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

Кафедра «Прикладная математика и информатика»
(наименование)

09.03.03 Прикладная информатика
(код и наименование направления подготовки / специальности)

Разработка программного обеспечения
(направленность (профиль) / специализация)

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА (БАКАЛАВРСКАЯ РАБОТА)

на тему Создание системы для организации корпоративной IP телефонии

Обучающийся А. С. Осипов
(Инициалы Фамилия) (личная подпись)

Руководитель Н. Н. Рогова
(ученая степень (при наличии), ученое звание (при наличии), Инициалы Фамилия)

Тольятти 2025

Аннотация

Тема бакалаврской работы – «Создание системы для организации корпоративной IP-телефонии».

Работа состоит из введения, трех глав, заключения, списка используемой литературы и источников, приложения с листингом кода.

Во введении определена цель работы и приводится объект исследования.

В первой главе рассматриваются существующие системы для организации корпоративной IP-телефонии, проводится их анализ, а также рассматриваются требования к функционалу разрабатываемых компонентов.

Во второй главе представлен процесс разработки компонентов системы IP-телефонии для предприятия, включающий проектирование с использованием UML-диаграмм, таких как диаграммы компонентов, развертывания, последовательности и деятельности. Далее, представлен прототип пользовательского интерфейса и построены концептуальная и логическая модели базы данных.

В третьей главе проведена оценка эффективности разработанных компонентов системы IP-телефонии для предприятия. Сделан вывод о работоспособности и готовности к вводу системы в промышленную эксплуатацию.

В заключении подведены итоги проведенного исследования, определены перспективы возможной оптимизации и улучшения разработанных компонентов.

Бакалаврская работа состоит из 78 страниц, включает 17 таблиц, 26 рисунков и приложение.

Оглавление

Введение

IP-телефония стала широко распространенной технологией голосовой связи, которая использует компьютерную сеть для передачи данных. Главным образом это обусловлено тем, что для внедрения VoIP технологии не требуется покупка дорогостоящего оборудования и прокладка телефонных проводов. IP-телефония может обеспечить практически любой вид обслуживания, в том числе без необходимости создания физических инфраструктур.

IP-телефония практически полностью заменила аналоговые АТС и обладает рядом преимуществ: простота внедрения, легкая масштабируемость, наличие дополнительных возможностей для бизнес-целей предприятия на базе телефонии, интеграция с CRM, управление большим потоком звонков, персонализаций опций.

Целью работы является создание системы для организации корпоративной IP-телефонии на базе FreeSWITCH с разработкой модуля IVR и телефонной книги предприятия, отвечающим бизнес-требованиям предприятия. Для достижения поставленной цели в данной работе будут рассмотрены основные задачи, которые необходимо решить:

- описание и анализ существующих разработок,
- сбор и анализ требований к реализации решения,
- проведение проектирования разрабатываемых компонентов,
- разработка модуля IVR и телефонной книги предприятия для платформы FreeSWITCH,
- тестирование работоспособности и оценка эффективности разработанных компонентов.

Объектом исследования выпускной квалификационной работы является процесс создания системы для организации корпоративной IP-телефонии на базе FreeSWITCH. Исследование темы ВКР проводилось в департаменте сервисных услуг и технической поддержки АО «Софтлайн Солюшн».

В работе использованы методология реинжиниринга бизнес-процессов

предприятия, а также методы и технологии проектирования автоматизированных информационных систем.

Важным этапом при создании системы для организации корпоративной IP-телефонии является апробация разрабатываемого решения, включающая следующие шаги:

- развертывание системы IP-телефонии на базе FreeSWITCH,
- создание перечня тестируемых функций разрабатываемых компонентов,
- проведение тестирования,
- сбор обратной связи от пользователей,
- анализ полученных результатов.

Данная работа состоит из введения, трех глав, заключения, списка используемой литературы и используемых источников, приложения.

Первая глава посвящена анализу предметной области и постановки задачи на разработку компонентов системы IP-телефонии.

Во второй главе описан процесс проектирования и разработки компонентов системы IP-телефонии для предприятия.

В третьей главе рассмотрены все этапы апробации разработанного решения.

Полученные в работе результаты могут использоваться при разработке и внедрении IP-телефонии в любых организациях. Применение разработанных компонентов для платформы FreeSWITCH имеет практическую ценность благодаря своей гибкости, экономичности и возможности интеграции с современными технологиями. Это может стать идеальным выбором для организаций, стремящихся улучшить свои коммуникационные процессы и оптимизировать затраты.

Глава 1 Анализ предметной области и постановка задачи на разработку компонентов системы IP-телефонии

1.1 Значение разрабатываемых компонентов для системы IP-телефонии предприятия

Разработка программного решения с применением систем IP-телефонии значительно расширяет возможности коммуникаций для предприятия.

На сегодняшний день технология Voice over IP (VoIP) остается развивающейся технологией в мире деловых коммуникаций, меняя и совершенствуя способы общения компаний со своими клиентами. Архитектура IP-телефонии — это топология сети, которая определяет, как аудио в реальном времени будет передаваться через сеть организации, а также конфигурация инфраструктуры для обеспечения VoIP-вызовов.

«По мере того, как сети сотовой связи окончательно перейдут на технологии выше 3G, а в фиксированной телефонии произойдет смена морально и физически устаревших ЦАТС (цифровых автоматических телефонных станций) синхронной коммутации на оборудование Softswitch, технологии IP-телефонии будут полностью доминировать при оказании услуг передачи речевых сообщений» [12, с. 272].

Обобщенная архитектура технологии IP-телефонии представлена на рисунке 1.

«В ее составе можно выделить два функциональных уровня:

- верхний – управление обслуживанием вызовов (сеансов связи);
- нижний – транспортная сеть с маршрутизацией пакетов IP» [12, с. 273].

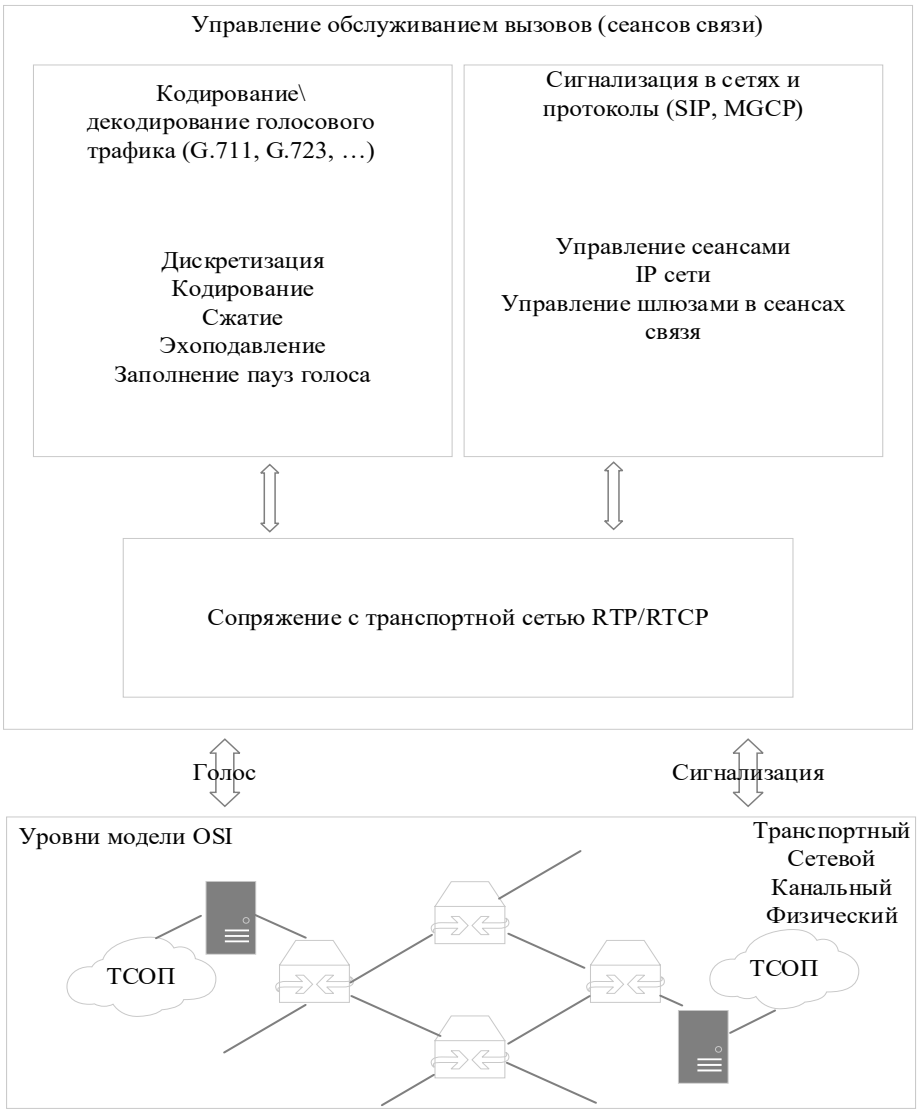

Рисунок 1 - Обобщенная архитектура IP-телефонии

«Верхний уровень решает две важные задачи:

– кодирование/декодирование речевых сигналов с помощью кодеков G.711, G.723, G.729 и т. п., а также подготовка к передаче речевых сигналов в транспортной сети;
– организация обслуживания вызовов (сеансов связи), под которой подразумевается управление установлением, поддержанием и разрушением телефонных соединений в ходе сеансов связи. По сути, это сигнализация в сети IP, поддерживаемая протоколами H.323, SIP, MGCP/MeGaCo, Q.931, СС № 7 и т. д.» [12, с. 274].

«Сигнализация в сетях IP-телефонии обеспечивает управление сеансами связи. Варианты реализации сигнализации в сетях IP поддерживаются протоколами уровней выше транспортного, такими как SIP, MGCP и т.п. Сигнализация IP-сетей обеспечивает управление не только сеансами телефонной, но и других видов связи: видео, электронной почты, файлового обмена, и многих других услуг. Носителями этих технологий являются аппаратно-программные средства, выполняющие роль оконечного абонентского оборудования, серверов современных услуг связи и шлюзового оборудования. Эти средства объединяются в единую сеть IP-телефонии локальными и транспортными сетями стека протоколов TCP/IP. Взаимоувязанная совокупность этих средств объединена между собой транспортными сетями стека протоколов TCP/IP. Управление сеансами связи в IP-телефонии осуществляется оборудованием Softswitch, в том числе IP-АТС» [12, с. 276].

Система IP-телефонии выступает не просто в роли маршрутизатора звонков, ее возможности гораздо шире с использованием дополнительного функционала, такого как интерактивное голосовое меню (IVR Interactive Voice Response) и телефонная книга предприятия. Существует множество различных решений IVR и телефонных книг предприятия для систем IP-телефонии, но наиболее востребованы модули, разработанные непосредственно под заказчика.

Задачей ВКР является создание системы для организации корпоративной IP-телефонии с разработкой модуля IVR и телефонной книги предприятия, отвечающих заявленным требованиям.

Предпосылками для инициирования проекта явились прекращение поддержки вендором Microsoft Skype for Business, а также необходимость в унификации текущего ИТ-решения коммуникаций, централизации администрирования и управления лицензиями, консолидации и централизации телефонных мощностей, повышении продуктивности работы сотрудников предприятия за счёт обеспечения пользователей корпоративных

информационных систем унифицированной, современной, удобной в использовании системы коммуникаций в реальном времени, предоставляющей возможности, перечисленные в функциональных требованиях.

Основными целями модернизации системы IP-телефонии являются: замена более не поддерживаемой вендором системы унифицированных коммуникаций Microsoft Skype for Business на систему IP-телефонии на базе решения с открытым исходным кодом FreeSWITCH; обеспечение сотрудникам компании бесперебойного доступа к сервисам голосовой связи с использованием технологий IP-телефонии; обеспечение возможности использовать на объектах внедрения телефонных аппаратов различных производителей и программных телефонов с расширенным функционалом; повышение отказоустойчивости и резервирования сервиса. В задачи данного мероприятия входит обеспечение качественной IP-телефонии на территории предприятия.

Переход с одной системы IP-телефонии (Microsoft Skype for Business) на другую (FreeSWITCH) приводит к необходимости создания своего отдельного модуля интерактивного голосового меню и телефонной книги предприятия. Microsoft Skype for Business уже содержит собственный модуль IVR с удобной панелью управления, а так же использует Microsoft Active Directory в качестве телефонной книги предприятия.

Внедрение FreeSWITCH предполагает разработку такого IVR модуля с компонентом DISA и телефонной книги предприятия отдельно, так как они не встроены в систему.

На рисунке 2 представлена диаграмма «Причина-следствие», показывающая взаимосвязь между проблемой необходимости замены существующей системы IP-телефонии и факторами, способствующими ей.

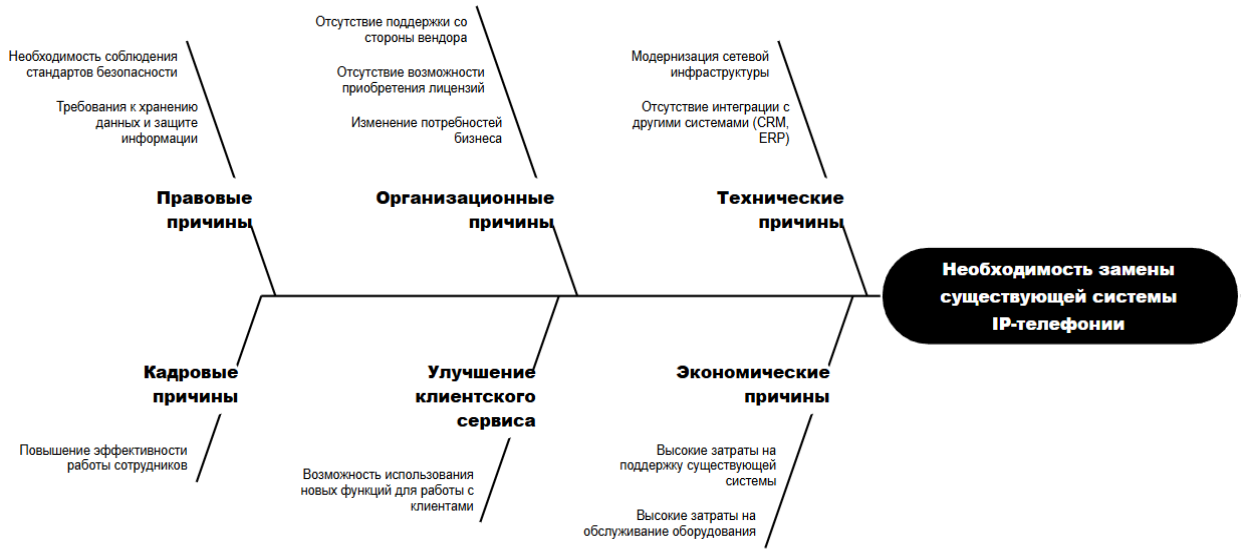

Рисунок 2 - Диаграмма «Причина-следствие»

Главной задачей создаваемой системы IP-телефонии является отказ от зарубежного платного погромного обеспечения путем перехода на бесплатное ПО с открытым исходным кодом с разработкой модуля IVR и телефонной книги предприятия. На момент перехода системы должны сосуществовать друг с другом. Для этого единой точкой входа будет являться внедряемая система IP-телефонии, которая будет маршрутизировать звонки в том числе в действующую систему. Для реализации данного проекта в качестве полноценной АТС в созданной системе IP-телефонии выбран FreeSWITCH.

Функционирование FreeSWITCH основано на протоколах VoIP и благодаря этому выбранная АТС может работать с практически любым оборудованием.

В период проведения исследования по теме ВКР осуществлен процесс миграции абонентов из текущей системы IP-телефонии на базе Microsoft Skype for Business в целевую систему IP-телефонии с открытым исходным кодом FreeSWITCH в целях создания адаптированной под бизнес-цели предприятия корпоративной IP-телефонии. В качестве процессной модели организации рассматривается процесс – функционирование системы IP-телефонии.

«На сегодняшний день наибольшее распространение получили несколько стандартов моделирования бизнес-процессов:

– семейство стандартов функционального моделирования IDEF (IDEF0, DFD, IDEF3),
– семейство стандартов ARIS (в частности, нотация eEPC),
– семейство стандартов UML для визуального моделирования, основанного на объектно-ориентированном подходе (Usecase diagram, Activity diagram)» [15, с. 6].

Жизненный цикл поступающего звонка представлен в виде схемы – функциональной модели TO-BE (рисунок 3):

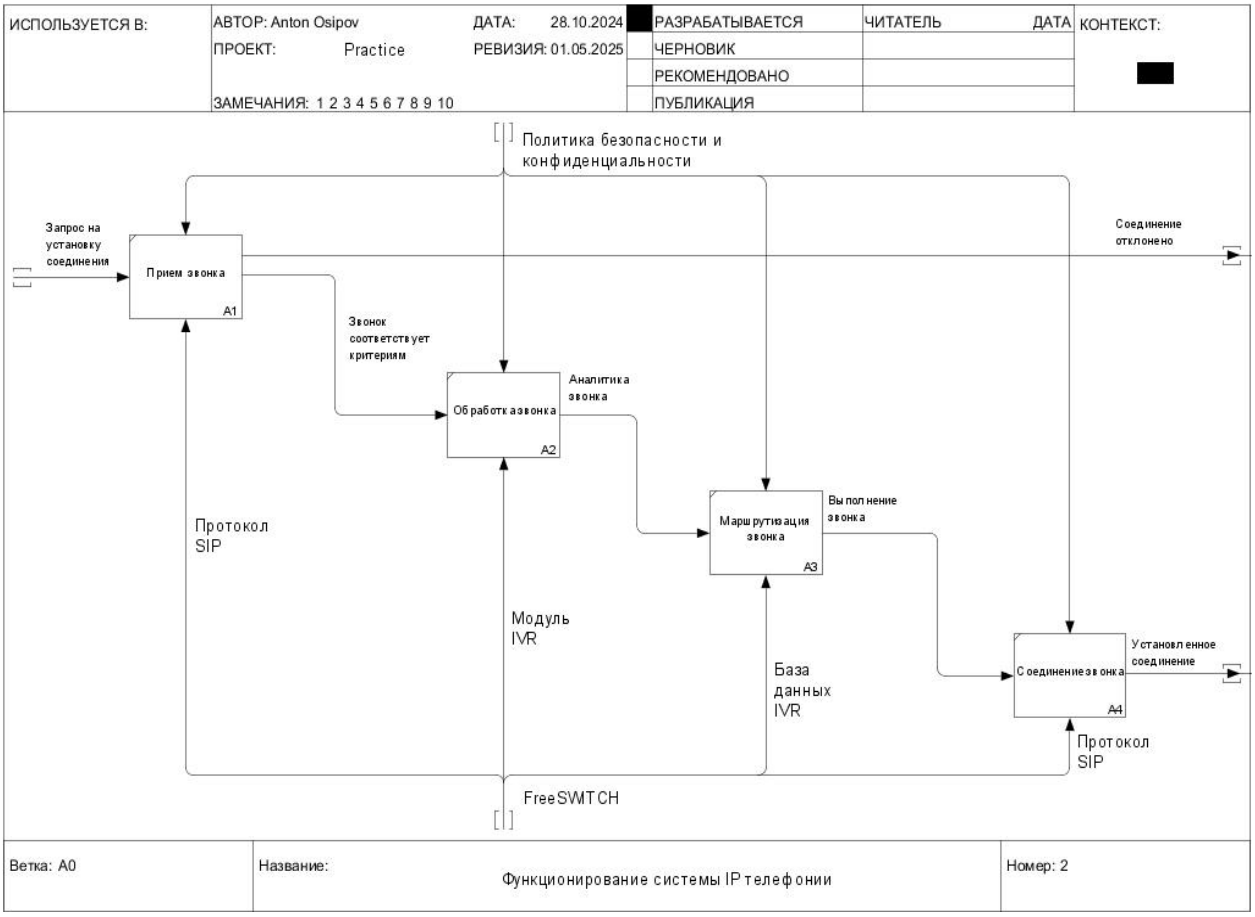

Рисунок 3 - Диаграмма функционирования системы IP-телефонии

Архитектура решения на момент пилотной миграции абонентов должна

включать в себя компоненты текущей и внедряемой системы IP-телефонии (сосуществование систем). Архитектура внедряемой системы IP-телефонии на площадке предприятия должна быть развернута в виртуальной среде Microsoft Hyper-V.

