Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Магнитогорский государственный технический университет

им. Г. И. Носова»

Многопрофильный колледж

ПЦК № 10 Информатики и вычислительной техники

**КУРСОВОЙ ПРОЕКТ**

**РАСЧЁТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

По ПМ.11 Разработка, администрирование и защита баз данных

МДК.11.01 Технология разработки и защиты баз данных

на тему: Разработка информационной системы для автоматизации деятельности провайдера телекоммуникационных услуг с использованием СУБД PostgreSQL

Исполнитель: Головин Егор Евгеньевич, студент 3 курса, группа ИСПпК-21-1

Руководитель: Мазнина Юлия Александровна, преподаватель МпК

(Ф.И.О., должность, уч. степень, уч. звание)

Работа допущена к защите “” 20 г.

(подпись)

Работа защищена “” 20 г. с оценкой

(оценка) (подпись)

Магнитогорск, 2024

Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Магнитогорский государственный технический университет   
им. Г. И. Носова»

Многопрофильный колледж

**КУРСОВОЙ ПРОЕКТ**

Тема: Разработка информационной системы для автоматизации деятельности жилищно-коммунальных организаций города с использованием СУБД PostgreSQL

Студент: Головин Егор Евгеньевич

Задание

Исходные данные:

краткое описание предметной области «Провайдеров телекоммуникационных услуг»

Состав и содержание проекта (работы):

Провести анализ предметной области " Провайдеров телекоммуникационных услуг " и имеющихся программных продуктов. Выполнить моделирование предметной области. Спроектировать и разработать базу данных. Реализовать разграничение прав доступа. Разработать десктопное приложение для работы с базой данных с использованием языка программирования C#, WPF и IDE Visual Studio. Подготовить руководство пользователя и руководство программиста.

Срок сдачи: «25» июля 2024 г.

Руководитель: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ / Мазнина Ю.А. /

Срок сдачи: «25» июля 2024 г.

Задание получил: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ / Головин Е.Е. /

«15» января 2024 г.

Магнитогорск, 2024

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 5](#_Toc171174579)

[1 ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ 6](#_Toc171174580)

[1.1 Анализ предметной области 6](#_Toc171174581)

[1.1.1 Предоставление телекоммуникационных услуг 6](#_Toc171174582)

[1.1.2 Основные процессы в сфере телекоммуникационных услуг 8](#_Toc171174583)

[1.1.3 Специалисты, работающие в сфере телекоммуникационных услуг 8](#_Toc171174584)

[1.1.4 Разнообразие действий участников в сфере телекоммуникаций 9](#_Toc171174585)

[1.1.5 Обзор провайдеров телекомуникационных услуг 10](#_Toc171174586)

[1.2 Анализ существующих программных продуктов 12](#_Toc171174587)

[1.2.1 Дом.ру 12](#_Toc171174588)

[1.2.2 Интерсвязь 13](#_Toc171174589)

[1.3 Постановка задачи 14](#_Toc171174590)

[1.4 Моделирование информационной системы 16](#_Toc171174591)

[1.5 Проектирование базы данных 19](#_Toc171174592)

[1.5.1 Концептуальная модель 19](#_Toc171174593)

[1.5.2 Концептуальная модель 21](#_Toc171174594)

[1.5.3 Словарь данных 24](#_Toc171174595)

[1.6 Выбор и обоснование средств разработки 27](#_Toc171174596)

[2 РАЗРАБОТКА ИНФОРМАЦИОННОЙ СИСТЕМЫ 29](#_Toc171174597)

[2.1 Создание базы данных 29](#_Toc171174598)

[2.2 Создание представлений 31](#_Toc171174599)

[2.3 Создание функций и хранимых процедур 34](#_Toc171174600)

[2.4 Создание триггеров 38](#_Toc171174601)

[2.5 Разграничение прав доступа, создание ролей и пользователей 40](#_Toc171174602)

[2.6 Разработка интерфейса приложения 42](#_Toc171174603)

[3 Разработка документации 49](#_Toc171174604)

[3.1 Руководство пользователя 49](#_Toc171174605)

[3.1.1 Введение 49](#_Toc171174606)

[3.1.2 Назначение и условия применения 49](#_Toc171174607)

[3.1.3 Подготовка к работе 50](#_Toc171174608)

[3.1.4 Описание операций 50](#_Toc171174609)

[3.2 Руководство программиста 53](#_Toc171174610)

[3.2.2 Характеристика ПО 54](#_Toc171174611)

[3.2.3 Развертывание ПО 54](#_Toc171174612)

[3.2.4 Запуск ПО 54](#_Toc171174613)

[3.2.5 Входные и выходные данные 55](#_Toc171174614)

[ЗАКЛЮЧЕНИЕ 56](#_Toc171174615)

[СПИСОК ИСТОЧНИКОВ 59](#_Toc171174616)

# ВВЕДЕНИЕ

В условиях стремительного развития телекоммуникационных технологий и увеличения объема информации, циркулирующей в глобальных сетях, возрастает потребность в эффективных системах управления и учета телекоммуникационных услуг. Современные информационные системы способны значительно повысить производительность и качество обслуживания, что важно для телекоммуникационных провайдеров, стремящихся удовлетворить растущие требования клиентов.

Данный курсовой проект посвящен созданию информационной системы «Портал телекоммуникационных услуг», которая позволит автоматизировать процессы управления услугами и учет клиентов. Такая система призвана улучшить координацию между различными отделами телекоммуникационных компаний, а также обеспечить надежное и оперативное обслуживание клиентов.

Основной целью проекта является разработка интегрированной информационной системы, способной автоматизировать учет услуг, управление ресурсами и взаимодействие с клиентами в телекоммуникационных компаниях. Для достижения этой цели необходимо решить следующие задачи:

1. Провести анализ текущих информационных систем в сфере телекоммуникаций для выявления их сильных и слабых сторон.
2. Спроектировать базу данных, которая будет обеспечивать хранение и управление информацией о провайдерах, услугах, тарифах и клиентах.
3. Разработать удобный и функциональный пользовательский интерфейс для различных категорий пользователей (менеджеры, администраторы, клиенты).
4. Внедрить и протестировать информационную систему для подтверждения ее работоспособности и эффективности.
5. Оценить экономическую и практическую целесообразность внедрения разработанной системы в условиях реальной эксплуатации.

Структура курсового проекта включает введение, обзор литературы, анализ существующих систем, проектирование базы данных, разработку приложения, тестирование системы и ее оценку, а также заключение. Каждый раздел направлен на выполнение конкретных задач проекта и охватывает основные аспекты разработки информационных систем в телекоммуникационной отрасли.

Создание современных информационных систем для телекоммуникационных компаний позволит улучшить качество предоставляемых услуг, повысить эффективность взаимодействия с клиентами и оптимизировать внутренние процессы, что в конечном итоге приведет к повышению конкурентоспособности и удовлетворенности клиентов.

# 1 ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ

## 1.1 Анализ предметной области

### 1.1.1 Предоставление телекоммуникационных услуг

Предоставление телекоммуникационных услуг заключается в обеспечении доступа к различным видам связи, таким как интернет, телевидение, телефония, видеонаблюдение и другие. Современные телекоммуникационные услуги бывают разнообразны по способу подключения, качеству, цене и функционалу. Каждая из них имеет свои преимущества и недостатки, а также требования к оборудованию и инфраструктуре. Для каких-то целей будет достаточно простого подключения по оптоволокну, для других потребуется беспроводная связь с высокой скоростью передачи данных и широким покрытием.

В первую очередь необходимо определить, какие телекоммуникационные услуги нужны пользователю, и отталкиваясь от этого выбора подбирать подходящего провайдера. На сегодняшний день телекоммуникационные услуги предоставляют многие крупные и мелкие компании, такие как Ростелеком, МТС, Билайн, Дом.ру, Акадо и другие. Они имеют разную тарифную политику, спектр услуг, качество обслуживания и репутацию. Телекоммуникационные услуги характеризуются скоростью интернета, количеством телеканалов, наличием дополнительных сервисов, стоимостью подключения и абонентской платы. Уже начиная с выбора телекоммуникационных услуг пользователь должен учитывать свои потребности и возможности. Для разных целей лучше всего подходят те или иные тарифы, необходимо сравнивать их характеристики и выгоду. Существуют простые и дешевые тарифы, они подходят для офисного или домашнего использования, например для просмотра веб-страниц, электронной почты и прочих базовых задач. Современные тарифы предлагают более высокую скорость интернета и больше телеканалов, они подходят для более сложных задач, таких как, просмотр онлайн-видео, игры, работа в облачных сервисах, видеозвонки и другие. Кроме того существуют совсем продвинутые тарифы, они идеально подходят для тех, кто хочет получить максимум возможностей по лучшей цене, они включают в себя интерактивное телевидение, видеонаблюдение, домофонию, кабельное телевидение и другие. Но, соответственно, чем больше телекоммуникационных услуг, тем они дороже, и требуют более сложного оборудования и настройки.

После выбора подходящих пользователю телекоммуникационных услуг, необходимо выбрать провайдера, который их предоставляет. Провайдеры отличаются друг от друга способом подключения, технологией передачи данных, уровнем клиентского сервиса, наличием дополнительных услуг и т.д. Здесь немаловажную роль играет географическое расположение пользователя, так как не все провайдеры работают во всех регионах и населенных пунктах. Впоследствии от выбора провайдера будет зависеть качество и стабильность связи, а также удобство и безопасность пользования телекоммуникационными услугами.

Так же как и телекоммуникационные услуги, провайдеры бывают различных видов и для различных целей. В большинстве случаев пользователю достаточно простого и надежного провайдера, но если есть необходимость, можно выбрать провайдера, специально предназначенного для работы с определенными видами связи. Такие провайдеры обычно имеют более широкий спектр услуг, более высокое качество связи, более гибкую тарифную политику, дополнительные сервисы и поддержку.

После выбора телекоммуникационных услуг, обычно выбирают вторую самую важную часть – способ подключения. Способ подключения – важный фактор, отвечающий за скорость, стабильность и безопасность связи. Но на самом деле способ подключения нужен не во всех случаях. Он необходим, только если пользователь желает получить максимальную производительность и качество связи или собирается использовать связь в удаленных или труднодоступных местах. В условиях города, обычно нет необходимости в специальном способе подключения, достаточно стандартного подключения по оптоволокну, кабелю.

Способ подключения также стоит выбирать под свои нужды, обычно это самая дорогая часть телекоммуникационных услуг, поэтому к выбору надо отнестись с умом. Способы подключения характеризуются типом передачи данных, дальностью действия, требованиями к оборудованию и настройке. Все эти характеристики напрямую влияют на эффективность и удобство пользования телекоммуникационными услугами.

Далее стоит определиться с выбором оборудования. Оборудование - необходимый элемент, в котором содержатся непосредственно те устройства, которые обеспечивают подключение и работу с телекоммуникационными услугами. Оборудование, как и все компоненты, выбирается под характеристики телекоммуникационных услуг, чем лучше оборудование, тем лучше связь. Существуют несколько типов оборудования, такие как модемы, роутеры, антенны и другие. Тип оборудования выбирается в зависимости от типа и способа подключения, а также от вида телекоммуникационных услуг. Также стоит учитывать совместимость оборудования с провайдером и другими устройствами. Количество и мощность оборудования выбирается в зависимости от нужд пользователя. Для просмотра веб-страниц и электронной почты будет достаточно простого модема и роутера, для просмотра интерактивного телевидения и видеонаблюдения желательно иметь телевизор и камеру. Для улучшения качества и стабильности связи, стоит выбирать оборудование с хорошими характеристиками и надежными производителями. Это следует сделать для того, чтобы избежать неприятных ситуаций и неожиданных расходов в будущем.

В предметной области предоставления телекоммуникационных услуг происходит постоянное развитие и совершенствование технологий, услуг и тарифов, а также конкуренция между различными провайдерами. Пользователи имеют возможность выбирать из множества вариантов подключения и пользования разными видами связи, такими как интернет, телевидение, телефония, видеонаблюдение и другие. Предметная область включает в себя не только техническую сторону, но и юридическую, финансовую, маркетинговую и другие аспекты деятельности провайдеров и пользователей телекоммуникационных услуг.

**1.1.2 Основные процессы в сфере телекоммуникационных услуг**

Основные процессы в предметной области предоставления телекоммуникационных услуг можно разделить на три группы: процессы подключения, процессы обслуживания и процессы развития. Процессы подключения включают в себя анализ потребностей и возможностей пользователя, выбор подходящих телекоммуникационных услуг и провайдера, заключение договора, установку и настройку оборудования, активацию и проверку качества связи. Процессы обслуживания включают в себя оплату абонентской платы, контроль за расходом трафика, получение технической поддержки, консультации и решение проблем, изменение тарифов и услуг, расторжение договора. Процессы развития включают в себя изучение рынка и потребностей пользователей, разработку и внедрение новых технологий, услуг и тарифов, проведение рекламных и маркетинговых кампаний, улучшение качества и надежности связи, расширение географии и спектра услуг.

**1.1.3 Специалисты, работающие в сфере телекоммуникационных услуг**

В предметной области предоставления телекоммуникационных услуг работают различные специалисты, которые выполняют разные функции и задачи:

* руководитель отдела по работе с клиентами – отвечает за организацию и контроль работы отдела, который занимается привлечением, консультированием и удержанием клиентов, а также обеспечивает высокий уровень клиентского сервиса и лояльности;
* менеджер по работе с клиентами - осуществляет непосредственный контакт с клиентами, предоставляет им информацию о телекоммуникационных услугах и провайдерах, помогает в выборе и подключении, заключает договоры, принимает оплату, решает вопросы и жалобы;
* руководитель отдела технической поддержки - отвечает за организацию и контроль работы отдела, который занимается установкой, настройкой, обслуживанием и ремонтом оборудования, а также предоставляет техническую помощь и консультации клиентам и сотрудникам;
* специалист технической поддержки - выполняет практические работы по установке, настройке, обслуживанию и ремонту оборудования, а также предоставляет техническую помощь и консультации клиентам и сотрудникам по телефону, электронной почте или удаленно;
* бухгалтер – ведет учет и контроль за финансовыми операциями, связанными с предоставлением телекоммуникационных услуг, такими как поступление и расход денежных средств, налоги, счета, отчеты и другие документы;
* директор по развитию – отвечает за разработку и реализацию стратегии развития предприятия, анализирует рынок и конкурентов, ищет новые возможности и партнеров, внедряет инновации и улучшения в телекоммуникационные услуги и процессы;
* сотрудник технического департамента – занимается исследованием, разработкой, тестированием и внедрением новых технологий, услуг и тарифов, а также обеспечивает их совместимость, безопасность и эффективность.

