ожидаемые результаты от разработки системы:

- улучшение клиентского опыта и удовлетворенности клиентов;
- сокращение времени реакции и улучшение времени решения проблем;
- более точное и полное отслеживание истории заявок;

- повышение прозрачности и коммуникации между сотрудниками и клиентами;

- сокращение усилий и ресурсов, затрачиваемых на обработку и управление заявками.

- улучшение управления приоритетами и соблюдение сроков выполнения задач.

Постоянными пользователями информационной системы будут сотрудники контакт центра.

На рисунке 6 представлена диаграмма вариантов использования [6].

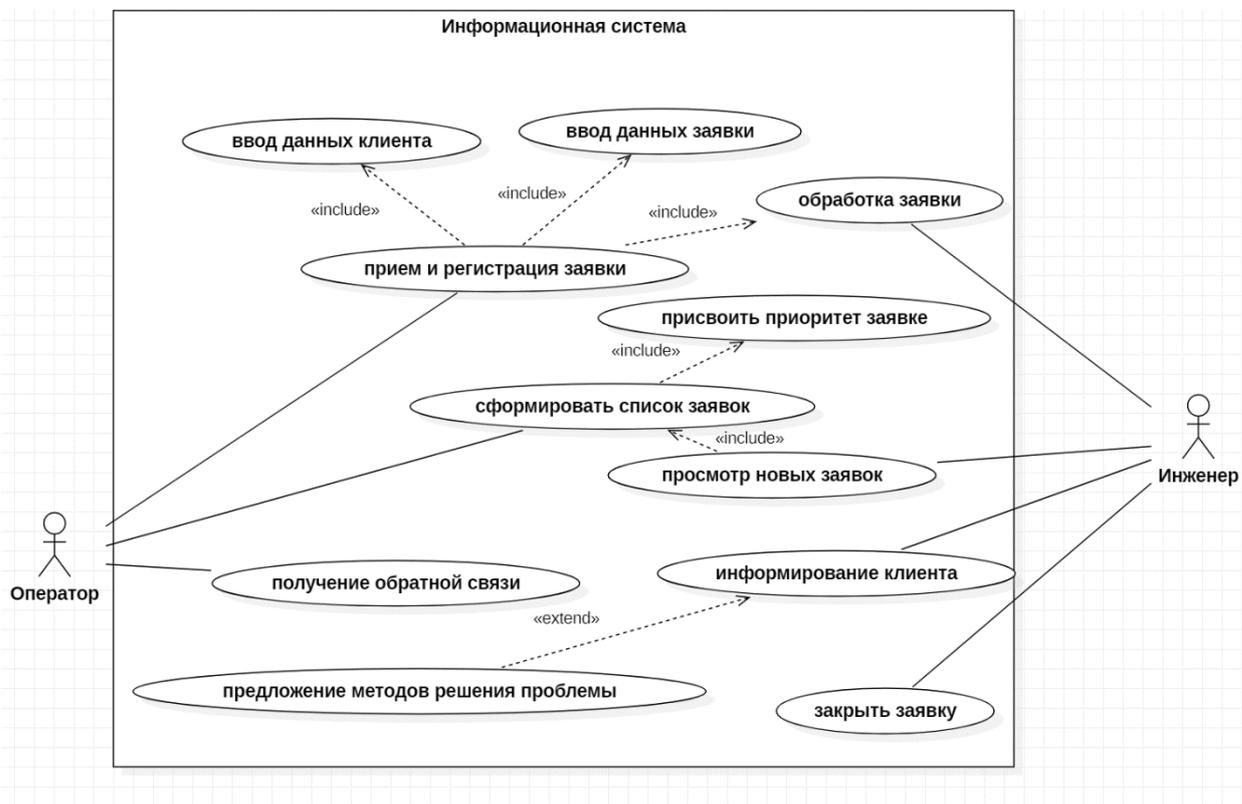


Рисунок 6 – Диаграмма вариантов использования

Основные функции информационной системы:

- учет заявок;
- регистрация клиентов и заявок;
- отслеживание выполнения заявки;

- формирование решений для устранения проблем.

Информационную систему будут использовать на протяжении всего рабочего дня.

#### **1.4 Анализ существующих разработок**

Рассмотрим некоторые аналогичные разработки, которые выполняют подобные функции.

Автоматизация деятельности интернет-провайдера на базе ПП «1С: Управление торговлей 8» [7].

Автоматизированы следующие функции:

- анализ кадрового состава;
- взаиморасчеты с покупателями;
- взаиморасчеты с поставщиками;
- кадровый учет;
- оформление заказов покупателей;
- оформление заказов поставщикам;
- планирование закупок;
- управление отношениями с клиентами (CRM);
- управление отношениями с поставщиками;
- управление складскими запасами;
- учет прихода ТМЦ;
- учет продаж ТМЦ.

«UserSide» — это отраслевое решение для операторов связи, которое разрабатывается с 2007 года. Веб-ориентированный программный комплекс, что позволяет упрощать, ускорять и эффективно управлять множеством процессов в компании. UserSide предлагает работу с сооружениями и линиями связи, оборудованием, взаимодействием с биллингом, учет клиентов, персонал, склад и многое другое. UserSide перезапустит механизм работы вашей компании, заставив функционировать её как единое целое.

Отраслевое решение именно для телекома, а не универсальный конструктор. ERP-система для операторов связи любого уровня с поддержкой большинства сфер его деятельности. Огромные возможности. Для тех, кто хочет упростить работу себе и своим сотрудникам. Решить проблемы по учёту инфраструктуры компании.

Преимущества:

- связь трёх измерений: тех.учёт – клиенты – биллинг;
- даёт возможность руководителю самого различного ранга контролировать эти процессы из любой точки мира;
- безопасность и конфиденциальность данных;
- помогает построению правильной стратегии развития компании и улучшение взаимоотношений с абонентами;
- разделение прав доступа сотрудников по географии и функционально;
- полный контроль сети, никакого хаоса, а чёткое понимание ситуации.

MICROSOFT DYNAMICS [8].

Microsoft Dynamics — это «семейство программных продуктов для управления отношениями с клиентами (CRM) и планирования ресурсов предприятия (ERP), разрабатываемых корпорацией Microsoft. Эти продукты предоставляют комплексные решения для автоматизации бизнес-процессов, повышения эффективности и улучшения взаимодействия с клиентами» [8].

Достоинства:

- интеграция с экосистемой Microsoft;
- гибкость;
- облачные решения;
- большое сообщество и поддержка.

Недостатки:

- сложность;
- системные требования;
- сложность пользовательского интерфейса.

Из рассмотренных выше аналогичных разработок ни одна из информационных систем и CRM-систем не подходит по требованиям заказчика [9].

Итоги сравнения программных продуктов показаны в таблице 1.

Таблица 1 – Сравнительная таблица функциональности систем

Показатель сравнения	1С: Управление торговлей 8	UserSide	Microsoft Dynamics
Простота интерфейса	-	+	+
Возможность ведения учета клиентов	+	+	+
Возможность ведения учета заявок	+	+	-
Доступность	-	-	-
Защита личных данных	+	-	-
Возможность расширения модуля	+	+	+
Наличие мобильной версии	-	-	-

На основе проведенного анализа аналогичных программ было принято решение о создании собственной системы, в которой будут включены все необходимые функции, согласно требованиям, контакт центра компании.

#### Выводы по первому разделу

В первом разделе проведено исследование деятельности компании провайдера, в частности отделов по работе и обслуживанию клиентов.

На основе исследований проведено моделирование бизнес-процессов.

Созданные диаграммы позволили выявить узкие места в процессах и разработать решение по автоматизации контакт центра компании с помощью внедрения информационной системы. Выполнена постановка задачи на разработку информационной системы.

Для подтверждения актуальности собственной разработки были рассмотрены несколько аналогичных программных продуктов, которые позволили убедиться в правильности принятого решения.

## 2 Проектирование информационной системы

### 2.1 Выбор технического обеспечения и средств разработки

Для хранения данных системы можно рассмотреть следующие СУБД (системы управления базами данных):

MySQL является одной из самых популярных открытых реляционных СУБД. Она обладает широкой поддержкой и активным сообществом разработчиков. MySQL предлагает высокую производительность, надежность и масштабируемость. Она «поддерживает широкий набор функций, таких как транзакции, индексы, полнотекстовый поиск и репликацию данных. MySQL также обладает хорошей совместимостью с различными программными языками и фреймворками разработки» [10].

PostgreSQL является мощной и расширяемой объектно-реляционной СУБД с открытым исходным кодом. Она предлагает множество продвинутых функций, включая поддержку сложных запросов, хранимых процедур и триггеров, полнотекстовый поиск, геоданные, репликацию и масштабирование. PostgreSQL также обладает высокой надежностью, поддержкой транзакций и многоуровневой безопасностью. Она широко используется в «критически важных системах, где требуется высокая производительность и целостность данных» [11].

MongoDB является документно-ориентированной NoSQL СУБД, которая хранит данные в формате JSON-подобных документов. Она отличается гибкостью и масштабируемостью, особенно при работе с большими объемами неструктурированных данных. MongoDB предлагает высокую производительность и горизонтальное масштабирование, поддерживая функции репликации и шардинга. Она также обладает «гибким языком запросов, индексацией и возможностью работать с геоданными» [12].