Внедряемая система IP-телефонии должна обеспечивать автоматизированный прием входящих внешних звонков и их маршрутизацию внутренним абонентам как внедряемой системы IP-телефонии, так и текущей системы IP-телефонии согласно меню IVR и телефонной книге предприятия. Исходящие звонки внутренних абонентов каждой из систем должны централизованно маршрутизироваться посредством внедряемой системы IP-телефонии

Системы IP-телефонии становятся стандартом для голосовой связи в крупных и малых предприятиях. VoIP имеет множество преимуществ, включая финансовую экономию, обмен медиа данными и проведение конференций в реальном времени с потоковым видео в рамках одной сессии. Системы IP-телефонии с открытым исходным кодом теперь открыло множество возможностей в VoIP-индустрии.

Несмотря на стремительное развитие цифровых средств коммуникации, телефония остается критически важным сервисом для предприятия и уход иностранных вендоров создал определенные сложности с поддержанием и развитием корпоративных систем IP-телефонии.

Внедрение системы IP-телефонии или миграция со старого решения VoIP на новое представляет собой значительный проект, а также множество изменений в том, как сотрудники выполняют свою работу. Им нужно будет приспособиться к новому способу выполнения задач, что никогда не бывает легко, а ИТ-отделу нужно будет сосредоточить свои усилия на новых задачах управления и устранения неполадок. Независимо от того, выполняется ли внедрение системы IP-телефонии или миграция на новое решение VoIP, для получения наилучшей производительности требуется планирование, активное управление и мониторинг.

«Когда организации начинают внедрение системы IP-телефонии, они рассчитывают на успех, в рамках их бюджетов, требований к результатам, ожиданий руководителей и конечных пользователей. Тем не менее, доля неудач таких проектов остается высокой.

Внедрение системы IP-телефонии заканчивается неудачей по множеству причин, включающих в себя: недостаток поддержки со стороны высшего руководства, нереалистичные ожидания, некорректное определение требований, неправильный выбор пакетов, различия между программными требованиями и бизнес-требованиями и недостаточные ресурсы» [8].

Разработка модуля IVR и телефонной книги под конкретные бизнес-требования предприятия решит задачу распределения звонков по отделам и менеджерам без участия секретаря с минимальными затратами. Бизнес-требования предполагают отказ от продуктов импортных вендоров без потери действующего функционала, финансовую экономию, возможность использования существующих голосовых шлюзов и телефонов, обновление продукта в долгосрочной перспективе. Кроме того, внедрение FreeSWITCH с разработанными телефонной книгой предприятия и модулем IVR с панелью его управления отвечает всем требованиям к хранению данных и защите информации. Так на сегодняшний день для государственных организаций в рамках существующих требований возможно использовать только решения, разработанные в России или на базе свободно распространяемого ПО, к которым относится FreeSWITCH.

Технология VoIP продолжает развиваться, она уже доказала свою эффективность и гибкость. Можно выделить 5 тенденций, которые будут определять технологию в будущем [21]:

– функции искусственного интеллекта. Программное обеспечение для анализа настроений отслеживает тон голоса звонящего во время разговора, помогая агентам и руководителям узнать, какие звонки могут быть неудачными и требуют внимания. А программы ИИ используются для создания письменных резюме конференц-звонков,

предоставляя сотрудникам еще один источник данных помимо самой записи звонка;

- будущее VoIP: поддержка гибридных моделей работы. Вместо настольных телефонов сотрудники могут войти в программные телефоны на своих ноутбуках или смартфонах и иметь полнофункциональный рабочий телефон, независимо от того, находятся ли они в офисе или дома. Видеоконференции позволяют сотрудникам продолжать проводить утренние совещания, стратегические сессии или обзоры спринта, даже когда некоторые члены команды не находятся в офисе. А виртуальные секретари и IVR могут направлять вызовы сотрудникам, работающим дома, в другом филиале офиса или во время командировок.
- мобильность и мобильные приложения. VoIP-приложения для настольных компьютеров и мобильных устройств являются мощными именно потому, что они работают без проблем на портативных устройствах. Больше не нужен физический настольный телефон — нужно просто запустить настольное приложение на ноутбуке, и будет собственный персонализированный и портативный бизнес-телефон.
- WebRTC и VoIP: мощное сочетание для деловых звонков. Это технология, которая обеспечивает передачу голоса, видео, текста и файлов (все формы общения в реальном времени) между веб-браузерами и мобильными приложениями. WebRTC определяет общий набор протоколов и стандартов, которые позволяют разным программам и интерфейсам взаимодействовать.
- безопасность. Шифрование телефонных звонков, многофакторная аутентификация, регулярное обучение сотрудников атакам социальной инженерии, тесты на проникновение и передовые антивирусные программы для обнаружения эксплойтов атак нулевого дня — все это важные меры, которые компании могут

использовать для защиты своих данных и укрепления своей безопасности. То же самое будет и с будущим платформ VoIP.

Создание системы IP-телефонии является важной задачей в условиях изменений потребностей бизнеса. Все функции, реализованные в действующей системе Microsoft Skype for Business, останутся доступными в системе FreeSWITCH с разработанным модулем IVR и телефонной книгой предприятия.

1.2 Основные функции разрабатываемых компонентов для системы IP-телефонии предприятия

Система IP-телефонии с модулем IVR и телефонной книгой предприятия выполняет множество функций.

К функциональным особенностям системы IP-телефонии относятся следующие характеристики:

- переадресация вызовов. Звонки можно легко перенаправить на мобильное устройство, телефон другого члена команды или даже в другой отдел, что обеспечивает плавное управление потоком вызовов и лучшее обслуживание клиентов;
- запись звонков. Юридическая и информативная запись разговоров — это не просто приятное дополнение, это важный инструмент. Запись звонков поддерживает контроль качества, разрешение споров и возможность пересматривать взаимодействия с клиентами для более глубокого понимания их потребностей;
- обмен сообщениями. Текстовое общение является необходимым дополнением к голосовым звонкам на современном рабочем месте. Системы VoIP преуспевают в этой области, предлагая функции обмена сообщениями, которые оптимизируют и улучшают повседневное взаимодействие;

– интерактивное голосовое меню – IVR. Система, которая позволяет клиентам выбирать варианты голосового меню и взаимодействовать с помощью голоса и цифровых клавиатур. Благодаря объединению вычислительных и телефонных технологий программное обеспечение IVR сокращает время ожидания в центре обработки вызовов, улучшает рабочие процессы обслуживания клиентов и способствует повышению их удовлетворенности;
– голосовая почта. Визуальные системы голосовой почты преобразуют входящие голосовые сообщения в текст, позволяя пользователям читать и управлять ими так же, как электронными письмами. Эта функция не только экономит время, но и интегрирует голосовые сообщения в привычный рабочий процесс цифрового обмена сообщениями;
– интеграция. Обычно решение IP-телефонии не существует изолированно. Оно может интегрироваться с другими важными бизнес-инструментами, объединяя и используя данные на разных платформах для более сплоченного и оптимизированного рабочего процесса;
– мобильные приложения. Эти приложения обеспечивают полную функциональность звонков, гарантируя, что предоставляется исключительный сервис независимо от местонахождения;
– поддержка программного телефона. Программные телефоны (софтфоны) превращают ноутбук или настольный компьютер в полнофункциональный телефон, используя услуги системы IP-телефонии без дополнительного оборудования;
– телефонная книга предприятия. Программные телефоны (софтфоны) и настольные IP-телефоны смогут отображать информацию о том, кто звонит. К наиболее важной информации о входящем звонке относятся номер телефона, ФИО, название компании и должности.

Разработанные для работы с FreeSWITCH модуль IVR и телефонная

книга предприятия используют базы данных для хранения информации. Обработка данной информации выполняется непосредственно компонентами системы. Так же модуль IVR имеет графический интерфейс пользователя для обработки данных администратором IVR.

На рисунке 4 представлена диаграмма потоков данных:

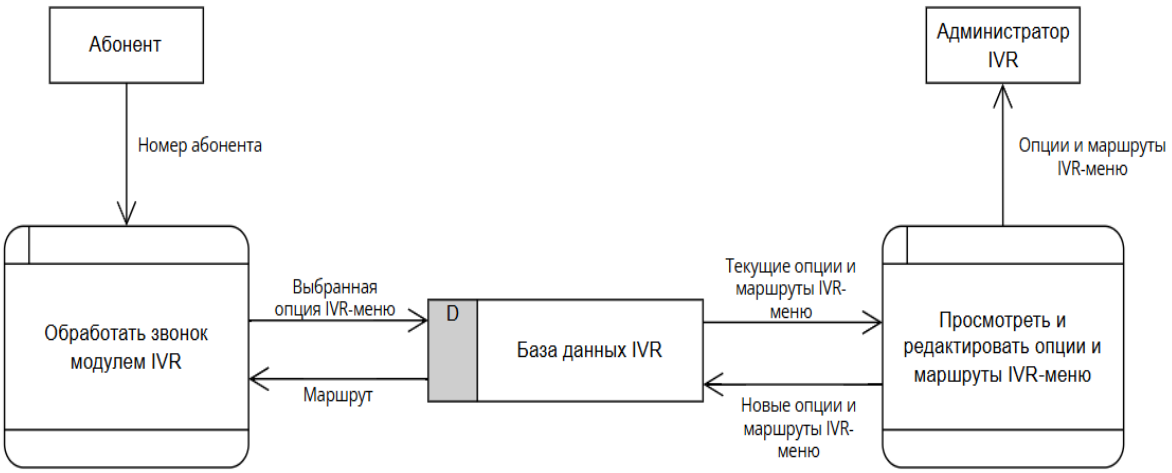

Рисунок 4 – Диаграмма потоков данных

Программное обеспечение помогает управлять ресурсами предприятия, автоматизируя функции коммуникации между клиентами и сотрудниками предприятия. Это делает любой операционный процесс более эффективным и продуктивным.

В таблице 1 представлены основные преимущества использования системы IP-телефонии с модулем IVR и телефонной книгой предприятия для различных систем управления предприятием

Таблица 1 – Преимущества использования систем IP-телефонии для систем предприятия

Система управления	Описание	Преимущества использования IP-телефонии
CRM-система	Система управления взаимоотношениями с клиентами	Интеграция звонков Повышение лояльности клиентов
PM	Система управления проектами	Упрощение коммуникации
HRM-система	Системы управления человеческими ресурсами	Коммуникация между сотрудниками
SCM	Система управления цепочками поставок	Координация действий с контрагентами
EMA	Система автоматизации маркетинга	Обратная связь от клиентов

Модуль IVR, как и любое разрабатываемое приложение, должен быть удобным и интуитивно понятным для абонентов и администратора IVR, который управляет сопоставлением опций меню маршрутам звонка. Голосовое меню должно быть кратким, рекомендуется использовать не более четырех опций в меню. Крайне важно использовать соответствующие голосовые подсказки. Необходимо предусмотреть вариант отсутствия выбора какой-либо из опций со стороны абонента и немедленное переключение на оператора в этом случае.

Пользовательский опыт (UX) и пользовательский интерфейс (UI) играют ключевые роли в модуле IVR и телефонной книге предприятия. Удобный интерфейс, не перегруженный незначительными опциями, позволит улучшить взаимодействия абонентов и администратора IVR с системой, снизит время ожидания, повысит удовлетворённость клиентов. Качественный UX/UI, несомненно, выделяется на фоне конкурентов, предлагает высокий уровень коммуникации.

Так как модуль IVR имеет графический интерфейс пользователя важно обеспечить безопасность данных и защиту от несанкционированного

доступа. Защита обеспечивается путем использования выделенного сервера для доступа к данным по шифрованным каналам связи. Доступ администратора к интерфейсу управления IVR также осуществляется по шифрованным каналам связи и требует авторизации.

Телефонная книга предприятия не имеет пользовательского интерфейса, защита данных обеспечивается на уровне SQL-сервера, требующего авторизацию для доступа к данным.

Для разрабатываемых модуля IVR и телефонной книги предприятия используется модель безопасности, представленная на рисунке 5.

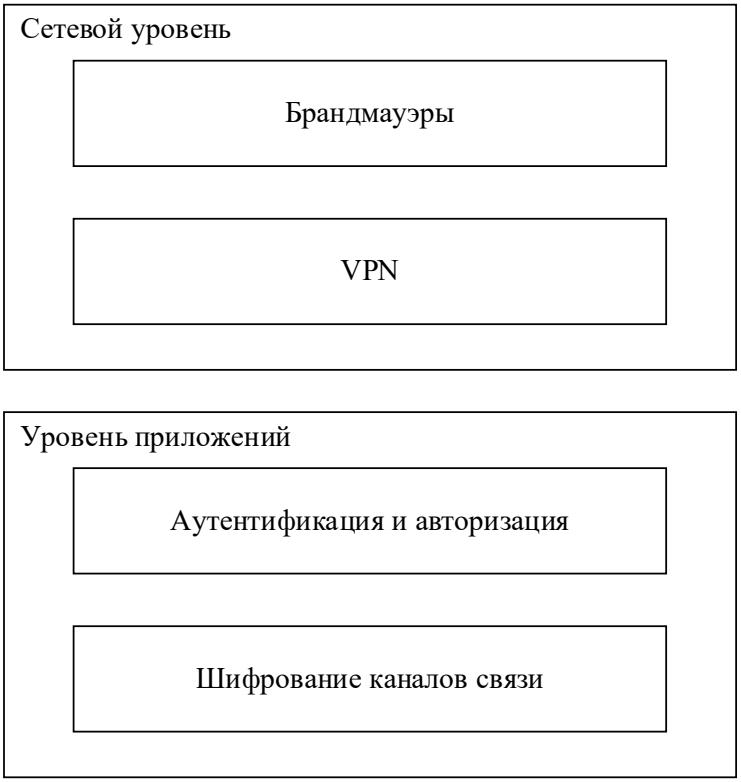

Рисунок 5 – Модель многослойной безопасности

Таким образом выявление основных функций программного обеспечения на этапе проектирования и их последующая реализация в разрабатываемом приложении способствует повышению его эффективности в процессе эксплуатации.

1.3 Обзор и анализ существующих систем для организации корпоративной IP-телефонии

На рынке довольно давно существуют как российские продукты, так и с открытым исходным кодом, способные полностью заменить иностранные решения корпоративного сегмента. Среди них выделяются такие как «ФЛАТ», «Сател» и системы с открытым исходным кодом - Asterisk, FreeSWITCH.

Системы IP-телефонии предлагают расширенные возможности связи за счет интеграции традиционной телефонии с инновационными IP-технологиями. Предприятия получают выгоду от оптимизированных операций в системах унифицированных коммуникаций.

Предоставляя надежные коммуникационные решения, системы IP-телефонии необходимы для современных предприятий. Эти системы объединяют функциональные возможности устаревших АТС с IP-технологиями, предоставляя многофункциональные услуги телефонии. Пользователи ценят гибкость, экономию средств и масштабируемость, которые обеспечивают платформы IP-телефонии. Предприятие может более эффективно управлять внутренними и внешними коммуникациями, что приводит к повышению производительности и оперативности.

В рамках исследования по теме ВКР были рассмотрены три решения систем IP-телефонии — это текущая система унифицированных коммуникаций Microsoft Skype for Business, российский продукт ФЛАТ, а также внедряемая система с открытым исходным кодом FreeSWITCH.

Рассмотрены следующие общие архитектурные и функциональные особенности систем IP-телефонии:

а) архитектурные:

1) сигнализация: управление сеансами связи - установлением, поддержанием и завершением звонков в ходе сеансов связи по протоколу SIP. Поддержка трансляции вызовов между разнородными сетями – ТСОП и IP, функционирование в роли

пограничного контроллера сессий, медиашлюза, медиасервера, сервера приложений;

2) кодирование/декодирование голосового трафика: поддержка кодеков G.711, G.723 и т. п.

б) функциональные:

1) безопасность: резервное копирование и хранение данных в зашифрованном виде, отслеживание и запись действий администраторов системы;
2) лицензирование: особенности, цены;
3) отчеты о звонках: детализация звонков по направлениям, номерам и длительности;
4) дополнительные виды обслуживания: удержание вызова, автодозвон, обратный звонок, перехват вызова, перевод и переадресация вызова;
5) унифицированные коммуникации: объединение голосовой, видео связи и обмен сообщениями на одной платформе;
6) автосекретарь: обеспечение автоматической обработки вызовов и маршрутизации в соответствии с заданными параметрами;
7) голосовая почта: отправка голосовых сообщений непосредственно в почтовые ящики электронной почты сотрудников;
8) телефонная книга: отображение номера телефона, ФИО, названия компании и должности звонящего.

В ходе исследования по теме ВКР рассмотрены преимущества системы IP-телефонии с открытым исходным кодом FreeSWITCH.