**1.1.4 Разнообразие действий участников в сфере телекоммуникаций**

В сфере телекоммуникаций работают различные специалисты, выполняющие разнообразные функции и задачи. Эти действия включают, но не ограничиваются следующими:

* Аналитики данных: занимаются сбором, обработкой и анализом больших объемов данных, полученных от клиентов и сетевых устройств. На основе анализа данных они разрабатывают рекомендации по улучшению качества услуг и увеличению эффективности работы компании.
* Маркетологи: разрабатывают и реализуют маркетинговые стратегии и кампании, направленные на привлечение новых клиентов и удержание существующих. Они анализируют рынок, изучают конкурентов и потребности клиентов, создают рекламные материалы и проводят промоакции.
* Инженеры по сетевым технологиям: отвечают за проектирование, установку, настройку и обслуживание сетевого оборудования и программного обеспечения. Они обеспечивают бесперебойную работу сети, устраняют технические проблемы и занимаются модернизацией инфраструктуры.
* Менеджеры по проектам: координируют и контролируют выполнение проектов в области телекоммуникаций. Они разрабатывают планы проектов, распределяют ресурсы, контролируют выполнение задач и сроки, а также взаимодействуют с клиентами и подрядчиками.
* Специалисты по обслуживанию клиентов: предоставляют информацию о телекоммуникационных услугах, помогают клиентам в выборе и подключении услуг, решают вопросы и жалобы, а также проводят опросы и акции для поддержания долгосрочных отношений с клиентами.

Таким образом, разнообразие действий участников в сфере телекоммуникаций охватывает широкий спектр задач, направленных на обеспечение качественного предоставления услуг и удовлетворение потребностей клиентов.

### 1.1.5 Обзор провайдеров телекомуникационных услуг

В настоящее время популярными провайдерами телекомуникационных услуг являются Дом.ру, Ростелеком, МТС. (рис. 1 – 3).

Основные возможности этих провайдеров включают предоставление услуг интернета, телефонии и интерактивного телевидения. Некоторые из них, такие как Дом.ру и Ростелеком, предлагают абонентам свободно выбирать оптимально подходящего им провайдера.

Сравнительный анализ нескольких самых используемых провайдеров телекомуникационных услуг представлен в таблице 1.

Таблица 1 – Сравнительный анализ популярных провайдеров телекомуникационных услуг

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование | Описание веб-сайта | Достоинства | Недостатки |
| Дом.ру | Дом.ру - это российский телекоммуникационный оператор услуг связи. Он предлагает широкий спектр услуг, включая интернет, телефонию и телевидение. | Оператор представлен в 56 городах России. Услуги доступны в большинстве крупных городов. | Некоторые пользователи отмечают проблемы со стабильностью подключения. |
| Ростелеком | Ростелеком - один из крупнейших провайдеров телекомуникационных услуг в России. Он предлагает услуги интернета, телефонии и телевидения. | Ростелеком предлагает высокое качество обслуживания. | Некоторые пользователи жалуются на высокие цены. |
| МТС | МТС - это российская телекомуникационная организация, которая предоставляет услуги жителям России и стран СНГ. Она предлагает такие услуги, как сотовая связь, проводная телефония, домашний интернет, мобильное, кабельное и цифровое телевидение. | МТС предлагает широкий спектр услуг и информации. | Некоторые пользователи отмечают проблемы со стабильностью подключения. |

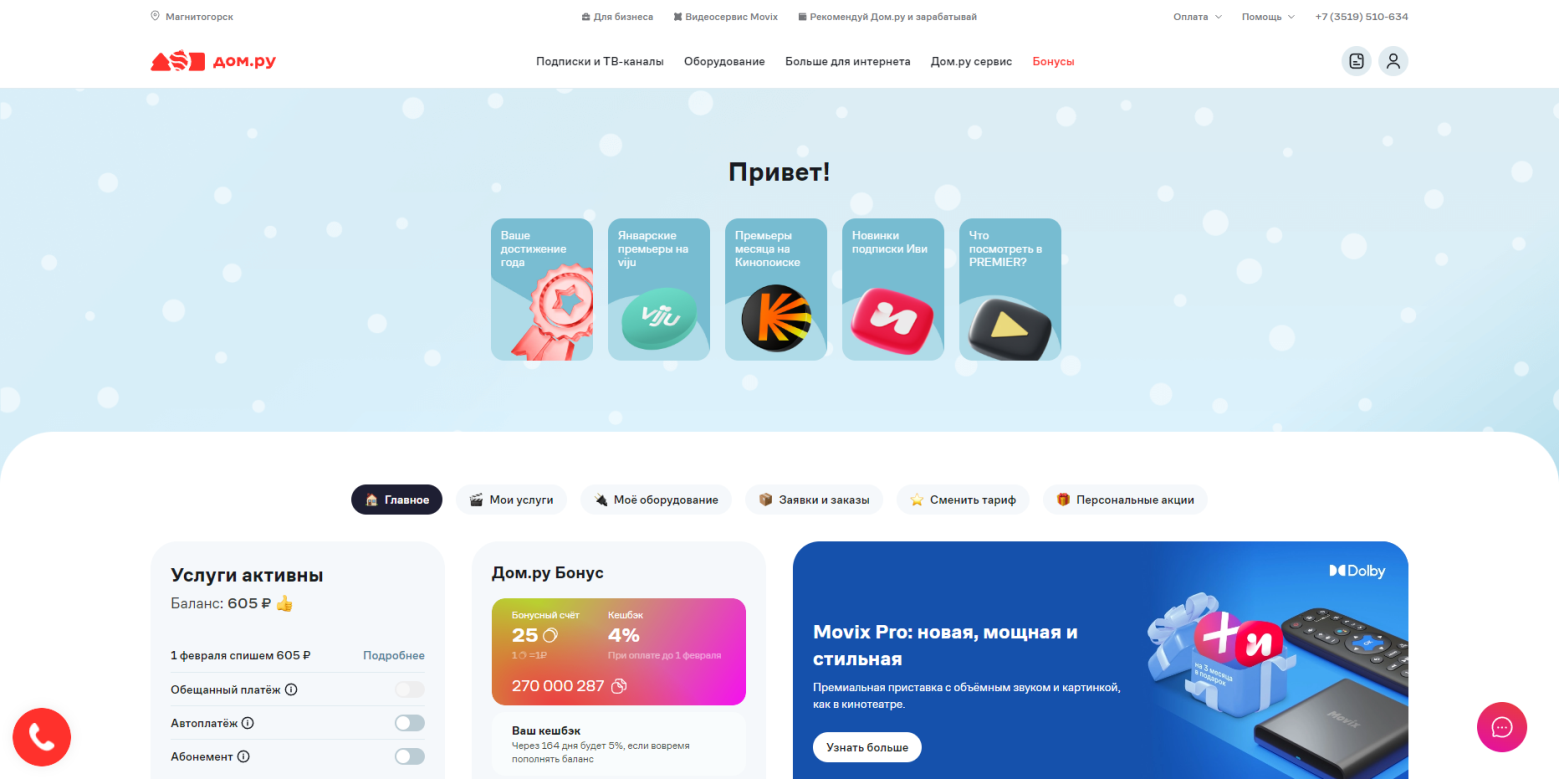


Рисунок 1 – Сайт Дом.ру

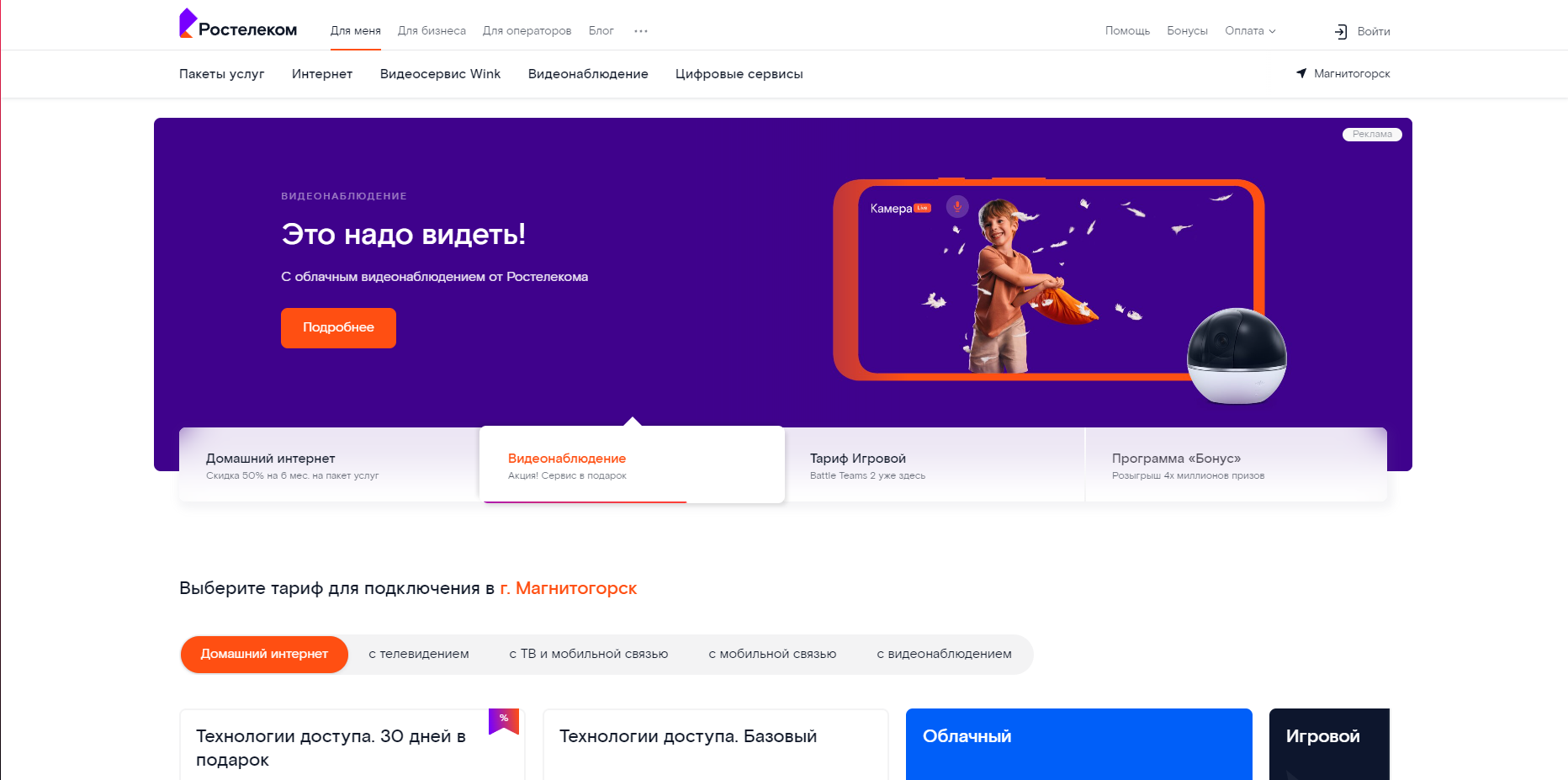


Рисунок 2 – Сайт Ростелеком

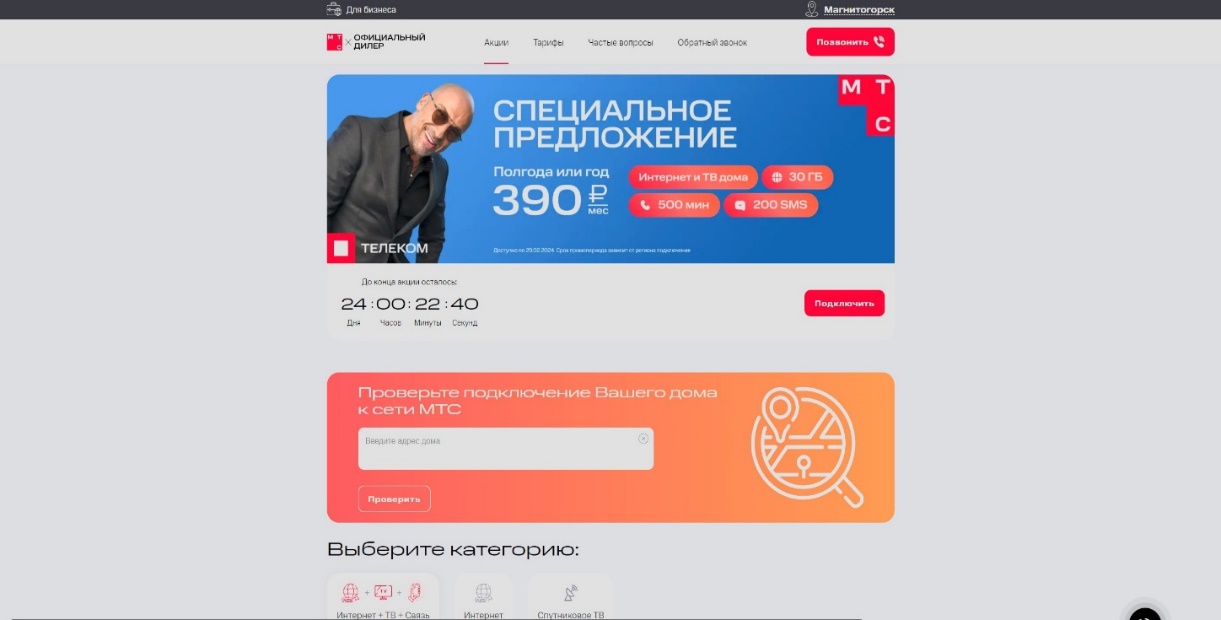


Рисунок 3 – Сайт МТС

## 1.2 Анализ существующих программных продуктов

### 1.2.1 Дом.ру

«Дом.ру» (рис. 4) – это мобильное приложение для управления услугами. После установки и регистрации пользователь попадает в меню «Главная». В приложении представлены различные функции, такие как автоплатеж, отложенный платеж и привязка банковских карт.

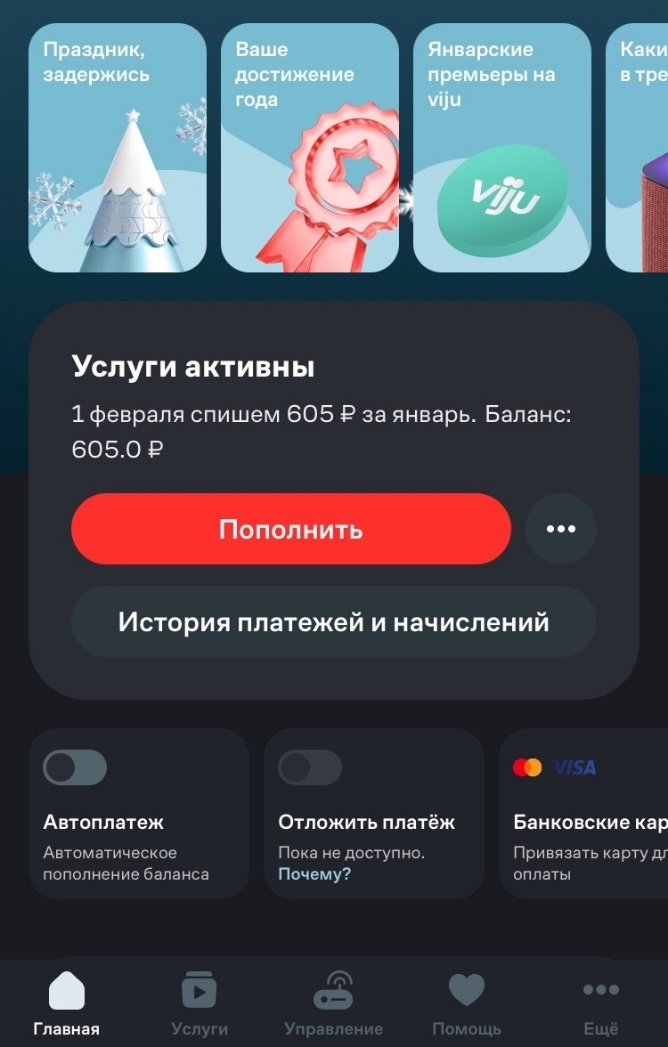


Рисунок 4 – Приложение «Дом.ру»

Основное назначение приложения - управление услугами и платежами. Приложение позволяет просматривать историю платежей и начислений, устанавливать автоматическое пополнение баланса и откладывать платежи. Также можно привязать банковскую карту для оплаты.

