|  |
| --- |
| МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ |
| ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ |
| «Национальный исследовательский ядерный университет «МИФИ» |
| **Технологический институт –** |
| филиал федерального государственного автономного образовательного учреждения высшего |
| образования «Национальный исследовательский ядерный университет «МИФИ» |

**(ТИ НИЯУ МИФИ)**

**Кафедра информационных технологий и прикладной математики**

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

к курсовой работе

по дисциплине «Разработка программных модулей»

**Разработка клиент-серверного приложения для автоматизации работы сервисного центра**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | | |  |
| **Руководитель**  преподаватель |  |  |  | А.А.Шляпкин |
| *(должность)* |  | *(подпись)* |  | *(И.О. Фамилия)* |
|  |  |  |  |  |
| **Студент**  ИСП-311 |  |  |  | В.Н.Котельников |
| *(группа)* |  | *(подпись)* |  | *(И.О. Фамилия)* |

Курсовая работа защищена с оценкой «\_\_\_\_\_\_\_\_\_\_\_\_» «\_\_\_\_\_\_» \_\_\_\_\_\_\_\_\_20\_\_ г.

Лесной 2024

**СОДЕРЖАНИЕ**

**Элементы оглавления не найдены.**

# **ВВЕДЕНИЕ**

Современная информационная технология ставит перед разработчиками задачу создания эффективных и удобных программных модулей, способных решать разнообразные задачи и обеспечивать эффективное взаимодействие с данными. В рамках курсовой работы по модулю "Разработка программных модулей" была поставлена задача разработать базу данных и клиент-серверное приложение, способное обеспечить хранение и обработку информации, а также обеспечить доступ к данным для конечных пользователей.

Целью данного проекта является создание программного решения, которое позволит эффективно управлять данными и обеспечить пользователей надежным и удобным интерфейсом для работы с информацией. В рамках работы была создана база данных, а также разработано клиент-серверное приложение, которое предоставляет доступ к этим данным через сеть.

Для реализации проекта, необходимо выполнить задачи:

* Изучить предметную область;
* Спроектировать базу данных;
* Создать базу данных;
* Разработать клиентскую часть приложения;
* Протестировать приложение.

# **Анализ предметной области**

## **Постановка задачи**

В рамках курсовой работы необходимо разработать систему, которая бы осуществляла:

1. **Учет клиентов и заявок**:
   * Регистрация новых клиентов в системе сервисного центра.
   * Ведение базы данных клиентов с контактной информацией и историей обращений.
   * Создание и учет заявок на обслуживание от клиентов.
2. **Планирование и управление задачами**:
   * Организация и распределение задач между сотрудниками сервисного центра.
   * Отслеживание статуса выполнения задач и сроков их выполнения.
   * Оповещение сотрудников о новых задачах или изменениях в статусе текущих.
3. **Финансовый учет**:
   * Возможность выставления счетов клиентам за оказанные услуги.
   * Возможность изменения цен на услуги.
4. **Аутентификация и безопасность**:
   * Реализация механизмов аутентификации пользователей и контроля доступа к функционалу приложения.
   * Защита конфиденциальной информации клиентов и компании.
5. **Удобный интерфейс пользователя**:
   * Создание интуитивно понятного пользовательского интерфейса для удобного взаимодействия с приложением.
   * Оптимизация процесса ввода данных и выполнения операций для повышения производительности сотрудников.

Необходимо реализовать ролевую модель доступа к приложению, учитывающую возможность работы с ним администраторов, операторов и мастеров.

## **Описание бизнес-процесса**

Цель приложения для сервисного центра - обеспечить эффективное управление процессом обслуживания клиентов, автоматизировать учет заявок, оптимизировать работу персонала и повысить качество обслуживания. Ключевые задачи включают в себя управление клиентами и заявками, планирование работ.

Новые сотрудники сервисного центра добавляются Администратором, указываются их роль, имя, фамилия, телефон, пароль, статус работы. Также, Администратор может менять любые данные работника при необходимости.

Администратор может добавлять новые услуги сервисного центра и менять их цену и название по необходимости.

Кроме того, администратор имеет доступ ко всему функционалу приложения и в определенных случаях может выполнять роль оператора.

Оператор принимает заказы по телефону и назначает мастера для выполнения заказа. В функционал оператора входит добавление новых клиентов, добавление новых заказов и услуг к ним, изменение даты выполнения заказа и передача заказа другому мастеру.

Мастеру передает заказ оператор по телефону или в мессенджере, однако, мастер также имеет доступ к приложению, где он может просматривать заказы и клиентов. Мастер отчитывается оператору о проделанной работе, после чего оператор закрывает дату заказа и добавляет к заказу выполненные оператором услуги.

# **Анализ инструментальных средств разработки**

В данной работе для хранения базы данных была использована база данных MSSQL (Microsoft SQL Server). Эта система управления базами данных (СУБД) предлагает высокую производительность и надежность. Она обладает расширенными возможностями по управлению данными, включая поддержку сложных запросов и масштабируемость. Кроме того, MSSQL имеет интеграцию с другими продуктами Microsoft и обширную документацию, что упрощает разработку и поддержку приложений.