Основными преимуществами являются:

– финансовая экономия. FreeSWITCH не полностью бесплатная система IP-телефонии: как и для любого другого программного решения, для работы этого программного обеспечения требуется

оборудование или сервер. Так же могут потребоваться аппаратные платы Digital T1 или голосовой шлюз для подключения к телефонной сети общего пользования (ТСОП) для связи с внешнем миром. Большинство коммерческих решений VoIP требуют платы за лицензию аппаратного обеспечения, плату за пользовательскую лицензию, плату за телефонную лицензию и так далее. Кроме того, может потребоваться покупка контрактов на поддержку и контрактов на обслуживание. Для системы IP-телефонии FreeSWITCH с открытым исходным кодом этого не требуется, но, если нужна какая-либо поддержка, доступны как платные, так и бесплатные варианты;

– гибкость. Система IP-телефонии FreeSWITCH с открытым исходным кодом можно использовать в многочисленных коммуникационных службах: IVR-сервер, сервер голосовой почты, шлюз VoIP, колл-центр, IP-АТС, программный коммутатор и конференц-мост — это то, что можно построить с использованием технологий VoIP с открытым исходным кодом. Их так же легко интегрировать. Это упрощает подключение телекоммуникационных продуктов;
– настройка (кастомизация). Система IP-телефонии FreeSWITCH основана на IP, поэтому может интегрироваться с такими приложениями, как CRM, click to dial (нажми чтобы позвонить), база данных, электронная почта, мобильный SIP номеронабиратель или любой программой, интегрированной с решением VoIP с открытым исходным кодом. Администраторы могут самостоятельно выполнить настройку интеграции посредством конфигурационных файлов, большинство этих настроек уже произведены в типичном дистрибутиве с открытым исходным кодом;
– долгосрочность. Решения VoIP с открытым исходным кодом, такие как FreeSWITCH, постоянно обновляются новейшими наборами функций, в основном для улучшения его производительности и исправления всех ранее обнаруженных проблем. В отличие от

основанных на Asterisk систем IP-телефонии, FreeSWITCH является более производительным и простым решением. Открытый исходный код FreeSWITCH обновляется автоматически и бесплатно, минуя пакеты обслуживания и обновления, которые обычно требуются для коммерческих систем IP-телефонии;

- простота. FreeSWITCH полностью бесплатна для использования в соответствии с условиями лицензии MPL. Нет никаких лицензионных сборов, блокировок поставщиков и сборов за обновление функций и добавление пользователей. Это позволяет FreeSWITCH быть простой и экономически эффективной платформой IP-телефонии, помогающей улучшить связь в бизнесе. Конфигурационные файлы имеют формат XML, что упрощает настройку администраторами не имеющих специальных знаний, требующимися большинством коммерческих систем IP-телефонии;
- кроссплатформенность. FreeSWITCH можно установить на ОС семейства Linux, Microsoft Windows, MacOS. В качестве серверной ОС рекомендуется использование Debian.

«FreeSWITCH — это решение IP-телефонии с открытым исходным кодом, разработанное на языке C. Оно позволяет реализовать связь с виртуальным телефоном через виртуальный коммутатор. FreeSWITCH можно использовать как простой коммутатор, АТС, шлюз или сервер приложений IVR, используя скрипты или XML-файлы для автоматизации определенных задач и разработки новых услуг.

FreeSWITCH работает на нескольких операционных системах, включая Windows, Mac OS X, Linux, BSD и обе платформы Solaris (32- и 64-разрядные). FreeSWITCH поддерживает стандартные и расширенные функции протокола SIP» [18, с. 52].

«В 2006 году группа бывших разработчиков Asterisk приняли решение разработать альтернативное решение – на свет появился FreeSWITCH. Вдохновленные модульной структурой веб – сервера Apache, команда

разработчиков преследовала цель улучшить параметры масштабируемости и стабильности работы на разных платформах.

FreeSWITCH создан по модели состояний, вследствие чего, каждый вызов(канал) работает по отдельному потоку данных. Для построения структуры использовались компоненты opensource решений, такие как, например, Sofia SIP – SIP UA с открытым исходным кодом, созданный компанией Nokia.

В сравнении с FreeSWITCH Asterisk использует общие ресурсы, включая программные потоки – это главная проблема при большой интенсивности вызовов.

Несмотря на сложность и многогранность программного кода, на котором написан Asterisk, он находит огромное множество применений в сети. С другой стороны, FreeSWITCH написан на C, структура которого более понятна и прозрачна. Потоки процессов выполняются последовательно и отдельно для каждого канала, что безусловно отличает FreeSWITCH от Asterisk. При этом, как правило, по этой причине FreeSWITCH требует больший объем оперативной памяти (RAM).

FreeSWITCH имеет хорошо документированный API (Application Programming Interface), сегментированный по ролям. Такая структура обеспечивает безопасное подключение к API в отличие от Asterisk, где более открытая конструкция API допускает вероятность внесения багов и ошибок.

Asterisk в свою очередь базируется на текстовых конфигурационных файлах, в то время как FreeSWITCH использует файлы формата .xml. Безусловно, с точки зрения работы с конфигами для админа, файлы текстового формата проще редактировать, однако, плюсы формата .xml всплывают на этапе автоматизации различных процессов» [11].

Сравнение основных характеристик рассматриваемых систем IP-телефонии приведено в таблице 2.

Таблица 2 – Сравнительная характеристика систем IP-телефонии

Производители	ФЛАТ	FreeSWITCH	Microsoft Skype for Business
Архитектурные особенности			
Поддержка SIP	да	да	да
Трансляция ТСОП - IP	при использовании стороннего медиашлюза	при использовании стороннего медиашлюза\плат расширения	нет
Пограничный контроллер сессий	да	да	нет
Медиасервер	да	да	да
Медиашлюз	да	да	нет
Сервер приложений	да	да	нет
Поддержка кодеков	G.711A, G.729	G.711A\U, G.729, OPUS, SILK и т.п.	G.711A\U, SILK
Функциональные особенности			
Резервное копирование и шифрование данных	да	да	да
Отслеживание и запись действий администратора	да	нет	нет
Лицензирование	каждый модуль	нет	сервер и клиенты
ДВО (удержание вызова, автодозвон, обратный звонок, перехват вызова, перевод и переадресация вызова)	да	да	только удержание вызова, перехват вызова, перевод и переадресация вызова
Видеосвязь и обмен сообщениями	да	да	да
Автосекретарь (голосовое меню)	да	разработанный модуль IVR	да
Голосовая почта	да	да	при интеграции с Microsoft Exchange
Телефонная книга	да	разработанная телефонная книга	да, Microsoft Active Directory

Microsoft Skype for Business является одной из самых современных платформ унифицированных коммуникаций, объединяющих в себе возможности систем IP-телефонии и видеосвязи, а так же обмена мгновенными сообщениями. Среди наиболее подходящих под разработанные требования аналогов была выделена система IP-телефонии с открытым исходным кодом FreeSWITCH. Данная система является наиболее близкой по

функциональности текущей - объединяет в себе традиционные функции систем телефонии, а также возможность видеозвонков и обмена мгновенными сообщениями.

Сравнение основных характеристик модулей IVR и телефонных книг предприятия рассматриваемых систем IP-телефонии приведено в таблицах 3 и 4. Анализ проводился на основе субъективной оценки по следующим критериям: функциональность, стоимость, удобство использования, поддержка и безопасность. Каждый критерий оценен по шкале от 1 до 5.

Таблица 3 – Сравнительная характеристика модулей IVR систем IP-телефонии

Критерии	ФЛАТ IVR	Разрабатываемый модуль IVR	Skype for Business IVR
Функциональность	5	4	5
Стоимость	2	5	1
Удобство использования	4	5	5
Поддержка	5	1	1
Безопасность	5	5	5

На рисунке 6 представлена диаграмма, визуализирующая проведенный сравнительный анализ.

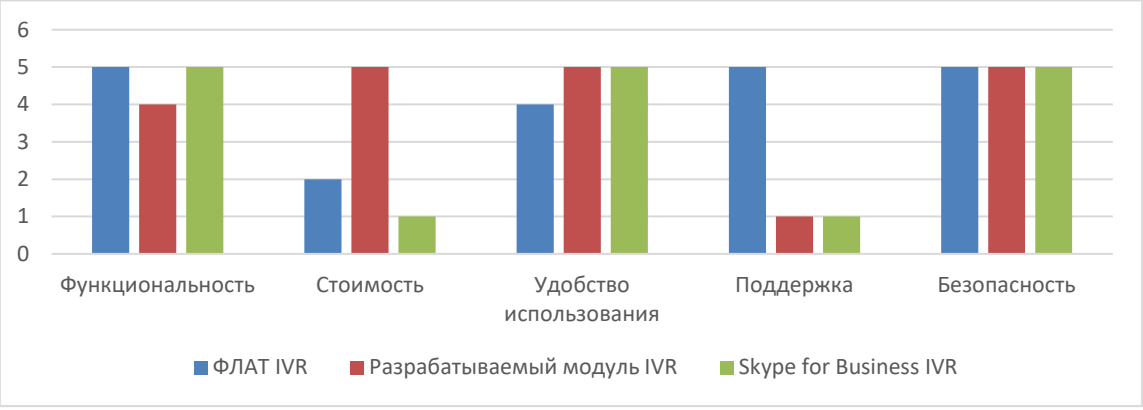

Рисунок 6 – Диаграмма сравнения характеристик модулей IVR

Таблица 4 – Сравнительная характеристика телефонных книг предприятия систем IP-телефонии

Критерии	Разрабатываемая телефонная книга	Skype for Business телефонная книга
Функциональность	4	5
Стоимость	5	1
Удобство использования	5	5
Поддержка	1	1
Безопасность	5	5

На рисунке 7 представлена диаграмма, визуализирующая проведенный сравнительный анализ.

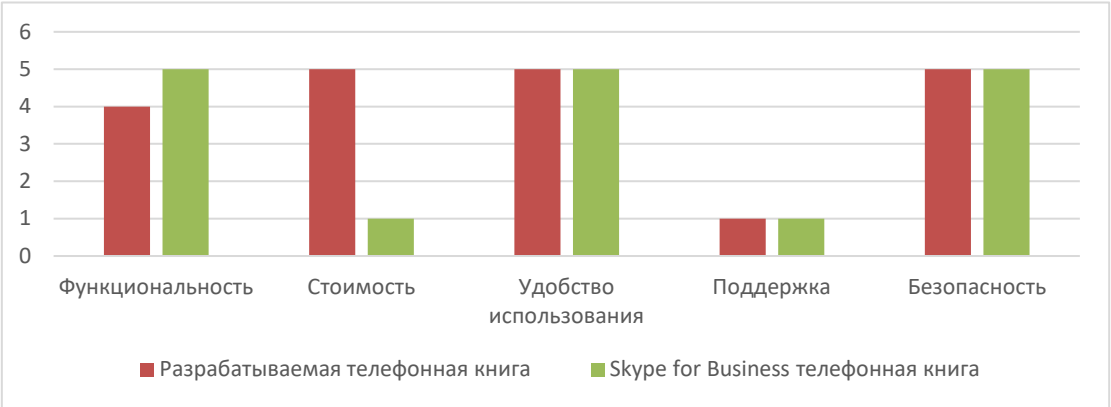

Рисунок 7 – Диаграмма сравнения характеристик модулей IVR

По результатам проведенного анализа различных систем IP-телефонии со встроенными IVR модулями и телефонными книгами предприятия определена задача создания собственного модуля голосового меню и телефонной книги предприятия, отличающихся низкой стоимостью реализации с сохранением основного функционала.

1.4 Требования к функционалу и интерфейсу разрабатываемых компонентов системы IP-телефонии предприятия

Основным требованием к создаваемой системе IP-телефонии является совместная работа двух систем: действующей Skype for Business и внедряемой FreeSWITCH. Это связано с необходимостью сохранения работоспособности системы в переходный период. Впоследствии возможен как полный перевод рабочего процесса на новую платформу, так и сохранение старой системы, как дублирующего, поддерживающего варианта. Окончательное решение будет принято заказчиком после опытной эксплуатации системы IP-телефонии FreeSWITCH. Опциональными требованиями к внедряемой системе со стороны заказчика являются:

- выбор направления звонка,
- переключение звонка на оператора по таймауту,
- проигрывание мелодии в период ожидания соединения,
- перевод звонка на внутренний номер абонента,
- отображение номера, ФИО, названия компании и должности звонящего абонента.

Кроме того, внедряемая система должна отвечать системным и функциональным требованиям, которые были рассмотрены ранее.

Необходимыми условиями реализации решения являются:

- обеспечение сохранности получаемой и передаваемой информации, защита от преднамеренных и непреднамеренных информационных воздействий;
- соответствие требованиям законодательства Российской Федерации о защите информации и о персональных данных.

Функциональные и нефункциональные требования к модулю IVR и телефонной книге предприятия внедряемой системы FreeSWITCH определены в методологии FURPS+.

«Классификация требований к системе FURPS+ была разработана

Робертом Грэйди (Robert Grady) из Hewlett-Packard и предложена в 1992 году. Сокращение FURPS+ расшифровывается как:

а) functionality, функциональность

б) usability, удобство использования

в) reliability, надежность

г) performance, производительность

д) supportability, поддерживаемость

е) ограничения, такие как:

1) ограничения проектирования, design

2) ограничения разработки, implementation

3) ограничения на интерфейсы, interface

4) физические ограничения, physical» [13].

В рамках выполнения работ по проекту выявлены следующие требования (таблицы 5–10).

Таблица 5 – Функциональные требования к разрабатываемым компонентам

Определение правила	Требования
Управление пользователями	Возможность входа в панель управления IVR-меню администратора.
Управление IVR-меню	Возможность просмотра, редактирования и удаления текущих опций и маршрутов IVR-меню, добавления новых.
Голосовое приветствие	Проигрывание голосового приветствия при поступлении звонка от абонента.
IVR-меню	Перевод звонка по заданному маршруту в соответствии с выбранной абонентом опцией.
Оператор	Перевод звонка оператору по истечению времени ожидания выбора опции абонентом.
DISA	Возможность донабора внутреннего номера абонента при условии выбранной опции запуска системы DISA.
Телефонная книга	Отображение на экране IP-телефона или софтфона информации об абоненте (его номер телефона, ФИО, название компании и должности).

Таблица 6 – Требования к удобству использования разрабатываемых компонентов

Определение правила	Требования
Интерфейс управления IVR-меню	Должен быть интуитивно понятным, простым, не перегруженным незначительными деталями для администратора IVR, иметь адаптивный дизайн для работы на различных устройствах и подсказки для облегчения взаимодействия.
Интерфейс IVR-меню	Меню должно быть кратким, интуитивно понятным абонентам, с голосовыми подсказками, переводом оператору по истечении времени ожидания ввода опции меню, возможностью донабора внутреннего номера абонента.
Интерфейс телефонной книги	Должен быть простым, не перегруженным незначительными деталями, иметь возможность отображения данных на оконечных абонентских устройствах.

Таблица 7 – Требования к надежности разрабатываемых компонентов

Определение правила	Требования
Безопасность	Интерфейс управления IVR-меню должен запрашивать авторизацию. Данные должны передаваться по зашифрованным каналам связи.
Частота сбоев	Crash Rate не должен превышать 1-2%.
Допустимое время простоя	Разрабатываемые компоненты должны быть доступны в соответствии с режимом работы предприятия.
Время восстановления работы системы после сбоя	Функциональность разрабатываемых компонентов должна быть восстановлена в течение 10 минут.

Таблица 8 – Требования к производительности разрабатываемых компонентов

Определение правила	Требования
Время отклика системы	Разрабатываемые компоненты должны поддерживать одновременную работу не менее 100 абонентов без существенного влияния на их производительность.
Пропускная способность	Для разрабатываемых компонентов не требуется выделение отдельной полосы пропускания трафика.
Потребление ресурсов	Для разрабатываемых компонентов не требуется выделение высокопроизводительных ресурсов.

Таблица 9 – Требования к поддерживаемости разрабатываемых компонентов

Определение правила	Требования
Установка	Разрабатываемые компоненты должны поставляться в виде исходного кода, при внедрении которого производится внедрение функционала данных модулей.
Документация	Должна быть предоставлена справочная документация по работе с интерфейсом управления IVR-меню и внесения изменений в данные телефонной книги предприятия.
Поддержка	Наличие системы поддержки администратора IVR, телефонной книги предприятия.
Сопровождение	Должен быть разработан регламент обслуживания, включающий документы «Восстановление после сбоев», «Выполнение резервного копирования и восстановления», «Выполнение типовых операций администрирования».
Интеграция	Разрабатываемые компоненты должны интегрироваться с внедряемой системой IP-телефонии FreeSWITCH.

Таблица 10 – Ограничения разрабатываемых компонентов

Группа	Ограничение
Ограничения проектирования	В качестве системы хранения для модуля IVR должна использоваться база данных SQLite, для телефонной книги предприятия база данных PostgreSQL. Схемы сетевого взаимодействия должны быть созданы с использованием Microsoft Visio. Эксплуатационная документация должна быть разработана с использованием Microsoft Word.
Ограничения реализации	Код разрабатываемых компонентов должен быть реализован, используя HTML, JavaScript и T-SQL.
Ограничение интерфейсов	Разрабатываемые компоненты должны работать, используя протокол TCP. Должна обеспечиваться поддержка защищенного протокола HTTPS. Должен использоваться формат данных JSON.
Физические ограничения	Разрабатываемые компоненты должны поддерживать работу как на физическом сервере, так и в виде виртуальной машины популярных систем виртуализации (Microsoft Hyper-V, VMWare).

На рисунке 8 представлена диаграмма вариантов использования UML:

Рисунок 8 – Диаграмма вариантов использования UML

Внешний абонент совершает звонок по номеру, закрепленному в конфигурации системы IP-телефонии FreeSWITCH. Система IP-телефонии выполняет вызов модуля IVR на базе конфигурации закрепленного за ним номера. Модуль предоставляет вызывающему абоненту меню выбора направления звонка – соединение с отделом продаж, отделом технической поддержки или вызов системы DISA для прямого набора номера внутреннего

абонента. В случае, если внешний абонент не совершает ввод на номеронабирателе телефона, звонок переадресуется оператору. Меню выбора направления звонка может редактировать администратор IVR. Внутреннему абоненту предоставляется информация о звонящем из телефонной книги предприятия.

На рисунке 9 представлена диаграмма последовательности, показывающая взаимодействие между объектами в системе в процессе выполнения сценариев.