SQL Server — это реляционная система управления базами данных (СУБД), разработанная компанией Microsoft. Она предлагает широкий набор

функциональности и инструментов для хранения, управления и обработки данных.

SQL Server способен обрабатывать большие объемы данных и обеспечивать высокую производительность даже при интенсивных рабочих нагрузках. Он поддерживает параллельную обработку запросов, индексирование данных и оптимизацию запросов для улучшения производительности.

Обладает многоуровневой системой безопасности, позволяющей управлять доступом к данным и обеспечивать конфиденциальность, целостность и доступность информации. Она поддерживает аутентификацию, авторизацию и шифрование данных для защиты от несанкционированного доступа.

SQL Server предлагает функции резервного копирования, восстановления и репликации данных для обеспечения высокой доступности и отказоустойчивости. Это включает функциональность автоматического переключения на резервные серверы при отказе основного сервера. Интегрируется с другими продуктами и инструментами Microsoft, такими как Visual Studio, Azure и SharePoint. Он также поддерживает язык программирования T-SQL (Transact-SQL) для написания хранимых процедур, триггеров и пользовательских функций, что обеспечивает расширяемость функциональности.

MongoDB обычно используется с языками, которые лучше поддерживают документоориентированные данные, такие как JavaScript (Node.js), Python или Ruby. PostgreSQL имеет отличную поддержку для C# и предоставляет более стабильные и обширные драйверы, позволяющие эффективно работать с базой данных из приложений на C#, однако эта СУБД является сложной для обучения.

Представим СУБД в сравнительной таблице 2.

Таблица 2 – Результаты сравнения баз данных

Основные отличия	MongoDB	SQL Server	PostgreSQL
Доступность	-	+	+
Хранение большого объема данных	+	+	+
Скорость обработки операций с большими данными	+	+	-
Защита от несанкционированного доступа	+	+	+
Простота использования	-	+	-

Для написания системы нам нужно выбрать оптимальный язык программирования, чтобы реализовать весь функционал. В качестве языка программирования будет использован язык C#.

Язык программирования C# имеет несколько достоинств, которые делают его популярным и предпочтительным выбором для разработки различных типов приложений:

- простота и легкость изучения;
- объектно-ориентированное программирование;
- мощная интеграция с платформой .NET;
- безопасность типов;
- многопоточность и асинхронное программирование;
- инструменты разработки и поддержка;
- платформенная независимость.

Для реализации приложения выбрана интегрированная среда разработки программного обеспечения Visual Studio 2019.

Диаграмма размещения [13] показана на рисунке 7, где показаны основные узлы системы, взаимодействующие между собой в процессе работы.

Требования к программно-техническому обеспечению:

- интеграция с коммуникационными каналами;
- управление запросами клиентов;
- интеграция с базой данных;
- аналитика и отчетность;
- масштабируемость;

- автоматизация процессов;
- безопасность и конфиденциальность.

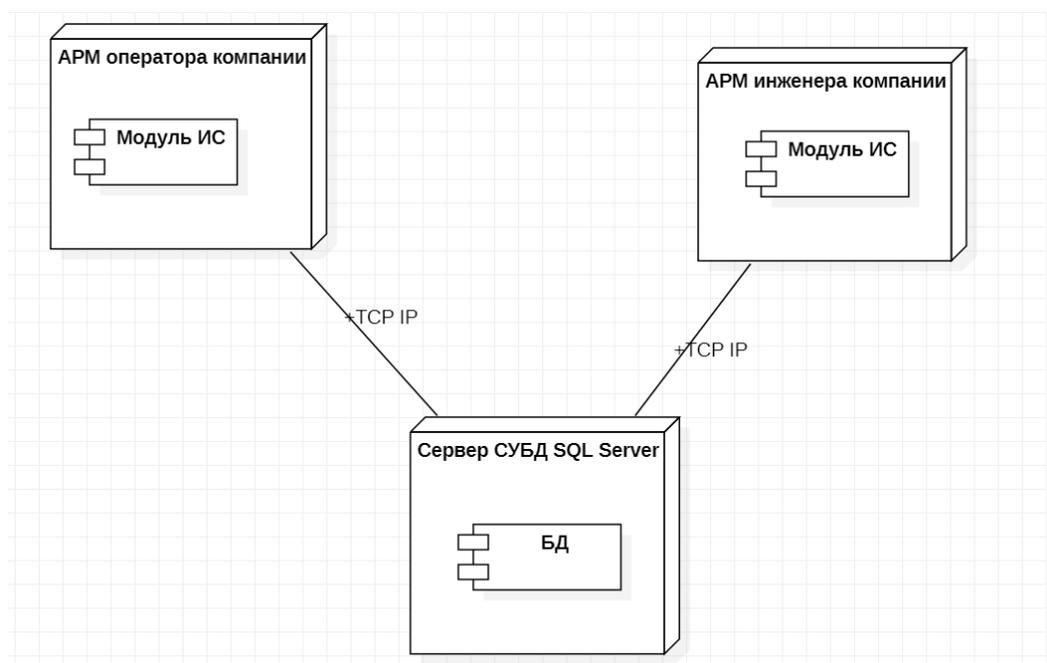


Рисунок 7 – Диаграмма размещения

#### Аппаратное обеспечение:

- Процессор: Intel Pentium G4400 1 шт.
- Чипсет: Intel® H110
- ОЗУ: 4GB (1x4GB) Unbuffered DIMM DDR4 (частота шины 2133MHz)

Kingston, объем памяти (макс.) 32GB, слотов памяти 2шт.

- Диски: 250GB (1x500GB) SATA 6Gb/s 3.5" Seagate;
- Отсеков: 3.5" 3шт., 5.25" 2шт;
- Сеть: на мат. плате: 1Gb/s 1шт;
- Блок питания: 1 шт. 450W.

#### Программное обеспечение:

- операционная система – Windows 7 корп.;
- методы защиты информации – антивирус NOD32;
- пакет офисных программ Microsoft Office [14].

После определения всех необходимых требований и определения аппаратного обеспечения можно приступать к проведению анализа документации и проектирования моделей базы данных.

## **2.2 Характеристика входной и выходной информации**

Входящие документы в контакт центре провайдерской компании играют важную роль в обеспечении эффективной работы и обслуживания клиентов. Эти документы представляют собой информацию и запросы, поступающие от клиентов, партнеров и других сторон, и могут иметь различные форматы, включая письма, электронные сообщения, факсы, заявки и т. д.

Входные документы:

- заявления на подключение или отключение услуг;
- жалобы на качество услуг, сбои в работе сети или другие проблемы;
- запросы на техническую поддержку;
- запросы о переносе телефонных номеров от других провайдеров;
- отзывы о работе контакт центра.

Исходящие документы в контексте провайдерской компании представляют собой информацию и сообщения, которые отправляются от компании клиентам, партнерам, регулирующим органам и другим заинтересованным сторонам.

Исходящие документы:

- счета и счета-фактуры;
- уведомления о изменениях;
- подтверждения и уведомления о состоянии счета;
- письма технической поддержки;
- заявления и договоры.

Для эффективного управления исходящими документами провайдерские компании могут использовать специализированные системы управления контентом, автоматизированные процессы печати и доставки, а

также механизмы электронной почты и электронных уведомлений. Важно, чтобы исходящие документы были правильно сформулированы, точны и своевременно доставлены получателям, чтобы обеспечить качественное обслуживание клиентов и соблюдение законодательных требований.

В таблице 3 представлены реквизиты входных документов.

Таблица 3 – Характеристика входной информации

Входные данные	Объем в год	Источники получения
Данные о клиенте	100000-150000	Личные данные клиента
Данные о заявке	2000-3000	Оператор
Список неисправностей	10-20	Каталог проблемных ситуаций

Исходящая информация в виде документов представлена в таблице 4.

Таблица 4 – Описание выходных документов

Документ	Форма представления	Назначение
Отзывы клиентов	Бумажный, электронный	Оператор
Выполненные заявки	Бумажный, электронный	Оператор

Исходящие документы оператор или инженер могут менять в соответствии с входящей формой документов.

### 2.3 Описание структуры базы данных

На основе проведенного анализа процесса обслуживания клиентов компании провайдера в контакт центре по техническим причинам были выделены основные объекты, которые будут хранить данные в базе.

Для учета клиентов потребуются сведения, которые будут храниться в таблице «Клиенты» и содержать информацию о клиенте.

Таблица «Клиенты»:

- ID клиента;
- имя клиента;
- контактная информация клиента.

Чтобы клиент мог подать заявку в случае прецедента, должны быть объекты «Заявки» и «Статус заявки».

Таблица «Заявки»:

- ID заявки;
- клиент;
- тема заявки;
- описание заявки;
- статус заявки;
- приоритет заявки;
- дата создания заявки;
- дата обновления заявки.

Таблица «Статусы заявок»:

- ID статуса;
- название статуса.

Для осуществления выполнения заявки должна быть таблица с информацией о сотрудниках.

Таблица «Сотрудники»:

- ID сотрудника;
- фамилия сотрудника;
- имя сотрудника;
- отчество сотрудника;
- должность сотрудника;
- контактная информация сотрудника.

Формирование комментариев должно выноситься в отдельную таблицу.

