|  |  |
| --- | --- |
| **Gerb-BMSTU_01** | **Министерство науки и высшего образования Российской Федерации**  Калужский филиал  федерального государственного бюджетного  образовательного учреждения высшего образования  ***«Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)»***  ***(КФ МГТУ им. Н.Э. Баумана)*** |

**ФАКУЛЬТЕТ** \_ ***ИУК «Информатика и управление»\_\_\_\_\_\_\_\_\_***

**КАФЕДРА** \_\_ ***ИУК5 «Системы обработки информации»***

**РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

**к курсовой работе на тему:**

**«**Приложение для организации и проведения похорон**»**

по дисциплине ***Архитектура автоматизированных систем обработки информации и управления***

Студент гр. ИУК5-51Б \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (Новгородский С.Д)

(подпись) (Ф.И.О.)

Руководитель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (Смирнов М.Е.)

(подпись) (Ф.И.О.)

Оценка руководителя \_\_\_\_\_ баллов \_\_\_\_\_\_\_\_\_\_\_

30-50 (дата)

Оценка защиты \_\_\_\_\_ баллов \_\_\_\_\_\_\_\_\_\_\_

30-50 (дата)

Оценка проекта \_\_\_\_\_ баллов \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(оценка по пятибалльной шкале)

Комиссия: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_)

(подпись) (Ф.И.О.)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_)

(подпись) (Ф.И.О.)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_)

(подпись) (Ф.И.О.)

Калуга, 2021

Калужский филиал   
федерального государственного бюджетного образовательного учреждения высшего образования

***«Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)»   
(КФ МГТУ им. Н.Э. Баумана)***

УТВЕРЖДАЮ

Заведующий кафедрой **\_\_ИУК5\_\_\_**

\_\_\_\_\_\_\_\_\_\_\_(Е.В. Вершинин)

«\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_20\_\_\_г.

**ЗАДАНИЕ**

**на выполнение курсовой работы**

по дисциплине ***Архитектура автоматизированных систем обработки информации и управления***

Студент Новгородский С.Д. ИУК5-51Б

(фамилия, инициалы, индекс группы)

Руководитель Смирнов М.Е.

(фамилия, инициалы)

График выполнения проекта: 25% к\_4\_нед., 50% к\_7\_нед., 75% к\_10\_нед., 100% к\_14\_нед.

***1. Тема курсового проекта***

***Приложение для организации и проведения похорон***

***2. Техническое задание***

*Выполнить исследование и описание предметной области, произвести анализ объекта автоматизации, разработать методы и способы решения технических задач. Разработать и реализовать структуру базы данных, интерфейс приложения и взаимодействие с базой данных*

***3. Оформление курсового проекта***

3.1. Расчетно-пояснительная записка на\_\_\_\_\_\_\_\_ листах формата А4.

3.2. Перечень графического материала КП (плакаты, схемы, чертежи и т.п.)\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Дата выдачи задания «\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2021г.

Руководитель курсового проекта \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/\_Смирнов М.Е.\_/

(подпись) (Ф.И.О.)

Задание получил \_\_\_\_\_\_\_\_\_\_\_\_\_\_ /\_Новгородский С.Д. \_/ «\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_ 2021\_г.

(подпись) (Ф.И.О.)

**Содержание**

[1.Техническое задание 4](#_Toc89877955)

[2. Научно – исследовательская часть 8](#_Toc89877956)

[2.1 Постановка задачи проектирования. 8](#_Toc89877957)

[2.2 Описание предметной области. 8](#_Toc89877958)

[2.3 Анализ аналогов. 9](#_Toc89877959)

[2.4 Перечень задач, подлежащих решению в процессе разработки. 11](#_Toc89877960)

[2.5 Обоснование выбора инструментов и платформы для разработки. 11](#_Toc89877961)

[3. Проектно-конструкторская часть 13](#_Toc89877962)

[3.1 Разработка структуры приложения 13](#_Toc89877963)