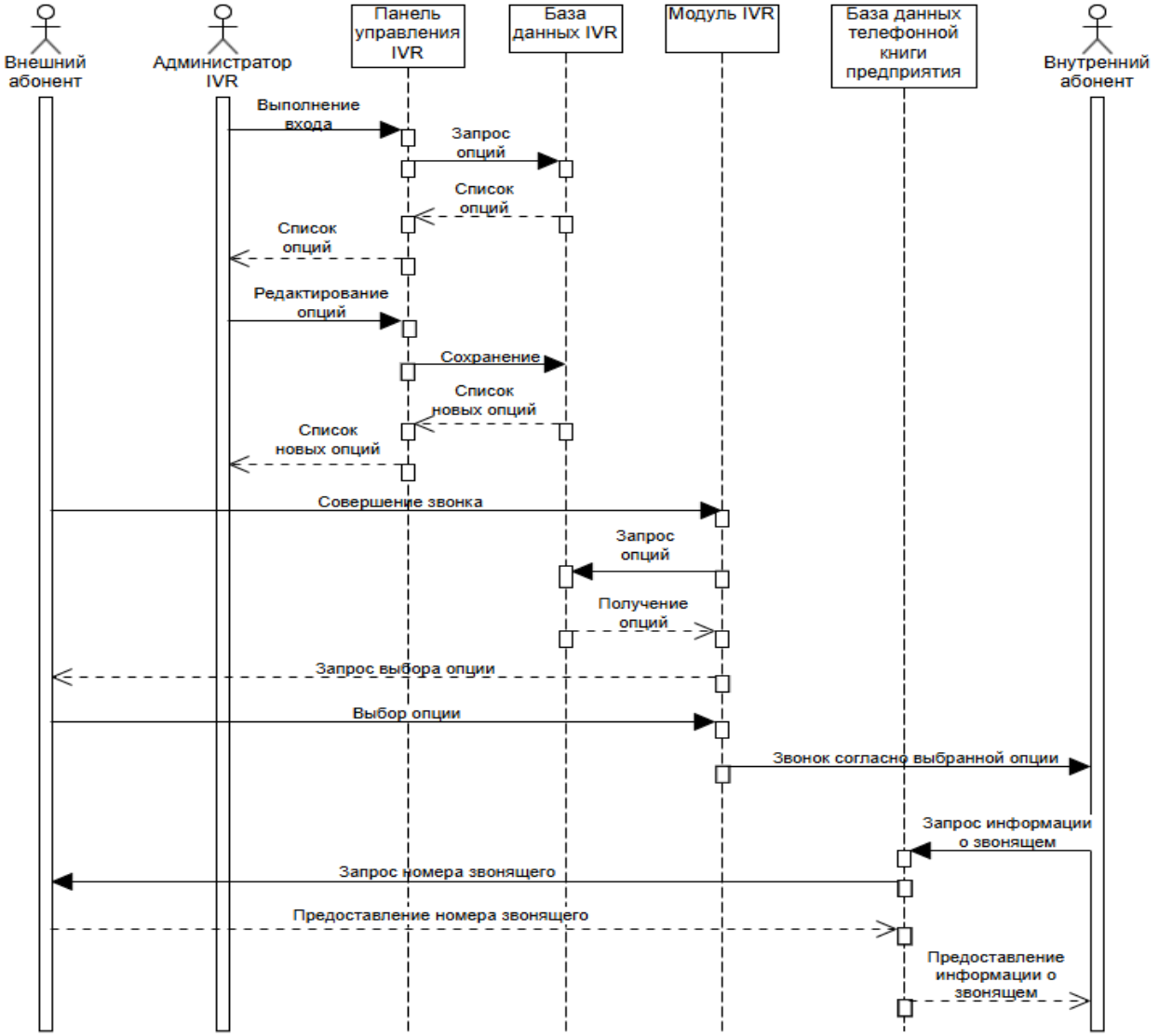

Рисунок 9 – Диаграмма последовательности

Выводы по главе 1

В первой главе:

- определены цели и задачи исследования;
- построена процессная модель функционирования системы IP-телефонии в нотациях IDEF0 и DFD;
- проведен анализ существующих систем для организации корпоративной IP-телефонии;
- рассмотрены преимущества системы IP-телефонии с открытым исходным кодом FreeSWITCH;
- поставлена задача создания собственного модуля IVR и телефонной книги предприятия с сохранением основного функционала;
- определены функциональные и нефункциональные требования к разрабатываемым компонентам в методологии FURPS+;
- построена UML диаграмма вариантов использования, отражающая взаимодействие пользователей с разрабатываемыми компонентами.

Глава 2 Процесс разработки компонентов системы IP-телефонии для предприятия

2.1 Проектирование разрабатываемых компонентов для системы IP-телефонии предприятия

Процесс создания системы IP-телефонии включал в себя проектирование IVR модуля FreeSWITCH, с отдельным компонентом DISA и графическим интерфейсом управления. К проектированию модуля системы применимо объектно-ориентированное проектирование. Так же была спроектирована база данных телефонной книги предприятия.

Исходя из требований к функционалу и интерфейсу разрабатываемых компонентов системы IP-телефонии предприятия, можно сделать вывод, что модуль IVR включает интерфейс управления, базу данных и логику работы с абонентами через FreeSWITCH. Интерфейс управления состоит из двух сервисов – отдельного фронтенд, работающего с клиентом в виде веб-браузера администратора и бэкенд, работающего с базой данных. Все сервисы, включая базу данных могут располагаться на разных серверах. В данном случае архитектура разрабатываемых компонентов одновременно является клиент-серверной и сервисно-ориентированной.

Клиент-серверная архитектура – это модель организации вычислительных систем, в которой задачи распределены между клиентами и серверами. В такой архитектуре клиент, обычно являющийся пользователем или программой, запрашивает услуги или ресурсы у сервера, который отвечает на запросы, предоставляя необходимые данные или функциональность. Разделение на клиентскую и серверную части позволяет распределять нагрузку и улучшать масштабируемость. Среди преимуществ клиент-серверной архитектуры следует выделить хранение всех данных на сервере и возможность объединить клиентов, использующих разные аппаратные платформы и операционные системы.

Сервис-ориентированная архитектура (SOA) предполагает, что функции приложения предоставляются в виде независимых сервисов. Эти сервисы могут взаимодействовать друг с другом через стандартизированные интерфейсы и протоколы такие как SOAP и REST. Основная цель SOA — обеспечить возможность повторного использования и гибкость в разработке и интеграции программных компонентов. Сервисы могут работать на различных платформах и быть написаны на разных языках программирования. К преимуществам сервис-ориентированной архитектуры относится возможность обновления сервисов без необходимости обновления всей системы и легкость масштабирования под меняющиеся нагрузки.

Логическое моделирование модуля IVR в FreeSWITCH включает в себя проектирование модуля, который позволяет пользователям взаимодействовать с телефонной системой путем нажатия клавиш номеронабирателя телефона, а администратору управлять IVR-меню.

Основные компоненты логического моделирования IVR в FreeSWITCH:

- сценарий IVR (определение сценария, который используется для взаимодействия с пользователем, включает последовательность вопросов и ответов, а также возможные пути, которые может выбрать пользователь);
- голосовые сообщения (приветствия, инструкции, проигрывание музыки в период ожидания);
- обработка ввода пользователя (ввод пользователем своих ответов осуществляется путем нажатия клавиш номеронабирателя телефона – DTMF (Dual-Tone Multi-Frequency);
- логика обработки (разработка логики для обработки пользовательского ввода и принятия решений на основе полученных данных - переход к другим вопросам, выполнение действий (например, перевод звонка на оператора) или завершение сеанса);

– обработка ввода администратора (просмотр, редактирование, добавление и удаление пунктов меню в графическом интерфейсе управления);
– интеграция с базами данных и API (возможность интеграции с внешними системами для получения информации или выполнения действий на основе пользовательского ввода).

Разработанный модуль IVR с компонентом DISA и графическим интерфейсом управления, взаимодействует с администратором посредством веб-интерфейса и системой IP-телефонии FreeSWITCH посредством команд интерфейса сценариев API.

Большинство команд FreeSWITCH API реализовано в модуле mod_commands. Так же некоторые модули FreeSWITCH добавляют команды, которые выполняются через FSAPI. Механизм FSAPI принимает строку текста в качестве входных данных и выполняет определенное действие. Возвращаемое значение также является строкой, которая может быть любого размера, от одного символа до нескольких страниц текста, в зависимости от функции, которая была вызвана входной строкой. Основным преимуществом функций FSAPI является то, что модуль может использовать их для вызова процедур в другом модуле без прямой ссылки на фактический скомпилированный код (что позволяет избежать внезапных несовместимостей и сбоев). Наиболее наглядным примером является интерфейс командной строки FreeSWITCH или CLI, который использует функции FSAPI для передачи команд API FreeSWITCH.

Рассмотрим выполнение команды отображения статуса системы status в командной строке FreeSWITCH CLI.

«При вводе команды status и нажатия клавиши Enter, слово status используется для поиска функции status FSAPI из модуля, в котором она реализована. Затем вызывается базовая функция (с передачей ей аргументов, если они были введены, в данном случае ни одного), и ядро запрашивает сообщение о статусе. После получения данных о статусе вывод записывается

в поток, который печатает строку» [16, с. 20].

FreeSWITCH является модульной системой, которая обеспечивает интерфейс разработчикам для масштабирования системы вокруг стабильного обменного ядра.

Разработанная база данных телефонной книги предприятия взаимодействует с системой IP-телефонии используя встроенный модуль mod_cidlookup FreeSWITCH.

mod_cidlookup — API поиска Caller*ID. Этот модуль FreeSWITCH позволяет искать сопоставление номера и имени в локальной базе данных.

«Модули системы критически важны и добавляют некоторые из ключевых функций, которые делают FreeSWITCH мощной платформой. Основная роль модулей FreeSWITCH заключается в том, чтобы взять определенные общие технологии связи и нормализовать их в общую абстрактную сущность, которая называется сеансом. Сеанс представляет собой соединение между FreeSWITCH и определенным протоколом. Существует несколько модулей, которые поставляются с FreeSWITCH. Они реализуют несколько протоколов, таких как SIP, H.323, Jingle, Verto, WebRTC и другие» [16, с. 11].

Для различных задач, таких как обработка голосовых звонков, автоматизация взаимодействия с пользователями и реализация голосовых меню используется языковой модуль FreeSWITCH.

«Языковой модуль - отдельный тип модуля, который не имеет прямого интерфейса с FreeSWITCH, как файлы и конечные точки, но все же предлагает чрезвычайно мощное соединение с существующей технологией. Языковые модули встраивают язык программирования, такой как Lua, JavaScript, Perl, C# и т. д., в FreeSWITCH, и передают функциональность между ядром и средой выполнения языка. Языковые модули обычно регистрируются в ядре с интерфейсом приложения и интерфейсом FSAPI и выполняются из диалплана. Модули языка предлагают множество возможностей. Используя модули языка, можно создавать приложения для общения в реальном времени на

стандартном языке программирования, используя его библиотеки для манипуляций данными и устаревшего интерфейса» [16, с. 22].

FreeSWITCH поддерживает несколько языков программирования для написания сценариев и обработки событий. Основные из них: Lua, JavaScript, Python, Perl, C/C++.

Кроме того, FreeSWITCH может интегрироваться с другими языками через API и модули, что делает его очень гибким инструментом для разработки VoIP-приложений.

«В объектно-ориентированном девелопменте программных продуктов много лет успешно используется унифицированный язык моделирования UML (Unified Modeling Language), применяемый на этапах анализа, проектирования и разработки предметно-ориентированных информационных систем. Язык UML является общецелевым языковым средством визуального моделирования, который был разработан для спецификации, графической визуализации, проектирования, разработки и документирования компонентов прикладного программного обеспечения и предметно-ориентированных информационных систем» [10, с. 6].

«Диаграмма компонентов (Component diagram) — статическая структурная диаграмма, показывает разбиение программной системы на структурные компоненты и связи (зависимости) между компонентами. В качестве физических компонентов могут выступать файлы, библиотеки, модули, исполняемые файлы, пакеты и т. п.

Диаграмма компонентов описывает особенности физического представления системы. Диаграмма компонентов позволяет определить архитектуру разрабатываемой системы, установив зависимости между программными компонентами, в роли которых могут выступать исходный, бинарный и исполняемый коды» [14, с. 36].

На рисунке 10 представлена диаграмма компонентов:

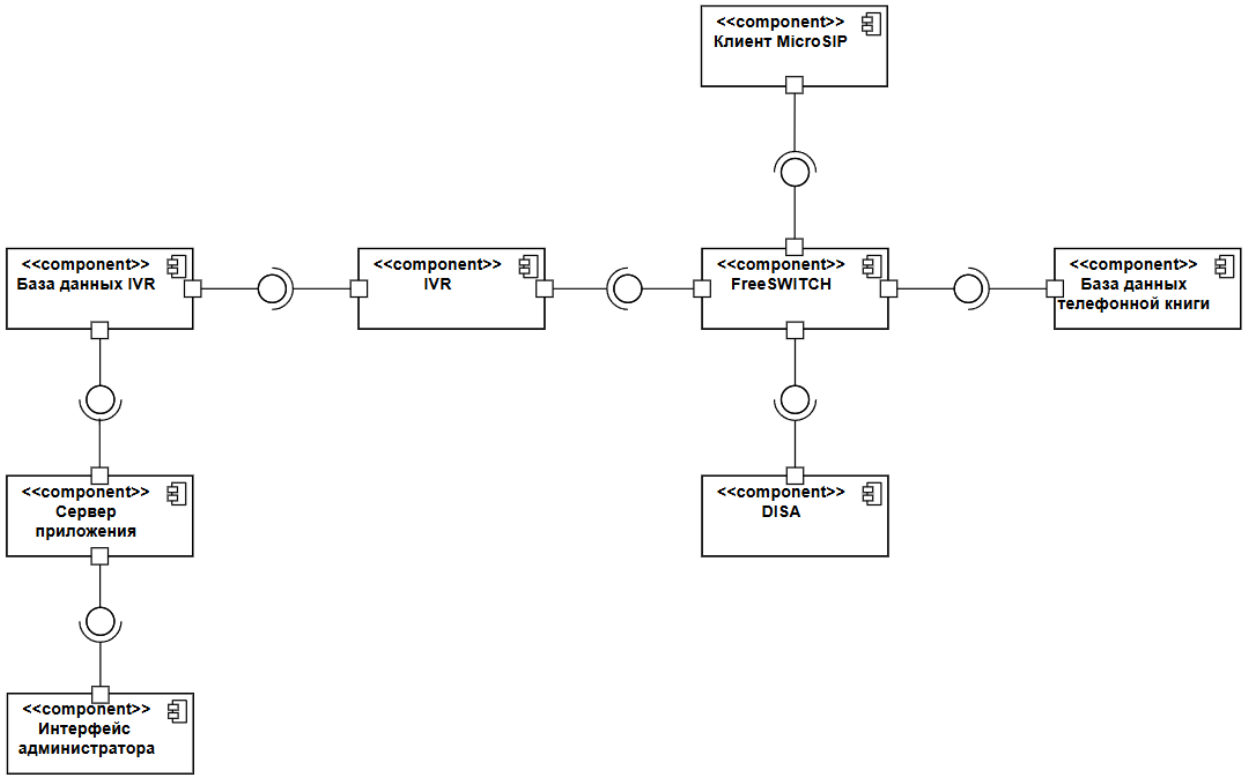

Рисунок 10 – Диаграмма компонентов

«Диаграмма развертывания в UML моделирует физическое развертывание артефактов на узлах. Например, чтобы описать веб-сайт, диаграмма развертывания должна показывать, какие аппаратные компоненты («узлы») существуют (например, веб-сервер, сервер базы данных, сервер приложения), какие программные компоненты («артефакты») работают на каждом узле (например, веб-приложение, база данных), и как различные части этого комплекса соединяются друг с другом (например, JDBC, REST, RMI).

Диаграммы развертывания используются для моделирования статического вида системы с точки зрения развертывания. Это представление в первую очередь обращено на распределение, поставку и установку частей, из которых состоит физическая система» [14, с. 52].

На рисунке 11 представлена диаграмма развертывания.

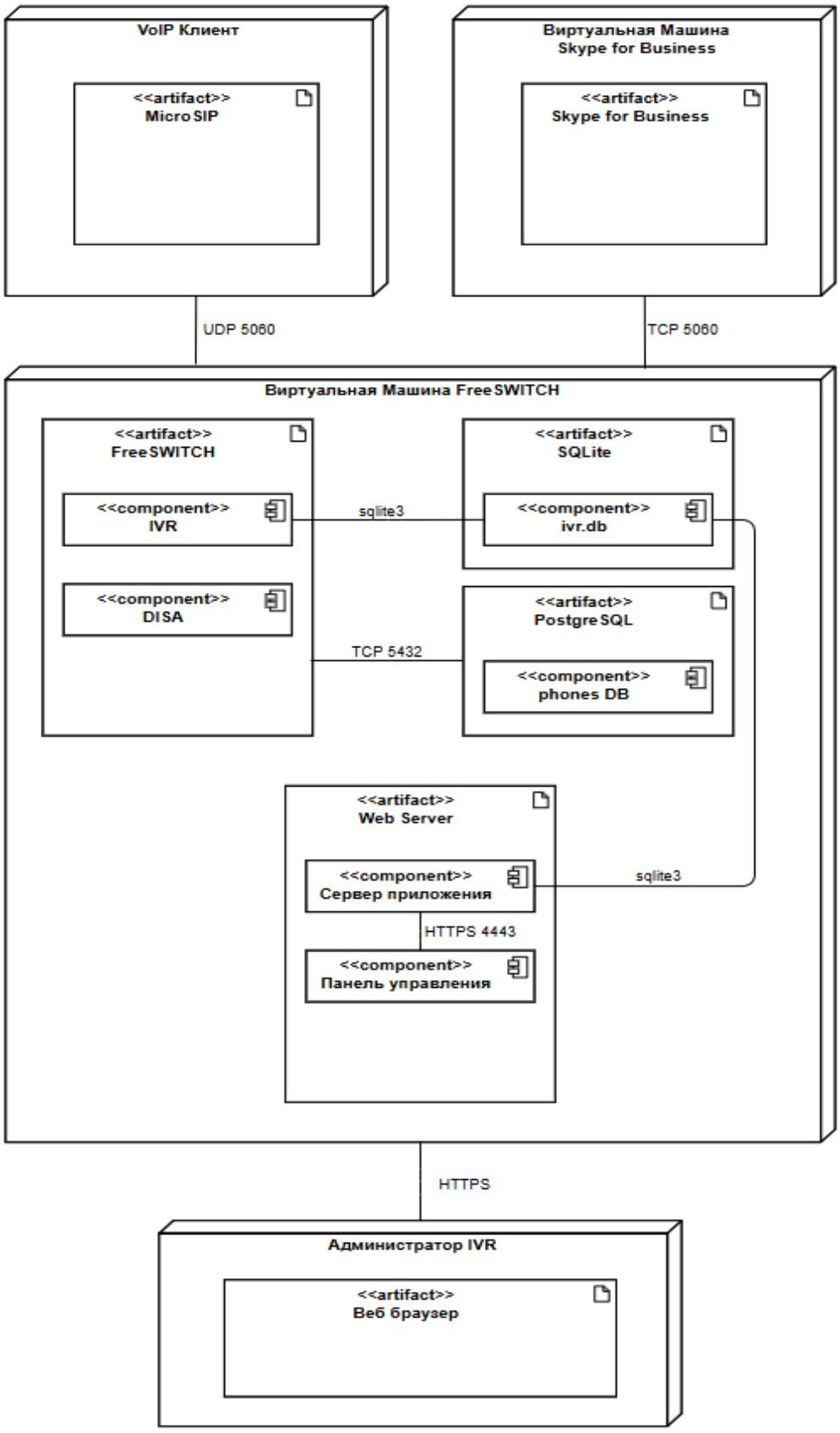

Рисунок 11 – Диаграмма развертывания

На рисунке 12 представлена диаграмма пользовательского интерфейса:

Рисунок 12 - Диаграмма пользовательского интерфейса

На рисунке 13 представлен прототип модели интерфейса:

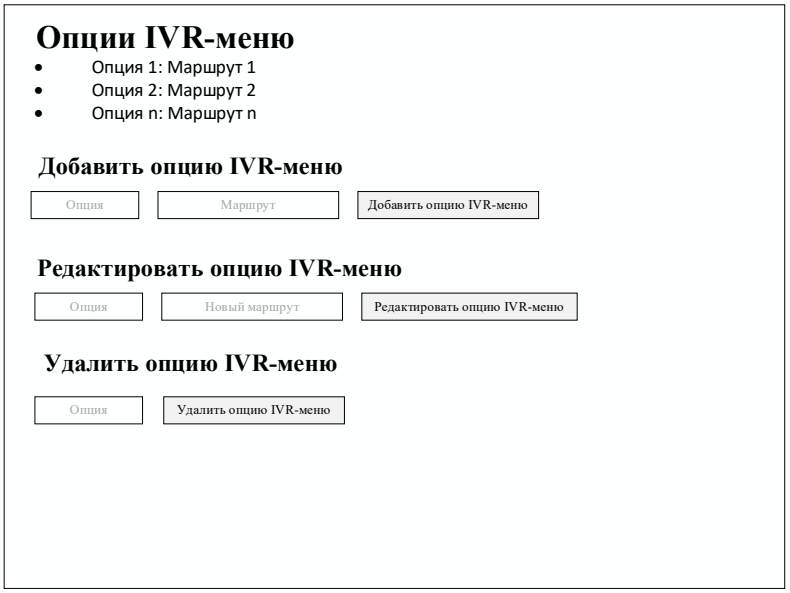

Рисунок 13 - Прототип пользовательского интерфейса

Разработанный модуль IVR с компонентом DISA и графическим интерфейсом управления, предполагает хранение данных о выбираемых абонентом опциях и соответствующим им маршрутам в базе данных. Администратор IVR может вносить изменения в эту базы данных, используя веб-интерфейс.