Таблица «Комментарии»:

- ID комментария;
- Заявка;

- автор комментария;
- текст комментария;
- дата комментария.

Приоритеты заявок должны быть вынесены в отдельную таблицу для обеспечения соблюдения правила 3НФ.

Таблица «Приоритеты заявок»:

- ID приоритета;
- название приоритета.

На рисунке 8 показана ER-модель [14]. «ER-модель данных является концептуальной моделью для описания структуры данных в информационных системах. Она представляет собой графическую нотацию, которая используется для визуального представления сущностей (entities), их атрибутов (attributes) и отношений (relationships) между ними» [14].

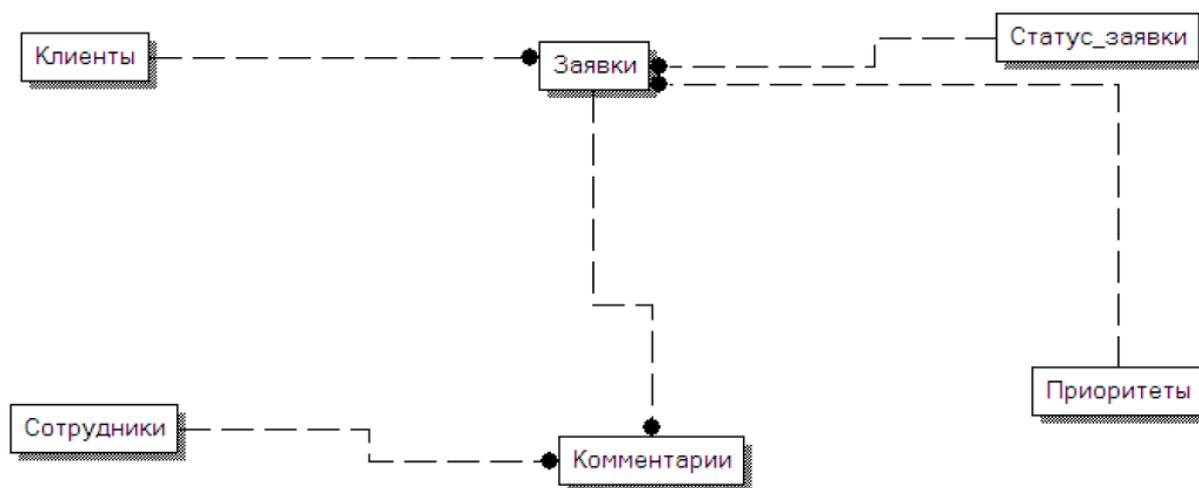


Рисунок 8 – ER-модель данных

ER-модель позволяет визуализировать структуру данных, определить иерархии, связи и зависимости между сущностями. Это позволяет проектировщикам баз данных лучше понимать требования к системе, анализировать поток данных и разрабатывать более эффективные и оптимизированные структуры баз данных.

ER-модель является основой для дальнейшего проектирования физической базы данных, где концептуальная модель переводится в специфический синтаксис и схему базы данных конкретной СУБД, такой как SQL Server, MySQL или Oracle.

На рисунке 9 представлена даталогическая модель БД [15].

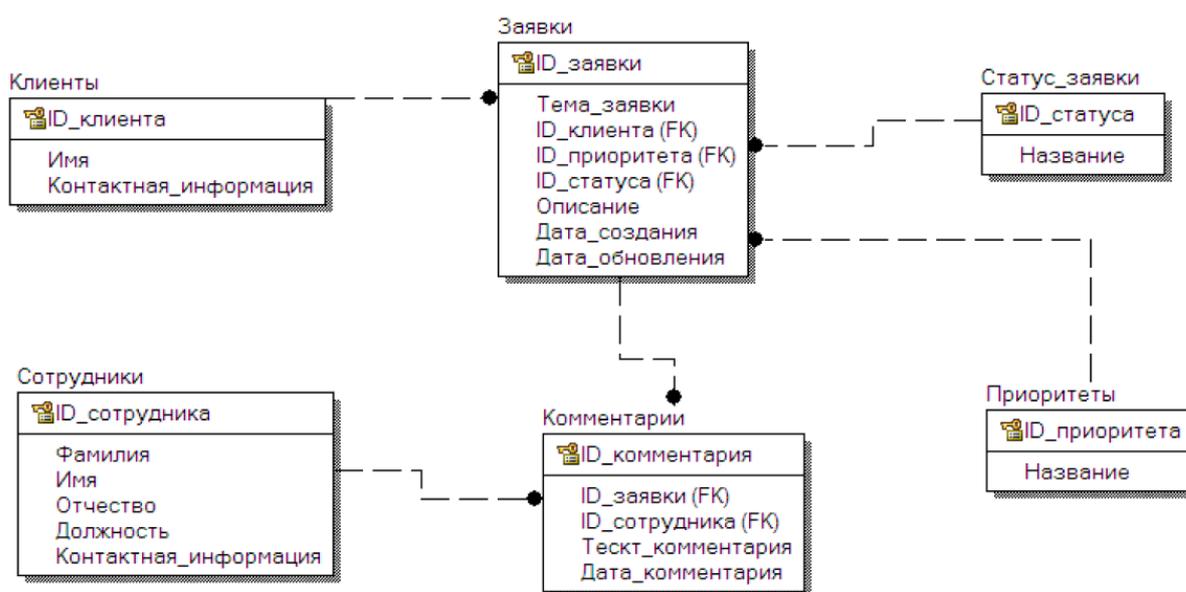


Рисунок 9 - Логическая модель данных

Даталогическая модель базы данных (DBDL) — это «конкретная реализация ER-модели данных, которая определяет структуру базы данных на уровне таблиц, полей, ключей и связей. Даталогическая модель служит основой для создания физической базы данных и определяет точные правила и форматы для хранения данных» [15].

Даталогическая модель базы данных обычно представляется в виде диаграммы базы данных, которая визуально отображает таблицы, их столбцы, ключи и связи. Для создания даталогической модели используются специальные инструменты для моделирования баз данных, такие как ER-диаграммы или CASE-среды разработки.

Физическая модель базы данных для СУБД SQL Server представлена на рисунке 10.

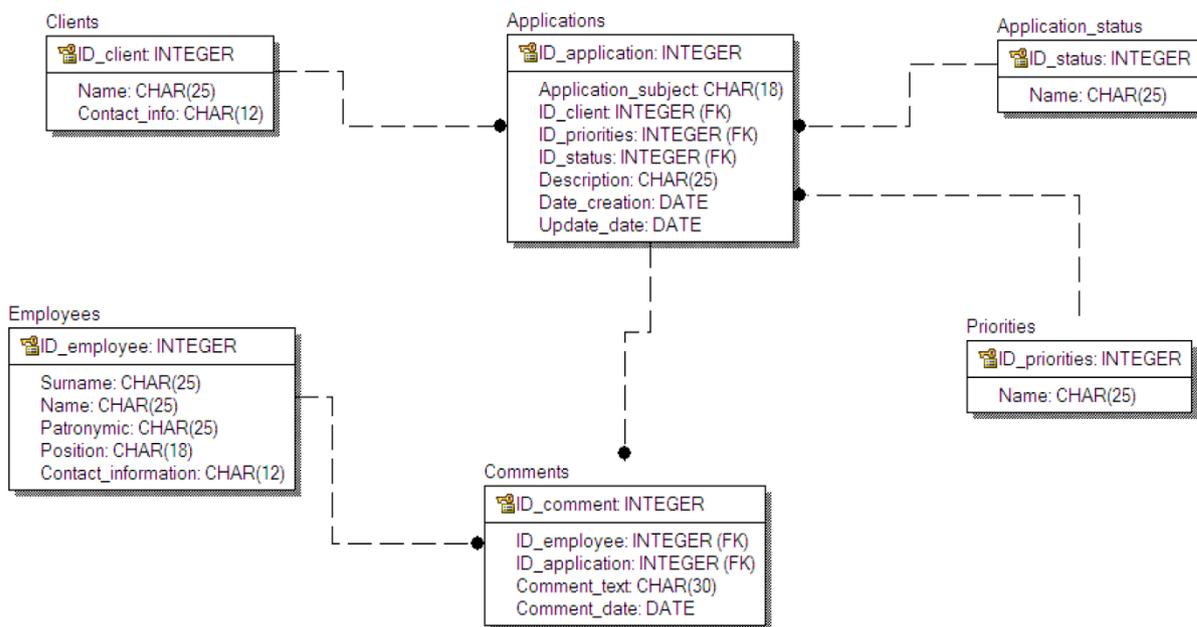


Рисунок 10 - Физическая модель данных

Физическая модель данных — это реализация даталогической модели данных на конкретной СУБД (системе управления базами данных) и определение способа хранения данных на физическом уровне. Физическая модель определяет структуру таблиц, индексы, ограничения целостности, оптимизацию запросов и другие аспекты, связанные с физическим хранением данных.

Физическая модель данных разрабатывается с учетом требований производительности, доступности, безопасности и других факторов, связанных с конкретной СУБД и окружением, в котором будет работать система. Она служит основой для создания и настройки физической базы данных и обеспечивает эффективное хранение и обработку данных в соответствии с потребностями приложения.