[3.2 Разработка алгоритмов обработки информации. 13](#_Toc89877964)

[3.2 Разработка архитектуры приложения 19](#_Toc89877965)

[3.4 Разработка графического интерфейса взаимодействия пользователя с системой. 20](#_Toc89877966)

[4. Проектно-технологическая часть 24](#_Toc89877967)

[4.1 Проектирование начального и тестового наполнения базы данных. Порядок развертывания системы 24](#_Toc89877968)

[4.2 Руководство пользователя 29](#_Toc89877969)

[Заключение 37](#_Toc89877970)

[Список литературы 38](#_Toc89877971)

# 1.Техническое задание

**1.1 Общие сведения.**

**1.1.1 Полное наименование системы и ее условное обозначение.**

Разработка прикладного программного обеспечения для записи представителя погибшего на ритуальные услуги.

**1.1.2 Шифр темы или шифр (номер) договора.**

Шифры темы или договора отсутствуют.

**1.1.3 Наименование предприятий (объединений) разработчика и заказчика (пользователя) системы и их реквизиты.**

Высшее учебное заведение Калужский филиал МГТУ имени Н. Э. Баумана

**1.1.4 Перечень документов, на основании которых создается система, кем и когда утверждены эти документы.**

На основании учебного плана кафедры ИУК5.

**1.1.5 Плановые сроки начала и окончания работы по созданию системы.**

Начало работы: 01.09.2021

Окончание работы: 10.12.2021

**1.1.6. Сведения об источниках и порядке финансирования работ.**

Финансирование работы отсутствует.

**1.2. Назначение и цели создания (развития) системы.**

**1.2.1 Назначение системы.**

Прикладное программное обеспечение необходимо для создания записей представителей усопшего на необходимые ритуальные услуги в заданный день

**1.2.2 Цели создания системы.**

Целью курсовой работы является формирование практических навыков по разработке и реализации программного приложения с использованием баз данных.

**1.3. Характеристика объектов автоматизации.**

**1.3.1 Краткие сведения об объекте автоматизации или ссылки на документы, содержащие такую информацию.**

Объектом автоматизации является набор функций, обеспечивающие быстрый доступ к специалисту который смог бы предоставить услуги.

**1.3.2 Сведения об условиях эксплуатации объекта автоматизации и характеристиках окружающей среды.**

Сведения об условиях эксплуатации объект автоматизации, отсутствуют

На окружающую среду объект автоматизации влияние не оказывает

# 

**1.4. Требования к системе.**

**1.4.1. Требование к структуре и функционированию системы.**

Функциональная структура системы должна включать основные методы для записи представителей усопшего на запланированное на определенную дату захоронение, а так же представления дополнительных услуг

**1.4.2. Требования к защите информации от несанкционированного доступа.**

Требования к защите не требуются так как в программе не будет системы авторизации и аутентификации пользователя, вся информация будет предоставляется без регистрации, из-за не частого обращения одного пользователя.

**1.4.3. Требования к составу и параметрам технических средств.**

Для использования программы необходимы средства ввода и вывода информации (мышь, клавиатура и монитор)

**1.5. Состав и содержание работ по создание системы**

Задание на курсовую работу определяет общие требования на состав и содержание курсовой работы как на научно-техническую продукцию, каковой и является результат выполнения курсовой работы.

