Министерство образования и науки Российской Федерации

**Муромский институт (филиал)**

федерального государственного бюджетного образовательного учреждения

высшего профессионального образования

**«Владимирский государственный университет**

**имени Александра Григорьевича и Николая Григорьевича Столетовых»**

(МИ (филиал) ВлГУ)

Факультет ИТР

Кафедра ПИн

*КУРСОВАЯ РАБОТА*

По дисциплине СУБД

Тема Информационная система охранного агентства

Руководитель

(оценка)  (уч. степень, звание)

Колпаков А. А.

(подпись) (дата)

Члены комиссии Студент ПИн-122

(группа)

Коротков А. М.

(подпись) (дата)

(подпись) (дата) (подпись) (дата)

Муром 2024 г.

В работе представлена информационная система охранного агентства. Приложение разработано на платформе Windows. Цель работы - рассмотреть принципы разработки информационных систем и разработать программу на основе этих принципов.

Для разработки были выбраны язык программирования – C# и среда разработки – Microsoft Visual Studio Community 2022. Для создания базы данных использовалась СУБД PostgreSQL.

В разработанном приложении доступны возможность добавления, удаления, изменения и поиска записей в базе данных. Проведены тестирование и отладка программы.

The paper presents the information system of a security agency. The application is developed on the Windows platform. The purpose of the work is to consider the principles of information systems development and develop a program based on these principles.

The programming language – C# and the development environment – Microsoft Visual Studio Community 2022 were chosen for development. The PostgreSQL DBMS was used to create the database.

The developed application provides the ability to add, delete, modify and search records in the database. The program has been tested and debugged.

**Содержание**

[Введение 6](#_Toc440226278)

[1. Анализ технического задания 8](#_Toc440226279)

[2. Разработка моделей данных. 10](#_Toc440226280)

[2.1 IDEF0 диаграммы 10](#_Toc440226282)

[2.2 DFD диаграммы 11](#_Toc440226283)

[2.3 Физическая и логическая модель 11](#_Toc440226284)

[3. Разработка и реализация ИС. 12](#_Toc440226285)

[3.1 Создание базы данных 12](#_Toc440226286)

[3.2 Программирование ИС 17](#_Toc440226287)

[4. Тестирование ИС. 19](#_Toc440226288)

[Заключение 24](#_Toc440226289)

[Список используемой литературы 25](#_Toc440226290)

[Приложение А 26](#_Toc440226291)

[Приложение Б 28](#_Toc440226291)

**Введение**

АИС (автоматизированная информационная система) — это совокупность устройств и программных средств, которые предназначены для автоматизации деятельности предприятия или организации. Они связаны с хранением, передачей, поиском и обработкой всех объёмов информации.

В автоматизированных информационных системах за хранение любой информации отвечают:

1. На физическом уровне:

а. внешние накопители;

б. встроенные устройства памяти (RAM);

в. массивы дисков;

2. На программном уровне:

а. СУБД;

б. файловая система ОС;

в. системы хранения мультимедиа, документов и т. д.

На сегодняшний день достаточно широко применяются разнообразные программные средства при работе с компьютером. В их числе находятся и автоматизированные информационные системы. Информационная система или ИС – это система обработки, хранения и передачи какой-либо информации, которая представлена в определенной форме.

Основная цель АИС — хранение, обеспечение эффективного поиска и передачи информации по соответствующим запросам для наиболее полного удовлетворения информационных запросов большого числа пользователей.

Автоматизация информационных процессов включает в себя эффективное решение следующих задач:

1) Сведение к минимуму рутинных задач в рабочем процессе.

2) Уменьшение трудозатрат, которые определяются при выполнении традиционных информационных операций и процессов.

3) Скорость выполнения процессов преобразования и обработки информации значительно возрастают.

4) Увеличение показателей качества и оперативности обслуживания клиентов предприятия или организации.

5) Точность информации увеличивается (как учетной, так и отчетной), а проведение статистического анализа получает более широкие возможности.

6) Обеспечение больших организационных возможностей и эффективности применения информационных ресурсов, благодаря использованию современных информационных технологических решений.

7) Полная замена или качественная модернизация отдельных элементов в традиционных технологических решениях.

8) Расширение возможностей для эффективного обмена информацией, участия в проектах, которые будут обеспечивать развитие и интеграцию.

В качестве классического примера современной информационной системы, стоит упомянуть банковские системы, АС управления предприятиями, системы резервирования железнодорожных или авиационных билетов и т. д.

Целью курсовой работы является создание информационной системы для добавления, удаления, изменения и поиска записей в базе данных.

Задачи, работающие для этой цели:

1. Проектирование логической и физической моделей базы данных.

2. Реализация физической модели в одной из современных СУБД.

3. Подключение БД к программе с помощью средств среды разработки.

4. Написание обработчиков событий и функций для добавления, удаления, изменения и поиска записей в БД.

1. **Анализ технического задания**

В данной курсовой работе поставлена задача создания информационной системы охранного агентства. Для разработки такой системы необходимы СУБД для создания базы данных и среда разработки для создания интерфейса и функций по работе с БД.