Проектируемая база данных для модуля IVR состоит из одной таблицы menu_options (опции меню) с двумя столбцами option (varchar(10)) и route (varchar(120)).

Типы данных, которые используются для полей таблицы menu_options базы данных IVR, приведены ниже (таблица 11).

Таблица 11 – Типы данных для полей таблицы menu_options

Имя	Тип данных	Ключ	Null
option	varchar(10)	–	–
route	varchar(120)	–	–

Поля option и route используют тип данных varchar. Этот тип данных позволяет хранить строки переменной длины до 65535 символов различной длины, разрешает использование специальных символов, которые могут присутствовать в данных.

Текущая система IP-телефонии предполагает использование телефонной книги предприятия. При поступлении звонка от внутреннего абонента, кроме номера вызывающего абонента высвечивается его ФИО, название компании и должность. Внедряемая система IP-телефонии не имеет подобного функционала, но он должен быть разработан согласно бизнес-требованиям. Поэтому в рамках исследования по теме ВКР была разработана база данных телефонной книги предприятия для внедряемой системы IP-телефонии.

Логическое проектирование базы данных для FreeSWITCH включает несколько ключевых этапов, таких как:

– «сопоставление концептуальной модели с логической моделью,
– проверка логической модели с помощью нормализации,
– проверка ограничений целостности логической модели,
– проверка логической модели на соответствие требованиям пользователя» [5, с. 122].

База данных телефонной книги предприятия phones состоит из двух таблиц:

а) data (данные абонентов) со столбцами:
   1) id (SERIAL) – идентификатор данных абонента и основной ключ таблицы,
   2) name (text) – ФИО абонента,
   3) org (text) – название компании и должности абонента.

б) numbers (телефонные номера абонентов) со столбцами:
   1) id (SERIAL) – идентификатор номера абонента и основной ключ таблицы,
   2) data_id (integer) – внешний ключ таблицы data,
   3) number (text) – телефонный номер абонента.

Типы данных, которые используются для полей таблицы data базы данных phones, приведены ниже (таблица 12):

Таблица 12 – Типы данных для полей таблицы data

Имя	Тип данных	Ключ	Null
id	serial	primary key	not null
name	text	–	–
org	text	–	–

Типы данных, которые используются для полей таблицы numbers базы данных phones, приведены ниже (таблица 13):

Таблица 13 – Типы данных для полей таблицы numbers

Имя	Тип данных	Ключ	Null
id	serial	primary key	not null
data_id	integer	foreign key	–
number	text	–	–

Тип данных serial для поля id является псевдотипом. Он позволяет создать автоинкрементный целочисленный столбец, обычно используемый для первичных ключей. Он автоматически генерирует последовательность чисел, которые увеличиваются на единицу для каждой новой строки. Эта функция упрощает процесс создания уникальных идентификаторов для каждой строки в таблице.

Поля name, org и number используют тип данных text. Он позволяет хранить строки различной длины, разрешает использование специальных символов, которые могут присутствовать в данных.

Поле data_id имеет тип данных integer – целочисленный тип. Используется для внешнего ключа.

Связь между таблицами организована при помощи внешнего ключа (data_id) в таблице телефонных номеров абонентов (numbers), который связывает поле данной таблицы с основным ключом (id) таблицы данных абонента (data). Через внешний ключ так же контролируется удаление данных в таблице телефонных номеров абонентов (numbers): при удалении данных абонента (name), удаляется телефонный номер данного абонента (number).

Логическая модель данных базы данных телефонной книги предприятия, разрабатываемой в рамках исследования по теме ВКР, представлена на диаграмме «сущность – связь» (ERD) ниже. Диаграмма построена с использованием бесплатного онлайн-сервиса online.visual-paradigm.com.

ERD в нотации Чена отражена на рисунке 14.

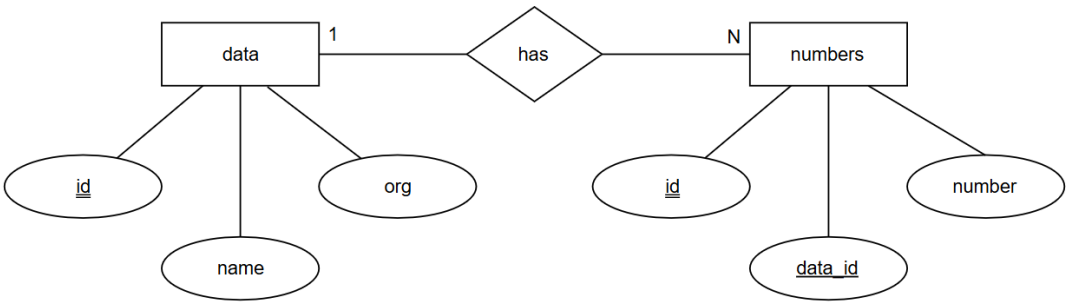

Рисунок 14 – ER-диаграмма в нотации Чена

«Логическое проектирование переводит программно-независимую концептуальную модель в программно-зависимую модель. Последний шаг в процессе логического проектирования заключается в проверке всех определений логической модели по всем требованиям к данным, транзакциям и безопасности пользователя» [5, с. 123].

Созданная ER-диаграмма представлена на рисунке 15:

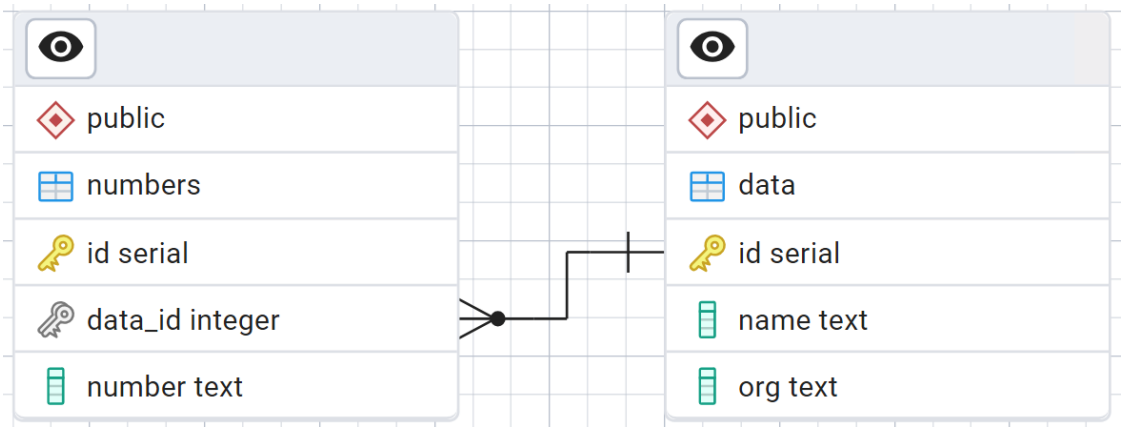

Рисунок 15 – ER-диаграмма

Внешний входящий звонок обрабатывается модулем IVR по следующему алгоритму (рисунок 16):

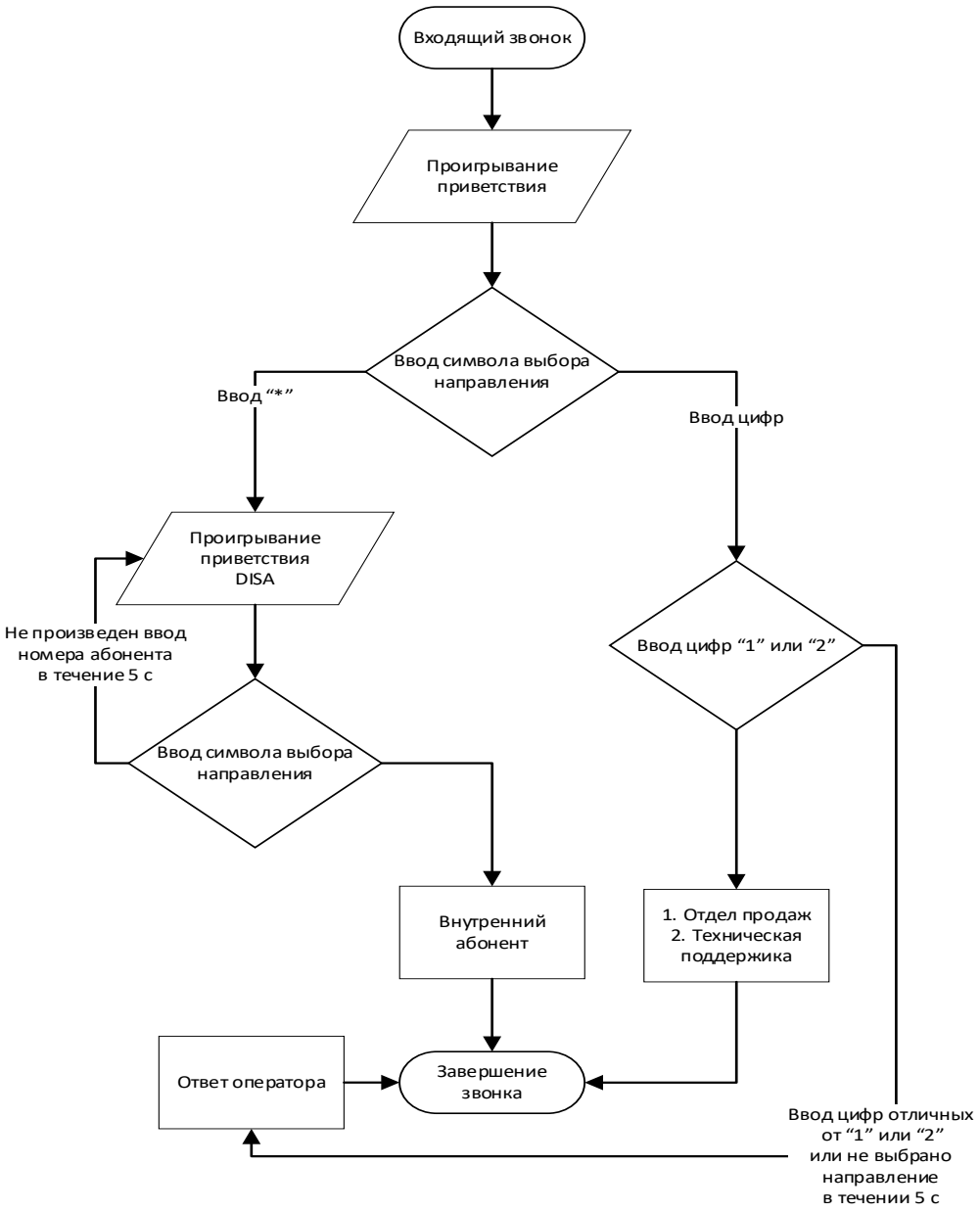

Рисунок 16 – Алгоритм входящего звонка, проходящего через модуль IVR

При поступлении входящего звонка вызывающему абоненту проигрывается приветственное сообщение, предлагающее выбрать направление звонка: цифра «1» – для соединения с отделом продаж, цифра «2» – для соединения с технической поддержкой, символ «*» – для выбора и соединения с внутренним абонентом. В случае, если вызывающий абонент не выбрал направление звонка (не нажал требуемую клавишу с цифрой или символом) в течение 5 секунд, звонок маршрутизируется оператору. При нажатии клавиши с цифрой «1» звонок маршрутизируется внутреннему

абоненту отдела продаж, при нажатии клавиши с цифрой «2» звонок маршрутизируется внутреннему абоненту отдела технической поддержки. Если будет нажата любая другая цифра, звонок маршрутизируется оператору. В случае, если вызывающий абонент нажмет клавишу с символом «*», звонок маршрутизируется на систему DISA, используя которую вызывающий абонент может набрать внутренний номер абонента для соединения с ним. Если внутренний номер не будет набран в течение 5 секунд, система DISA сообщает о некорректном вводе и предлагает заново ввести внутренний номер абонента.

Управление модулем IVR осуществляется администратором по следующему алгоритму (рисунок 17):

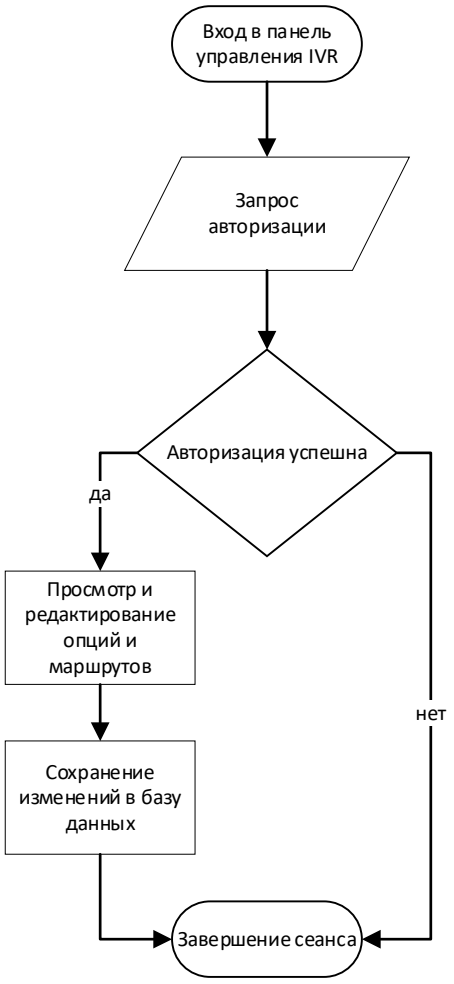

Рисунок 17 – Алгоритм управления администратором модуля IVR

При входе администратора в панель управления IVR, веб-интерфейс запрашивает авторизацию. Если авторизация успешна, администратор имеет возможность просматривать, а также редактировать опции меню и маршруты с сохранением изменений в базу данных, после чего завершить сеанс работы. В случае неуспешной авторизации сеанс работы завершится до предоставления возможности администратору просматривать и редактировать опции меню и маршруты.

Работа телефонной книги предприятия осуществляется по следующему алгоритму (рисунок 18):

Рисунок 18 – Алгоритм работы телефонной книги предприятия

При поступлении входящего звонка, FreeSWITCH определяет номер звонящего и отправляет запрос в базу данных телефонной книги предприятия для получения ФИО, названия компании и должности соответствующих номеру звонящего, в результате чего на экране IP-телефона или софтфона

внутреннего абонента высвечивается полученная из базы данных телефонной книги предприятия информация, включающая номер, ФИО, название компании и должности звонящего. Если номер звонящего отсутствует в базе данных, на экране IP-телефона или софтфона внутреннего абонента высвечивается только номер звонящего.

«Диаграмма деятельности — инструмент моделирования, основной целью которого является создание стандартного набора условных обозначений, понятных всем бизнес-пользователям. Бизнес-пользователи включают в себя бизнес-аналитиков, создающих и улучшающих процессы, технических разработчиков, ответственных за реализацию процессов, и менеджеров, следящих за процессами и управляющих ими» [14, с. 44].

На рисунках 19–21 представлены диаграммы деятельности:

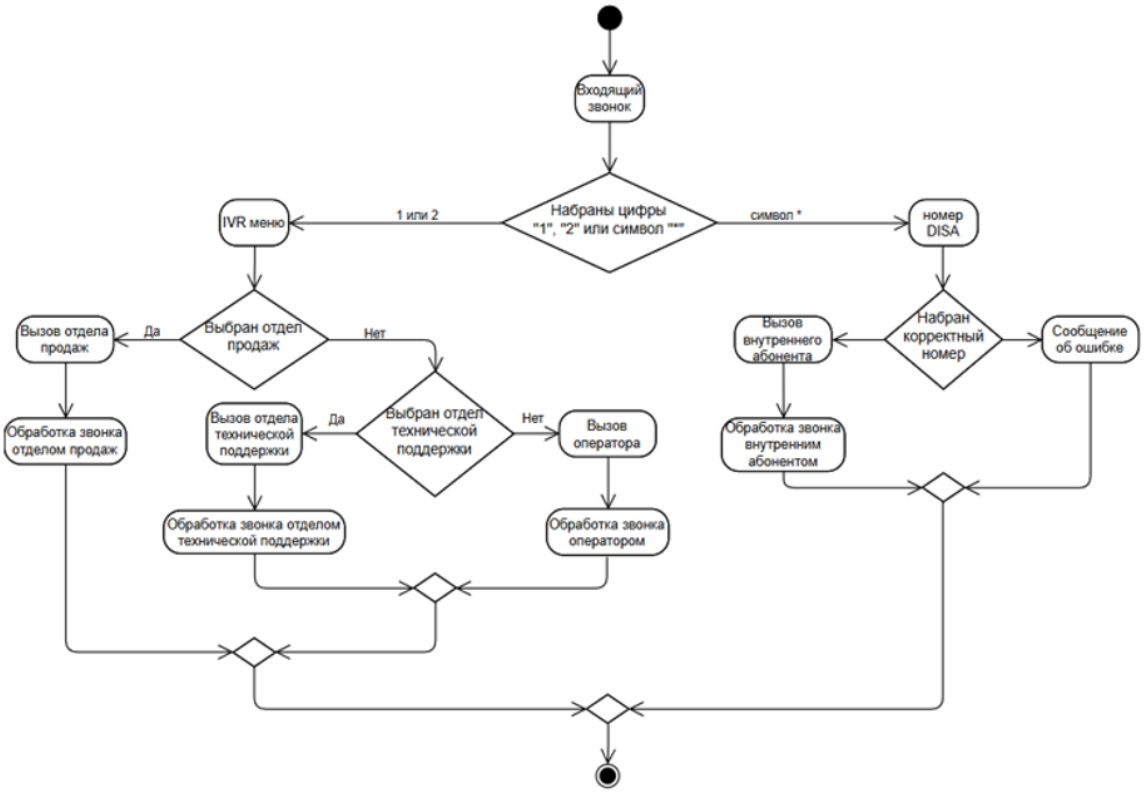

Рисунок 19 – Диаграмма деятельности модуля IVR

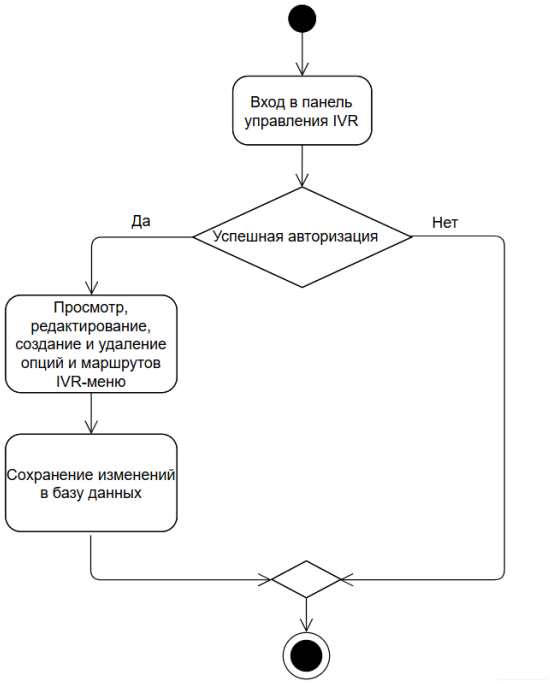

Рисунок 20 – Диаграмма деятельности управления модулем IVR

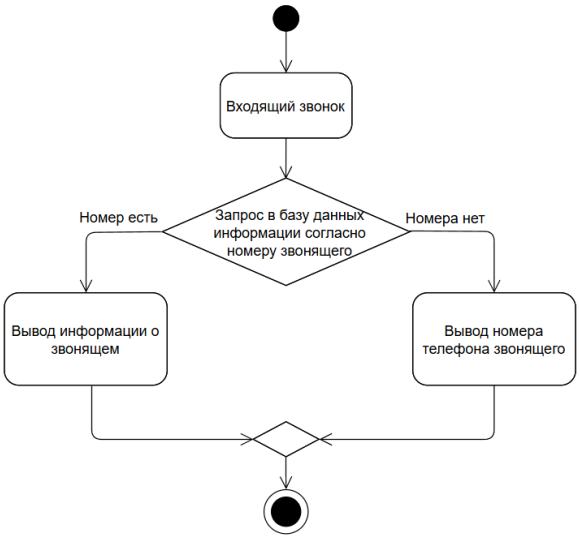

Рисунок 21 – Диаграмма деятельности телефонной книги предприятия

На рисунке 22 представлена схема обмена данными.