Благодаря тому, что данные хранятся в облаке, их не нужно синхронизировать и загружать на дополнительные устройства: достаточно просто установить приложение на устройство и авторизоваться. В случае со стационарным устройством достаточно зайти в браузер и авторизоваться.

### 1.2.2 Интерсвязь

«Интерсвязь» (рис. 5) – это многофункциональное мобильное приложение от провайдера телекомуникационных услуг.

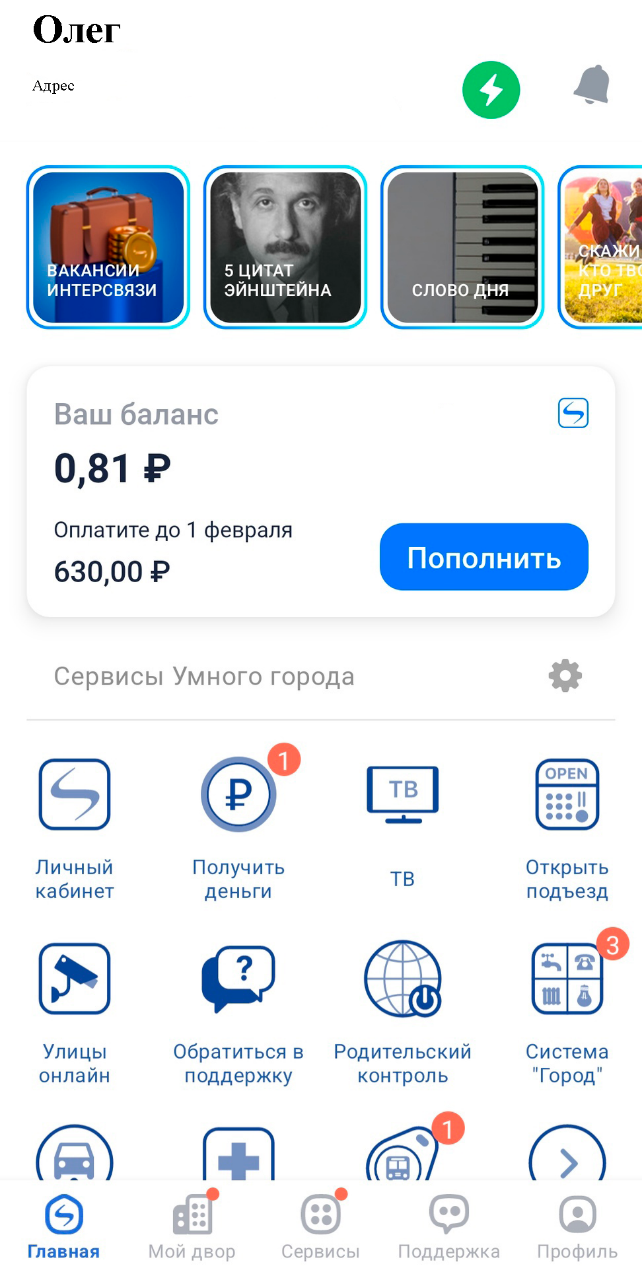


Рисунок 5 – Приложение Интерсвязь

Оно предоставляет доступ к различным услугам и информации, включая управление финансами, просмотр телевизионных каналов и доступ к городским услугам. Также можно использовать его для получения поддержки и управления личным кабинетом. Это удобный инструмент для управления услугами на одной платформе.

## 1.3 Постановка задачи

Целью курсового проекта является разработка информационной системы «Портал телекомуникационных услуг» для автоматизированного учета и управления услугами телекомуникационных провайдеров.

Для достижения поставленной цели курсового проекта необходимо выполнить ряд задач:

1. провести анализ предметной области;
2. провести анализ существующих программных средств;
3. разработать проектную документацию;
4. выполнить моделирование и проектирование программного обеспечения;
5. выбрать средства и технологии для разработки программного обеспечения;
6. установить и настроить платформу проекта;
7. спроектировать и разработать базу данных;
8. разработать графический интерфейс;
9. разработать функциональные модули приложения;
10. провести тестирование и отладку веб-приложения;
11. написать документацию программного продукта.

В рамках курсового проекта планируется разработать АРМ для менеджера и администратора.

Функционал АРМ менеджера:

1. Управление заявками пользователей: прием, обработка и контроль выполнения заявок.
2. Ведение клиентской базы данных: добавление, изменение и удаление информации о клиентах.
3. Формирование отчетов: создание отчетов о заявках, клиентах, предоставленных услугах и т.д.
4. Управление договорной документацией: оформление, изменение и хранение договоров с клиентами.
5. Интерактивная поддержка клиентов: предоставление информации и консультаций по услугам.
6. Настройка тарифных планов: разработка и внедрение новых тарифных планов, изменение существующих.
7. Мониторинг качества предоставляемых услуг: сбор и анализ данных о качестве услуг и удовлетворенности клиентов.

Функционал АРМ администратора:

1. разработка модуля провайдеров для управления информацией о провайдере;
2. разработка модуля услуг для управления информацией об услугах;
3. разработка модуля тарифов для управления информацией о тарифах.

При разработке функциональных требований к приложению необходимо учесть ряд пожеланий. Приложение должно иметь не перегруженный интерфейс и минимально необходимое, но достаточное количество функций для управления услугами телекоммуникационных провайдеров.

Интерфейс должен быть интуитивно понятным и удобным. Пользователь должен сразу понимать, как использовать функционал приложения. Также интерфейс должен привлекать внимание пользователя. Для хранения, редактирования, удаления и добавления информации необходимо использовать базу данных.

Приложение должно использовать современные технологии защиты информации. В информационной системе должно быть предусмотрено разграничение прав доступа к ее функциональности. В зависимости от уровня доступа должен выводиться соответствующий интерфейс взаимодействия.

Для администраторов и пользователей необходимо предоставить возможность авторизации и регистрации, предусмотреть разные роли для взаимодействия с приложением. Администратор должен иметь свой личный кабинет, в котором он сможет модерировать приложение, работать с базой данных, удалять старую информацию и заменять её на новую.

Работа с учетными записями должна включать в себя:

1. авторизацию учётной записи;
2. регистрацию учётной записи;
3. редактирование учётной записи;
4. выдачу привилегий/прав учётной записи.

## 1.4 Моделирование информационной системы

На рисунках 6 – 10 представлены диаграммы, созданные в рамках моделирования предметной области.

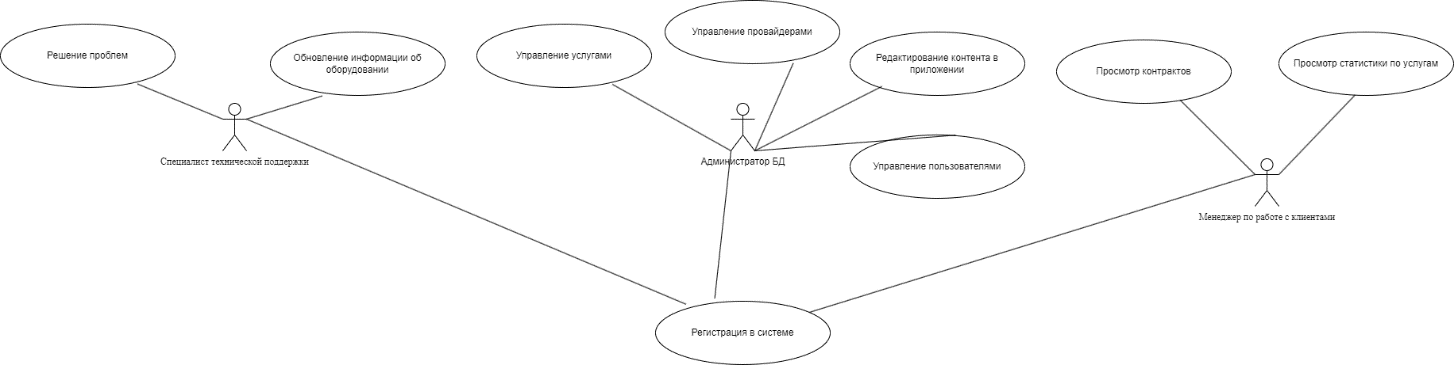


Рисунок 6 – Use Case диаграмма

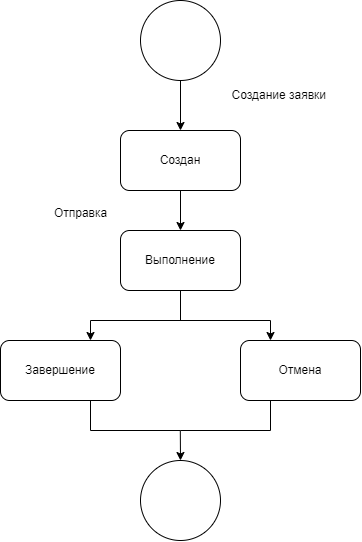


Рисунок 7 – Диаграмма состояний заявки на подключение

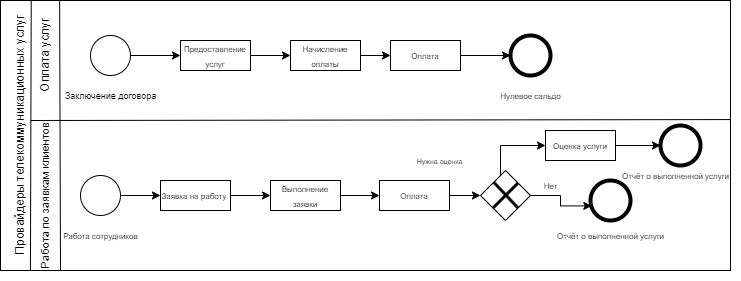


Рисунок 8 – Диаграмма бизнес-процесса «Обработка заявок на оказание услуг»

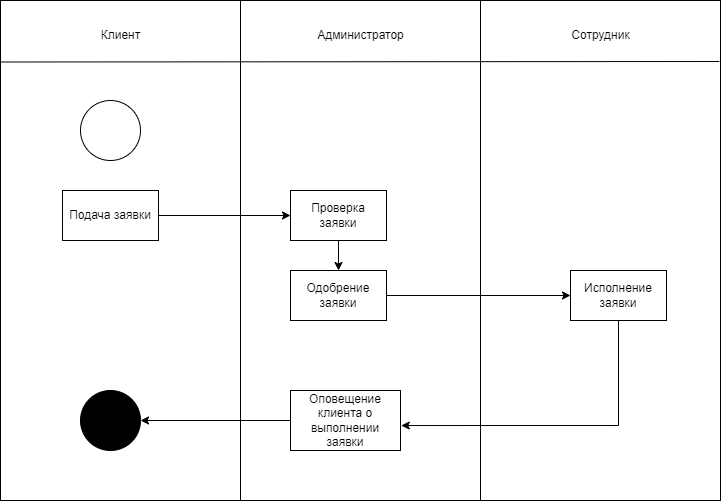


Рисунок 9 – Диаграмма активности

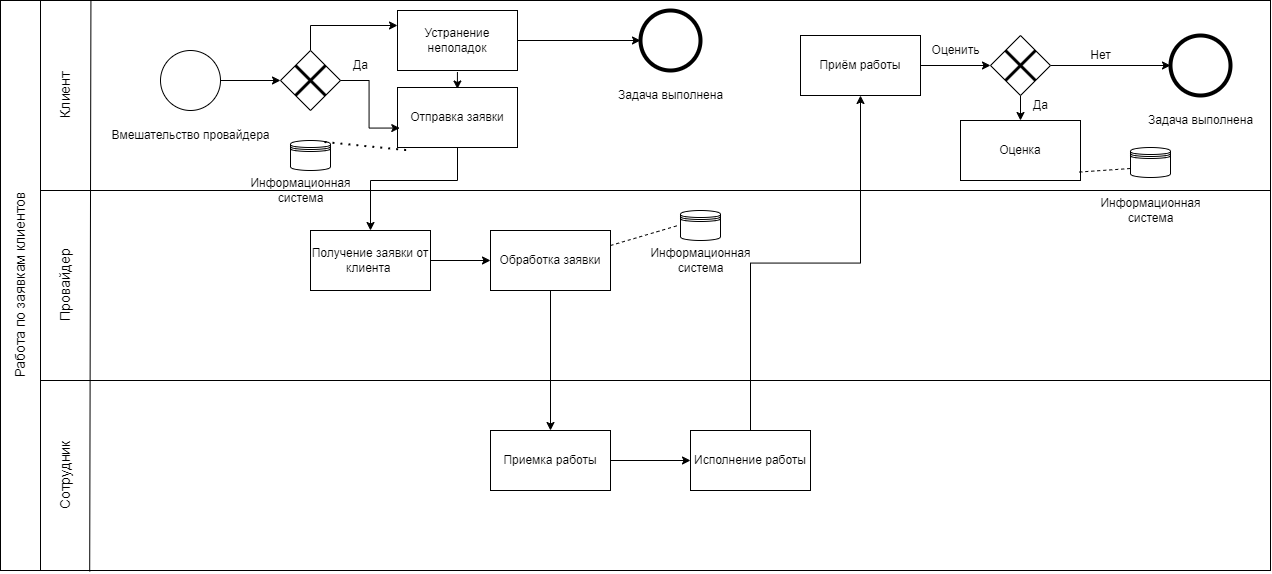


Рисунок 10 – Общая диаграмма бизнес-процессов в нотации BPMN

## 1.5 Проектирование базы данных

**1.5.1 Концептуальная модель**

ER-модель, или модель «сущность-связь», – это концептуальная модель данных, которая описывает данные как сущности, атрибуты и связи:

* сущности – это объекты или концепции, которые могут быть явно идентифицированы. Например, в базе данных университета студенты, курсы и преподаватели могут быть сущностями;
* атрибуты – это свойства или характеристики сущностей;
* связи – это способы, которыми сущности связаны друг с другом;
* ER-модель часто используется для проектирования баз данных, позволяя легко визуализировать структуру данных. Это особенно полезно при проектировании больших и сложных баз данных.