Согласно заданию, в программе нужно учесть следующие особенности:

- Агентство занимается охраной юридических и физических лиц.

- Необходимо вести учет сотрудников агентства, охраняемых объектов, заключенных договоров, выручки за заданный промежуток времени (произвольные даты, месяц, квартал, год).

- Хранимая информация о сотрудниках: ФИО, должность, дата поступления на работу, оклад, образование. Заказчики –адрес, номер счета (при наличии), ФИО представителя (при наличии), телефон, вид услуг. Вид предоставляемой услуги – наименование услуги (охрана объекта, установка сигнализации и др.), стоимость вида услуги, ФИО заказчика, дата заказа, дата выполнения заказа.

- Должна быть возможность получения сводных отчетов на основе информации из БД: объем выручки, количество договоров, статистика по сотрудникам, статистика вызовов охраны.

- В БД предусмотреть хранение изображений (минимум в одном поле) в соответствии с тематикой курсовой работы.

- В БД должны быть реализованы хранимые процедуры и/или триггеры.

- В БД должно быть занесено суммарно не менее 50 записей.

Исходя из данных требований к приложению, база данных должна содержать несколько таблиц и данные таблицы должны быть связаны для целостности системы. Для разработки базы данных в рамках поставленной задачи можно использовать PostgreSQL, которая как раз относится к СУБД реляционного типа.

Вот некоторые преимущества PostgreSQL:

1. Бесплатность. PostgreSQL распространяется на условиях лицензии, которая является бесплатной и открытой.
2. Простота использования. PostgreSQL имеет простой и интуитивно понятный интерфейс, который позволяет пользователям управлять базой данных без особых сложностей.
3. Высокая надёжность и безопасность. PostgreSQL является одной из самых надёжных СУБД.
4. Высокая производительность. PostgreSQL может обрабатывать тысячи запросов в секунду и поддерживает широкий спектр функциональности.
5. Широкие возможности. PostgreSQL имеет множество дополнительных модулей и расширений, которые позволяют пользователям увеличить её функциональность.
6. Открытый исходный код и сообщество разработчиков. PostgreSQL — это СУБД с открытым исходным кодом, что позволяет любому желающему просмотреть исходный код PostgreSQL и внести свои изменения.

Для разработки программы, работающей с базой данных, нужно использовать объектно – ориентированный подход. Разработка приложения будет вестись на языке C#.

Для того чтобы создать программу, необходимо учесть то, что она создается, прежде всего, для пользователя, и поэтому немаловажным требованием к программе должен стать удобный и интуитивно понятный интерфейс. Необходимо предусмотреть все возможности управления приложением, чтобы упростить работу пользователя и максимально обеспечить эффективность работы.

Программа должна правильно работать с данными, т.е всегда должен выводится нужный результат, требуемый пользователю. Приложение должно мгновенно реагировать на действия пользователя и в зависимости от запроса с его стороны формировать выходной результат.

1. **Разработка моделей данных**

Данный этап является самым важным при создании АИС. Здесь выделяются сущности, атрибуты сущностей и связи между сущностями. На основе полученной диаграммы “Сущность – связь” строятся функциональные модели системы и диаграмма потоков данных.

2.1 IDEF0 диаграммы

Диаграмма IDEF0 — это графическое описание модели предметной области или её части.

Главными компонентами IDEF0-диаграммы являются блоки. Они отображают некоторые работы, функции, процессы, задачи, которые происходят или выполняются в течение определённого времени и имеют некоторые результаты.

Взаимодействие функций с внешним миром и между собой описывается с помощью дуг (связей). В IDEF0 различают пять типов дуг:

1. Вход (Input).
2. Управление (Сontrol).
3. Выход (Output).
4. Механизм (Mechanism).
5. Вызов (Call).

Модель в нотации IDEF0 представляет собой совокупность иерархически упорядоченных и взаимосвязанных диаграмм. Каждая диаграмма описывает отдельные компоненты системы и располагается на отдельном листе.

IDEF0 диаграмма представлена в приложении А рис. 1.

* 1. DFD диаграммы

DFD — общепринятое сокращение от англ. data flow diagrams — диаграммы потоков данных.

Так называется методология графического структурного анализа, описывающая внешние по отношению к системе источники и адресаты данных, логические функции, потоки данных и хранилища данных, к которым осуществляется доступ.

Диаграмма потоков данных — один из основных инструментов структурного анализа и проектирования информационных систем.

DFD диаграмма представлена в приложении А рис. 2.

* 1. Физическая и логическая модель

ER-диаграмма (также известная как диаграмма «Сущность-связь» или ERD) — это разновидность блок-схемы, которая показывает, как разные «сущности» (люди, объекты, концепции и так далее) связаны между собой внутри системы.

ER-диаграммы применяются для проектирования и отладки реляционных баз данных в сфере образования, исследования и разработки программного обеспечения и информационных систем для бизнеса.

Основная концепция ER-диаграмм заключается в выделении ключевых сущностей, их атрибутов и связей.

В ходе анализа технического задания было установлено, что база данных будет реляционной. Под этим понимается, что сущности – таблицы, атрибуты – поля. Для каждого поля указывается тип данных. Идентифицирующие атрибуты сущности становятся первичными ключами.