Логическая, даталогическая и физическая модели были разработаны с использованием программы AllFusion Data Modeler 7.3.

Использование AllFusion Data Modeler 7.3 для моделирования данных в проекте разработки базы данных предлагает несколько преимуществ:

- визуальное моделирование;
- улучшенная продуктивность;
- обеспечение целостности данных;
- анализ и оптимизация;
- документирование и совместная работа;
- интеграция с другими инструментами.

Выводы по второму разделу

Второй раздел работы включает выполнение проектирования системы контакт центра. В ходе проектирования были рассмотрены входящие и исходящие документы, которые поступают и формируются в течение рабочего дня в контакт центре. Во втором разделе также были рассмотрены и выбраны инструменты для хранения данных, разработана структура базы данных и описаны основные сущности, которые будут содержать необходимую информацию для осуществления процессов контакт-центра компании. Было проведено моделирование моделей базы данных на разных уровнях и представлены наглядные диаграммы «сущность-связь», логической и физической моделей.

### **3 Разработка информационной системы контакт центра**

#### **3.1 Функции системы и описание интерфейса информационной системы**

На основе требований к разработке информационной системы контакт центра, функциональность системы будет выглядеть таким образом:

- ведение данных о клиентах;
- ввод и хранение данных о заявках;
- ведение информации о статусе заявке;
- ввод данных о приоритете заявки.

На диаграмме представлены основные функции:

- работа с клиентами;
- работа с заявками;
- работа с приоритетами;
- работа со статусом заявки.

Основные функции имеют разветвление на несколько подфункций, которые позволят пользователю (сотрудник контакт центра) управлять данными.

На рисунке 11 показана диаграмма потоков данных.

Оператор контакт центра работает с клиентами выполняет первоначальные функции по приему заявок и регистрации данных клиента и заявки. Также занимается отправкой заявок дальше инженерам отдела, которые непосредственно и обрабатывают заявку с учетом приоритета и сложности.

Инженер непосредственно занимается обработкой заявок, диагностикой и устранением неполадок либо консультированием клиента.

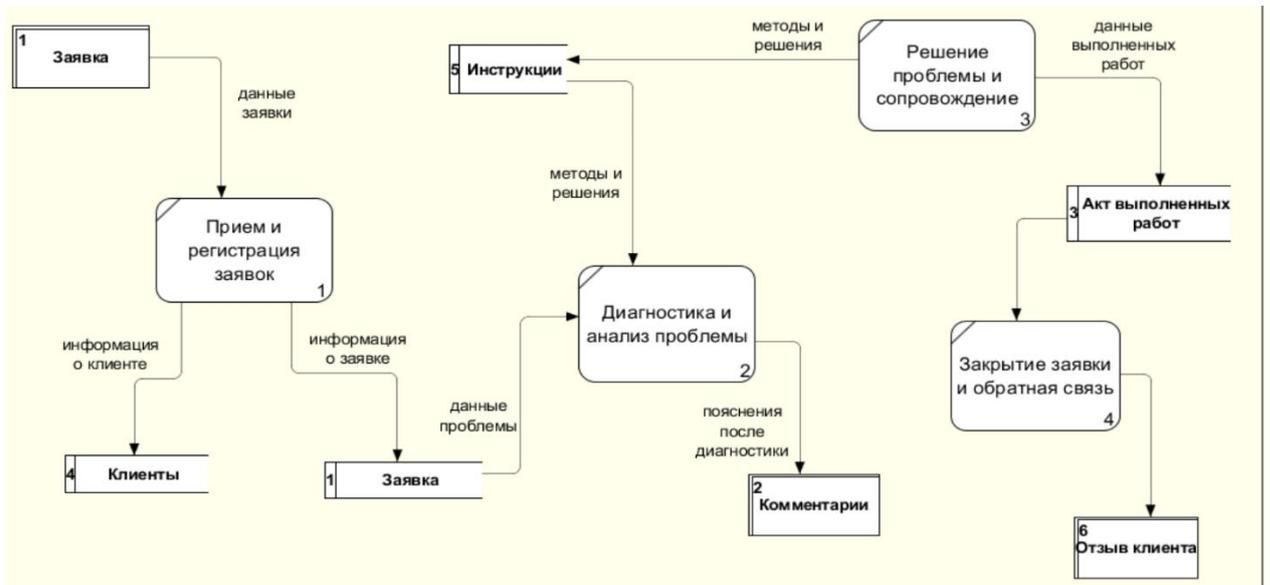


Рисунок 11 – Диаграмма потоков данных

Построим диаграмму [15] последовательности одного из процессов, возьмем «Прием и регистрация заявки», на рисунке 12 показана диаграмма последовательности этого процесса.

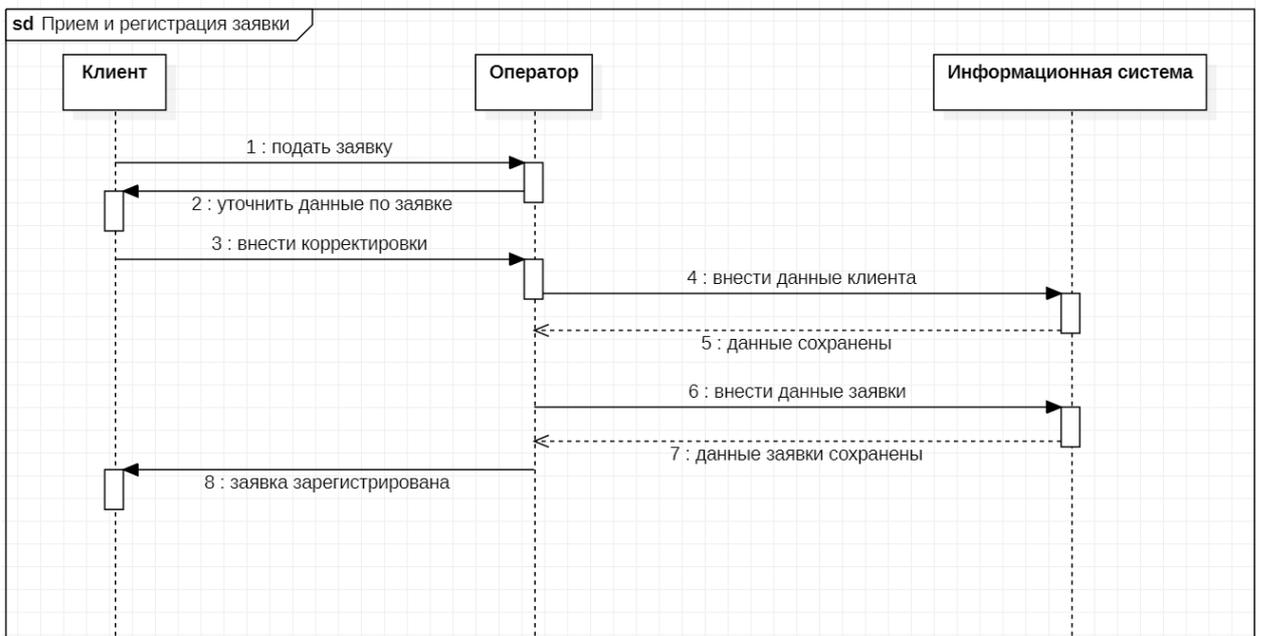


Рисунок 12 – Описание действий оператора при приемке и регистрации заявки

«Проектирование интерфейса пользователя подразумевает навигационное меню в виде кнопок, которое облегчит оператору и инженеру поиск информации и ввод данных в формы.

### **3.2 Структурная схема разрабатываемой системы**

Для проектирования и разработки также требуется выбрать методологию, по которой команда разработчиков или один программист будет выстраивать свою работу.

DevOps — это методология разработки программного обеспечения, объединяющая разработку (Development) и операционное обслуживание (Operations). Она направлена «на улучшение сотрудничества, коммуникации и интеграции между разработчиками программного обеспечения и операционными специалистами с целью достижения более быстрой и эффективной поставки программного продукта» [14].

Методология DevOps включает в себя следующие основные принципы:

- совместная ответственность;
- автоматизация;
- непрерывная поставка;
- обратная связь;
- гибкость и масштабируемость.

Контакт центры обычно сталкиваются с частыми изменениями требований и необходимостью оперативного внедрения новых функций и исправлений. «Методология DevOps обеспечивает непрерывную поставку программного обеспечения, позволяя быстро и безопасно внедрять изменения в ИС контакт центра» [15].

Автоматизированные процессы развертывания, тестирования и мониторинга позволяют сократить время доставки изменений, уменьшить вероятность ошибок и повысить качество ИС контакт центра.

Методология DevOps «способствует формированию культуры совместной ответственности, где разработчики, операционные специалисты и команда поддержки контакт центра работают вместе для достижения общих целей. Это способствует повышению эффективности команды и снижению риска возникновения проблем в ИС контакт центра». [16].

Пользователь должен иметь возможность зарегистрироваться в системе контакт центра. Делается это через веб-интерфейс приложения.

При регистрации пользователь предоставляет свои персональные данные, такие как имя, адрес электронной почты, пароль для будущей авторизации.

Рассмотрим более подробно процесс регистрации нового клиента в автоматизированной информационной системе, с помощью построения алгоритма и представления в виде блок-схемы процесса.