Логическая ER-модель (Entity-Relationship Model) – это абстрактное и концептуальное представление данных. Она определяет структуру данных, не вдаваясь в детали того, как данные будут храниться в физической базе данных. В логической ER-модели данные представлены в виде сущностей (объектов), атрибутов (характеристик или свойств этих объектов) и связей между ними.

Физическая ER-модель (Entity-Relationship Model) – это детализированная версия ER-модели, которая включает в себя все сущности, атрибуты и связи, найденные в реальном мире. Она также включает в себя ключи первичного и внешнего ключа для каждой сущности, а также правила, которые определяют, как операции обновления и удаления данных будут влиять на данные в связанных таблицах.

Логическая и физическая ER-модели различаются по уровню абстракции и назначению. Логическая модель представляет структуру данных на концептуальном уровне, независимо от физической реализации. Она включает сущности, атрибуты и связи, и используется для общения с пользователями и определения требований. Физическая модель фокусируется на реализации структуры данных в конкретной СУБД, учитывая детали хранения и оптимизации. Она включает таблицы, колонки, индексы, типы данных, первичные и внешние ключи. Логическая модель абстрактна и ориентирована на бизнес-требования, тогда как физическая модель конкретна и направлена на реализацию и производительность.

Для создания информационной системы была применена реляционная модель данных. Это обусловлено её способностью эффективно представлять взаимосвязи между сущностями и их атрибутами, характерными для данной предметной области.

Концептуальная модель предметной области представлена на рисунке 6.

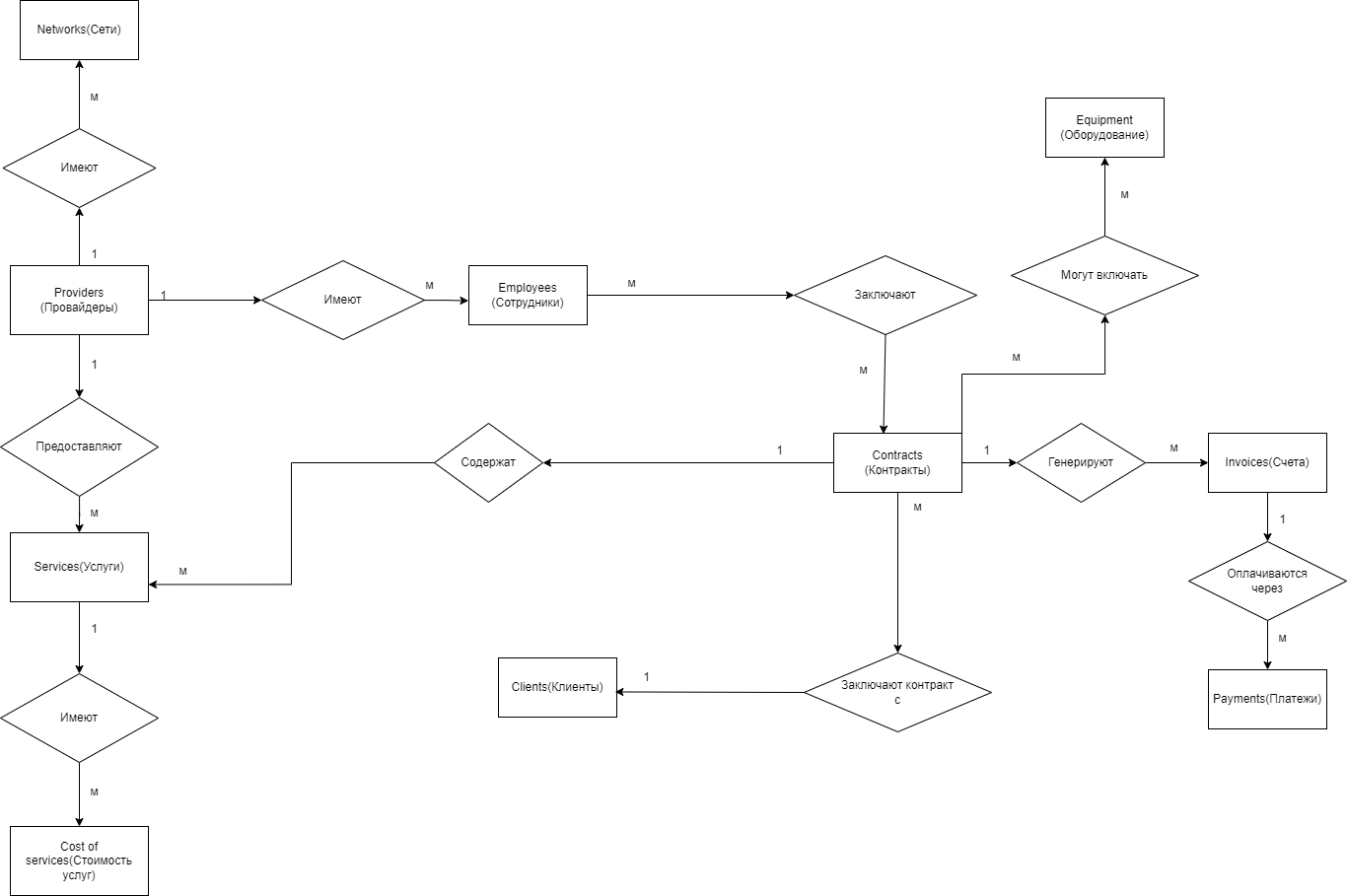


Рисунок 6 – Схема взаимосвязей сущностей в информационной системе

Исходя из анализа ER-модели предметной области, был выявлен следующий комплект таблиц:

– сущность «Провайдеры» связана с сущностью «Сотрудники» связью один ко многим;

* сущность «Провайдеры» связана с сущностью «Сети» связью один ко многим;
* сущность «Провайдеры» связана с сущностью «Услуги» связью один ко многим;
* сущность «Услуги» связана с сущностью «Стоимость услуг» связью один ко многим;
* сущность «Счета» связана с сущностью «Платежи» связью один ко многим;
* сущность «Сотрудники» связана с сущностью «Контракты» связью многие ко многим;
* сущность «Контракты» связана с сущностью «Клиенты» связью многие к одному;
* сущность «Контракты» связана с сущностью «Счета» связью один ко многим;
* сущность «Контракты» связана с сущностью «Оборудование» связью многие ко многим.

**1.5.2 Концептуальная модель**

После определения набора таблиц, был проведен процесс нормализации для устранения избыточности данных и обеспечения целостности информации. Это позволило избежать дублирования информации и улучшить производительность базы данных.

Были установлены требования к поддержке целостности данных, включая ограничения целостности, каскадное удаление, автоматическое обновление и другие.

На основе результатов нормализации и определения требований к целостности данных, была разработана окончательная версия логической модели данных для информационной системы.

В итоге в базе данных получилось 10 таблиц, из них 3 справочные таблицы и 7 являются операционными (рис. 7).

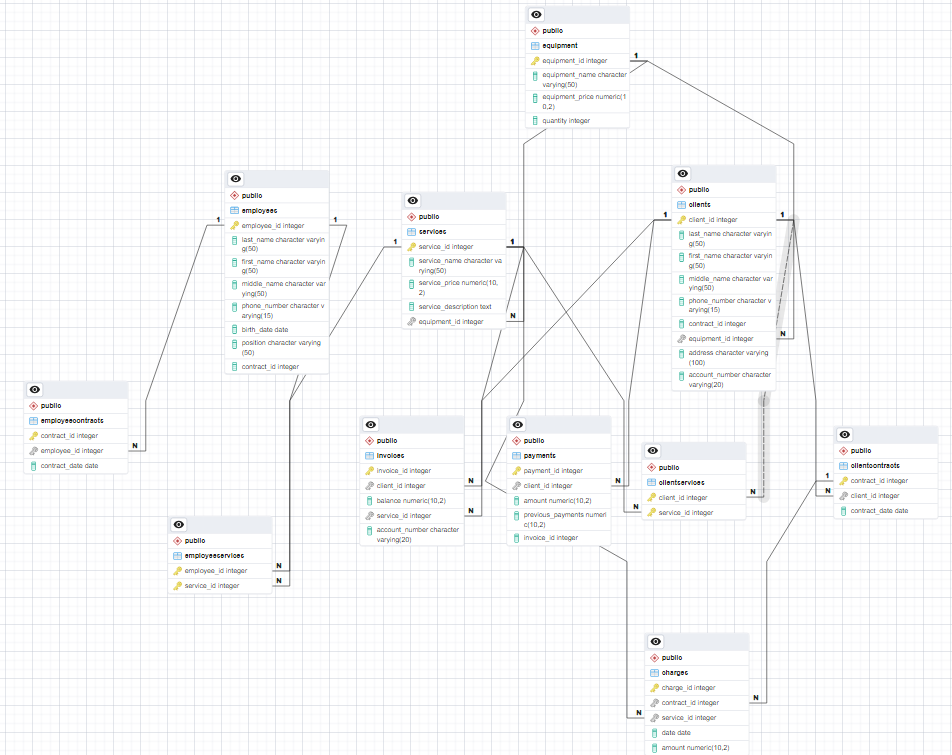


Рисунок 7 – Логическая структура БД

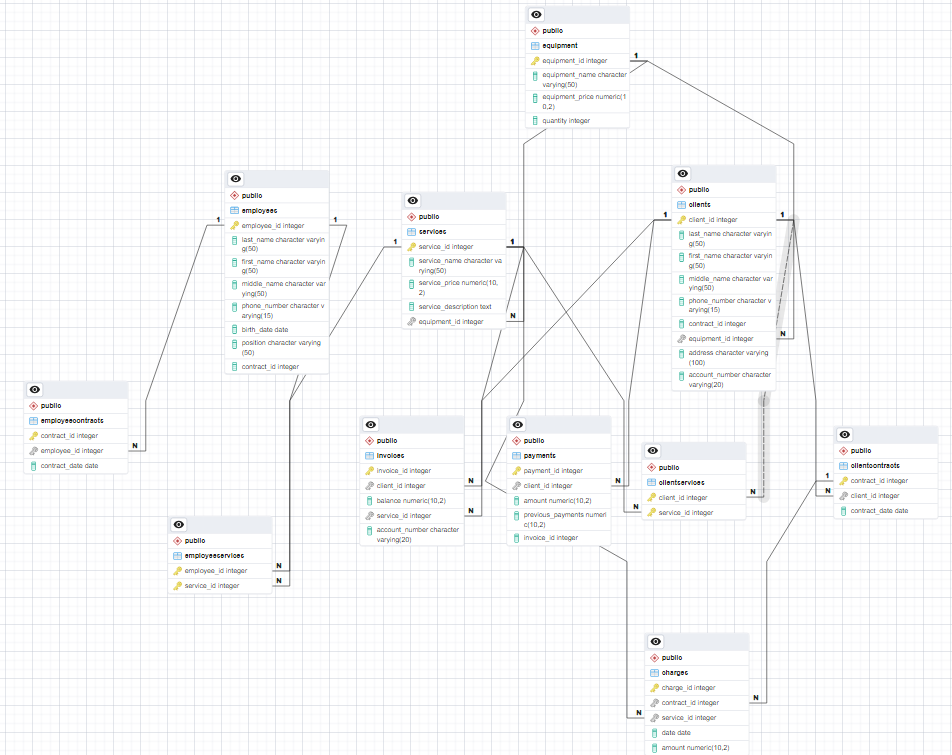


Рисунок 8 – Физическая структура БД

### 1.5.3 Словарь данных

В разделе рассматриваются основные аспекты проектирования логической и физической структуры базы данных, обеспечивающие эффективное хранение и обработку данных в информационной системе. Основные элементы структуры представлены в таблице 2 - таблице 11.

Таблица 2 – Clients

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Имя атрибута | Тип данных | Тип PostgreSQL | Длина | Ограничения | Описание поля |
| client\_id | Числовой | SERIAL |  | PK | Идентификатор клиента |
| last\_name | Строковый | character varying | 50 | NN | Фамилия клиента |
| first\_name | Строковый | character varying | 50 | NN | Имя клиента |
| middle\_name | Строковый | character varying | 50 |  | Отчество клиента |
| phone\_number | Строковый | character varying | 15 |  | Номер телефона |
| Address | Строковый | character varying | 100 |  | Адрес |
| account\_number | Строковый | character varying | 20 |  | Номер счета клиента |
| equipment\_id | Числовой | INT |  | FK | Идентификатор оборудования |
| contract\_id | Числовой | INT |  |  | Идентификатор контракта |

Таблица 3 – Employees

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Имя атрибута | Тип данных | Тип PostgreSQL | Длина | Ограничения | Описание поля |
| employee\_id | Числовой | SERIAL |  | PK | Идентификатор сотрудника |
| last\_name | Строковый | character varying | 50 | NN | Фамилия сотрудника |
| first\_name | Строковый | character varying | 50 | NN | Имя сотрудника |
| middle\_name | Строковый | character varying | 50 |  | Отчество сотрудника |
| phone\_number | Строковый | character varying | 15 |  | Номер телефона |
| birth\_date | Дата | DATE | 100 |  | |  | | --- | | Дата рождения |  |  | | --- | |  | |
| position | Строковый | |  | | --- | |  |  |  | | --- | | VARCHAR | | 50 |  | Должность |
| contract\_id | Числовой | INT |  |  | Идентификатор контракта |

Таблица 4 – Equipment

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Имя атрибута | Тип данных | Тип PostgreSQL | Длина | Ограничения | Описание поля |
| equipment\_id | Числовой | SERIAL |  | PK | Идентификатор оборудования |
| equipment\_name | Строковый | VARCHAR | 50 |  | Наименование оборудования |
| equipment\_price | Дробное число | DECIMAL(10, 2) |  |  | Цена оборудования |
| quantity | Числовой | INT |  |  | Количество оборудования |

Таблица 5 – Payments

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Имя атрибута | Тип данных | Тип PostgreSQL | Длина | Ограничения | Описание поля |
| payment\_id | Числовой | SERIAL |  | PK | Идентификатор платежа |
| client\_id | Числовой | INT |  | REFERENCES Clients(client\_id) | Идентификатор клиента |
| amount | Дробное число | DECIMAL(10, 2) |  |  | Сумма платежа |
| previous\_payments | Дробное число | DECIMAL(10, 2) |  |  | Сумма предыдущих платежей |
| invoice\_id | Числовой | INT |  |  | Идентификатор счета |

Таблица 6 – Invoices

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Имя атрибута | Тип данных | Тип PostgreSQL | Длина | Ограничения | Описание поля |
| invoice\_id | Числовой | SERIAL |  | PK | Идентификатор счета |
| client\_id | Числовой | INT |  | REFERENCES Clients(client\_id) | Идентификатор клиента |
| balance | Дробное число | DECIMAL(10, 2) |  |  | Баланс счета |
| account\_number | Строковый | VARCHAR | 20 |  | Номер счета |
| service\_id | Числовой | INT |  | REFERENCES Services(service\_id) | Идентификатор услуги |