ER диаграмма представлена в проложении А рис. 3.

**3. Разработка и реализация ИС**.

3.1 Создание базы данных.

База данных будет создана в программе СУБД PostgreSQL. Для её создания нужно использовать физическую модель.

Перед тем как связать таблицы их нужно создать и для каждого поля указать тип. Названия таблиц базы данных:

* clients хранит информацию о клиентах.
* employees хранит информацию о сотрудниках.
* s\_services хранит информацию о заказах охраны.
* m\_services хранит информацию о заказах на установку охранной системы.
* schedules хранит информацию о расписаниях.
* s\_service\_employee нужна для создании связи «многие ко многим» для таблиц s\_services и employees.
* m\_service\_employee нужна для создании связи «многие ко многим» для таблиц m\_services и employees.

Типы полей для таблиц представлены ниже (Рис. 4-10).

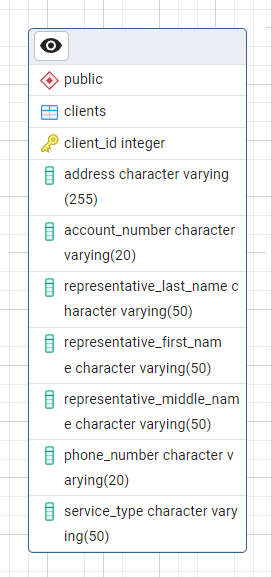


Рисунок 4. Таблица «Клиенты».

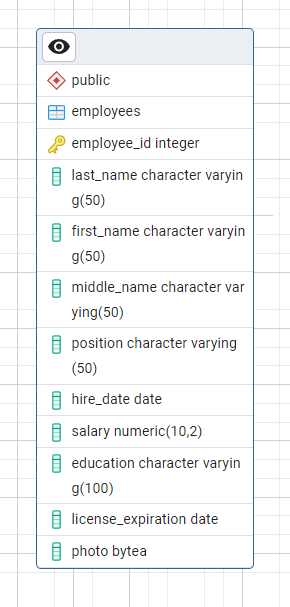


Рисунок 5. Таблица «Сотрудники».

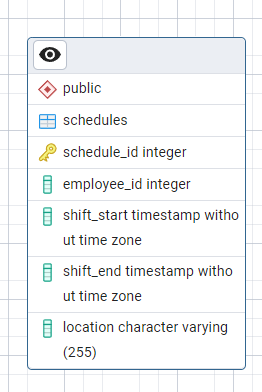


Рисунок 6. Таблица «Расписания».

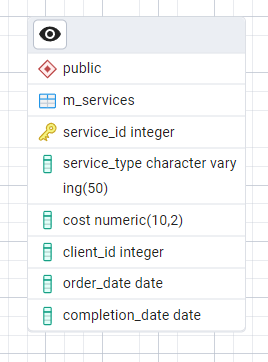


Рисунок 7. Таблица «Услуги».

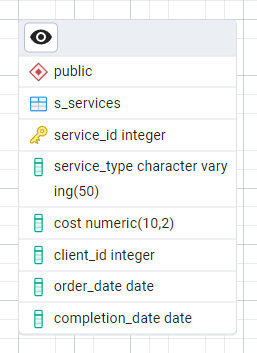


Рисунок 8. Таблица «Услуги».

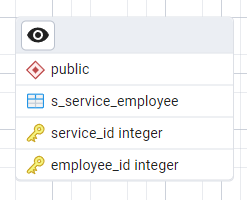


Рисунок 9. Таблица для связи «многие ко многим».

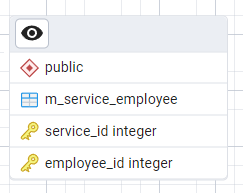


Рисунок 10. Таблица для связи «многие ко многим».

После создания таблиц их необходимо связать и получить схему (Рис. 11).