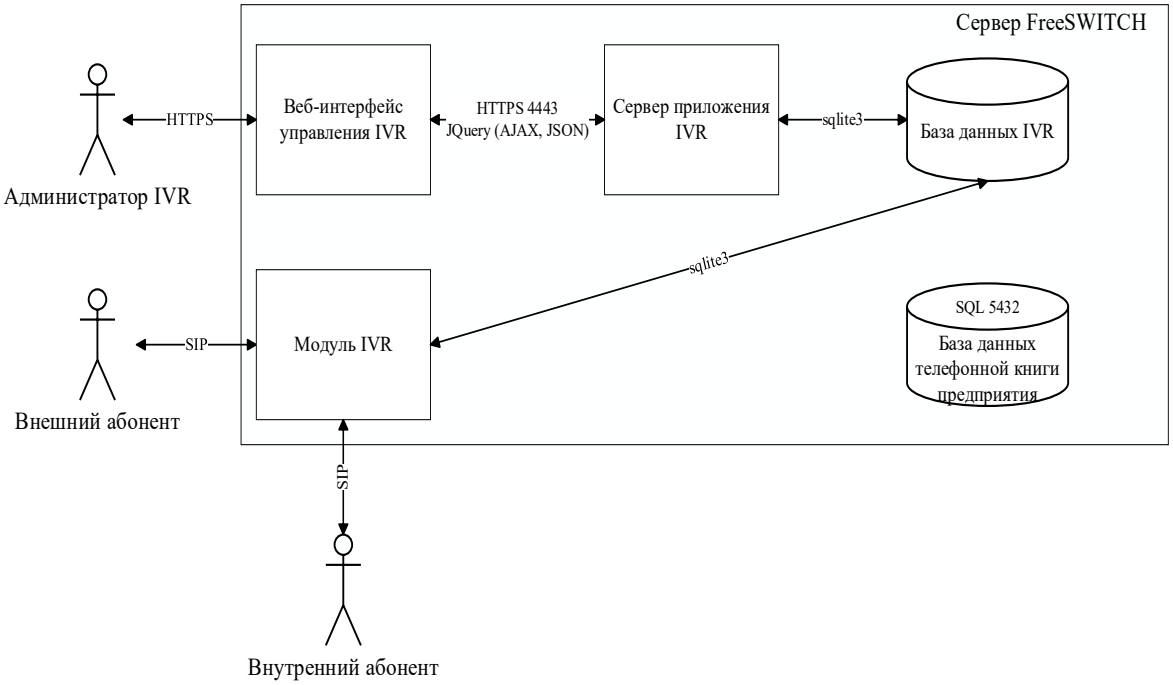

Рисунок 22 – Схема обмена данными

Администратор IVR подключается к веб-интерфейсу управления используя защищенный канал данных при помощи протокола HTTPS по стандартном порту (TCP 443). Веб-интерфейс управления подключается к серверу приложения IVR, используя защищенный канал данных при помощи протокола HTTPS по порту TCP 4443. Сервер приложения IVR представляет собой RESTful сервер, предоставляющий REST API клиенту REST – веб-интерфейсу управления IVR. Для обмена данными между веб-интерфейсом и сервером приложения IVR используются AJAX и JSON через библиотеку JQuery. Сервер приложения обменивается с базой данной IVR используя API SQLite через библиотеку sqlite3. Абоненты взаимодействует с модулем IVR посредством протокола SIP по стандартному порту (UDP 5060) через систему IP-телефонии FreeSWITCH, которая является платформой для всех разрабатываемых компонентов. База данных телефонной книги взаимодействует с платформой, используя протокол SQL по порту TCP 5432.

2.2 Выбор технологий и инструментов для разработки компонентов системы IP-телефонии

Важным фактором при выборе языка программирования является конкретные потребности проекта.

В рамках исследования по теме ВКР рассмотрено несколько популярных языков программирования: Python, C++, JavaScript; проведен их сравнительный анализ по различным критериям с целью определения соответствия потребностям проекта.

Python славится своей простотой и читаемостью. Синтаксис Python напоминает английский язык, что делает его легким в освоении. Этот язык идеален для начинающих разработчиков.

Производительность: Python, как язык интерпретации, может быть менее производительным по сравнению с компилируемыми языками, но его удобство в использовании часто перевешивает этот недостаток. Python активно используется в области науки о данных и искусственного интеллекта.

Популярность: Python занимает верхние строчки рейтинга языков программирования благодаря своей универсальности, применяется в веб-разработке, науке о данных, искусственном интеллекте и других областях.

C++ считается языком высокой сложности. Его мощные возможности предоставляют разработчикам полный контроль над ресурсами компьютера, но требуют более глубокого понимания.

Производительность: C++ известен своей высокой производительностью и эффективностью. Этот язык широко используется в системном программировании.

Популярность: C++ остается популярным в высокопроизводительных областях разработки, несмотря на более сложный синтаксис по сравнению с некоторыми современными языками.

JavaScript является относительно легким для изучения языком. Он используется в основном для веб-разработки, что делает его ключевым для

фронтенда и бэкенда веб-приложений.

Производительность: JavaScript предоставляет высокую производительность в веб-среде. С появлением средств оптимизации и новых версий языка его применение расширилось и в другие области разработки.

Популярность: JavaScript — один из самых популярных языков программирования в мире, особенно благодаря развитию фронтенд-фреймворков и библиотек, таких как React и Angular.

Используемая платформа FreeSWITCH для разрабатываемых компонентов накладывает определенные ограничения на выбор языка программирования. Хотя платформа и поддерживает достаточно большое количество языков программирования для разработки под нее решений, наиболее популярным остается JavaScript, движок которого доступен в FreeSWITCH «из коробки».

«Языковой модуль JavaScript FreeSWITCH изначально использовал движок Mozilla SpiderMonkey JavaScript (ECMAScript). Он поддерживает все стандартные элементы языка JavaScript, например, циклы «for» и «while», регулярные выражения и т. д.

На данный момент FreeSWITCH поддерживает движок Google V8 JavaScript (ECMAScript). Он предоставляется из модуля mod_v8. В текущем релизе FreeSWITCH mod_v8 является движком JavaScript по умолчанию» [20].

Кроме удобства работы с FreeSWITCH, следует отметить, что для разработки графического интерфейса управления IVR на базе веб-сайта так же будет использоваться язык программирования JavaScript, который является наиболее подходящим для этой задачи.

«Возросшая скорость выполнения кода JavaScript позволила использовать этот язык в масштабируемых проектах, а также в высоконагруженных сервисах. Одним из современных примеров использования JavaScript является Node.js. Программная платформа разработана Райаном Далем на языках C, C++ и JavaScript и использует движок Google V8. Node.js позволяет разрабатывать серверные приложения (особенно

веб-приложения, взаимодействующие с клиентом в реальном времени) с использованием JavaScript и, в отличие от браузерного JavaScript, позволяет обращаться к файловой системе. Это ознаменовало появление языка JavaScript за пределами веб-браузеров» [6, с. 270].

«Node.js стала платформой, которая позволила использовать JavaScript в качестве языка программирования общего назначения для многих разновидностей платформ, включая небольшие встроенные устройства. Модули ввода-вывода Node.js в комбинации с высокоэффективной машиной V8 были сравнимы по возможностям и часто превосходили по эффективности другие динамические языки для создания приложений, такие как Python и Ruby» [9, с. 18].

В JavaScript используется модель прототипного наследования: каждый объект наследует свойства и методы объекта - прототипа. В JavaScript нет классической реализации классов. Классы это лишь надстройка над прототипным наследованием. Классы являются особым типом функций.

«Среди разработчиков ПО часто употребляется термин «фреймворк» (framework), который в какой-то мере подменяет термин «технология» и более привычное понятие — «набор для разработки программного обеспечения» (software development kit). Фреймворки — это сложные среды разработки программного обеспечения. Они включают множество разноплановых компонент. В их состав входят: отладчики, компиляторы, библиотеки с описаниями интерфейсов, редакторы и другие компоненты. Все они предназначены для упрощения разработки приложений. Фреймворки составляют основу процесса разработки, и их использование делает этот процесс более простым и интересным. Без фреймворков каждое приложение пришлось бы писать с нуля. Язык JavaScript уже давно вышел за рамки средства для «оживления» HTML-страниц. Наряду с JavaScript ряд фреймворков ориентирован на использование языка TypeScript от Microsoft, который отличается большей строгостью и находит своих пользователей среди сторонников традиционного объектно-ориентированного подхода,

характерного для языков C++ или Java. Однако отметим, что TypeScript транслируется в JavaScript, который, в конечном счёте, и реализуется интерпретатором» [4, с. 79].

В рамках исследования по теме ВКР рассмотрено несколько популярных фреймворков и библиотек, которые могут ускорить разработку, проведен их сравнительный анализ по различным критериям с целью определения целесообразности использования в коде разрабатываемых компонентов.

Django — это высокоуровневый фреймворк для веб-разработки на языке Python. Он следит за принципом "батарейки включены" и предлагает множество встроенных функций, таких как ORM, аутентификация, админ-панель и многое другое. Django идеально подходит для создания сложных и масштабируемых приложений. Одним из ключевых преимуществ Django является его мощная система администрирования, которая позволяет быстро создавать интерфейсы для управления данными. Это делает Django отличным выбором для проектов, требующих сложного управления контентом, таких как новостные сайты или блоги.

Удобство использования: легко освоить благодаря отличной документации и встроенным инструментам. Django предоставляет множество готовых решений, что делает его очень удобным для новичков.

Производительность: Высокая производительность благодаря встроенным оптимизациям и кешированию. Django использует мощные механизмы кеширования, которые позволяют значительно ускорить обработку запросов.

Популярность: Большое и активное сообщество. Django имеет множество готовых решений и библиотек, что делает его очень удобным для разработчиков.

Laravel — это фреймворк для веб-разработки на языке PHP. Он предлагает множество встроенных инструментов и библиотек, таких как ORM Eloquent, система маршрутизации и шаблонизатор Blade. Laravel

известен своей элегантностью и простотой использования. Одним из ключевых преимуществ Laravel является его мощная система миграций, которая позволяет легко управлять изменениями в базе данных.

Удобство использования: легко освоить благодаря отличной документации и сообществу. Laravel имеет большое и активное сообщество, что обеспечивает наличие множества обучающих материалов и готовых решений.

Производительность: Средняя производительность, но предлагает множество встроенных инструментов для улучшения. Laravel включает в себя мощные механизмы кеширования и оптимизации запросов к базе данных.

Популярность: очень большое и активное сообщество. Laravel имеет множество готовых решений и библиотек, что делает его очень удобным для разработчиков.

Express.js — это минималистичный фреймворк для Node.js. Он предоставляет базовые инструменты для создания серверных приложений и позволяет разработчикам выбирать дополнительные модули по мере необходимости. Express.js идеально подходит для создания быстрых и легковесных приложений. Одним из ключевых преимуществ Express.js является его асинхронная природа, что делает его отличным выбором для приложений с высокой нагрузкой.

Удобство использования: Прост в использовании, но требует больше ручной работы. Express.js предоставляет базовые инструменты, что делает его очень гибким, но требует больше усилий для настройки.

Производительность: Высокая производительность благодаря асинхронной природе Node.js. Express.js использует асинхронные операции ввода-вывода, что делает его очень быстрым и эффективным.

Популярность: Большое и активное сообщество. Express.js имеет множество готовых решений и библиотек, что делает его очень удобным для разработчиков.

Учитывая использование языка программирования JavaScript в рамках разработки компонентов системы IP-телефонии FreeSWITCH, а также требования проекта, в коде использовался фреймворк Express.js, а также библиотека Core.DB платформы FreeSWITCH, которые упрощают разработку клиент-серверной модели. В коде так же используются вспомогательные библиотеки, модули и middleware (связующее программное обеспечение):

- библиотека «JQuery» - работа с AJAX, JSON;
- middleware «Body parser» - работа с запросами HTTP;
- модуль «sqlite3» - работа с базой данных SQLite;
- middleware «CORS» - работа с заголовками HTTP;
- модуль «fs» - работа с файловой системой;
- модуль «https» - работа с HTTPS.

Значительная часть программного обеспечения, будь то ПО для банков, страховых организаций, фондов, денежного рынка, платежных сервисов, разрабатывается на основе базы данных. Выбор системы управления базами данных (СУБД) зависит от ряда факторов, включая тип данных, производительность, масштабируемость, стоимость, безопасность и другие специфические требования проекта или бизнеса.

В рамках исследования по теме ВКР рассмотрено несколько СУБД для разрабатываемых компонентов, такие как MySQL, PostgreSQL и SQLite.

MySQL - одна из самых популярных открытых СУБД. Поддерживает различные операционные системы. Хорошо документирована и имеет обширное сообщество пользователей. Отлично подходит для небольших и средних проектов.

PostgreSQL - открытая СУБД, но с большим уклоном в расширенные функции. Поддерживает множество типов данных и сложные запросы. Высокая надежность и целостность данных. Хорошо подходит для больших проектов, где требуется масштабируемость и поддержка сложных структур данных.

SQLite — это система управления реляционными базами данных с открытым исходным кодом. Используя этот инструмент, можно изменять размеры файлов на меньшие с меньшим количеством метаданных. Он надежен и в основном используется для мобильных приложений, поэтому является наиболее используемым движком базы данных. Он может обрабатывать HTTP-запросы с низким и средним трафиком и может использоваться в качестве временного набора данных. Он совместим с Windows, Mac, Linux и Solaris.

Используемая платформа FreeSWITCH накладывает ограничения на выбор СУБД. Платформа поддерживает «из коробки» только SQLite. Так как модуль IVR разрабатывается на языке программирования JavaScript и не требует хранения и обработки большого объема данных, для работы с ним была выбрана СУБД SQLite. Для телефонной книги предприятия была выбрана СУБД PostgreSQL, которая требует более сложной организации структуры данных.

Выбор инструментов программирования и языков программирования зависит от множества различных факторов. Конкретные требования проекта, безусловно, являются наиболее важным аспектом всего процесса разработки программного обеспечения. Есть и другие факторы, которые можно учитывать, чтобы сравнить различные инструменты разработки и выбрать идеальную платформу, которая поможет в таких процедурах, как создание исходного кода, отладка, всесторонняя проверка кода, автоматизация процессов, создание веб-сервисов и общее управление проектами.

В рамках исследования по теме ВКР рассмотрено несколько инструментов разработки программного обеспечения.

GitHub — это платформа, которая в основном предназначена для групп разработчиков программного обеспечения для совместной разработки. Он поддерживает как проверку кода, так и управление им благодаря своим расширенным функциям. Более 56 миллионов разработчиков и более 3 миллионов компаний используют этот известный инструмент разработки

программного обеспечения. Они предоставляют услуги нескольким известным компаниям, включая Adobe, Dell Technologies и Ford.

Chrome DevTools предназначен для веб-разработчиков для написания веб-приложений, веб-сервисов и тестирования. Различные инструменты отладки встроены прямо в браузер Google Chrome для облегчения веб-разработки. Chrome DevTools — один из лучших инструментов разработки программного обеспечения для разработчиков веб-сайтов, поскольку он разработан непосредственно Google.

Visual Studio Code — один из самых популярных редакторов кода среди разработчиков программного обеспечения. Он доступен для всех основных операционных систем, поскольку эту платформу разработки программного обеспечения используют многие опытные и начинающие разработчики программного обеспечения. Это мощный редактор кода с открытым исходным кодом, который включает в себя все основные функции, которые можно ожидать от лучшего инструмента разработки программного обеспечения. Он включает в себя интегрированный CLI, возможность выделения синтаксиса, функции отступов, проверку кода и компиляцию кода.

Модуль IVR со встроенным компонентом DISA, отвечающий требованиям и бизнес-целям предприятия, разработан на языке JavaScript в среде разработки Visual Studio Code, являющейся самым удобным и универсальным инструментом разработки программного обеспечения. Для отладки разрабатываемого кода использовался Chrome DevTools.

Для реализации цели проекта выбран язык JavaScript, как наиболее гибкий и доступный.

2.3 Реализация функциональных требований к разработке компонентов системы IP-телефонии

Модуль IVR состоит из четырех компонент: обработки звонков, веб-интерфейса управления, сервера приложения и базы данных.

Компонент обработки звонков представляет собой исполняемый код JavaScript, работающий с API FreeSWITCH, состоит из функций: захвата введенных цифр с номеронабирателя телефона (getDigits), получения маршрута выбранной опции из базы данных (getRoute), запуска системы DISA (disaStart), захвата введенных цифр с номеронабирателя телефона системы DISA (captureInput) и перевода звонка DISA (disaTransferCall).

«Функция в JavaScript – это фрагмент кода, который можно вызывать через имя и параметры из любого места основного сценария. Функции можно описать двумя способами – по аналогии с классическими языками (Pascal, C++) и по новому стандарту ES6 в стрелочном виде. В классическом виде функция описывается с помощью ключевого слова function, за которым следует уникальное имя. Затем в круглых скобках указываются параметры, разделяемые запятыми. Они называются формальными параметрами или аргументами. Функция может не иметь параметров вообще. Команды и операторы, позволяющие выполнять действия над параметрами функции, записываются в фигурных скобках. Условно функции можно поделить на два вида – не возвращающие результат (подпрограммы) и возвращающие результат с помощью ключевого слова return. После команды return происходит выход из функции. В основном скрипте вызов функции происходит по ее имени с заменой параметров на фактические значения. Они называются фактическими параметрами» [2, с. 119].