Таблица 7 – ClientContracts

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Имя атрибута | Тип данных | Тип PostgreSQL | Длина | Ограничения | Описание поля |
| contract\_id | Числовой | SERIAL |  | PK | Идентификатор контракта |
| client\_id | Числовой | INT |  | REFERENCES Clients(client\_id) | Идентификатор клиента |
| contract\_date | Дата | DATE |  |  | Дата заключения контракта |
| termination\_date | Дата | DATE |  |  | Дата завершения контракта |

Таблица 8 – EmployeeContracts

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Имя атрибута | Тип данных | Тип PostgreSQL | Длина | Ограничения | Описание поля |
| contract\_id | Числовой | SERIAL |  | PK | Идентификатор контракта |
| employee\_id | Числовой | INT |  | REFERENCES Employees(employee\_id) | Идентификатор сотрудника |
| contract\_date | Дата | DATE |  |  | Дата заключения контракта |
| termination\_date | Дата | DATE |  |  | Дата завершения контракта |

Таблица 9 – ClientServices

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Имя атрибута | Тип данных | Тип PostgreSQL | Длина | Ограничения | Описание поля |
| client\_id | Числовой | INT |  | REFERENCES Clients(client\_id) | Идентификатор клиента |
| service\_id | Числовой | INT |  | REFERENCES Services(service\_id) | Идентификатор услуги |

Таблица 10 – EmployeeServices

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Имя атрибута | Тип данных | Тип PostgreSQL | Длина | Ограничения | Описание поля |
| employee\_id | Числовой | INT |  | REFERENCES Employees(employee\_id) | Идентификатор сотрудника |
| service\_id | Числовой | INT |  | REFERENCES Services(service\_id) | Идентификатор услуги |

Таблица 11 – Charges

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Имя атрибута | Тип данных | Тип PostgreSQL | Длина | Ограничения | Описание поля |
| charge\_id | Числовой | SERIAL |  | PK | Идентификатор начисления |
| contract\_id | Числовой | INT |  | REFERENCES ClientContracts(contract\_id) | Идентификатор контракта |
| service\_id | Числовой | INT |  | REFERENCES Services(service\_id) | Идентификатор услуги |
| Date | Дата | DATE |  |  | Дата начисления |
| Amount | Дробное число | DECIMAL(10, 2) |  |  | |  | | --- | |  |  |  | | --- | | Сумма начисления | |

## 1.6 Выбор и обоснование средств разработки

Для разработки информационной системы выбрана СУБД PostgreSQL.

Использование C# и WPF в корпоративной разработке открывает новые горизонты для создания масштабируемых и надежных приложений на платформе .NET. C# - это не просто язык программирования, это целый мир возможностей. Его гибкость и мощь делают его незаменимым инструментом. Обширная экосистема C#, включающая разнообразные инструменты и библиотеки, помогает разработчикам достичь своих целей.

Использование PostgreSQL как СУБД предлагает надежное решение для хранения и обработки данных. PostgreSQL обладает всеми необходимыми функциями, включая поддержку транзакций, многопользовательский доступ, а также инструменты для обеспечения безопасности и масштабируемости. Благодаря открытому исходному коду и активному сообществу пользователей, PostgreSQL становится идеальным выбором для проектов любого размера.

# 2 РАЗРАБОТКА ИНФОРМАЦИОННОЙ СИСТЕМЫ

## 2.1 Создание базы данных

Скрипты для создания таблиц базы данных представлены в листингах 1–11.

Листинг 1 – Создание таблиц / таблицы Clients

CREATE TABLE Clients (

client\_id SERIAL PRIMARY KEY,

last\_name VARCHAR(50),

first\_name VARCHAR(50),

middle\_name VARCHAR(50),

phone\_number VARCHAR(15),

contract\_id INT,

address VARCHAR(100),

account\_number VARCHAR(20),

equipment\_id INT REFERENCES Equipment(equipment\_id)

);

Листинг 2 – Создание таблиц / таблицы Employees

CREATE TABLE Employees (

employee\_id SERIAL PRIMARY KEY,

last\_name VARCHAR(50),

first\_name VARCHAR(50),

middle\_name VARCHAR(50),

phone\_number VARCHAR(15),

birth\_date DATE,

position VARCHAR(50),

contract\_id INT

);

Листинг 3 – Создание таблиц / таблицы Services

CREATE TABLE Services (

service\_id SERIAL PRIMARY KEY,

service\_name VARCHAR(50),

service\_price DECIMAL(10, 2),

service\_description TEXT,

equipment\_id INT REFERENCES Equipment(equipment\_id)

);

Листинг 4 – Создание таблиц / таблицы Equipment

CREATE TABLE Equipment (

equipment\_id SERIAL PRIMARY KEY,

equipment\_name VARCHAR(50),

equipment\_price DECIMAL(10, 2),

quantity INT

);

Листинг 5 – Создание таблиц / таблицы Payments

CREATE TABLE Payments (

payment\_id SERIAL PRIMARY KEY,

client\_id INT REFERENCES Clients(client\_id),

amount DECIMAL(10, 2),

previous\_payments DECIMAL(10, 2),

invoice\_id INT

);

Листинг 6 – Создание таблиц / таблицы Invoices

CREATE TABLE Invoices (

invoice\_id SERIAL PRIMARY KEY,

client\_id INT REFERENCES Clients(client\_id),

balance DECIMAL(10, 2),

account\_number VARCHAR(20),

service\_id INT REFERENCES Services(service\_id)

);

Листинг 7 – Создание таблиц / таблицы ClientContracts

CREATE TABLE ClientContracts (

contract\_id SERIAL PRIMARY KEY,

client\_id INT REFERENCES Clients(client\_id),

contract\_date DATE,

termination\_date DATE

);

Листинг 8 – Создание таблиц / таблицы EmployeeContracts

CREATE TABLE EmployeeContracts (

contract\_id SERIAL PRIMARY KEY,

employee\_id INT REFERENCES Employees(employee\_id),

contract\_date DATE,

termination\_date DATE

);

Листинг 9 – Создание таблиц / таблицы ClientServices

CREATE TABLE ClientServices (

client\_id INT REFERENCES Clients(client\_id),

service\_id INT REFERENCES Services(service\_id),

PRIMARY KEY (client\_id, service\_id)

);

Листинг 10 – Создание таблиц / таблицы EmployeeServices

CREATE TABLE EmployeeServices (

employee\_id INT REFERENCES Employees(employee\_id),

service\_id INT REFERENCES Services(service\_id),

PRIMARY KEY (employee\_id, service\_id)

);

Листинг 11 – Создание таблиц / таблицы Charges

CREATE TABLE Charges (

charge\_id SERIAL PRIMARY KEY,

contract\_id INT REFERENCES ClientContracts(contract\_id),

service\_id INT REFERENCES Services(service\_id),

date DATE,

amount DECIMAL(10, 2)

);

Листинг 12 – Создание таблицы/таблицы audit\_log

CREATE TABLE audit\_log (

id SERIAL PRIMARY KEY,

operation\_type VARCHAR(50),

table\_name VARCHAR(50),

changed\_by VARCHAR(50),

changed\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP

);

## 2.2 Создание представлений

Представлена методология создания представлений в базе данных, упрощающая доступ к данным и улучшающая производительность системы. Примеры создания представлений можно найти в листинге 13 – 17 (рис.9 – 13)

Листинг 13 – Просмотр всех категорий услуг

CREATE VIEW ServiceCategories AS

SELECT service\_id, service\_name, service\_description

FROM Services;

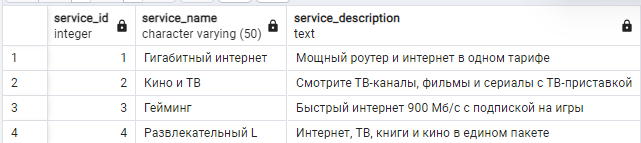


Рисунок 9 – Название всех услуг

Листинг 14 – Просмотр услуги с указанием цены и категории (рис.10)

CREATE VIEW ServiceDetails AS

SELECT service\_name AS "Услуга", service\_price AS "Цена", service\_description AS "Категория"

FROM Services;

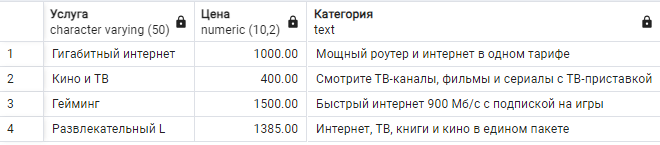


Рисунок 10 – Услуги с указанием цены и категории

Листинг 15 – Просмотр клиентов с указанием номера счета, адреса(рис.11)

CREATE VIEW ClientDetails AS

SELECT last\_name, first\_name, middle\_name, phone\_number, account\_number, address

FROM Clients;

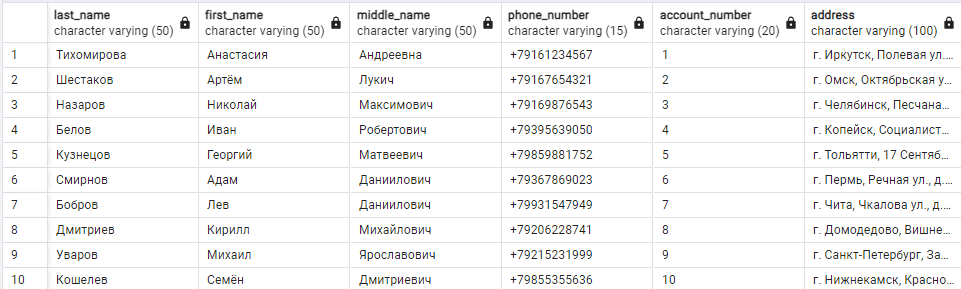


Рисунок 11 – Клиенты с указанием номера счета, адреса

Листинг 16 – Просмотр договоров на оказание услуг указанием клиента, сотрудника, заключившего договор.(рис.12)

CREATE VIEW ServiceContracts AS

SELECT

ClientContracts.contract\_id AS "Номер договора",

CONCAT(Clients.last\_name, ' ', Clients.first\_name, ' ', Clients.middle\_name) AS "Клиент",

CONCAT(Employees.last\_name, ' ', Employees.first\_name, ' ', Employees.middle\_name) AS "Сотрудник",

ClientContracts.contract\_date AS "Дата договора"

FROM

ClientContracts

JOIN

Clients ON ClientContracts.client\_id = Clients.client\_id

JOIN

EmployeeContracts ON ClientContracts.contract\_id = EmployeeContracts.contract\_id

JOIN

Employees ON EmployeeContracts.employee\_id = Employees.employee\_id;

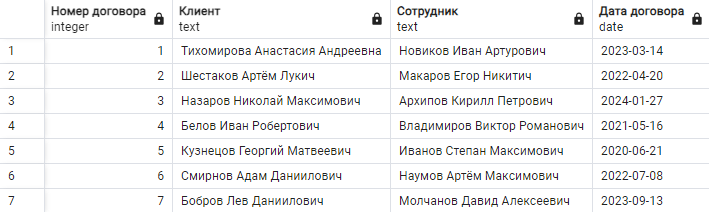


Рисунок 12 – Договоры на оказание услуг указанием клиента, сотрудника, заключившего договор

Листинг 17 – Просмотр оборудования, имеющееся в наличии(рис.13)

CREATE VIEW AvailableEquipment AS

SELECT equipment\_name AS "Наименование оборудования", quantity AS "Количество в наличии"

FROM Equipment

WHERE quantity > 0;

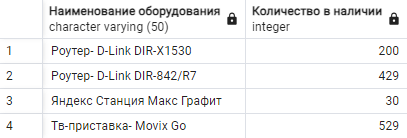


Рисунок 13 – Оборудование, имеющееся в наличии

## 2.3 Создание функций и хранимых процедур

Раздел посвящен разработке функций и хранимых процедур, обеспечивающих выполнение повторяющихся задач и бизнес-логики на стороне сервера базы данных. Соответствующие примеры представлены в листинге 18 – 24 (рис.14 – 20)

Листинг 18 – Функция. Получить список услуг и реализованных товаров по определенному договору

CREATE FUNCTION get\_services\_and\_goods(contract\_id INT)

RETURNS TABLE(service\_name VARCHAR(50), equipment\_name VARCHAR(50)) AS $$

BEGIN

RETURN QUERY

SELECT Services.service\_name, Equipment.equipment\_name

FROM ClientContracts

JOIN ClientServices ON ClientContracts.client\_id = ClientServices.client\_id

JOIN Services ON ClientServices.service\_id = Services.service\_id

JOIN Equipment ON Services.equipment\_id = Equipment.equipment\_id

WHERE ClientContracts.contract\_id = get\_services\_and\_goods.contract\_id;

END; $$ LANGUAGE plpgsql;

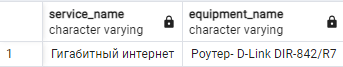


Рисунок 14 – Список услуг и реализованных товаров по определенному договору

Листинг 19 – Функция. Получить топ 5 клиентов с наибольшей суммой договора(рис.15)

CREATE OR REPLACE FUNCTION get\_top\_5\_clients() RETURNS TABLE (

client\_id INT,

total\_contract\_amount DECIMAL

) AS $$

BEGIN

RETURN QUERY

SELECT c.client\_id, SUM(ch.amount) as total\_contract\_amount

FROM Clients c

JOIN Charges ch ON c.contract\_id = ch.contract\_id

GROUP BY c.client\_id

ORDER BY total\_contract\_amount DESC

LIMIT 5;

END; $$

LANGUAGE plpgsql;

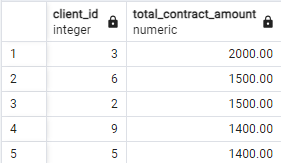


Рисунок 15 – 5 клиентов с наибольшей суммой договора

Листинг 20 – Функция. Вывести все оплаты за определенный период.(рис.16)

CREATE OR REPLACE FUNCTION get\_payments\_in\_period(\_start\_date DATE, \_end\_date DATE)

RETURNS TABLE (

payment\_id INT,

client\_id INT,

amount DECIMAL,

payment\_date DATE

) AS $$

BEGIN

RETURN QUERY

SELECT Payments.payment\_id, Payments.client\_id, Payments.amount, Charges.date AS payment\_date

FROM Payments

JOIN Charges ON Payments.payment\_id = Charges.charge\_id

WHERE Charges.date BETWEEN \_start\_date AND \_end\_date;

END; $$

LANGUAGE plpgsql;

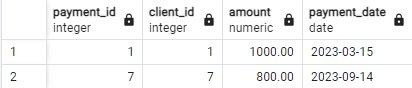


Рисунок 16 – все оплаты за определенный период

Листинг 21 – Функция. Вывести все начисления и оплаты по клиенту.(рис.17)

CREATE OR REPLACE FUNCTION get\_charges\_and\_payments(\_client\_id INT)