Последовательность функции «Регистрация нового клиента»:

- открыть форму регистрации клиента;
- заполнить все поля данными о клиенте;
- нажать кнопку «ОК»;
- при пустом поле выдать сообщение (Поле не заполнено!);
- после заполнения всех полей добавлять запись в БД «Клиенты»;
- после окончания ввода нажать «Сохранить».

Когда данные будут сохранены, система выведет пользователю сообщение указывающее, что данные успешно сохранены, после чего пользователь может закрывать форму регистрации клиента. Все данные будут храниться в базе данных, откуда будет возможность получить информацию о клиенте, который уже обращался ранее в контакт центр компании.

Блок-схема алгоритма функции «Регистрация нового клиента» представлен на рисунке 13.

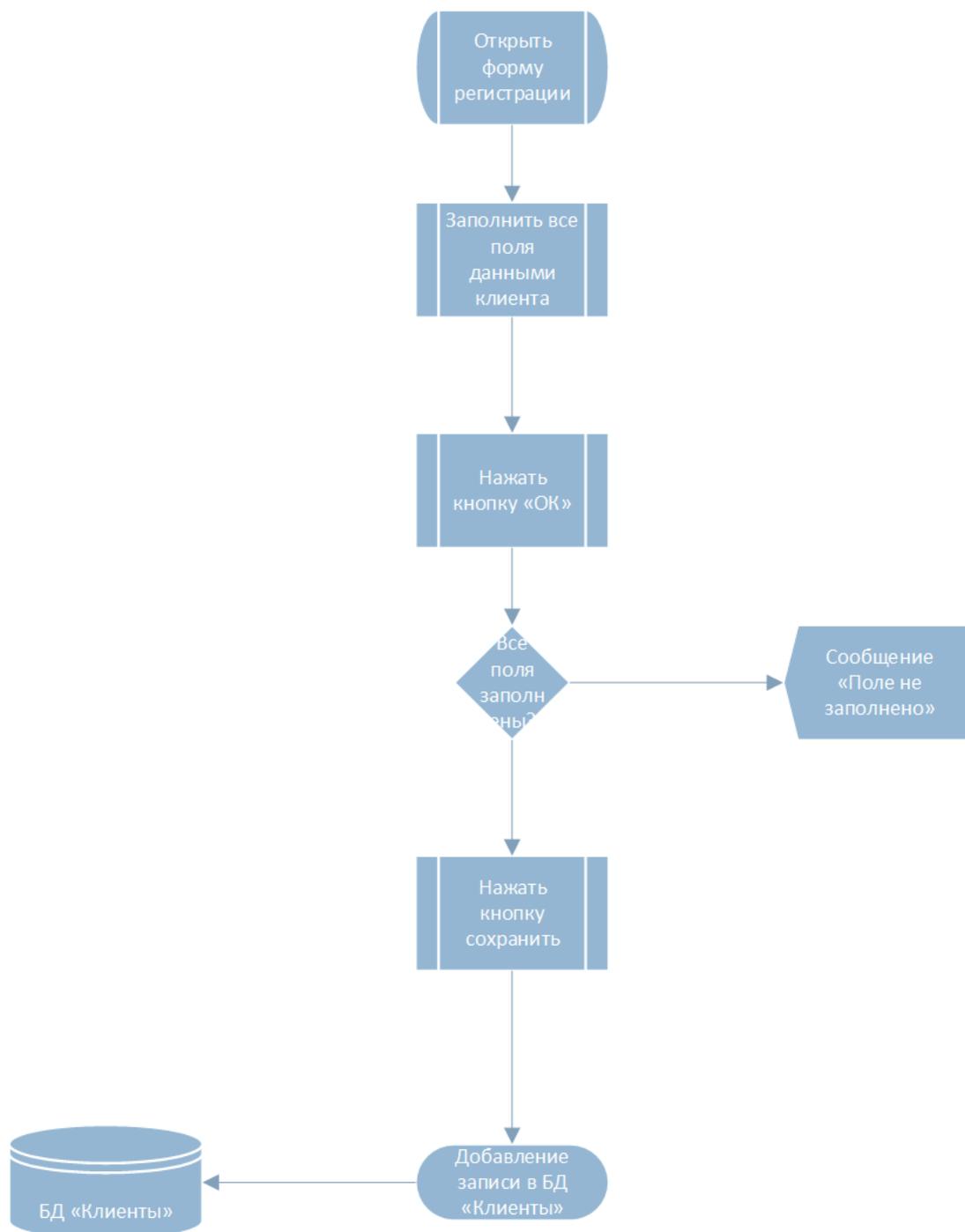


Рисунок 13 – Блок-схема алгоритма «Регистрация нового клиента»

На рисунке 14 показана блок-схема процесса обработки заявок в контакт центре компании.

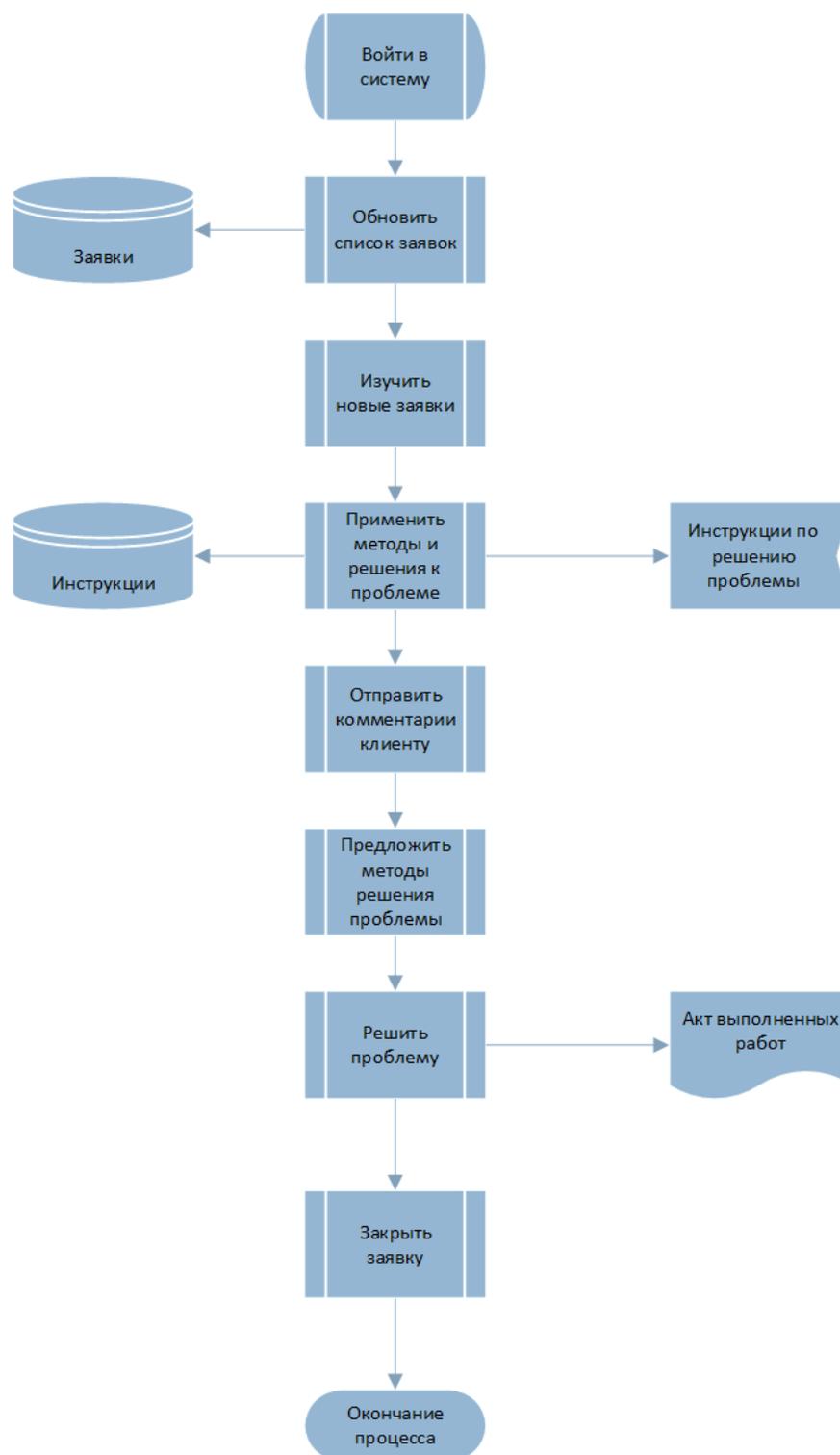


Рисунок 14 – Блок-схема алгоритма «Обработки заявок в контакт центре»

Перед началом реализации информационной системы были определены основные требования и цели ИС контакт центра. Осуществлено

взаимодействие с заинтересованными сторонами, включая менеджеров контакт центра, операторов и клиентов, чтобы полностью понять их потребности. Проанализированы существующие процессы, системы и данные, чтобы определить области, требующие улучшения и автоматизации.

На основе требований разработан список функциональности, которую должна предоставлять ИС контакт центра. Разработана архитектура системы, определены компоненты, модули, база данных и взаимодействие с другими системами.

На данном этапе начинается активная стадия разработки ИС контакт центра. Выбрана методология разработки, чтобы разделить работу на итерации или спринты [17].