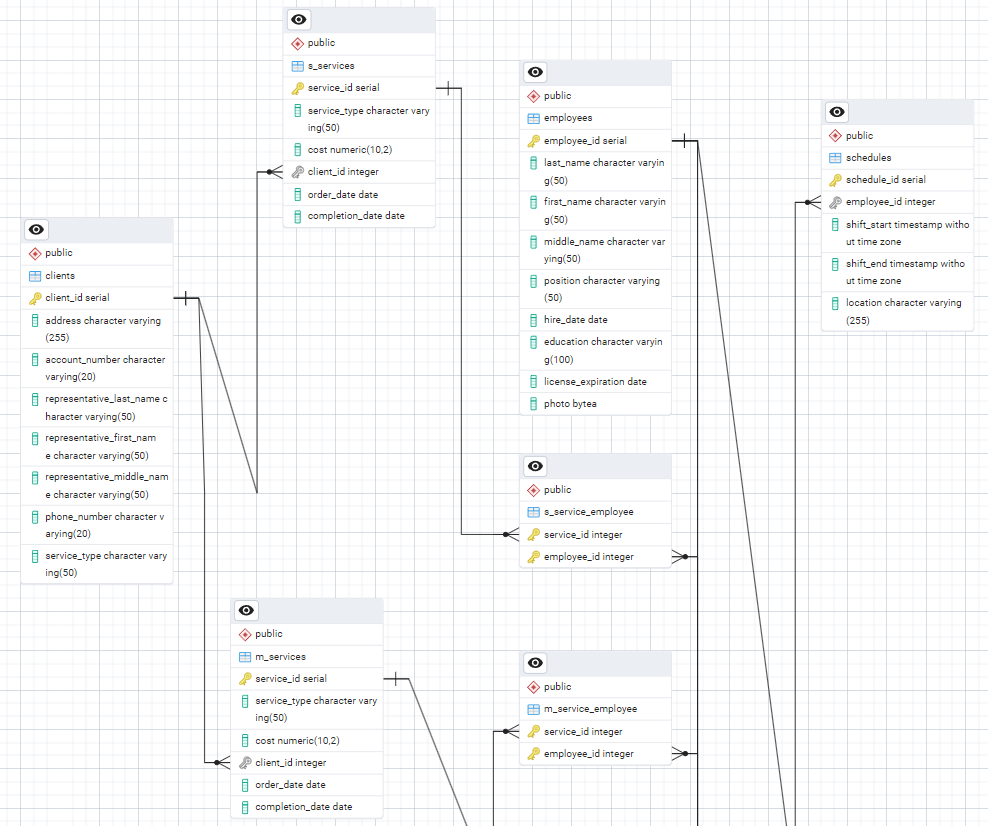


Рисунок 11. Схема таблиц.

3.2 Программирование ИС

Для создания системы будет использоваться среда разработки Microsoft Visual Studio Community 2022. В данной программе уже по умолчанию добавлены компоненты и библиотеки классов для работы с базой данных.

После создания проекта нужно подключить базу данных (Рис. 12). Для этого есть пакет NuGet Npgsql.

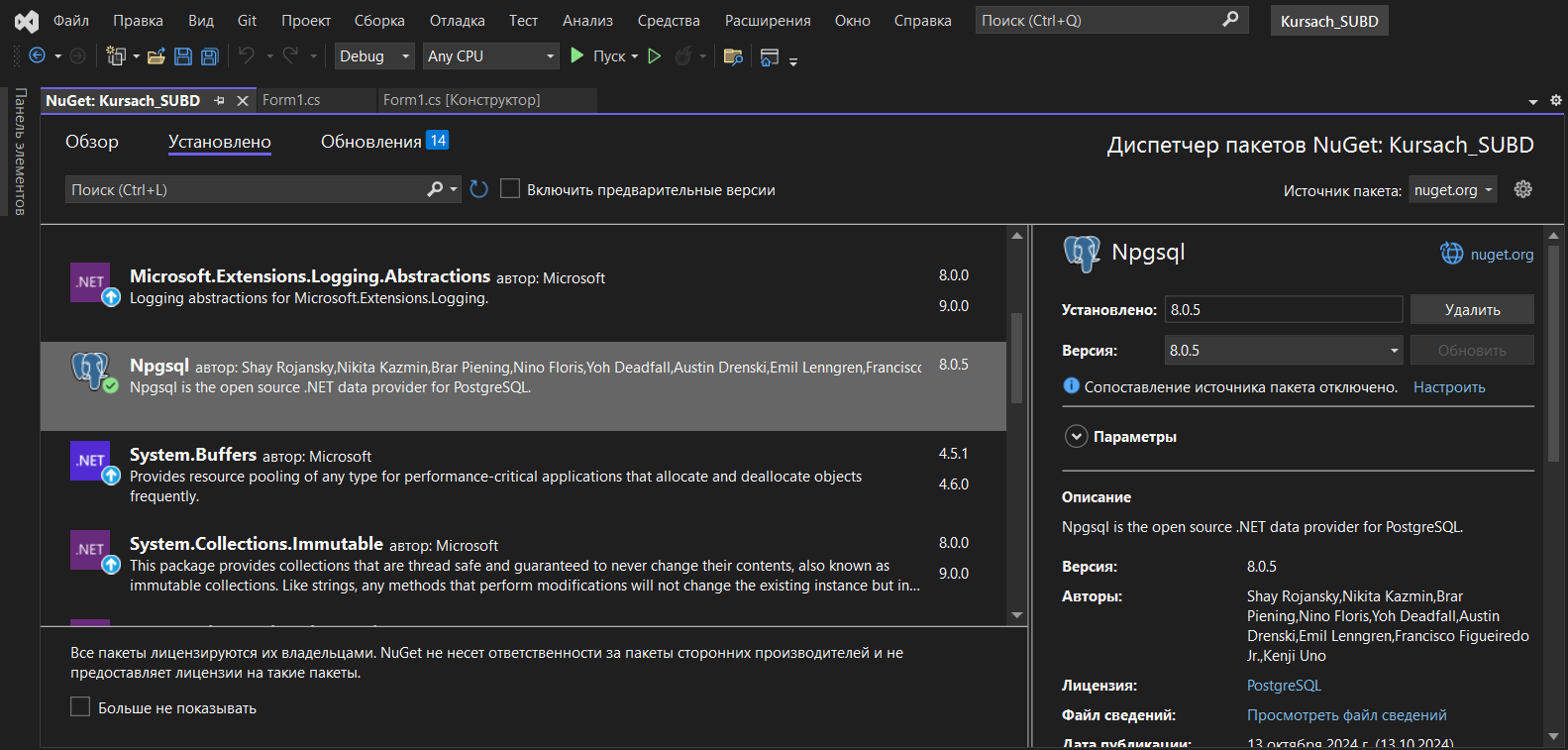


Рисунок 12. Пакет Npgsql.