RETURNS TABLE (

charge\_id INT,

charge\_amount DECIMAL,

payment\_id INT,

payment\_amount DECIMAL

) AS $$

BEGIN

RETURN QUERY

SELECT Charges.charge\_id, Charges.amount AS charge\_amount, Payments.payment\_id, Payments.amount AS payment\_amount

FROM Charges, Payments

WHERE Charges.contract\_id = (SELECT contract\_id FROM Clients WHERE client\_id = \_client\_id)

AND Payments.client\_id = \_client\_id;

END; $$

LANGUAGE plpgsql;

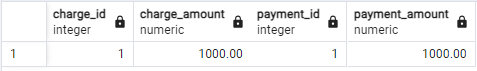


Рисунок 17 – все начисления и оплаты по клиенту

Листинг 22 – Процедура. Увеличить стоимость услуг на указанный процент.(рис.18)

CREATE OR REPLACE PROCEDURE increase\_service\_price(p\_percent FLOAT)

LANGUAGE plpgsql

AS $$

BEGIN

UPDATE Services SET service\_price = service\_price \* (1 + p\_percent / 100);

COMMIT;

END; $$

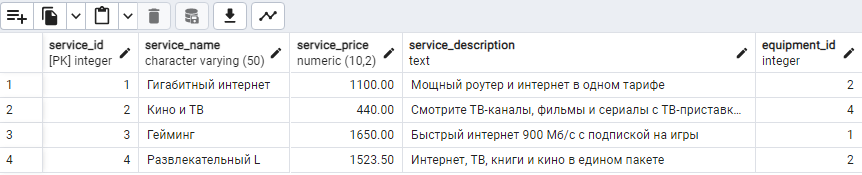


Рисунок 18 – Увеличение стоимости услуг на указанный процент

Листинг 23 – Процедура. Задать указанную стоимость указанной услуге.(рис.19)

CREATE OR REPLACE PROCEDURE set\_service\_price(p\_service\_id INT, p\_new\_price DECIMAL)

LANGUAGE plpgsql

AS $$

BEGIN

UPDATE Services SET service\_price = p\_new\_price WHERE service\_id = p\_service\_id;

COMMIT;

END; $$

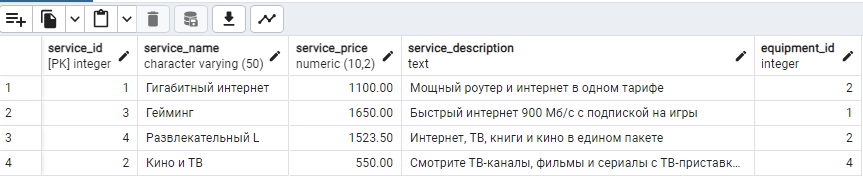


Рисунок 19 – Задана указанная стоимость услуге

Листинг 24 – Процедура. Биллинг на текущий месяц. Сделать начисления на текущий месяц.(рис.20)

CREATE OR REPLACE PROCEDURE billing\_for\_current\_month()

LANGUAGE plpgsql

AS $$

DECLARE

cur CURSOR FOR SELECT client\_id FROM Clients;

current\_client\_id INT;

current\_month INT := EXTRACT(MONTH FROM CURRENT\_DATE);

current\_year INT := EXTRACT(YEAR FROM CURRENT\_DATE);

BEGIN

OPEN cur;

LOOP

FETCH NEXT FROM cur INTO current\_client\_id;

EXIT WHEN NOT FOUND;

INSERT INTO Charges (contract\_id, service\_id, date, amount)

SELECT cc.contract\_id, cs.service\_id, CURRENT\_DATE, s.service\_price

FROM ClientContracts cc

JOIN ClientServices cs ON cc.client\_id = cs.client\_id

JOIN Services s ON cs.service\_id = s.service\_id

WHERE cc.client\_id = current\_client\_id AND

EXTRACT(MONTH FROM cc.contract\_date) <= current\_month AND

EXTRACT(YEAR FROM cc.contract\_date) <= current\_year AND

(cc.termination\_date IS NULL OR

EXTRACT(MONTH FROM cc.termination\_date) >= current\_month AND

EXTRACT(YEAR FROM cc.termination\_date) >= current\_year);

END LOOP;

CLOSE cur;

COMMIT;

END; $$

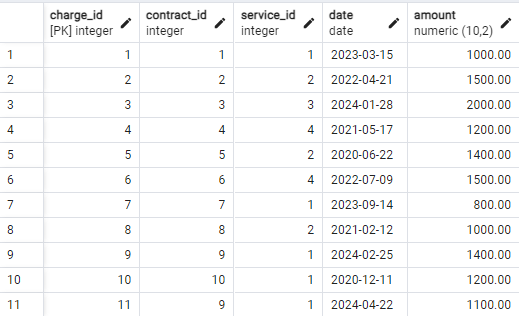


Рисунок 20 – Биллинг на текущий месяц

## 2.4 Создание триггеров

Описан процесс создания триггеров, которые автоматически выполняют определенные действия при изменении данных в таблицах базы данных. Примеры триггеров приведены в листинге 25 – 28

Листинг 25 – Триггер для автоматического обновления баланса счета после внесения платежа

CREATE OR REPLACE FUNCTION update\_balance\_after\_payment()

RETURNS TRIGGER AS $$

BEGIN

UPDATE Invoices

SET balance = balance - NEW.amount

WHERE client\_id = NEW.client\_id;

RETURN NEW;

END;

$$ LANGUAGE plpgsql;

CREATE TRIGGER trg\_update\_balance\_after\_payment

AFTER INSERT ON Payments

FOR EACH ROW

EXECUTE FUNCTION update\_balance\_after\_payment();

Листинг 26 – Автоматическое создание записи в таблице Charges при добавлении новой услуги клиенту

CREATE OR REPLACE FUNCTION add\_charge\_for\_new\_service()

RETURNS TRIGGER AS $$

BEGIN

INSERT INTO Charges (contract\_id, service\_id, date, amount)

VALUES (NEW.contract\_id, NEW.service\_id, CURRENT\_DATE, (SELECT service\_price FROM Services WHERE service\_id = NEW.service\_id));

RETURN NEW;

END;

$$ LANGUAGE plpgsql;

CREATE TRIGGER trg\_add\_charge\_for\_new\_service

AFTER INSERT ON ClientServices

FOR EACH ROW

EXECUTE FUNCTION add\_charge\_for\_new\_service();

Листинг 27 – Триггер для уменьшения количества оборудования на складе при его продаже

CREATE OR REPLACE FUNCTION update\_equipment\_quantity()

RETURNS TRIGGER AS $$

BEGIN

UPDATE Equipment

SET quantity = quantity - NEW.quantity

WHERE equipment\_id = NEW.equipment\_id;

RETURN NEW;

END;

$$ LANGUAGE plpgsql;

CREATE TRIGGER trg\_update\_equipment\_quantity

AFTER INSERT ON Charges

FOR EACH ROW

EXECUTE FUNCTION update\_equipment\_quantity()

Листинг 28 – Триггер для аудита вставки данных

CREATE OR REPLACE FUNCTION audit\_insert() RETURNS TRIGGER AS $$

BEGIN

INSERT INTO audit\_log (operation\_type, table\_name, changed\_by)

VALUES ('INSERT', TG\_TABLE\_NAME, current\_user);

RETURN NEW;

END;

$$ LANGUAGE plpgsql;

CREATE TRIGGER trg\_audit\_insert

AFTER INSERT ON Clients

FOR EACH ROW EXECUTE FUNCTION audit\_insert();

## 2.5 Разграничение прав доступа, создание ролей и пользователей

Таблица 12 – Роли и права

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Роль | Роль в БД | Функции в приложении | Справочные таблицы | | | | Операционные таблицы | | | | | | | Представления | Хранимые функции и процедуры | Триггеры |
| Clients | Employees | Services | Equipment | **Payments** | **Invoices** | **ClientContracts** | **EmployeeContracts** | **ClientServices** | **EmployeeServices** | Charges | В конце приложения делать | В конце приложения делать | В конце приложения делать |
| Администратор | AdminDB | Доступ ко всей базе данных | All | All | All | All | All | All | All | All | All | All | All | Полный доступ | Полный доступ | Полный доступ |
| Менеджер по работе с клиентами | ClientManager | Просмотр договоров | Read | Read | Read | Read | Read, Update | Read, Update | Read | Read | Read | Read | Update |  |  |  |
| Специалист технической поддержки | TechSupport | Просмотр договоров, вывод начислений и оплат клиента | Read | Read | Read | Read | Read, Update | Read, Update | Read, Update | Read | Read | Read | Read, Update |  |  |  |

Рассмотрены методы разграничения прав доступа и создание ролей и пользователей для обеспечения безопасности и контроля доступа к данным в системе. Подробные сведения представлены в листинге 29 – 31

Листинг 29 – Роль Админ БД

CREATE ROLE AdminDB WITH LOGIN PASSWORD 'admin\_password';

GRANT ALL PRIVILEGES ON DATABASE kursovaya TO AdminDB;

Листинг 30 – Роль Менеджер по работе с клиентами

CREATE ROLE ClientManager WITH LOGIN PASSWORD 'clientmanager\_password';

GRANT SELECT ON TABLE Clients TO ClientManager;

GRANT SELECT ON TABLE ClientContracts TO ClientManager;

Листинг 31 – Роль Специалист технической поддержки

CREATE ROLE TechSupport WITH LOGIN PASSWORD 'techsupport\_password';

GRANT SELECT ON TABLE Clients TO TechSupport;

GRANT SELECT ON TABLE Equipment TO TechSupport;

GRANT SELECT ON TABLE ClientContracts TO TechSupport;

## 2.6 Разработка интерфейса приложения

Интерфейс пользователя (UI) – это средство взаимодействия между пользователем и программным обеспечением. Интерфейс позволяет пользователям взаимодействовать с приложением, вводить данные, выполнять команды и получать результаты. В зависимости от способов взаимодействия, интерфейсы могут быть различных видов: командный интерфейс, графический интерфейс и другие.

Виды интерфейсов

1. Командный интерфейс (CLI):

Командный интерфейс позволяет пользователям взаимодействовать с системой путем ввода текстовых команд. Этот тип интерфейса используется в терминалах и командных строках, таких как командная строка Windows и терминал Linux.

1. Графический интерфейс (GUI):

Графический интерфейс предоставляет пользователю элементы управления, такие как окна, кнопки, меню, текстовые поля и т.д., для взаимодействия с приложением. GUI является наиболее распространенным видом интерфейса, используемым в современных приложениях.

1. Интерфейс на основе жестов:

Этот тип интерфейса использует сенсорные экраны и жесты для управления приложением. Примеры включают интерфейсы смартфонов и планшетов.

1. Голосовой интерфейс:

Позволяет пользователям взаимодействовать с системой с помощью голосовых команд. Примеры включают виртуальных помощников, таких как Siri, Alexa и Google Assistant.

1. Интерфейс дополненной реальности (AR) и виртуальной реальности (VR):

Эти интерфейсы предоставляют пользователям возможность взаимодействия с цифровым контентом в дополненной или виртуальной среде. Примеры включают очки дополненной реальности и VR-шлемы.

Выбор интерфейса для приложения

Для разработки данного приложения был выбран графический интерфейс с использованием технологии Windows Presentation Foundation (WPF). WPF – это мощная платформа для создания приложений с богатым пользовательским интерфейсом на базе Windows. Она предоставляет широкие возможности для разработки гибких и настраиваемых графических интерфейсов, включая поддержку 2D и 3D графики, анимации и мультимедийного контента.

Преимущества использования WPF:

* Гибкость и мощность: WPF позволяет создавать сложные и настраиваемые интерфейсы с использованием XAML (Extensible Application Markup Language) и C#;
* Поддержка графики и мультимедиа: WPF поддерживает работу с 2D и 3D графикой, анимацией и мультимедийным контентом, что делает интерфейсы более привлекательными и интерактивными;
* Стили и шаблоны: WPF предоставляет мощные механизмы для настройки внешнего вида и поведения элементов интерфейса с помощью стилей и шаблонов;
* Привязка данных: WPF обеспечивает удобные механизмы для привязки данных, что упрощает работу с данными и обновление интерфейса в реальном времени.

Для прототипирования интерфейса используется Wireframe, который позволяет визуализировать расположение и функциональность элементов управления на экране (рис.21 – 23)

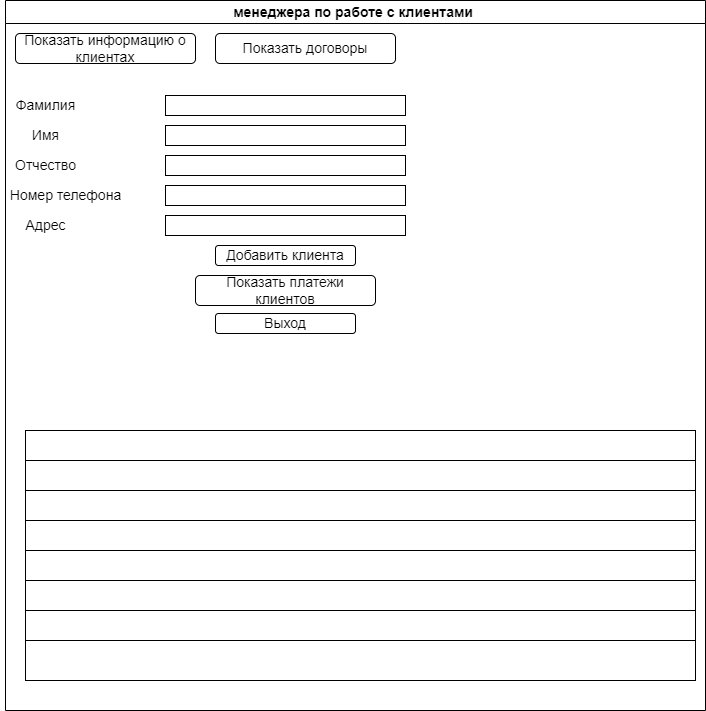


Рисунок 21 – Wireframe окна менеджера по работе с клиентами

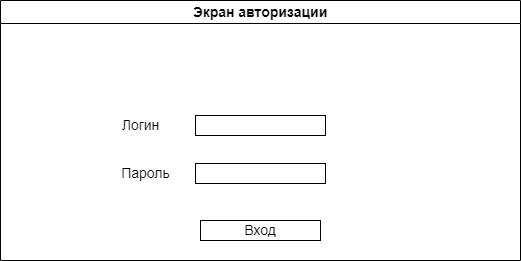


Рисунок 22 – Wireframe окна авторизации

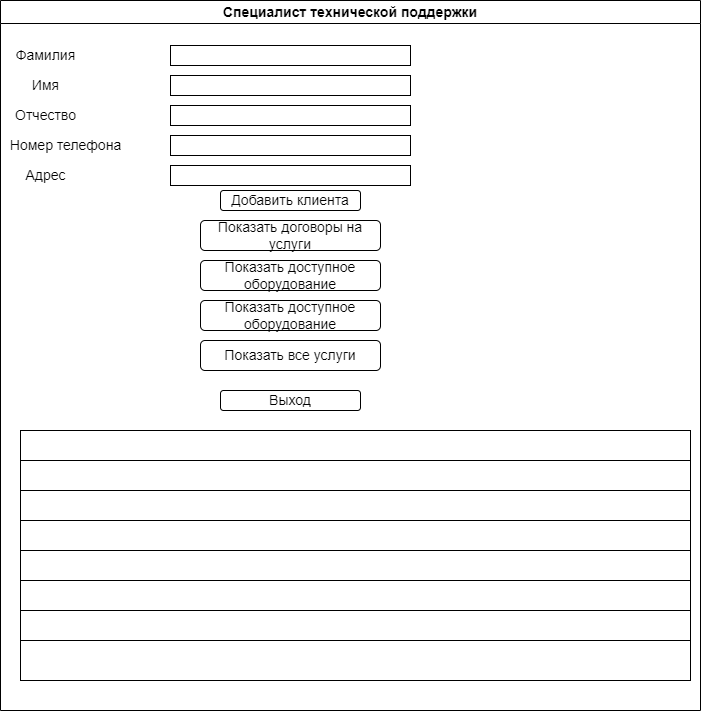


Рисунок 23 – Wireframe окна специалиста технической поддержки

LoginButton\_Click – Эта функция обрабатывает нажатие кнопки "Вход" в окне аутентификации. Она проверяет учетные данные пользователя и перенаправляет его на соответствующее окно в зависимости от роли.