Чтобы использовать модуль IVR в FreeSWITCH создан extension, который обрабатывает входящий звонок на определенный номер и вызывает JavaScript-код компонента обработки звонков, включающий:

- функцию getDigits компонента обработки звонков, которая считывает вводимые знаки с номеронабирателя телефона и добавляет их к переменной;
- функцию getRoute компонента обработки звонков, которая получает маршрут из базы данных согласно сохраненным в переменной данным;

- метод answer объекта session, отвечающий на звонок компонента обработки звонков;
- метод streamFile объекта session, проигрывающий приветствие компонента обработки звонков;
- метод collectInput объекта session, ожидающий вводимые знаки с номеронабирателя телефона в течение 5 секунд;
- условие перенаправления звонка в зависимости от введенных знаков;
- метод execute объекта session, осуществляющий перевод звонка в зависимости от введенных знаков или абоненту отдела продаж, или абоненту отдела технической поддержки, или оператору;
- функцию disaStart, запускающую систему DISA;
- функцию captureInput осуществляющую считывание набранных знаков системы DISA;
- функцию disaTransferCall осуществляющую звонок абоненту в зависимости от набранных знаков, в случае некорректного ввода повторно проигрывается сообщение и перезапускается система DISA.

Импорт библиотек FreeSWITCH не осуществляется, так как FreeSWITCH предоставляет необходимые функции через API (объект session).

«Объект Session предоставляет методы, которые позволяют взаимодействовать с голосовым каналом.

Создание объекта Session:

s = new Session(); – создает пустой объект Session, который предоставляет методы, позволяющие взаимодействовать с голосовым каналом.

s = new Session(uuid); – создает объект Session из существующего uuid, позволяет взаимодействовать с существующим голосовым каналом» [17].

Особенности использования некоторых методов объекта представлены ниже [19]:

Session.streamFile можно использовать для воспроизведения

аудиофайлов, поскольку он может автоматически определять рекомендуемую кодировку. Недостаток использования Session.streamFile в том, что он использует функции обратного вызова для обработки получаемых им тонов DTMF, которые будут выполняться не по порядку с остальной частью программы; таким образом, он не может контролировать поведение IVR. streamFile(file,,,); можно настроить с помощью функции, которая будет вызываться каждый раз, когда пользователь нажимает клавишу.

GetDigits вернет либо количество цифр, указанное numDigits, либо количество цифр до указания клавиши-терминатора в переменной mTerminator. Если mTerminator имеет «+» в спецификаторе, то getDigits вернет цифры, включая клавишу-терминатор. Следующие два поля используются для указания тайм-аутов. iTimeout представляет тайм-аут, который следует ждать, пока первая цифра не будет считаться нажатой, а dTimeout представляет количество времени, которое допускается между двумя последовательными цифрами.

Компонент веб-интерфейса управления представляет собой фронтенд веб-сервер nginx защищенный требованием авторизации (.htpasswd) и протоколом HTTPS (сертификатом и его закрытым ключом). Веб-сервер представлен единственной страницей веб-сайта, содержащей код HTML, состоящий из элементов управления и JavaScript кода, работающего с API сервера приложения.

В коде HTML присутствуют теги заголовков (<h1-h2>), ввода значений (<input>) и кнопок (<button>). JavaScript-код состоит из событий .on("click" [, eventData], handler), методов ajax() и ready() библиотеки JQuery.

Компонент сервера приложения представляет собой бэкенд веб-сервер выполняющий код JavaScript, предоставляющий API веб-интерфейсу управления и взаимодействующий с базой данных SQLite. Он состоит из экземпляра класса Database модуля sqlite3, который соединяется с базой данных SQLite, методов фреймворка Express.js GET, POST, PUT, DELETE для манипуляции данными и методов json() интерфейса Response для работы с

JSON, методов db.all() и db.run() модуля sqlite3 для манипуляции данными в базе данных SQLite, метода fs.readFileSync модуля fs, использующегося для чтения сертификата и его закрытого ключа по заданному пути файловой системы, метода https.createServer модуля https для запуска сервера по протоколу HTTPS.

Компонент базы данных представляет собой базу данных SQLite, содержащую единственную таблицу menu_options с двумя столбцами option и route.

Телефонная книга предприятия представляет собой базу данных PostgreSQL состоящей из двух таблиц data со столбцами id, name и org и numbers со столбцами id, data_id и number, к которой обращается встроенный модуль mod_cidlookup FreeSWITCH в момент совершения звонка за информацией о звонящем.

В приложении А приведен листинг кода.

Выводы по главе 2

Во второй главе:

- выполнена визуализация структуры разрабатываемых компонентов, отвечающих заданным требованиям, с использованием диаграмм UML;
- описан пользовательский интерфейс разрабатываемых компонентов;
- разработан модуль IVR с компонентом DISA и веб-интерфейсом управления на языке JavaScript с использованием фреймворка Express.js, библиотеки JQuery, модулей sqlite3, fs и https, middleware Body parser и CORS, а также API FreeSWITCH;
- выполнено логическое проектирование базы дынных и описаны типы данных на основе разработанной концептуальной модели;
- создана база данных телефонной книги предприятия с использованием СУБД PostgreSQL.

Глава 3 Оценка эффективности компонентов системы IP-телефонии для предприятия

3.1 Тестирование компонентов системы IP-телефонии

Существует множество методов тестирования программного обеспечения, с помощью которых можно убедиться, что изменения в коде или настройке будут работать как ожидалось.

Среди них можно выделить [7]:

- модульные тесты (модульные тесты, или unit тесты, работают на очень низком уровне, близко к исходному коду приложения. Они заключаются в тестировании отдельных методов и функций классов, компонентов или модулей, используемых в ПО);
- интеграционные тесты (в ходе интеграционного тестирования проверяется, хорошо ли работают вместе различные модули и сервисы, используемые приложением);
- функциональные тесты (в функциональных тестах основное внимание уделяется бизнес-требованиям к приложению);
- сквозные тесты (сквозное тестирование копирует поведение пользователя при работе с ПО в контексте всего приложения. Оно обеспечивает контроль того, что различные схемы действий пользователя работают должным образом);
- приемочное тестирование (приемочные тесты — это формальные тесты, которые проверяют, отвечает ли система требованиям бизнеса. При этом во время тестирования должно быть запущено само приложение, и основное внимание уделяется воспроизведению поведения пользователей);
- тестирование производительности (в тестах производительности оценивается работа системы при определенной рабочей нагрузке. С

помощью таких тестов можно оценить надежность, скорость, масштабируемость и отзывчивость приложения);

- smoke-тестирование (smoke-тесты — это базовые тесты, которые проверяют основные функциональные возможности приложения).

Для того чтобы оценить эффективность реализации проектного решения в рамках исследования по теме ВКР определены следующие цели тестирования разрабатываемых компонентов:

- проверка соответствия бизнес-требованиям заказчика,
- выявление ошибок в работе функций компонентов,
- проверка пользовательских сценариев,
- ввод в промышленную эксплуатацию.

Указанные цели тестирования достигаются выполнением функционального теста разработанных компонентов в соответствии с бизнес-требованиями проекта.

Существуют следующие виды функционального тестирования:

- тестирование на соответствие требованиям (каждая функция должна работать в соответствии с требованиями к разрабатываемым компонентам);
- тестирование пользовательского интерфейса (проверяется работа визуальных элементов - полей ввода, кнопок, ссылок, и их взаимодействия);
- тестирование удобства использования (проводится анализ, на сколько интерфейс интуитивно понятен и удобен пользователю);
- тестирование интеграции (проверяется взаимодействие разрабатываемых компонентов).

В рамках исследования по теме ВКР использовался функциональный тест базы данных телефонной книги предприятия и компонент модуля IVR: обработки звонков, веб-интерфейса управления, сервера приложения и базы данных.

В таблице 14 приведен список тестируемых функций с указанием

критериев успешности.

Таблица 14 – Перечень тестируемых функций с критериями успешности

Тестируемая функция	Критерий успешности
Вход в панель управления модуля IVR	Веб-интерфейс управления запрашивает авторизацию
Загрузка списка опций и маршрутов IVR-меню	В веб-интерфейсе управления IVR отображается список текущих опций и маршрутов меню IVR
Добавление опций и маршрута IVR-меню	Опция и маршрут успешно добавлены, отображается список опций и маршрутов меню IVR, включающий вновь созданный элемент
Редактирование опций и маршрута IVR-меню	Опция и маршрут успешно отредактированы, отображается список опций и маршрутов меню IVR, включающий отредактированный элемент
Удаление опций IVR-меню	Опция успешно удалена, отображается список опций и маршрутов меню IVR, исключающий удаленный элемент
Прием звонков модулем IVR	Звонок принят модулем IVR, проиграно приветственное сообщение
Распределение звонков по маршрутам в соответствии с выбранной опцией меню	При нажатии клавиши соответствующей опции на номеронабирателе телефона звонок переадресовывается согласно заданному маршруту
Отображении информации о звонящем	При приеме звонка у абонента на IP-телефоне или софтфоне отображается информация о звонящем, соответствующая входящему номеру

При выполнении тестирования функции входа в панель управления IVR проверяется наличие запроса авторизации пользователя и в случае корректного ввода регистрационных данных успешный вход в веб-интерфейс панели управления IVR.

При выполнении тестирования функции загрузки списка опций и маршрутов IVR-меню проверяется корректность загрузки информации из базы данных IVR и ее отображения в веб-интерфейсе управления.

При выполнении тестирования функций добавления, редактирования и удаления опций и маршрутов IVR-меню проверяется корректность внесения

изменений информации в базу данных IVR с последующим ее отображением в веб-интерфейсе управления.

При выполнении тестирования функции приема и распределения звонка проверяется корректность работы разработанного компонента звонка модуля IVR с базой данных IVR и API FreeSWITCH.

При выполнении тестирования функции отображения информации о звонящем проверяется корректность работы компонента телефонной книги предприятия с разработанной базой данных телефонной книги предприятия посредством встроенного модуля mod_cidlookup FreeSWITCH.

3.2 Оценка удобства и функциональности приложения разработанных компонентов системы IP-телефонии

Оценка удобства использования интерфейсов — важнейший аспект для обеспечения качественного взаимодействия пользователя с продуктом.

На рисунке 23 представлено окно запроса авторизации при входе в веб-интерфейс управления модулем IVR.

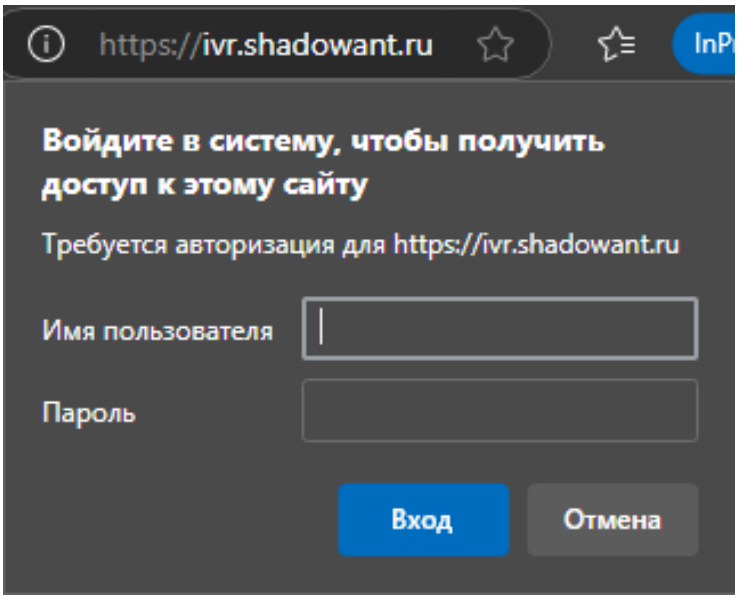

Рисунок 23 – Запрос авторизации веб-интерфейса управления модулем IVR

В соответствии с бизнес-требованиями проекта к разрабатываемым компонентам интерфейс управления IVR-меню должен быть интуитивно понятным, простым и иметь адаптивный дизайн для работы на различных устройствах. IVR-меню должно быть кратким, интуитивно понятным абонентам, с голосовыми подсказками.

На рисунке 24 представлен веб-интерфейс управления модулем IVR, включающий загрузку, добавление, редактирование и удаление опций и маршрутов IVR-меню.

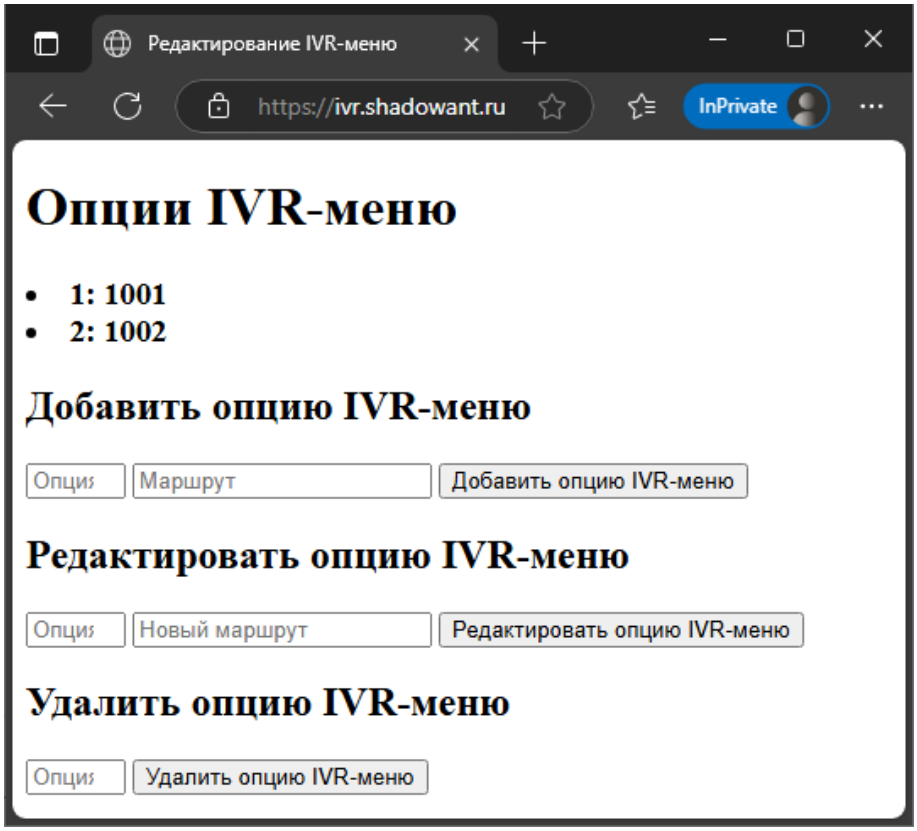

Рисунок 24 – Веб-интерфейс управления модуля IVR

На рисунке 25 представлен прием, и распределение звонков модулем IVR по маршрутам в соответствии с выбранной опцией.

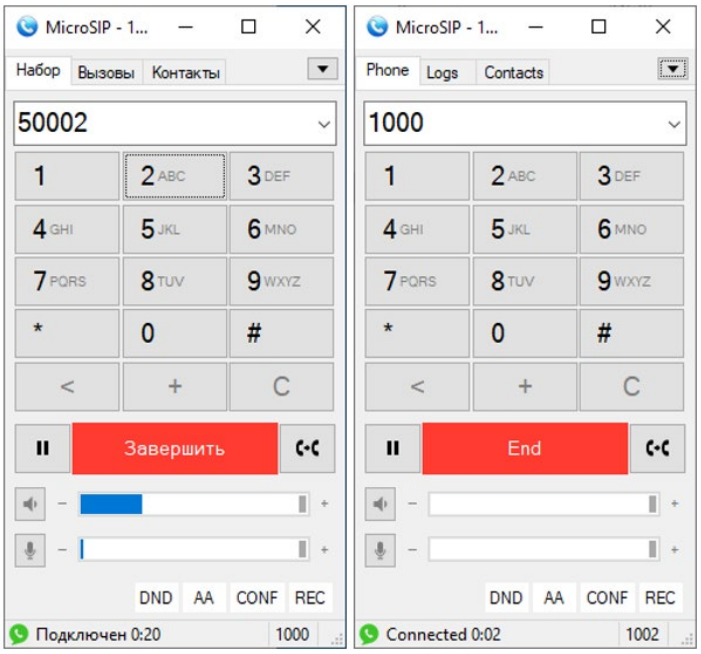

Рисунок 25 – Прием и распределение звонков модулем IVR

Интерфейс телефонной книги предприятия должен иметь возможность отображения данных на оконечных абонентских устройствах.

На рисунке 26 представлен результат работы базы данных телефонной книги предприятия.

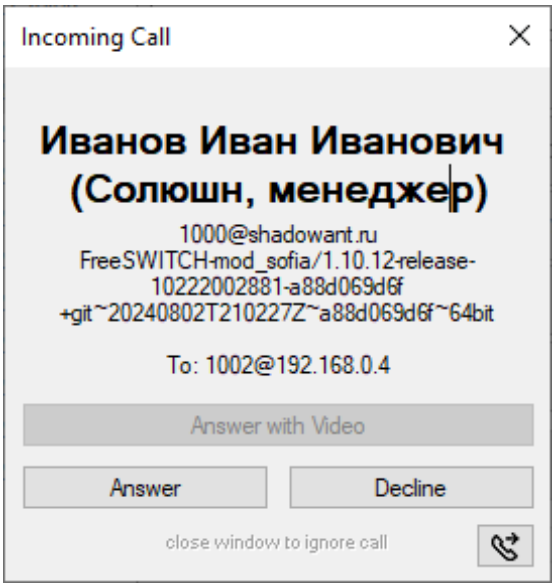

Рисунок 26 – Интерфейс отображения данных абонента

В таблице 15 приведен список тестируемых интерфейсов и их оценка.

Таблица 15 – Перечень тестируемых интерфейсов и их оценка

Тестируемый интерфейс	Удобство использования	Функциональность	Эстетика интерфейса	Производительность
Интерфейс управления IVR-меню	5	5	4	5
IVR-меню	5	5	5	5

Таблица составлена с учетом опроса администратора IVR, интервьюирования абонентов и наблюдением за пользователями во время их взаимодействия с разрабатываемыми компонентами.

3.3 Анализ результатов использования разработанных компонентов системы IP-телефонии

Процесс обработки тестовых данных состоит из следующих шагов:

- регистрация абонентов на создаваемой и существующей системах IP-телефонии,
- совершение звонка между абонентами создаваемой системой IP-телефонии и звонка между абонентом существующей системы IP-телефонии абоненту создаваемой системы IP-телефонии,
- вход по заданному адресу в панель управления IVR-меню,
- совершение звонка по номеру модуля IVR,
- фиксация результатов обработки.

Исходные данные для проведения тестирования созданной системы приведены в таблице 16.