Чтобы вывести информацию из базы данных необходимо использовать компонент DataGridView и написать метод.

Листинг кода метода вывода информации из таблицы «Сотрудники» в компонент DataGridView:

|  |
| --- |
| private void LoadEmployeeData()  {  using (var conn = new NpgsqlConnection("server=localhost; database=Kursach\_SUBD; user Id=postgres; password=1234"))  {  try  {  DataSet dataset = new DataSet();  conn.Open();  NpgsqlCommand command = new NpgsqlCommand("SELECT \* FROM employees", conn);  NpgsqlDataAdapter da = new NpgsqlDataAdapter(command);  da.Fill(dataset, "employee\_table");  dataGridViewEmployees.DataSource = dataset;  dataGridViewEmployees.DataMember = "employee\_table";  }  catch (Exception ex)  {  MessageBox.Show("Ошибка загрузки данных сотрудников: " + ex.Message);  }  }  } |

Таким же способом нужно реализовать вывод информации из оставшихся таблиц.

Для реализации запросов на добавление, удаление и обновление записей в таблицах необходимо добавить на форму компоненты Button и в обработчике событий прописать создание еще одной формы для удобства.

Также необходимо обработать исключения для каждого запроса к базе.

Весь исходный код программы представлен в приложении Б.

**4. Тестирование ИС**

При завершении разработки программы необходимо провести проверку на её работоспособность при различных действиях пользователей. Программа должна корректно отображать и выполнять действия пользователя в любом случае. В исключительных моментах должны выводиться окна с сообщениями, поясняющие ситуацию.

Проверим, как программа будет реагировать на некоторые действия пользователя. Давайте попробуем удалить сотрудника. При нажатии на кнопку “Удалить сотрудника” перед пользователем открывается сообщение (Рис.13) о том, что действительно ли он хочет удалить данную запись. При подтверждении удаления, запись с выбранным номером сотрудника удаляется из таблиц.

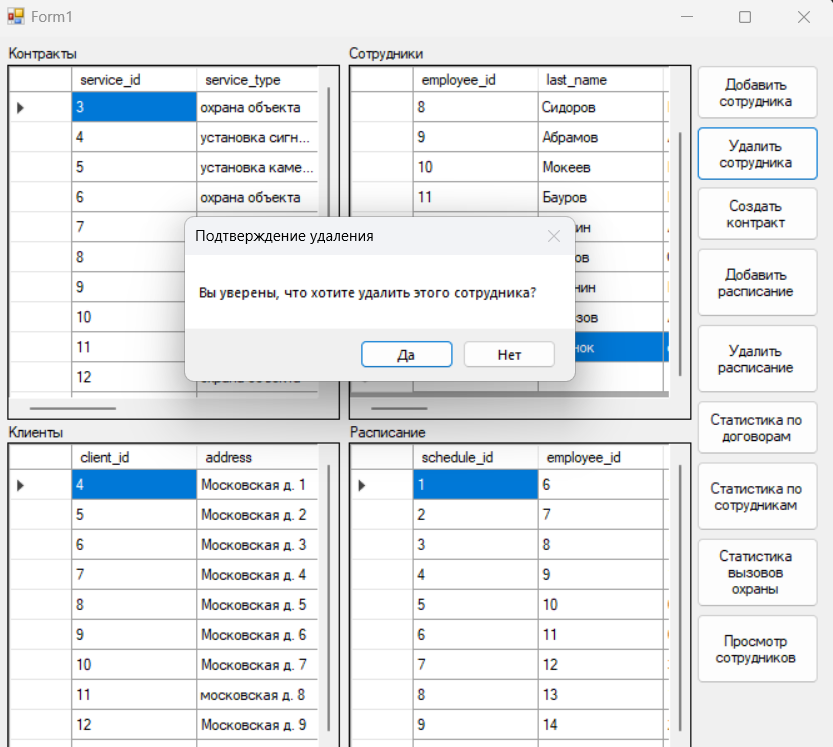


Рисунок 13. Удаление сотрудника.