Для данной курсовой работы использовался язык программирования C#. Выбор этого языка обусловлен его широким спектром возможностей и простотой в изучении. C# предоставляет мощные средства для разработки различных типов приложений, включая веб-приложения, настольные приложения и мобильные приложения. Кроме того, C# является основным языком для разработки приложений на платформе .NET, что обеспечивает высокую совместимость и интеграцию с другими технологиями Microsoft.

В качестве среды программирования была выбрана Visual Studio 2022. Это интегрированная среда разработки, которая предоставляет широкий набор инструментов для разработки приложений на языке C#. Visual Studio обладает удобным интерфейсом, обширными возможностями отладки и интегрированным поддержкой различных технологий, что делает ее популярным выбором среди разработчиков.

# **Разработка логической модели базы данных**

Логическая модель данных охватывает все сущности, их атрибуты и связи между ними, отражая бизнес-информацию и правила. По завершении проектирования логической модели она применяется для создания физической модели базы данных.

Исходя из анализа предметной области, можно выделить следующие сущности: Сотрудник, Клиент, Заказ, Услуга.

Логическая модель, созданная для целевой базы данных, соответствует третьей нормальной форме (3НФ) и отвечает следующим требованиям: каждая запись уникальна, имена столбцов (атрибутов) уникальны, отсутствуют составные столбцы, значения атрибутов атомарны, не ключевые поля функционально зависят только от первичного ключа. Полученная логическая модель представлена на Рисунке 1.

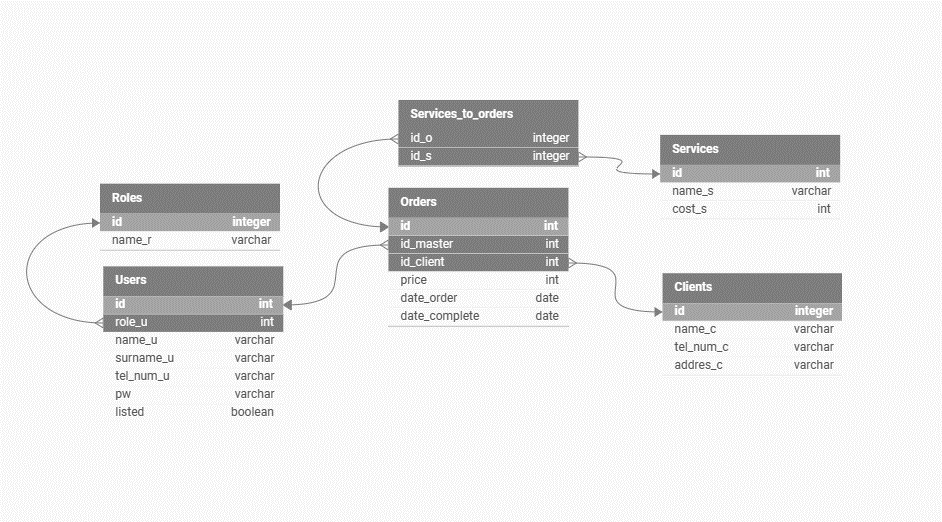


Рисунок 1 – Логическая модель БД «sc\_x99»

# **Создание базы данных и разработка клиентского приложения**

## **Создание базы данных**

После создания логической модели БД на сайте dbdesigner.net на локальном сервере MSSQL была создана база данных, состоящая из 6 таблиц: Clients, Orders, Roles, Services, Services\_to\_Orders, Users, эти таблицы можно увидеть на Рисунке 2.

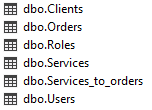


Рисунок 2 – Таблицы БД «sc\_x99»

Таблица department сдержит информацию об отделе музея, таблица event –информацию о мероприятии, таблица item – информацию об экспонате, таблица exposition объединяет таблицы event и item, таблицы payment, status, author, post, type\_of\_event являются справочниками и хранят информацию о типе оплаты билета, статусе мероприятия, авторе экспоната, должности сотрудника, типе мероприятия соответственно, таблицы person и staff хранят данные о людях: сотрудниках и зарегистрированных пользователях, таблица staff\_event объединяет таблицы staff и event, таблица ticket хранит информацию о билете, таблица reviews - информацию об оценке, таблица result - информацию об результатах мероприятия

Также в базе были групповые роли users и admins для разграничения ролей.

Структура созданных используемых таблиц представлена в Приложении А (Таблицы А.1 -А.15).

Получившаяся база данных полностью удовлетворяет требования, предъявляемые к БД, такие как уменьшение излишней избыточности данных, целостность, простота обновления данных и др.

## **Взаимосвязь базы данных и приложения**

Взаимосвязь базы и приложения осуществляется за счёт базовых компонентов библиотеки Npgsql: NpgsqlConnection, NpgsqlCommand, NpgsqlDataSet, NpgsqlDataAdapter, NpgsqlDataReader. Общение с базой обеспечивается за счёт использования встроенных в компоненты функций и свойств, а также динамического изменения данных свойств.

## **Программирование клиентского приложения**

После создания физической модели БД необходимо разработать клиентское приложение. Процесс разработки состоит из нескольких этапов:

1. Создание графического пользовательского интерфейса;
2. Программирование функций для работы с БД (сортировка, поиск вставка, удаление и изменение данных) и элементов управления.

Под созданием графического пользовательского интерфейса понимают создание форм и размещение на них компонентов, через которые будет производиться управление приложением.

Внешний вид главной формы представлен на Рисунке 3.