LoadClientDetails\_Click – Эта функция обрабатывает нажатие кнопки для загрузки данных о клиентах.

LoadServiceContracts\_Click – Эта функция обрабатывает нажатие кнопки для загрузки данных о договорах на услуги.

LoadClientPayments\_Click – Эта функция обрабатывает нажатие кнопки для загрузки данных о платежах клиентов за определенный период.

AddClient\_Click – Эта функция обрабатывает нажатие кнопки для добавления нового клиента в базу данных.

ExitButton\_Click – Эта функция обрабатывает нажатие кнопки "Выход", закрывая текущее окно и открывая главное окно приложения.

LoadServiceCategories\_Click – Эта функция обрабатывает нажатие кнопки для загрузки данных о категориях услуг.

LoadAvailableEquipment\_Click – Эта функция обрабатывает нажатие кнопки для загрузки данных о доступном оборудовании.

AddClientButton\_Click – Эта функция обрабатывает нажатие кнопки для добавления нового клиента в базу данных (дублируется в двух местах кода).

Скрипты для создания интерфейса приложения представлены в листингах 32 – 34.

Листинг 32 – Окно авторизации

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="Auto" />

<RowDefinition Height="Auto" />

<RowDefinition Height="Auto" />

<RowDefinition Height="\*" />

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="Auto" />

<ColumnDefinition Width="\*" />

</Grid.ColumnDefinitions>

<Label Grid.Row="0" Grid.Column="0" Margin="10" VerticalAlignment="Center">Логин:</Label>

<TextBox x:Name="UsernameTextBox" Grid.Row="0" Grid.Column="1" Margin="10" VerticalAlignment="Center"/>

<Label Grid.Row="1" Grid.Column="0" Margin="10" VerticalAlignment="Center">Пароль:</Label>

<PasswordBox x:Name="PasswordBox" Grid.Row="1" Grid.Column="1" Margin="10" VerticalAlignment="Center"/>

<Button Grid.Row="2" Grid.Column="1" Margin="276,0,350,0" VerticalAlignment="Center" Width="100" Click="LoginButton\_Click">Вход</Button>

</Grid>

Листинг 33 – Окно менеджера по работе с клиентами

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="Auto" />

<RowDefinition Height="Auto" />

<RowDefinition Height="Auto" />

<RowDefinition Height="Auto" />

<RowDefinition Height="Auto" />

<RowDefinition Height="\*" />

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="Auto" />

<ColumnDefinition Width="\*" />

</Grid.ColumnDefinitions>

<Button Grid.Column="0" Margin="10,0,10,0" VerticalAlignment="Center" Click="LoadClientDetails\_Click">Показать информацию о клиентах</Button>

<Button Grid.Column="1" Margin="10,0,10,0" VerticalAlignment="Center" Click="LoadServiceContracts\_Click">Показать договоры</Button>

<Button Grid.Row="5" Grid.Column="1" Margin="10,73,10,0" VerticalAlignment="Top" Click="LoadClientPayments\_Click">Показать платежи клиентов</Button>

<Label Grid.Row="1" Grid.Column="0" Margin="10" VerticalAlignment="Center">Фамилия:</Label>

<TextBox x:Name="LastNameTextBox" Grid.Row="1" Grid.Column="1" Margin="10" VerticalAlignment="Center"/>

<Label Grid.Row="2" Grid.Column="0" Margin="10" VerticalAlignment="Center">Имя:</Label>

<TextBox x:Name="FirstNameTextBox" Grid.Row="2" Grid.Column="1" Margin="10" VerticalAlignment="Center"/>

<Label Grid.Row="3" Grid.Column="0" Margin="10" VerticalAlignment="Center">Отчество:</Label>

<TextBox x:Name="MiddleNameTextBox" Grid.Row="3" Grid.Column="1" Margin="10" VerticalAlignment="Center"/>

<Label Grid.Row="4" Grid.Column="0" Margin="10" VerticalAlignment="Center">Номер телефона:</Label>

<TextBox x:Name="PhoneNumberTextBox" Grid.Row="4" Grid.Column="1" Margin="10" VerticalAlignment="Center"/>

<Label Grid.Row="5" Grid.Column="0" Margin="10,10,10,0" VerticalAlignment="Top">Адрес:</Label>

<TextBox x:Name="AddressTextBox" Grid.Row="5" Grid.Column="1" Margin="10,10,10,0" VerticalAlignment="Top"/>

<Button Grid.Row="5" Grid.Column="1" Margin="10,48,10,0" VerticalAlignment="Top" Click="AddClient\_Click">Добавить клиента</Button>

<DataGrid x:Name="dataGrid" Margin="10,136,10,10" Grid.Row="5" Grid.ColumnSpan="2"/>

<Button Grid.Row="5" Grid.Column="1" Margin="10,103,10,0" VerticalAlignment="Top" Click="ExitButton\_Click">Выход</Button>

</Grid>

Листинг 34 – Окно специалиста технической поддержки

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="Auto" />

<RowDefinition Height="Auto" />

<RowDefinition Height="Auto" />

<RowDefinition Height="Auto" />

<RowDefinition Height="Auto" />

<RowDefinition Height="Auto" />

<RowDefinition Height="\*" />

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="Auto" />

<ColumnDefinition Width="\*" />

</Grid.ColumnDefinitions>

<Label Grid.Row="0" Grid.Column="0" Margin="10" VerticalAlignment="Center">Фамилия:</Label>

<TextBox x:Name="LastNameTextBox" Grid.Row="0" Grid.Column="1" Margin="10" VerticalAlignment="Center"/>

<Label Grid.Row="1" Grid.Column="0" Margin="10" VerticalAlignment="Center">Имя:</Label>

<TextBox x:Name="FirstNameTextBox" Grid.Row="1" Grid.Column="1" Margin="10" VerticalAlignment="Center"/>

<Label Grid.Row="2" Grid.Column="0" Margin="10" VerticalAlignment="Center">Отчество:</Label>

<TextBox x:Name="MiddleNameTextBox" Grid.Row="2" Grid.Column="1" Margin="10" VerticalAlignment="Center"/>

<Label Grid.Row="3" Grid.Column="0" Margin="10" VerticalAlignment="Center">Номер телефона:</Label>

<TextBox x:Name="PhoneNumberTextBox" Grid.Row="3" Grid.Column="1" Margin="10" VerticalAlignment="Center"/>

<Label Grid.Row="4" Grid.Column="0" Margin="10" VerticalAlignment="Center">Адрес:</Label>

<TextBox x:Name="AddressTextBox" Grid.Row="4" Grid.Column="1" Margin="10" VerticalAlignment="Center"/>

<Button Grid.Row="5" Grid.Column="1" Margin="10" VerticalAlignment="Center" Click="AddClientButton\_Click">Добавить клиента</Button>

<Button Grid.Row="6" Grid.Column="1" Margin="10,65,10,0" VerticalAlignment="Top" Click="LoadServiceCategories\_Click">Показать все услуги</Button>

<Button Grid.Row="6" Grid.Column="1" Margin="10,34,10,0" VerticalAlignment="Top" Click="LoadAvailableEquipment\_Click">Показать доступное оборудование</Button>

<Button Grid.Row="6" Grid.Column="1" Margin="10,2,10,0" VerticalAlignment="Top" Click="LoadServiceContracts\_Click">Показать договоры на услуги</Button>

<DataGrid x:Name="dataGrid" Margin="10,126,10,10" Grid.Row="6" Grid.ColumnSpan="2"/>

<Button Grid.Row="6" Grid.Column="1" Margin="10,96,10,0" VerticalAlignment="Top" Click="ExitButton\_Click">Выход</Button>

</Grid>

# 3 Разработка документации

## 3.1 Руководство пользователя

3.1.1 Введение

3.1.1.1 Область применения

Приложение предназначено для телекоммуникационных компаний и применяется в процессах управления заявками на выполнение различных услуг. Система может использоваться на всех этапах обслуживания клиентов, от регистрации заявок до их выполнения и контроля качества оказанных услуг.

3.1.1.2 Краткое описание возможностей

Приложение предоставляет следующие возможности:

* Регистрация и учет заявок клиентов на предоставление телекоммуникационных услуг;
* Назначение заявок исполнителям и контроль за их выполнением;
* Управление статусами заявок;
* Генерация отчетов по заявкам и их статусам.

3.1.1.3 Уровень подготовки пользователя

Для работы с приложением пользователям необходимо иметь базовые знания работы с компьютером и операционной системой Windows. Опыт работы с аналогичными системами управления заявками будет преимуществом, но не является обязательным.

3.1.1.4 Перечень эксплуатационной документации

Для корректной работы с приложением рекомендуется ознакомиться с следующими документами:

* Руководство пользователя операционной системы Windows;
* Руководство по эксплуатации приложения для управления заявками.

3.1.2 Назначение и условия применения

3.1.2.1 Назначение

Приложение предназначено для автоматизации процессов управления заявками в телекоммуникационных компаниях, что позволяет улучшить качество обслуживания клиентов и повысить эффективность работы сотрудников.

3.1.2.2 Условия применения

Приложение предназначено для работы на компьютерах под управлением операционной системы Windows. Требуется подключение к локальной сети для взаимодействия с базой данных.

3.1.3 Подготовка к работе

3.1.3.1 Состав и содержание дистрибутивного носителя данных

Для установки приложения необходимы:

* Операционная система Windows;
* Инсталляционные файлы приложения для управления заявками;
* Инструкции по установке и настройке приложения.

3.1.3.2 Порядок загрузки данных и программ

* Установите операционную систему Windows (если еще не установлена);
* Установите приложение, следуя инструкциям инсталлятора;
* Настройте подключение к базе данных, используя параметры, указанные в руководстве по эксплуатации.

3.1.3.3 Порядок проверки работоспособности

* Запустите приложение;
* Проверьте успешное подключение к базе данных;
* Создайте тестовую заявку и убедитесь в ее корректном сохранении и отображении в системе.

3.1.4 Описание операций

3.1.4.1 Операции менеджера по работе с клиентами

Менеджер по работе с клиентами в приложении выполняет следующие основные операции (рис.24 – 25):

1. Добавление нового клиента:

* Менеджер вводит фамилию, имя, отчество, номер телефона и адрес нового клиента в соответствующие текстовые поля;
* После ввода данных нажимает кнопку "Добавить клиента";
* Приложение сохраняет данные клиента в базу данных, отображает сообщение об успешном добавлении и обновляет таблицу клиентов.

1. Просмотр информации о клиентах:

* Менеджер может просматривать информацию о всех клиентах, хранящихся в базе данных;
* Для этого используется кнопка "Показать информацию о клиентах", после нажатия которой в таблице выводится информация обо всех клиентах.

1. Просмотр договоров на услуги:

* Менеджер может просматривать заключенные договора на услуги;
* Для этого используется кнопка "Показать договоры", после нажатия которой в таблице выводится информация обо всех договорах.

1. Просмотр платежей клиентов:

* Менеджер может просматривать информацию о платежах клиентов за определенный период;
* Для этого используется кнопка "Показать платежи клиентов", после нажатия которой в таблице выводится информация о платежах.

1. Выход из системы:

* Менеджер может выйти из текущей сессии и вернуться на экран входа в систему;
* Для этого используется кнопка "Выход", которая закрывает текущее окно и открывает окно входа в систему.

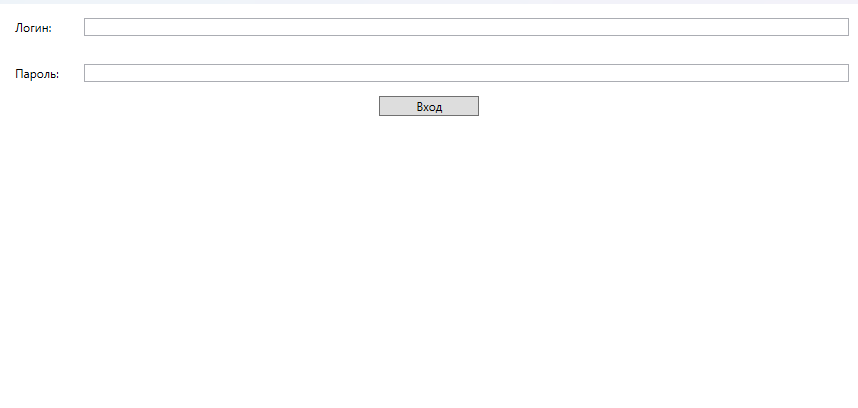


Рисунок 24 – Окно авторизации

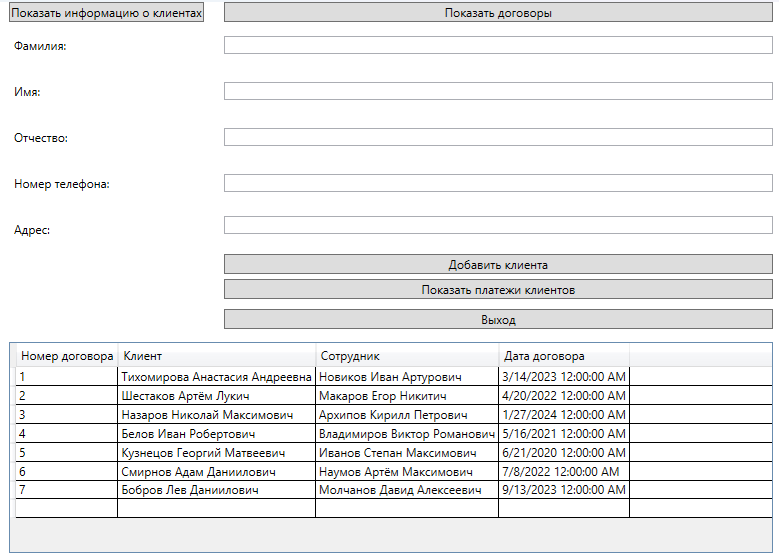


Рисунок 25 – Окно менеджера

3.1.4.2 Описание операций специалиста технической поддержки

Специалист технической поддержки выполняет следующие операции в приложении(рис.26):

1. Добавление нового клиента:
   * + Специалист вводит фамилию, имя, отчество, номер телефона и адрес нового клиента в соответствующие текстовые поля;
     + После ввода данных нажимает кнопку "Добавить клиента";
     + Приложение сохраняет данные клиента в базу данных, отображает сообщение об успешном добавлении и обновляет таблицу клиентов.
2. Просмотр доступных услуг:
   * + Специалист может просматривать все доступные услуги;
     + Для этого используется кнопка "Показать все услуги", после нажатия которой в таблице выводится информация обо всех услугах.
3. Просмотр доступного оборудования:
   * + Специалист может просматривать все доступное оборудование;
     + Для этого используется кнопка "Показать доступное оборудование", после нажатия которой в таблице выводится информация обо всем оборудовании.
4. Просмотр договоров на услуги:
   * + Специалист может просматривать заключенные договора на услуги;
     + Для этого используется кнопка "Показать договоры на услуги", после нажатия которой в таблице выводится информация обо всех договорах на услуги.
5. Выход из системы:
   * + Специалист может выйти из текущей сессии и вернуться на экран входа в систему;

* Для этого используется кнопка "Выход", которая закрывает текущее окно и открывает окно входа в систему.