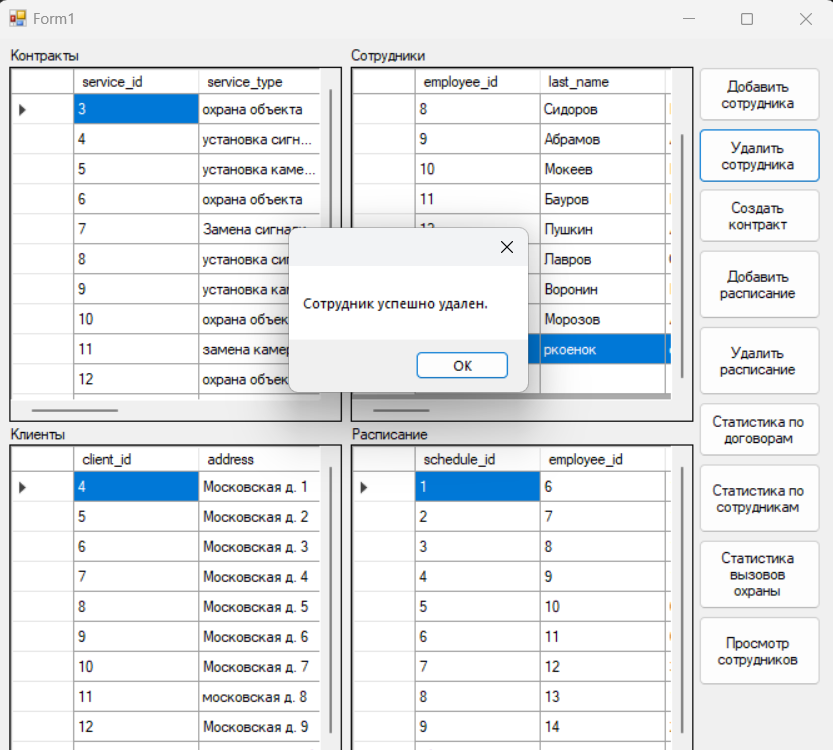


Рисунок 14. Удаление сотрудника.

Когда пользователь создает контракт, вместе с добавлением записи в таблицу «Контракты» автоматически должна создаваться запись в таблицу «Клиенты» и «Услуги». Также должна быть возможность привязать сотрудников к заказу. Например (Рис.15).

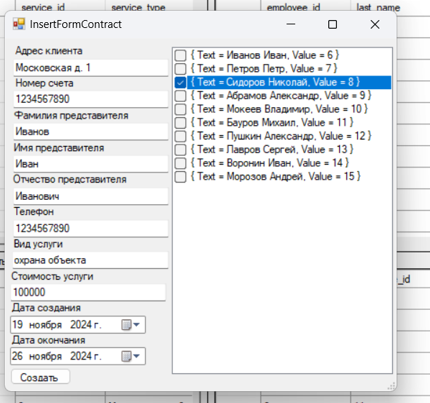


Рисунок 15. Создание контракта.

После того, как пользователь создал заказ и привязал к нему сотрудников, он может посмотреть различную статистику: статистика по договорам, статистика по сотрудникам, статистика вызовов охраны, а также просмотр сотрудников, связанных с выбранным заказом (рис. 16-19).

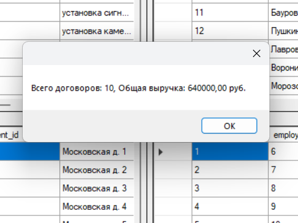


Рисунок 16. Статистика по договорам.

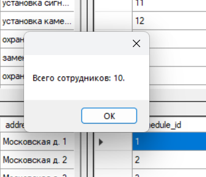


Рисунок 17. Статистика по сотрудникам.

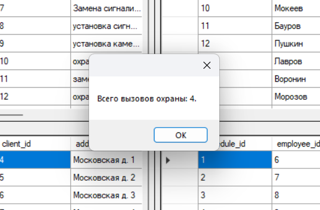


Рисунок 18. Статистика вызовов охраны.

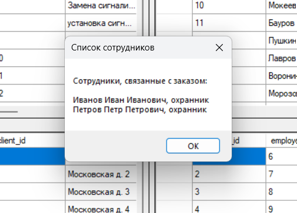


Рисунок 19. Просмотр сотрудников.

Версия приложения для сотрудников охраны имеет ограниченный функционал: отображаются только таблицы с сотрудниками, расписаниями и заказами и можно посмотреть расписание для отдельного сотрудника (Рис. 20).

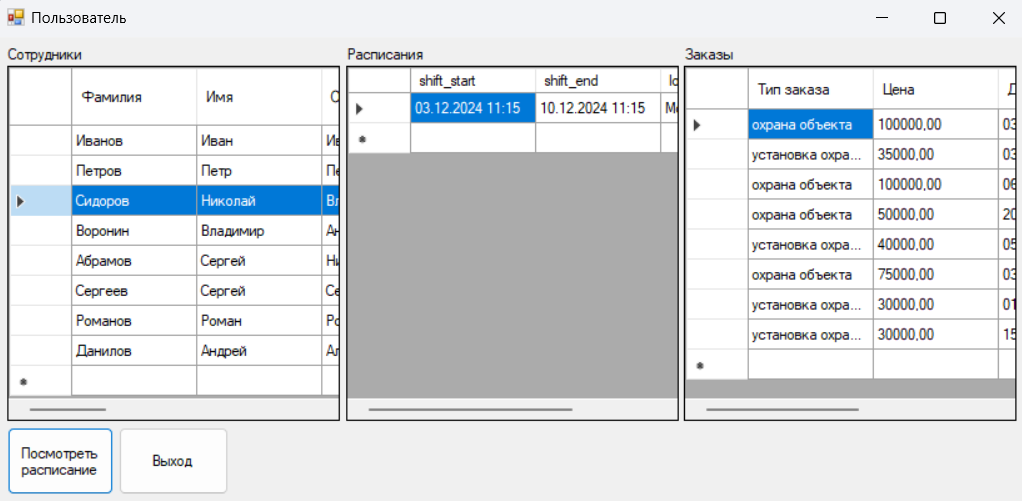


Рисунок 20. Интерфейс приложения для сотрудника охраны.