### **3.3 Описание программных модулей**

Нам потребуется разработка классических приложений .Net, выбираем и устанавливаем пакет инструментов.

Система будет написана на основе шаблона MVVM [19].

MVVM (Model-View-ViewModel) — это шаблон проектирования, который используется для разделения логики представления (View) и бизнес-логики (Model) в приложении, а также для обеспечения связи и взаимодействия между ними с помощью промежуточного компонента, называемого ViewModel.

Общее описание каждой части MVVM:

Модель (Model). Модель представляет собой бизнес-данные, логику и операции, связанные с обработкой данных. Она отвечает за получение, хранение и обработку данных приложения. Модель может включать классы, структуры, сервисы или другие компоненты, которые представляют бизнес-логику и данные.

Представление (View). Представление отвечает за отображение пользовательского интерфейса и взаимодействие с пользователем. Это может

быть веб-страница, окно приложения, форма или другой компонент, с которым пользователь взаимодействует. Представление отвечает за отображение данных из модели и отправку пользовательских действий обратно во ViewModel.

ViewModel. ViewModel служит промежуточным слоем между представлением и моделью. Он отвечает за предоставление представлению необходимых данных из модели и обработку пользовательских действий из представления. ViewModel представляет данные и команды, которые могут быть привязаны к элементам пользовательского интерфейса представления. Он обеспечивает разделение логики представления от бизнес-логики и позволяет более простое тестирование и поддержку приложения.

Связь между компонентами MVVM обычно осуществляется с помощью привязки данных и команд. Привязка данных позволяет представлению автоматически отображать данные из ViewModel, а также обновлять модель при изменении пользовательского ввода. Команды позволяют представлению вызывать методы или действия во ViewModel для обработки пользовательских действий.

Преимущества использования MVVM включают лучшую разделенность ответственности между компонентами, более легкое тестирование, повышенную переиспользуемость кода и улучшенную поддерживаемость приложения. MVVM также способствует лучшей масштабируемости и расширяемости, поскольку каждый компонент может разрабатываться независимо от других.

Для хранения данных была выбрана СУБД SQL Server [20].

Развернуть базу данных можно разными способами.

Установка с «использованием установочного пакета. Большинство СУБД предоставляют официальные установочные пакеты, которые можно скачать с официальных веб-сайтов разработчиков» [21]. Эти пакеты обычно содержат установщик, который проводит через процесс установки и настройки СУБД на сервере. Во время «установки может потребоваться

указать параметры, такие как путь установки, порты, настройки безопасности и другие настройки в соответствии с требованиями конфигурации» [22].

На рисунке 15 показана схема взаимодействия данных.

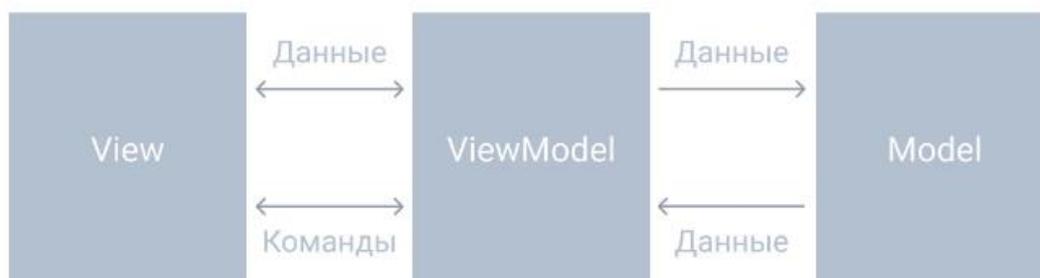


Рисунок 15 – Взаимодействие данных при использовании MVVM

Использование контейнеров, к примеру Docker. Контейнеры предоставляют портативное окружение, включающее все необходимое для работы СУБД, включая зависимости и конфигурацию. Можно «использовать инструменты управления контейнерами, такие как Docker, для развертывания и управления контейнером, содержащим СУБД. Это упрощает процесс развертывания и обеспечивает консистентное окружение для СУБД» [23].

Облачные сервисы. Многие облачные провайдеры, такие как Amazon Web Services (AWS), Microsoft Azure и Google Cloud Platform (GCP), предлагают готовые облачные сервисы для развертывания СУБД. Эти сервисы позволяют создавать экземпляры баз данных с помощью нескольких щелчков мыши и настраивать параметры, такие как масштабирование, резервное копирование и безопасность. «Облачные сервисы обеспечивают высокую доступность, масштабируемость и управление СУБД с минимальными затратами на инфраструктуру» [24].

Развертывание на сервере вручную. В некоторых случаях можно развернуть СУБД на сервере вручную, установив и настроив все необходимые компоненты вручную. Этот подход требует знания и опыта в установке и

настройке СУБД на конкретной операционной системе и аппаратной платформе. Руководство по установке и документация, предоставленные разработчиками СУБД, могут помочь в этом процессе.

### 3.4 Схема работы информационной системы

При запуске приложения пользователь (оператор или инженер) вводит логин и пароль, автоматизированная информационная система выглядит как пользовательский интерфейс с кнопками.

На рисунке 16 показана форма с данными клиентов компании.

Список абонентов						
Список услуг						
Список заявок						
Подключенные услуги						
Id	FirstName	Name	Patronymic	Address	Phone	
6	Иванченко	Олеся	Сергеевна	ул.Строителей, д.26, кв. 33	89234102585	
7	Митрикина	Ольга	Артемовна	ул.Смирнова, д.212, кв. 3	89234114463	
8	Смирнов	Игорь	Петрович	ул.Ленина, д.126, кв. 125	89234102011	
9	Потехина	Лариса	Анатольевна	ул.Сталина, д.12, кв. 1	89614205841	
10	Ивашин	Лев	Петрович	ул.Смолина, д.36, кв. 16	89614110202	
11	Иванченко	Олеся	Сергеевна	ул.Строителей, д.26, кв. 33	89234102585	
12	Митрикина	Ольга	Артемовна	ул.Смирнова, д.212, кв. 3	89234114463	
13	Смирнов	Игорь	Петрович	ул.Ленина, д.126, кв. 125	89234102011	
14	Потехина	Лариса	Анатольевна	ул.Сталина, д.12, кв. 1	89614205841	
15	Ивашин	Лев	Петрович	ул.Смолина, д.36, кв. 16	89614110202	

Рисунок 16 – Форма с данными клиентов

При формировании запросов можно получить отчет любой информации из базы данных, на рисунке 17 показан запрос, который выводит список заявок.

```

private void Button_Click_2(object sender, RoutedEventArgs e)
{
    using (Provider db = new Provider())
    {
        var requests = db.Requests.SqlQuery("SELECT * FROM
Requests").ToList();
        foreach (var request in requests)
        {
            db.Entry(request).Reference("Subscriber").Load();
            db.Entry(request).Reference("Service").Load();
        }
    }
}

```

Рисунок 17 – Запрос базы данных

На рисунке 18 показан отчет Заявок

Название заявки	Дата заявки	Фамилия абонента	Название услуги	
Подключение	21.10.2021	Иванченко	Интернет200	
Ремонт	11.10.2021	Митрикина	Интернет и ТВ	
Определение неисправности	15.10.2021	Митрикина	ТВ	
Подключение онлайн-кинотеатра	12.10.2021	Смирнов	Интернет100	
Подключение умного экрана	14.10.2021	Потехина	Интернет300	
Подключение	21.10.2021	Иванченко	Интернет200	
Ремонт	11.10.2021	Митрикина	Интернет и ТВ	
Определение неисправности	15.10.2021	Митрикина	ТВ	
Подключение онлайн-кинотеатра	12.10.2021	Смирнов	Интернет100	
Подключение умного экрана	14.10.2021	Потехина	Интернет300	

Рисунок 18 – Отчет по заявкам

Информационная система содержит данные о сотрудниках, клиентах, заявках и т.п. Данные хранятся в базе данных, с помощью запросов к базе пользователь получает списки необходимых данных. Основной функцией системы является вести учет заявок и их обслуживание в срок.

Метод MainWindow() позволяет автоматически генерировать данные в базу данных, метод private static void InitializeDbData() позволяет добавлять данные в базу без создания Sql запросов, рисунок 19.

```
private static void InitializeDbData()
{
    using (Provider db = new Provider())
    {
        db.Employees.RemoveRange(db.Employees);
        db.Requests.RemoveRange(db.Requests);
        db.Services.RemoveRange(db.Services);
        db.ServiceConnections.RemoveRange(db.ServiceConnections);
        db.Subscribers.RemoveRange(db.Subscribers);
        db.SaveChanges();
        //context.YourTable.RemoveRange(context.YourTable);
    }
}

Provider provider = new Provider();

Subscriber subscriber1 = new Subscriber()
{
    FirstName = "Иванченко",
    Name = "Олеся",
    Patronymic = "Сергеевна",
    Address = "ул.Строителей, д.26, кв. 33",
    Phone = "89234102585"
};
```

Рисунок 19 – Добавление данных в базу

Получение данных из базы происходит по средствам sql запросов и обработчика событий созданных кнопок интерфейса пользователя, рисунок 20.

```

1 reference
private void Button_Click_0(object sender, RoutedEventArgs e)
{
    using (Provider db = new Provider())
    {
        var subscribers = db.Subscribers.SqlQuery("SELECT * FROM Subscribers").ToList();
        subscriberDataGrid.Columns.Clear();
        subscriberDataGrid.AutoGenerateColumns = true;
        subscriberDataGrid.ItemsSource = subscribers;
    }
}

1 reference
private void Button_Click_1(object sender, RoutedEventArgs e)
{
    using (Provider db = new Provider())
    {
        var services = db.Services.SqlQuery("SELECT * FROM Services").ToList();
        subscriberDataGrid.Columns.Clear();
        subscriberDataGrid.AutoGenerateColumns = true;
        subscriberDataGrid.ItemsSource = services;
    }
}

```

Рисунок 20 – Получение данных из базы данных

В интерфейсе пользователя при нажатии на кнопку происходит обработка метода, описанного на рисунке 18, и пользователь получает данные в виде списка содержащихся данных в базе как показано на рисунке 21.