Таблица 16 – Таблица исходных данных

Компонент\Система	Skype for Business	FreeSWITCH
FQDN сервера	skype.sa.local	shadowant.ru
IP-адрес сервера	10.0.0.6	192.168.0.5
Порт	TCP 5061	UDP 5060
SIP-домен	shadowant.ru	shadowant.ru
Диапазон номеров	2XXX	1XXX, 5000 IVR
Тестовые пользователи	test1 (tel: 2001)	Иванов Иван Иванович, Солюшн, менеджер (tel: 1000) Петров Петр Петрович, Солюшн, бухгалтер (tel: 1001)
Модуль IVR		
URL панели управления	https://ivr.shadowant.ru	
Логин и пароль	admin	
Телефонная книга предприятия		
IP сервера PostgreSQL	192.168.0.5	
Порт сервера PostgreSQL	5432	
База данных	phones	
Логин и пароль	freeswitch	

Для проведения тестирования выполнена установка tcpdump на сервер FreeSWITCH для снятия дампа сетевого трафика с последующим анализом в Wireshark.

«Tcpdump – это утилита командной строки с открытым исходным кодом и мощным инструментом анализа, которая используется для сбора данных сетевого трафика. С помощью ее опций можно задать как простые параметры (количество пакетов, которые необходимо перехватить), так и сложные (из заголовков пакетов можно выбрать определенные биты для анализа установленных флагов). Tcpdump имеет сложный язык фильтрации, поэтому для того, чтобы получить набор сетевых данных для анализа, необходимо понимать, как фильтровать данные при сборе» [1, с. 594].

«Wireshark – это программа-анализатор сетевых протоколов, имеющая пользовательский графический интерфейс, задача которой состоит в мониторинге сетевого трафика в реальном времени, детальном отображении принятых и отправленных пакетов данных, а также сохранении собранных данных для последующего анализа. В Wireshark реализована мощная система

поиска и фильтрации пакетов по множеству критериев» [3, с. 13].

В качестве абонента FreeSWITCH использован программный клиент MicroSIP, в качестве абонента Skype for Business использован программный клиент Skype for Business. Для тестирования функций модуля IVR и телефонной книги предприятия использовались абоненты FreeSWITCH (MicroSIP). Тестирование веб-интерфейса управления IVR-меню выполнялось в браузерах Google Chrome и Microsoft Edge.

Результаты проведенного тестирования создаваемой системы IP-телефонии FreeSWITCH приведены в таблице 17.

Таблица 17 – Результаты тестирования

Тест	Результат
Тестирование веб-интерфейса управления модуля IVR	успешно
Тестирование работы модуля IVR	успешно
Тестирование работы компонента DISA	успешно
Звонок тестовому абоненту Skype for Business от тестового абонента FreeSWITCH	успешно
Работа компонента телефонной книги предприятия	успешно

Выводы по главе 3

В третьей главе:

- определены цели тестирования разрабатываемых компонентов;
- составлен тест-кейс проверки функциональности с критериями успешности;
- выполнен функциональный тест в соответствии с целями тестирования;
- проведена оценка удобства использования и функциональности пользовательских интерфейсов;
- проведен анализ результатов тестирования разработанных компонентов.

Заключение

Широкое распространение IP-телефонии связано с развитием облачных технологий. Наблюдается процесс замещения аналоговой телефонии виртуальными АТС. Со временем VoIP-технологии смогут полностью заменить традиционные.

В рамках исследования по теме ВКР решена задача создания системы для организации корпоративной IP-телефонии на базе FreeSWITCH. Были разработаны модуль IVR, включающий компонент DISA, и телефонная книга предприятия, отвечающие бизнес-требованиям организации.

Выбор телефонной платформы FreeSWITCH связан с необходимостью перевода предприятия с решения на базе Skype for Business в связи с уходом вендора с отечественного рынка и закрытием корпоративного сервиса Skype for Business в 2021 году. FreeSWITCH является современным и профессиональным бизнес-решением для компаний. Это открытая телефонная платформа, распространяемая в исходных кодах. Она легкодоступна для разработки модулей под самые различные требования.

В ходе исследования по теме ВКР был проведен анализ существующих систем. Учитывая их преимущества и недостатки, обоснована целесообразность разработки собственных компонентов для платформы FreeSWITCH – модуля IVR и телефонной книги предприятия, а также определены функциональные и нефункциональные требования к ИТ-решению.

В процессе разработки компонентов системы IP-телефонии для предприятия выполнено проектирование, реализация функциональных требований и тестирование ИТ-решения.

В результате проведенного исследования по теме ВКР были решены следующие задачи:

– выполнено проектирование разрабатываемых компонентов,

– разработан модуль IVR с максимально простым и интуитивно понятным интерфейсом,
– разработана телефонная книга предприятия,
– проведено функциональное тестирование работоспособности решения,
– проведена оценка удобства и функциональности приложения разработанных компонентов системы IP-телефонии.

Предлагаемое ИТ-решение полностью работоспособно, отвечает всем заявленным требованиям и готово к вводу в промышленную эксплуатацию.

Несмотря на достигнутые результаты в ходе решения задач в рамках исследования по теме ВКР, существует перспектива дальнейшего исследования и разработки для совершенствования разрабатываемых компонентов.

В части модуля IVR и телефонной книги предприятия возможно рассмотреть разработку компонентов не только для платформы FreeSWITCH, но также и для других платформ. Оформление интерфейса управления модулем IVR можно сделать в соответствии с корпоративным стилем организации, а также добавить возможность управления голосовыми приветствиями и возможными подпунктами IVR-меню.

Список используемой литературы и используемых источников

1. Актуальные проблемы инфотелекоммуникаций в науке и образовании: материалы конференции. — Санкт-Петербург: СПбГУТ им. М. А. Бонч-Бруевича, 2023 — Том 1–2023. С. 590-595.

2. Васильева, И. И. Теория и практика программирования: учебно-методическое пособие / И. И. Васильева, С. Е. Попов. — Елец: ЕГУ им. И. А. Бунина, 2022. 229 с.

3. Гольдштейн, Б. С. Сетевые анализаторы IP сетей : учебное пособие / Б. С. Гольдштейн, В. Ю. Гойхман, Ю. В. Столповская. — Санкт-Петербург: СПбГУТ им. М. А. Бонч-Бруевича, 2013. 55 с.

4. Заяц, А. М. Инструментальные средства инфокоммуникационных систем. Теория и практика / А. М. Заяц, А. А. Логачев. — Санкт-Петербург: Лань, 2023. 208 с.

5. Мамедли, Р. Э. Системы управления базами данных: учебник для вузов / Р. Э. Мамедли. — Санкт-Петербург: Лань, 2024. 228 с.

6. Мир науки без границ: материалы 5-й Всероссийской научно-практической конференции (с международным участием) для молодых ученых, Тамбовский государственный технический университет, 16 февраля, 2018: материалы конференции. — Тамбов: ТГТУ, 2018. С. 269–270.

7. Питтет С. Различные виды тестирования ПО [Электронный ресурс] // URL: https://www.atlassian.com/ru/continuous-delivery/software-testing/types-of-software-testing (дата обращения: 15.03.2025).

8. Принзо Р. 6 лучших практик для успешного внедрения системы управления проектами [Электронный ресурс] // https://www.advanta-group.ru/blog/6-praktik-dla-uspesnogo-vnedrenia-sistemy/ (дата обращения: 19.03.2025).

9. Саблина, В. А. Основы программирования на JavaScript: учебное пособие / В. А. Саблина, Е. А. Трушина. — Рязань: РГРТУ, 2022. 96 с.

10. Смоленцева, Т. Е. Проектирование предметно-ориентированных информационных систем: учебно-методическое пособие / Т. Е. Смоленцева, Р. А. Исаев. — Москва: РТУ МИРЭА, 2022. 69 с.

11. Сравнение FreeSWITCH и Asterisk. [Электронный ресурс] // URL: https://wiki.merionet.ru/articles/sravnenie-freeswitch-i-asterisk (дата обращения: 15.02.2025).

12. Средства и системы обработки, хранения и передачи информации: учебник / В. Т. Еременко, Н. А. Глинкин, Р. Б. Трегубов [и др.]. — Орел: ОГУ имени И. С. Тургенева, 2023. 360 с.

13. Требования к системе: классификация FURPS+ [Электронный ресурс] // URL: https://skillscup.com/furps/ (дата обращения: 09.11.2024).

14. Флегонтов, А. В. Моделирование информационных систем. Unified Modeling Language / А. В. Флегонтов, И. Ю. Матюшичев. — 3-е изд., доп. — Санкт-Петербург: Лань, 2023. 140 с.

15. Червяков, Ю.А. Бизнес-процессы предприятия, их оценка и методы описания // Научные записки ОрелГИЭТ. — 2017. № 3. С. 115–121.

16. Anthony Minessale II, Giovanni Maruzzelli. FreeSWITCH 1.8 / Packt Publishing Ltd, 2017.

17. Client and Developer Interfaces. JavaScript. Session. [Электронный ресурс] // URL: https://developer.signalwire.com/freeswitch/FreeSWITCH-Explained/Client-and-Developer-Interfaces/JavaScript/Session/ (дата обращения: 15.03.2025).

18. Michael E. Auer, Hanno Hortsch, Panarit Sethakul, The Impact of the 4th Industrial Revolution on Engineering Education: Proceedings of the 22nd International Conference on Interactive Collaborative Learning (ICL2019) – Volume 2. Advances in Intelligent Systems and Computing/ Springer Nature, 2020.

19. Javascript IVRs in FreeSWITCH [Электронный ресурс] // URL: https://www.onsip.com/voip-news/onsip-news/javascript-ivrs-in-freeswitch / (дата обращения: 30.01.2025).

20. JavaScript. V8 Supersedes SpiderMonkey [Электронный ресурс] // URL: https://developer.signalwire.com/freeswitch/FreeSWITCH-Explained/Client-and-Developer-Interfaces/JavaScript/ (дата обращения: 02.02.2025).

21. 6 VoIP Trends in 2024: Examining the Future of VoIP [Электронный ресурс] // URL: https://www.onsip.com/voip-resources/industry-news-trends/6-voip-trends-in-2024-examining-the-future-of-voip (дата обращения: 15.02.2025).

Приложение А

Листинг кода

Компонент обработки звонков (/etc/freeswitch/scripts/ivr.js)

```
// Подключение к базе данных
use("CoreDB");
var db = new CoreDB("ivr");
var sql;
var dtmf = new Object();

// В модуле IVR функция вызывается при каждом вводе цифры с панели
номеронабирателя телефона
function getDigits (session, type, data, arg) {
    if (type == "dtmf") {
        arg.digits += data.digit;
        if (arg.digits.length >= 1) {
            return false;
        }
    }
}

// Функция получения маршрута выбранной опции из базы данных
function getRoute(digits) {
    sql = "select * from menu_options where option = '" + digits +"';";
    db.prepare(sql);
    while(db.next()) {
        hash = db.fetch();
        if (hash["option"] == digits) {
            return hash["route"];
        } else {
            return false;
        }
    }
}

// Ответ на звонок модуля IVR
session.answer();
// Проигрывание приветствия модуля IVR
session.streamFile("/etc/freeswitch/sounds/greeting.wav", "");
while(session.ready()) {
    dtmf.digits = "";
    // Захват введенных цифр с номеронабирателя телефона модуля IVR
    if (dtmf.digits.length < 1) {
         session.collectInput(getDigits, dtmf, 5000);
    }
    // Перевод звонка оператору (номер 1001) по таймауту}
```

```
    if (!dtmf.digits) {
        session.execute("transfer", "1001 XML default");
    }
    //Перевод звонка согласно маршрута выбранной опции
    if(dtmf.digits) {
        route = getRoute(dtmf.digits);
        //Проверка существования опции или запуск системы DISA (клавиша "*")
        if (!route) {
            if ( dtmf.digits == "*" ) {
                //Запуск системы DISA (прямой набор номера внутреннего абонента)
                disaStart();
            }
        }
        function disaStart() {
            //Ответ на звонок системы DISA
            session.answer();
            //Приветствие системы DISA
            session.streamFile("/etc/freeswitch/sounds/extension.wav");
            captureInput();
        }
        //Функция захвата введенных цифр с номеронабирателя телефона системы DISA
        function captureInput() {
            //Ожидание ввода 4 цифр и #, либо 5 секунд
            const input = session.getDigits(4, "#", 5000);
            if (input) {
                disaTransferCall(input);
            } else {
                //Проигрывание сообщения при некорректном вводе
                session.streamFile("/etc/freeswitch/sounds/error.wav");
                //Перезапуск системы DISA
                disaStart();
            }
        }

        //Перевод звонка на введенный номер
        function disaTransferCall(extension) {
            session.execute("transfer", extension, "XML", "default");
        }
    }
    else {
        //Перевод звонка по маршруту согласно выбранной опции
        session.execute("transfer", route);
    }
}
```

Компонент веб-интерфейса управления (/var/www/html/index.html)

```
<!DOCTYPE html>
<html lang="ru">
    <head>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
        <title>Редактирование IVR-меню</title>
    <!-- Использование JQuery library -->
        <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
    </head>
    <body>
        <h1>Опции IVR-меню</h1>
        <h3 id="optionsList"></h3>

        <h2>Добавить опцию IVR-меню</h2>
        <input type="number" id="newOption" placeholder="Опция IVR-меню" min="0"
max="9">
        <input type="number" id="newRoute" placeholder="Маршрут">
        <button id="addOptionButton">Добавить опцию IVR-меню</button>

        <h2>Редактировать опцию IVR-меню</h2>
        <input type="number" id="editOption" placeholder="Опция IVR-меню" min="0"
max="9">
        <input type="number" id="editRoute" placeholder="Новый маршрут">
        <button id="editOptionButton">Редактировать опцию IVR-меню</button>

        <h2>Удалить опцию IVR-меню</h2>
        <input type="number" id="deleteOption" placeholder="Опция IVR-меню"
min="0" max="9">
        <button id="deleteOptionButton">Удалить опцию IVR-меню</button>

        <script>
            // Отображение текущих опций IVR-меню через back-end сервер
            const loadOptions = () => {
                $.get('https://ivr.shadowant.ru:4443/api/options', (data) => {
                    $('#optionsList').empty();
                    data.forEach(options => {
                        $('#optionsList').append("<li>" + (options.option) + ":
"+ (options.route) + "</li>");
                    });
                });
            };

            // Добавление опции IVR-меню через back-end сервер
            $('#addOptionButton').on('click', () => {
```

```
        const option = $('#newOption').val();
        const route = $('#newRoute').val();
        $.ajax({
            url: 'https://ivr.shadowant.ru:4443/api/options',
            type: 'POST',
            contentType: 'application/json',
            data: JSON.stringify({option,route}),
            success: () => {
                loadOptions();
                $('#newOption').val('');
                $('#newRoute').val('');
            },
            error: (err) => {
                console.error(err);
            }
        });
    });

    // Редактирование опции IVR-меню через back-end сервер
    $('#editOptionButton').on('click', () => {
        const option = $('#editOption').val();
        const route = $('#editRoute').val();
        $.ajax({
            url: 'https://ivr.shadowant.ru:4443/api/options/'+(option),
            type: 'PUT',
            contentType: 'application/json',
            data: JSON.stringify({route}),
            success: () => {
                loadOptions();
                $('#editOption').val('');
                $('#editRoute').val('');
            },
            error: (err) => {
                console.error(err);
            }
        });
    });

    // Удаление опции IVR-меню через back-end сервер
    $('#deleteOptionButton').on('click', () => {
        const option = $('#deleteOption').val();
        $.ajax({
            url: 'https://ivr.shadowant.ru:4443/api/options/'+(option),
            type: 'DELETE',
            success: () => {
                loadOptions();
                $('#deleteOption').val('');
            },
            error: (err) => {
                console.error(err);
            }
```

```
                });
            });

            // Обновление текущих опций IVR-меню после манипуляций через back-end
сервер
            $(document).ready(() => {
                loadOptions();
            });
        </script>
    </body>
</html>
```

Компонент сервера приложения (/var/www/back-end/server.js)

```
// Загрузка framework Express (Node.js web application framework)
const express = require('express');
// Загрузка middleware Body parser
const bodyParser = require('body-parser');
// Загрузка sqlite3 module (работа с базой данных SQLite)
const sqlite3 = require('sqlite3').verbose();
// Загрузка middleware CORS
const cors = require('cors');
// Загрузка fs module (работа с файловой системой)
const fs = require('fs');
// Загрузка https module (работа с HTTPS)
const https = require('https');

const app = express();
// Подключение к базе данных IVR
const db = new sqlite3.Database('/var/lib/freeswitch/db/ivr.db');

app.use(cors());
app.use(bodyParser.json());

// Получение опций и маршрутов IVR-меню из базы данных
app.get('/api/options', (req, res) => {
    db.all("SELECT * FROM menu_options", [], (err, rows) => {
        if (err) {
            res.status(500).json({error: err.message});
            return;
        }
        res.json(rows);
    });
});
```

```
¶
//·Добавление·опций·и·маршрутов·IVR-меню·в·базу·данных¶
app.post('/api/options',·(req,·res)·=>·{¶
°·°·const·{option,·route}·=·req.body;¶
°·°·db.run("INSERT·INTO·menu_options·(option,·route)·VALUES·(?,·?)",·[option,·
route],·function(err)·{¶
°·°·°·°·if·(err)·{¶
°·°·°·°·°·°·res.status(500).json({error:·err.message});¶
°·°·°·°·°·°·return;¶
°·°·°·°·}¶
°·°·°·°·res.json({option,·route});¶
°·°·});¶
});¶
¶
//·Редактирование·опций·и·маршрутов·IVR-меню·в·базе·данных¶
app.put('/api/options/:option',·(req,·res)·=>·{¶
°·°·const·{option}·=·req.params;¶
°·°·const·{route}·=·req.body;¶
°·°·db.run("UPDATE·menu_options·SET·route·=·?·WHERE·option·=·?",·[route,·option],·
function(err)·{¶
°·°·°·°·if·(err)·{¶
°·°·°·°·°·°·res.status(500).json({error:·err.message});¶
°·°·°·°·°·°·return;¶
°·°·°·°·}¶
°·°·°·°·res.json({option,·route});¶
°·°·});¶
});¶
¶
//·Удаление·опций·и·маршрутов·IVR-меню·из·базы·данных¶
app.delete('/api/options/:option',·(req,·res)·=>·{¶
°·°·const·{option}·=·req.params;¶
°·°·db.run("DELETE·FROM·menu_options·WHERE·option·=·?",·option,·function(err)·{¶
°·°·°·°·if·(err)·{¶
°·°·°·°·°·°·res.status(500).json({error:·err.message});¶
°·°·°·°·°·°·return;¶
°·°·°·°·}¶
°·°·°·°·res.json({deletedID:·option});¶
°·°·});¶
});¶
¶
//·Запуск·сервера¶
const·PORT·=·process.env.PORT·||·4443;¶
const·privateKey·=·fs.readFileSync('/etc/nginx/ssl/cert.key');¶
const·certificate·=·fs.readFileSync('/etc/nginx/ssl/cert.crt');¶
https.createServer({key:·privateKey,·cert:·certificate},·app).listen(PORT,·()·=>·
{¶
°·°·console.log('Сервер·запущен·на·https://localhost:'+(PORT));¶
°·°·});¶
```