В результате тестирования программы были найдены и исправлены недочёты по работе с программой.

**Заключение**

В результате разработки была создана программа, которая отвечает тем требованиям, которые были заложены перед её созданием. В ходе разработки были использованы внутренние ресурсы и библиотеки среды разработки Microsoft Visual Studio Community 2022. Компоненты упростили соединение с базой данных и вывод данных из таблиц.

На стадии разработки моделей данных были созданы логическая и физическая модели, а так же IDEF0 диаграммы и DFD диаграмма, которые значительно упрощают создание программы.

Самый длительный промежуток времени заняли разработка и программирование клиентского приложения.

В процессе программирования использовалось много сторонней литературы (в основном сайты, электронные книги). Литература помогла решить некоторые проблемы, которые возникали в ходе разработки программы.

**Список используемой литературы**

- Тарасов, С. В. СУБД для программиста. Базы данных изнутри / С. В. Тарасов. - Москва : СОЛОН-Пресс, 2018. - 320 c. - ISBN 978-2-7466-7383-0. - Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. - URL: https://www.iprbookshop.ru/90409.html (дата обращения: 23.08.2022). - Режим доступа: для авторизир. пользователей

- Прокушев, Я. Е. Базы данных : учебник с практикумом / Я. Е. Прокушев. - 2-е изд. - Санкт-Петербург : Интермедия, 2022. - 264 c. - ISBN 978-5-4383-0250-6. - Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. - URL: https://www.iprbookshop.ru/120171.html (дата обращения: 23.08.2022). – Режим доступа: для авторизир. пользователей

- Маркин, А. В. СУБД «Ред База Данных». Основы SQL : учебное пособие / А. В. Маркин. - Москва : Ай Пи Ар Медиа, 2022. - 460 c. - ISBN 978-5-4497-1605-7. - Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. - URL: https://www.iprbookshop.ru/119617.html (дата обращения: 23.08.2022). - Режим доступа: для авторизир. Пользователей.

- О частной детективной и охранной деятельности в Российской Федерации [Текст]: [Закон РФ № 2487-I от 11 марта 1992 г.](https://www.taggerd.su/info/zakon-rf-N-2487-ot-11.03.1992-g./) // Собрание законодательства. – 1992.

- ГОСТ Р 59044-2020. Реферат и аннотация. – Москва: Издательство стандартов, 2020.

**Приложение А. Модели данных**

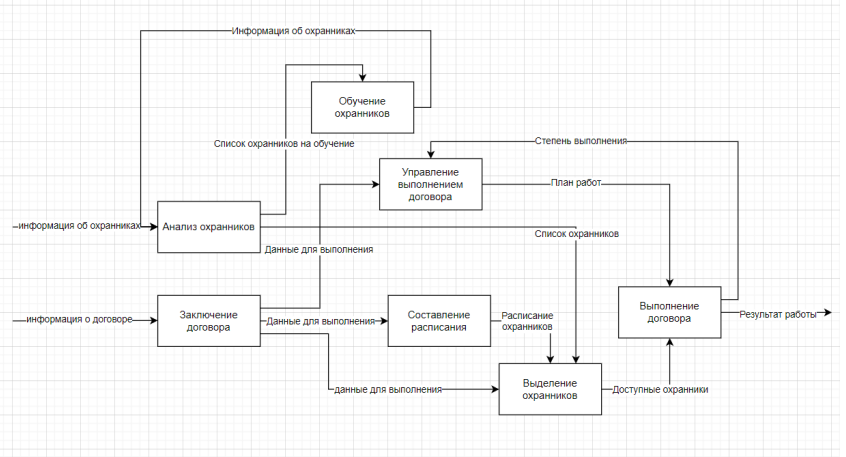


Рисунок 1. IDEF0-диаграмма.

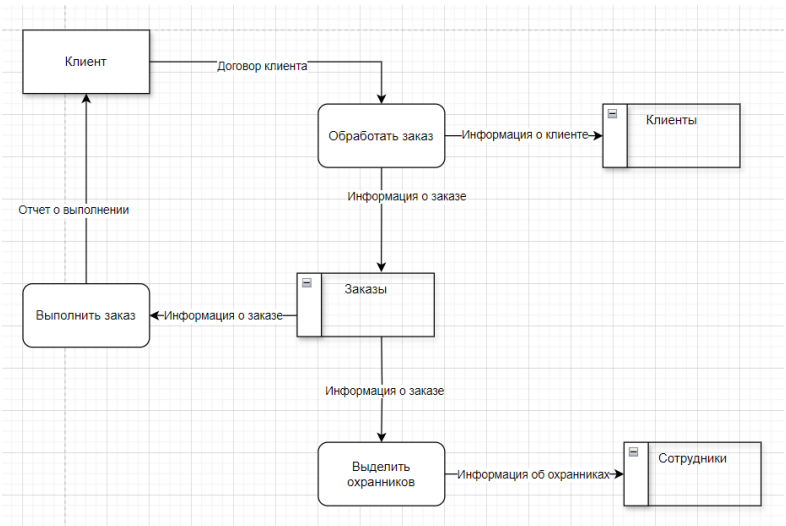


Рисунок 2. DFD-диаграмма.