Загрузка данных				
Список абонентов				
Список услуг				
Список заявок				
Подключенные услуги				
Название заявки	Дата заявки	Фамилия абонента	Название услуги	
Подключение	21.10.2021	Иванченко	Интернет200	
Ремонт	11.10.2021	Митрикина	Интернет и ТВ	
Определение неисправности	15.10.2021	Митрикина	ТВ	
Подключение онлайн-кинотеатра	12.10.2021	Смирнов	Интернет100	
Подключение умного экрана	14.10.2021	Потехина	Интернет300	

Рисунок 21 – Получение данных списка сделанных заявок

Регистрация заявок происходит с помощью формы, когда все данные будут добавлены нужно нажать кнопку «Добавить заявку», после чего она

сформируется в список, где можно будет добавить комментарий о поломке, рисунок 22.

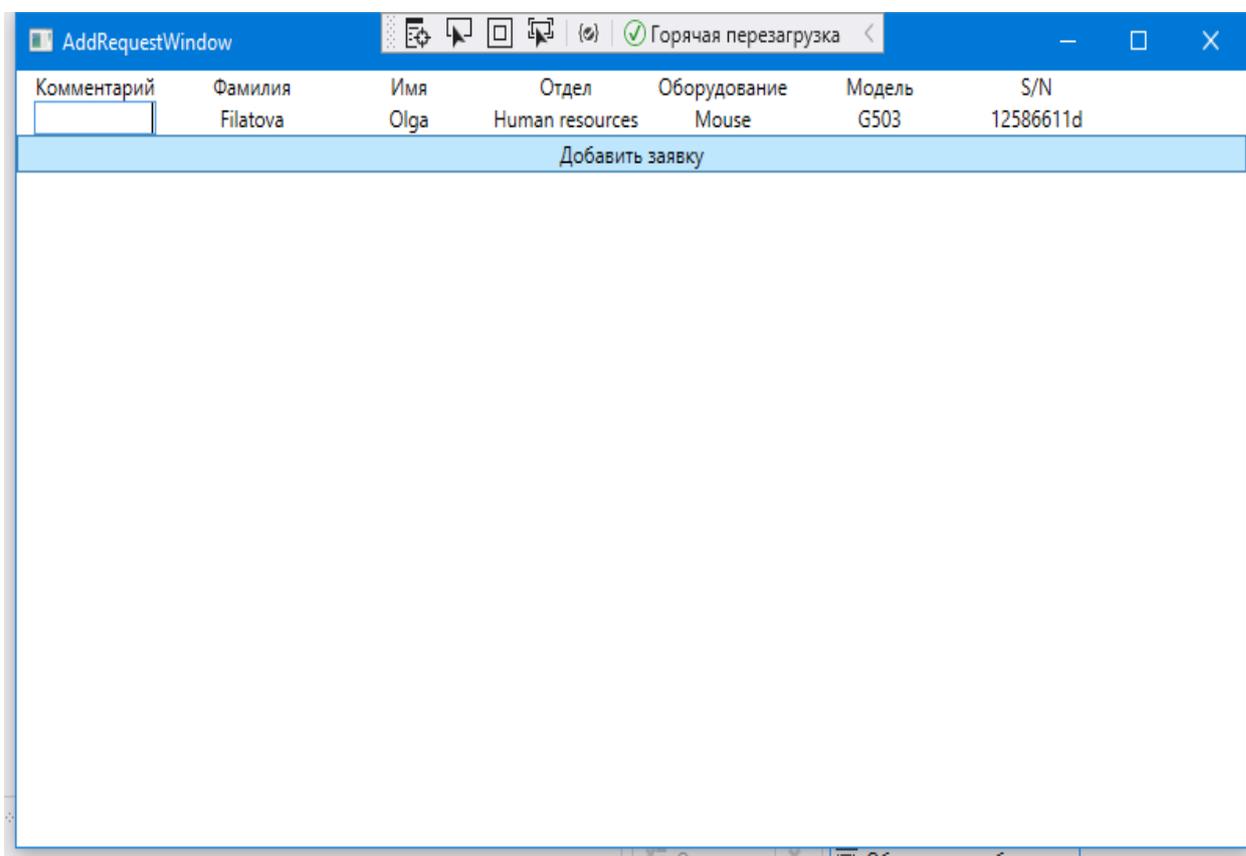


Рисунок 22 – Добавление комментария к заявке

После проверки всех данных и написания комментария к заявке при нажатии на кнопку «Добавить заявку», заявка будет добавлена в список всех заявок, которые будет видеть инженер контакт центра.

### 3.5 Тестирование информационной системы

Тестирование информационных систем является важным этапом их разработки и эксплуатации.

Тестирование позволяет выявить ошибки, дефекты и неполадки в информационных системах. Это могут быть ошибки программного кода,

некорректное поведение системы, несоответствие требованиям или неправильное функционирование при определенных условиях. Тестирование позволяет идентифицировать такие проблемы, чтобы разработчики могли исправить их до ввода системы в эксплуатацию.

Существует несколько видов тестирования информационных систем, которые выполняются на разных этапах жизненного цикла разработки и поддержки системы.

Система будет протестирована с помощью тестирования пользовательского интерфейса и системного тестирования.

Тестирование пользовательского интерфейса направлено на проверку удобства использования, доступности и соответствия дизайну системы. Целью является обнаружение проблем в навигации, взаимодействии, отображении данных и других аспектах пользовательского интерфейса.

Системное тестирование выполняется на уровне всей системы и проверяет, как система в целом функционирует и соответствует требованиям. Это включает проверку функциональности, производительности, безопасности и других аспектов работы системы в контексте ее окружения.

В процессе тестирования используются данные, характерные для системы в рабочем состоянии, т.е. данные для тестирования выбираются случайным образом. План проведения испытаний должен быть составлен заранее, обычно на этапе проектирования.

Для проведения тестирования было решено провести испытания класса В. Испытания класса В выполняются независимо от группы разработки и начинаются после того, как разработчики объявляют, что изделие готово к передаче потребителю.

В каталоге, содержащем загрузочный модуль системы «ИС контакт центра», должен находиться файл «Admin.cs», содержащий в себе все необходимые методы и данные. В случае отсутствия этого файла при запуске системы на экран выводится сообщение об отсутствии необходимого файла.

В разработанной системе пользователь имеет возможности изменять данные, которые уже хранятся в базе данных, пользователь имеет возможность просматривать данные списком с обновленными данными, которые обновляются автоматически при загрузке системы.

В ходе тестирования были выявлены следующие недостатки программы:

При отсутствии файла работа с системой невозможна. Вариант исправления: при отсутствии файла Admin.cs в указанной выше папке производить создание нового пустого файла или открыть папку Debug и запустить систему.

Система «ИС контакт центра» предназначена для осуществления обслуживания клиентов контакт центра по техническим вопросам компании провайдера.

Технические и программные средства, обеспечивающие выполнение данной программы: процессор Pentium IV или эквивалентный ему, не менее 80 Мб на жестком диске для самой программы, устройства ввода (клавиатура, мышь), устройство вывода (монитор) Программа работает под управлением операционной системы Windows 10.

Проверка программы производится в следующем порядке:

- запуск системы;
- проверка наличия загрузочного файла;
- если работа системы закончилась успешно и появилось окно графического интерфейса, то проверка считается успешно завершенной и следует выйти из программы;
- если работа программы завершилась не успешно, то следует обратиться к разработчику программы.

При получении диагностических и иных сообщений в ходе проверки программы следует обращаться к разделу «Сообщения системному программисту» данного руководства.

Чтобы протестировать корректность выполнения функций системы необходимо провести тестирование входных и выходных данных системы.

Рассмотрим список клиентов, которыми считаются абоненты компании провайдера, какие данные являются входящими и что пользователь получает на выходе после получения данных.

При внесении данных клиента указываются типы данных, которые должны соответствовать, рисунок 23.

```
public class Subscriber
{
    [Required]
    0 references
    public int Id { get; set; }

    [Required]
    5 references
    public string FirstName { get; set; }

    [Required]
    5 references
    public string Name { get; set; }

    [Required]
    5 references
    public string Patronymic { get; set; }

    [Required]
    5 references
    public string Address { get; set; }

    [Required]
    5 references
    public string Phone { get; set; }
}
```

Рисунок 23 – Добавление данных клиента в базу данных

На рисунке 23 показаны типы данных при внесении данных:

Id (идентификатор) – числовое значение (int);

FirstName (Фамилия) – строковое значение (string);

Name (Имя) – строковое значение (string);

Patronymic (Отчество) – строковое значение (string);

Address (Адрес) – строковое значение (string);

Phone (Телефон) – строковое значение (string).