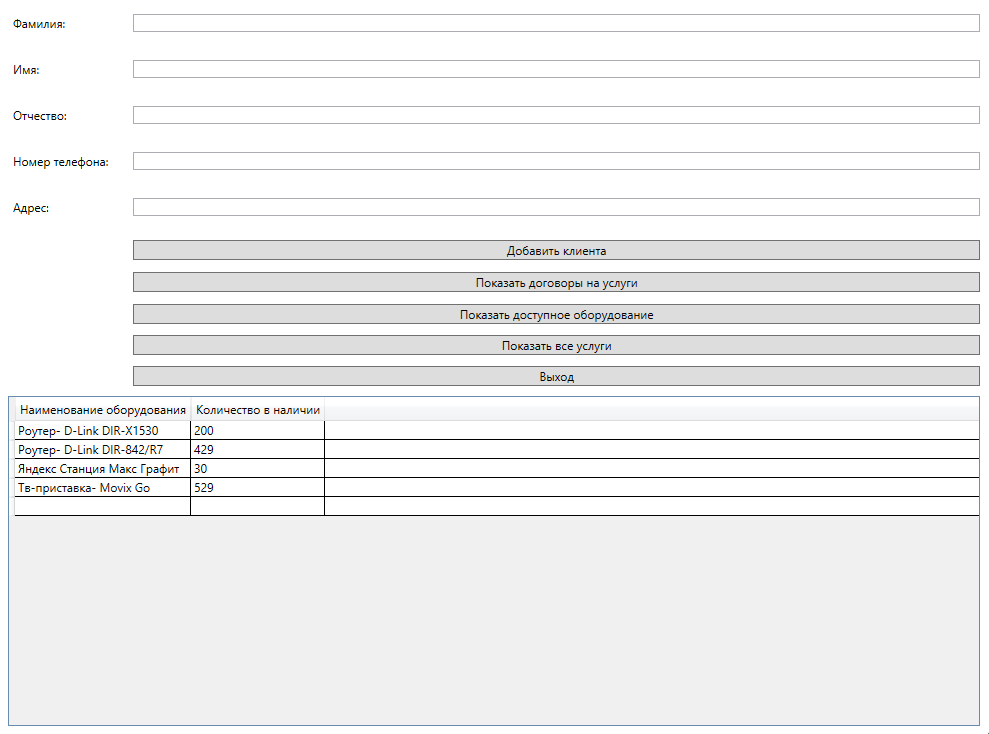


Рисунок 26 – Окно поддержки

## 3.2 Руководство программиста

3.2.1 Назначение и условия применения

Руководство программиста предназначено для разработчиков и администраторов системы управления заявками телекоммуникационных услуг. Данное руководство содержит описание функциональности системы, методы развертывания и запуска ПО, а также формат входных и выходных данных. Программное обеспечение разработано для работы в операционной системе Windows и использует технологию WPF (Windows Presentation Foundation) для создания графического интерфейса. В качестве системы управления базами данных используется PostgreSQL, обеспечивая надежное хранение и управление данными.

3.2.2 Характеристика ПО

Программное обеспечение для управления заявками телекоммуникационных услуг включает в себя несколько основных модулей;

* Модуль для менеджера по работе с клиентами: управление клиентскими данными, просмотр заявок и заключенных договоров;
* Модуль для специалиста технической поддержки: управление данными об услугах и оборудовании, просмотр и изменение заявок.

Основные технологии, используемые в разработке:

* Язык программирования: C#;
* Платформа: .NET Framework версии 4.6;
* GUI: Windows Presentation Foundation (WPF) версия 17.8.6;
* База данных: PostgreSQL версия 8.7.

3.2.3 Развертывание ПО

Для развертывания системы управления заявками необходимо выполнить следующие шаги:

* + - Установите операционную систему Windows, если она еще не установлена;
    - Установите PostgreSQL и настройте базу данных согласно прилагаемой инструкции;
    - Разверните серверную часть приложения, загрузив и настроив все необходимые компоненты;
    - Установите клиентскую часть приложения на рабочие станции пользователей;
* Проверьте подключение клиентской части к серверной части и базам данных.

3.2.4 Запуск ПО

Для запуска системы управления заявками выполните следующие действия:

* Откройте клиентское приложение;
* Введите учетные данные для входа в систему;
* Нажмите кнопку "Вход" для перехода к основному интерфейсу приложения;
* Выберите необходимую функцию из меню приложения для начала работы.

3.2.5 Входные и выходные данные

Входные данные:

* Данные о клиентах: ФИО, контактная информация, адрес;
* Данные о заявках: описание заявки, статус, назначенный исполнитель;
* Данные об услугах: название услуги, описание, стоимость;
* Данные об оборудовании: название оборудования, количество, статус.

Выходные данные:

* Отчеты о заявках: информация о выполненных, текущих и новых заявках;
* Информация о клиентах: список клиентов с их контактной информацией;
* Данные об услугах: полный список предоставляемых услуг;
* Данные об оборудовании: текущее состояние и количество оборудования.

# ЗАКЛЮЧЕНИЕ

В данной курсовой работе была разработана информационная система «Портал телекоммуникационных услуг», предназначенная для автоматизации учета и управления услугами телекоммуникационных компаний. Основной целью проекта было повышение эффективности и качества обслуживания клиентов через автоматизацию процессов управления услугами и учета клиентов. В ходе работы были выполнены следующие задачи, поставленные в разделе 1.3 "Постановка задачи":

1. Анализ предметной области и существующих решений:

* Изучены основные процессы в сфере телекоммуникационных услуг и ключевые участники;
* Проанализированы существующие программные продукты для выявления их сильных и слабых сторон.

1. Разработка проектной документации:

* Созданы концептуальные и физические модели базы данных;
* Описан словарь данных, содержащий все сущности и их атрибуты.

1. Проектирование и реализация базы данных:

* В базе данных было создано 10 таблиц, из которых 3 справочные и 7 операционных​(12)​;
* Разработаны и внедрены представления, хранимые процедуры и триггеры для обеспечения бизнес-логики и аудита изменений.

1. Разработка графического интерфейса и функциональных модулей приложения:

* С использованием технологии WPF создано десктопное приложение с удобным и интуитивно понятным интерфейсом;
* Реализованы основные функциональные модули, включая управление заявками, учет услуг и взаимодействие с клиентами.

1. Тестирование и отладка приложения:

* Приложение было тщательно протестировано для выявления и устранения ошибок;
* Выполнена оценка производительности и стабильности работы системы.

1. Документация программного продукта:

* Подготовлено руководство пользователя и программиста, включающее описание всех функциональных возможностей и инструкций по использованию системы.

Характеристики базы данных

База данных включает следующие компоненты:

1. Таблицы:

* 10 таблиц, из которых 3 справочные и 7 операционных​(12)​.

1. Хранимые функции и процедуры:

* Хранимые функции и процедуры для реализации бизнес-логики.

1. Триггеры:

* Триггеры для автоматического выполнения определенных действий и аудита изменений.

Реализация бизнес-логики

Бизнес-логика была частично реализована на стороне базы данных с помощью хранимых функций и процедур, что позволило повысить производительность и упростить поддержку приложения. Основная бизнес-логика, связанная с пользовательским интерфейсом, была реализована в приложении на языке программирования C# с использованием WPF.

Роли в системе

Система предусматривает несколько ролей пользователей, включая администратора, менеджера и клиента, каждая из которых имеет свои права доступа и функциональные возможности.

Функционал приложения

Приложение включает следующие основные функции:

* Управление заявками и услугами;
* Ведение учета клиентов и их данных;
* Генерация отчетов и статистических данных;
* Управление пользователями и ролями.

Преимущества приложения

Разработанное приложение предоставляет телекоммуникационным компаниям следующие преимущества:

* Автоматизация рутинных задач и процессов;
* Повышение точности и скорости обработки заявок;
* Улучшение качества обслуживания клиентов;
* Упрощение взаимодействия между различными отделами компании.

Дальнейшее развитие приложения

Для дальнейшего улучшения системы рекомендуется:

* Расширить функциональные возможности приложения;
* Интегрировать систему с другими корпоративными сервисами;
* Оптимизировать производительность и масштабируемость базы данных;
* Внедрить дополнительные меры безопасности и защиты данных.

# СПИСОК ИСТОЧНИКОВ

1. Шилдт Г. C# 8.0. Полное руководство : учебник. — Питер, 2020. — 896 с.
2. Троелсен Э. C# 9.0 и .NET 5.0 для профессионалов : учебник. — Питер, 2021. — 1504 с.
3. Албахари Д., Албахари Б. C# 9.0 in a Nutshell : справочник. — O'Reilly, 2021. — 1088 с.
4. Липперт Э. Язык программирования C# 8 и платформа .NET Core : учебное пособие. — БХВ-Петербург, 2019. — 704 с.
5. Фримен А., Робсон Р. C# для профессионалов : учебник. — Вильямс, 2020. — 896 с.
6. Грю Э., Жим Л. C# и платформа .NET Core : учебное пособие. — Питер, 2019. — 752 с.
7. Дейт К. PostgreSQL : введение в систему и язык SQL : учебник. — O'Reilly, 2021. — 420 с.
8. Харрисон С. PostgreSQL. Введение в разработку : учебное пособие. — Питер, 2020. — 480 с.
9. Бекстеин Х. PostgreSQL 12 High Performance : практическое руководство. — Packt Publishing, 2020. — 382 с.
10. Курячий Л. PostgreSQL 11. Руководство администратора : практическое руководство. — БХВ-Петербург, 2019. — 464 с.
11. Хесс Д. PostgreSQL. Эффективное использование : практическое руководство. — O'Reilly, 2020. — 380 с.
12. Лозинский Д. PostgreSQL 13 для профессионалов : учебник. — Питер, 2021. — 608 с.
13. Чен П. PostgreSQL. Архитектура и программирование : учебное пособие. — БХВ-Петербург, 2020. — 352 с.
14. Григорик Л. PostgreSQL. Разработка и оптимизация : практическое руководство. — Питер, 2020. — 416 с.
15. Линч М. PostgreSQL High Availability Cookbook : практическое руководство. — Packt Publishing, 2021. — 396 с.
16. Metanit: сайт о программировании. – URL: https://metanit.com/sharp/
17. PostgreSQL Documentation: официальный сайт. – URL: https://www.postgresql.org/docs/current/

# ПРИЛОЖЕНИЕ А ЛИСТИНГ СКРИПТОВ ПРОГРАММНОГО КОДА

Листинг 35 – Инициализация окна и загрузка клиентов

public partial class Window1 : Window

{

private string connectionString = "Host=127.0.0.1;Username=postgres;Password=12345;Database=KURSOVAYA\_RABOTA\_PROVIDERS";

public Window1()

{

InitializeComponent();

LoadClients();

}

private void LoadClients()

{

using (NpgsqlConnection connection = new NpgsqlConnection(connectionString))

{

try

{

connection.Open();

using (NpgsqlCommand command = new NpgsqlCommand("SELECT \* FROM Clients", connection))

{

NpgsqlDataAdapter dataAdapter = new NpgsqlDataAdapter(command);

DataTable dataTable = new DataTable();

dataAdapter.Fill(dataTable);

dataGrid.ItemsSource = dataTable.DefaultView;

}

}

catch (Exception ex)

{

MessageBox.Show("Error: " + ex.Message);

}

}

}

}

Листинг 36 – Обработка нажатия кнопки «Добавить клиента»

private void AddClientButton\_Click(object sender, RoutedEventArgs e)

{

string lastName = LastNameTextBox.Text;

string firstName = FirstNameTextBox.Text;

string middleName = MiddleNameTextBox.Text;

string phoneNumber = PhoneNumberTextBox.Text;

string address = AddressTextBox.Text;

if (string.IsNullOrWhiteSpace(lastName) || string.IsNullOrWhiteSpace(firstName) || string.IsNullOrWhiteSpace(phoneNumber))

{

MessageBox.Show("Пожалуйста, заполните все обязательные поля.");

return;

}

if (AddClientToDatabase(lastName, firstName, middleName, phoneNumber, address))

{

MessageBox.Show("Клиент успешно добавлен.");

LoadClients();

ClearInputFields();

}

else

{

MessageBox.Show("Ошибка при добавлении клиента.");

}

}

Листинг 37 – Добавление клиента в базу данных

private bool AddClientToDatabase(string lastName, string firstName, string middleName, string phoneNumber, string address)

{

using (NpgsqlConnection conn = new NpgsqlConnection(connectionString))

{

try

{

conn.Open();

using (NpgsqlCommand cmd = new NpgsqlCommand(

"INSERT INTO Clients (last\_name, first\_name, middle\_name, phone\_number, address) VALUES (@lastName, @firstName, @middleName, @phoneNumber, @address)", conn))

{

cmd.Parameters.AddWithValue("lastName", lastName);

cmd.Parameters.AddWithValue("firstName", firstName);

cmd.Parameters.AddWithValue("middleName", middleName);

cmd.Parameters.AddWithValue("phoneNumber", phoneNumber);

cmd.Parameters.AddWithValue("address", address);

cmd.ExecuteNonQuery();

return true;

}

}

catch (Exception ex)

{

MessageBox.Show("Error: " + ex.Message);

return false;

}

}

}

Листинг 38 – Обработка нажатия кнопки «Показать все категории услуг»

private void LoadServiceCategories\_Click(object sender, RoutedEventArgs e)

{

LoadData("SELECT \* FROM ServiceCategories");

}