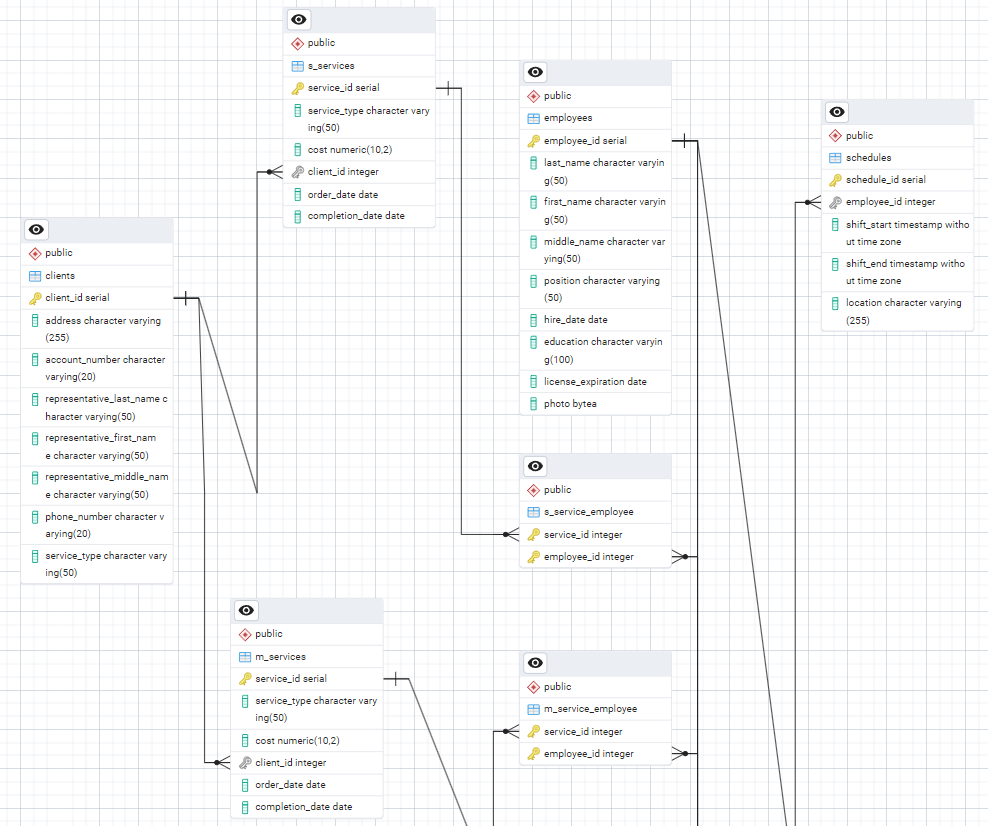


Рисунок 3. ER-модель.

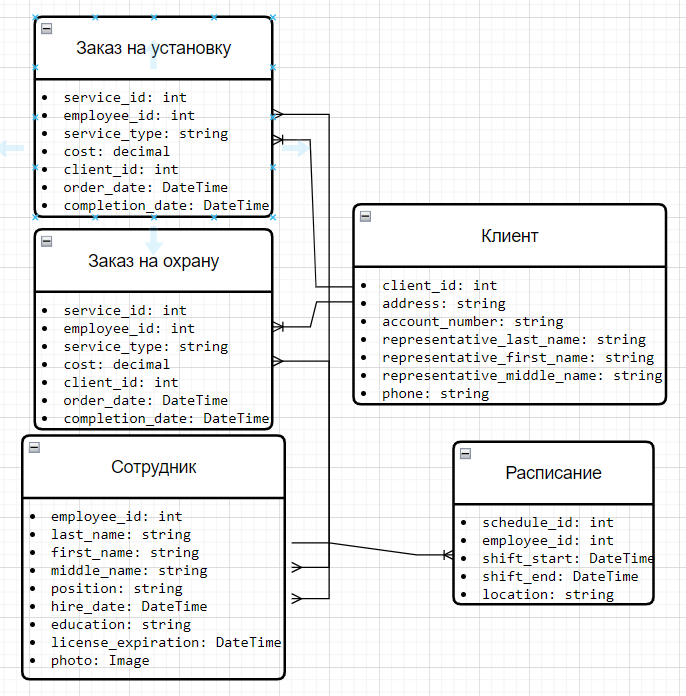


Рисунок 4. Логическая модель.

**Приложение Б. Текст программы**

Ссылка на репозиторий GitHub: <https://github.com/T2F2R/Kursach_SUBD>