При добавлении данных эти типы данных обязательно должны учитываться иначе программа просто выведет ошибку или исключение. На рисунке 24 показаны данные, добавленные в таблицу Клиенты.

```
Subscriber subscriber2 = new Subscriber()  
{  
    FirstName = "Митрикина",  
    Name = "Ольга",  
    Patronymic = "Артемовна",  
    Address = "ул.Смирнова, д.212, кв. 3",  
    Phone = "89234114463"  
};  
  
Subscriber subscriber3 = new Subscriber()  
{  
    FirstName = "Смирнов",  
    Name = "Игорь",  
    Patronymic = "Петрович",  
    Address = "ул.Ленина, д.126, кв. 125",  
    Phone = "89234102011"  
};
```

Рисунок 24 – Добавление данных клиента в базу данных с помощью создания нового объекта

На рисунке показаны данные и типы, которые к ним применены, после получения данных в виде списка ожидаем, что фамилия клиента будет выведена как строка, имя и отчество клиента также будут выведены в виде строки, адрес будет показан в виде строки с указанием улицы, дома и номера квартиры, телефон указан в кавычках как строковое значение, которое

позволяет вносить любые данные, как цифры, так и буквы. На рисунке 25 показан список полученных данных при проведении тестирования.

Id	FirstName	Name	Patronymic	Address	Phone	
1	Иванченко	Олеся	Сергеевна	ул.Строителей, д.26, кв. 33	89234102585	
2	Митрикина	Ольга	Артемовна	ул.Смирнова, д.212, кв. 3	89234114463	
3	Смирнов	Игорь	Петрович	ул.Ленина, д.126, кв. 125	89234102011	
4	Потехина	Лариса	Анатольевна	ул.Сталина, д.12, кв. 1	89614205841	
5	Ивашин	Лев	Петрович	ул.Смолина, д.36, кв. 16	89614110202	

Рисунок 25 – Список клиентов

Другие данные можно проверить по такому же сценарию входящих и исходящих данных.

На основе проведенного тестирования данных класса, проверки внесения данных с определенным типом данных и получение данных показало, что программа обрабатывает корректно и выводит корректные данные согласно присвоенному ему типу данных.

#### Выводы по третьему разделу

В третьем разделе работы описан интерфейс информационной системы контакт центра. Используя дерево функций, появилась возможность разработать схему информационной системы с помощью диаграмм потоков данных, последовательности и блок-схемы. Также были описаны алгоритмы выполнения функций системы, которые используются ежедневно оператором компании. На конечном этапе было проведено тестирование системы несколькими выбранными методами.

## Заключение

В ходе выполнения выпускной квалификационной работы было поставлено несколько задач, которые выполнялись поэтапно и логично.

На первоначальном этапе было проведено изучение деятельности компании провайдера, изучены основные бизнес-процессы, которые являются ключевыми для компании.

На основе изученных бизнес-процессов был выбран наиболее уязвимый процесс, который необходимо автоматизировать, в связи с чем были разработаны функциональные и нефункциональные требования информационной системы, которая в последствии обеспечит эффективность данного процесса.

Перед реализацией системы был проведен обзор аналогичных систем на рынке, решающих подобные задачи, анализ показал, что целесообразно будет разрабатывать свою систему «с нуля», спроектировав ее с необходимыми функциями и масштабом.

На этапе проектирования и подготовки были выбраны инструменты разработки, такие как ЯП C#, СУБД MS SQL Server и IDE Visual Studio 2019.

Разработана логическая, даталогическая и физическая модели базы данных, на основе чего и была реализована база данных в СУБД SQL Server.

Также были реализованы необходимые функции пользовательского интерфейса, для работы оператора и инженера компании провайдера.

На конечном этапе была разработана автоматизированная информационная система контакт центра компании и успешно протестирована.

Основная цель работы достигнута, все поставленные задачи выполнены в полном объеме.

## Список используемой литературы

1. Гвоздева, Т.В. Проектирование информационных систем. Методы и средства функционального проектирования /Т.В. Гвоздева. М.: Лань, 2019. – 458 с.
2. Глотова, С.А. Информационные системы /С.А. Глотова. М.: Темика, 2020. – 494 с.
3. Горбаченко В. И. Проектирование информационных систем с СА ERwin Modeling Suite 7.3: учебное пособие / В. И. Горбаченко, Г. Ф. Убиенных, Г. В. Бобрышева – Пенза: Изд-во ПГУ, 2019. – 154 с
4. Гусов К. Н., Андреев А. А., Бондаренко Э. Н. Трудовое право России. Учебник для бакалавров. — М.: Проспект, 2021. 592 с.
5. ГОСТ 24.702-85 Единая система стандартов автоматизированных систем управления. Эффективность автоматизированных систем управления. Основные положения.
6. ГОСТ 34.003-90 Автоматизированные системы. Термины и определения.
7. Методология Kanban: доски, принципы и возможности управления. [Электронный ресурс]. – Режим доступа: <https://skillbox.ru/media/>(дата обращения: 15.07.2023)
8. Онлайн курс обучения программированию: методологии разработки. [Электронный ресурс]. –Режим доступа: <https://javarush.ru/groups/posts/647-metodologii-razrabotki-po> (дата обращения: 15.07.2023)
9. Отличия, достоинства и недостатки базы данных PostgreSQL: что такое PostgreSQL. [Электронный ресурс]. – Режим доступа: <https://oracle-patches.com/common/3214-что-такое-postgresql> (дата обращения: 15.07.2023)
10. Петрова И.Р. Методология функционального моделирования IDEF0/ И.Р. Петрова, Р.Х. Фахртдинов, А.А. Сулейманова. – Казань: Казан. ун-т, 2018. – 68 с.

11. Семенова О.В. Информационные системы и программирование / О.В. Семенова. – М.: Инфра-М, 2020. – 214 с.
12. Хорев П.Б. Программно-аппаратная защита информации / П.Б. Хорев. – М.: Форум, 2019. – 352 с.
13. Худяков Г.А., Проектирование информационных систем / Г.А. Худяков. Учебное пособие, М.– Омега–Л, 2021. – 432 с.
14. Черушева Т. В. Проектирование программного обеспечения: учеб. пособие / Т. В. Черушева. – Пенза: Изд-во ПГУ, 2018. – 172 с.
15. Cross-Platform Desktop GUI Application using .Net Core [Электронный ресурс]. - Режим доступа: <https://yasar-yu.medium.com/cross-platform-desktop-gui-application-using-net-core-5894b0ad89a8> (дата обращения: 15.07.2023)
16. Docker: Развертывание контейнеров Docker в SQL Server [Электронный ресурс]. - Режим доступа: <https://docs.microsoft.com/ru-ru/sql/linux/sql-server-linux-docker-container-deployment?view=sql-server-ver15&pivots=cs1-bash> (дата обращения: 15.07.2023)
17. Firebase [Электронный ресурс]. Режим доступа: <https://a2is.ru/catalog/programmy-dlya-tantsevalnykh-studij-i-shkol-tantsev/firebase> (дата обращения 15.07.2023).
18. GeekBrains – Язык программирования C#. [Электронный ресурс]. - Режим доступа: <https://geekbrains.ru/posts/yazyk-programmirovaniya-c-sharp-istoriya-spezifika-mesto-na-rynke> (дата обращения: 15.07.2023)
19. Helpiks.org: Достоинства и недостатки языка [Электронный ресурс]. - Режим доступа: <https://helpiks.org/6-21879.html> (дата обращения: 15.07.2023)
20. Mobifitness [Электронный ресурс]. URL: <https://a2is.ru/catalog/crm-dlya-upravleniya-fitness-tsentrom/mobifitness> (дата обращения 15.07.2023).
21. Model-View-ViewModel (MVVM). [Электронный ресурс]. – Режим доступа: <https://learn.microsoft.com/en-us/dotnet/architecture/maui/mvvm> (дата обращения: 15.07.2023)

22. SoftClipper: что такое SQL Server. [Электронный ресурс]. – Режим доступа: <https://softclipper.net/foxpro-i-sql/sravnenie-baz-dannykh-microsoft-sql-server-i-microsoft-visual-foxpro.html> (дата обращения: 15.07.2023)

23. Sport Priority [Электронный ресурс]. URL: <https://a2is.ru/catalog/crm-dlya-upravleniya-fitness-tsentrom/sport-priority> (дата обращения 15.07.2023).

24. SQL Server: руководство по SQL Server. [Электронный ресурс]. – Режим доступа: <https://docs.microsoft.com/ru-ru/sql/sql-server/tutorials-for-sql-server-2016?view=sql-server-ver15> (дата обращения: 15.07.2023)

25. XRM Call center [Электронный ресурс] URL: <http://callcenter.xrm.ua/ru/> (дата обращения 15.07.2023).

26. Naumen Contact Center [Электронный ресурс] URL: <https://www.naumen.ru/products/phone/> (дата обращения 15.07.2023).