**1.5.1 Перечень документов, предъявляемых по окончании соответствующих стадий и этапов работ**

Таблица 1 – Этапы работ со сроком сдачи

|  |  |  |
| --- | --- | --- |
| **Стадии** | **Этапы работ** | **Сроки исполнения** |
| 1. Формирование требований к АС | 1.1. Обследование объекта и обоснование необходимости создания АС.  1.2. Формирование требований пользователя к АС.  1.3. Оформление отчёта о выполненной работе и заявки на разработку АС (тактико-технического задания) | 30.09.2021 |
| 2. Разработка концепции АС. | 2.1. Изучение объекта.  2.2. Проведение необходимых научно-исследовательских работ.  2.3. Разработка вариантов концепции АС, удовлетворяющего требованиям пользователя.  2.4. Оформление отчёта о выполненной работе. | 30.10.2021 |
| 3. Техническое задание. | Разработка и утверждение технического задания на создание АС. | 30.11.2021 |
| 4. Разработка системы. | 4.1. Разработка системы согласно техническому заданию. | 20.12.2021 |
| 5. Защита курсовой  работы. | 5.1 Подготовка презентации и речи для защиты курсовой работы  5.2 Защита курсовой работы | 10.12.2021 |

**1.6. Порядок контроля и приемки системы.**

**1.6.1 Общие требования к приемке работ по стадиям**

1 – 4-я неделя: Выбор и согласование темы с руководителем; проведение исследования по выбранной теме, обзор существующих программных продуктов; разработка технического задания. Выбор и согласование темы с руководителем; проведение исследования по выбранной теме, обзор существующих программных продуктов; разработка технического задания.

5 – 7-я недели: Разработка и реализация алгоритмов функционирования приложения, структуры, систем передачи информации, технологий обработки информации и интерфейса взаимодействия пользователя с системой. Выполнение и оформление проектно-конструкторской части работы.

9 – 10-я недели: Тестирование и отладка программного приложения. Разработка руководства пользователя и программиста. Написание проектно-технологической части расчетно-пояснительной записки.

11 – 14-я недели: Завершающее оформление документации согласно требованиям, ГОСТ и данного методического пособия. Подготовка доклада. Защита курсовой работы.

**1.7. Требования к составу и содержанию работ по подготовке объекта автоматизации к вводу системы в действие**

Коррекции и разрешение к показу программы приёмной комиссии является научный руководитель.

**1.8. Требование к документированию**

Проектная документация должна быть разработана в соответствии с ГОСТ 34.201-89

Отчетные материалы должны включать в себя текстовые материалы (предоставленные в виде бумажной копии и на цифровом носителе в формате MS Word).

# 2. Научно – исследовательская часть

# 2.1 Постановка задачи проектирования.

Задачей проектирования данного курсового проекта является формирование практических навыков по разработке и реализации программного обеспечения для записи и последующего обслуживания клиентов компании с использованием баз данных.

Необходимо разработать удобный пользовательский интерфейс. Интерфейс должен предоставлять возможность администратору добавлять, изменять и удалять клиентов, а так же организовывать похоронные процессии.

# 2.2 Описание предметной области.

Организация похорон – скорбное и очень ответственное дело, когда к каждому человеку, понесшему утрату, необходимо отнестись максимально деликатно и бережно, ритуальные услуги помогу организовать качественные похороны, а так же помогут организовать поминки.

Основными задачами администратора ритуальных услуг являются:

1. Организация беспрепятственной и безотлагательной записи клиентов на оказание им ритуальных услуг в автоматизированном режиме
2. Обеспечение регулирования потока клиентов с целью создание записи по дате.
3. Систематизированное хранение данных о клиентах и о проведенных, а также об актуальных заказов.

Администратор в праве:

* Получать, хранить и обрабатывать информацию добавленную в базу данных для качественного выполнения его функциональных обязанностей

Администратор должен:

* Производить бесперебойную запись клиентов.
* Добавлять новые материалы для гробов по мере их необходимости.
* Добавлять новые виды для памятников.
* Добавлять новые кладбища с указанием города их нахождения.
* Организовывать ритуальные процессии с указанием даты, место проведения ритуальной трапезы, памятника, материала для гроба и кладбища.
* Провести консультацию с клиентом

В свою очередь, клиент обязан: дать право на обработку персональных данных, а так же на хранение их в базе данных компании.

# 2.3 Анализ аналогов.

Аналогами моего приложения являются веб-сайты

1. Ритуальное агентство «Вознесение» (Рисунок 1)

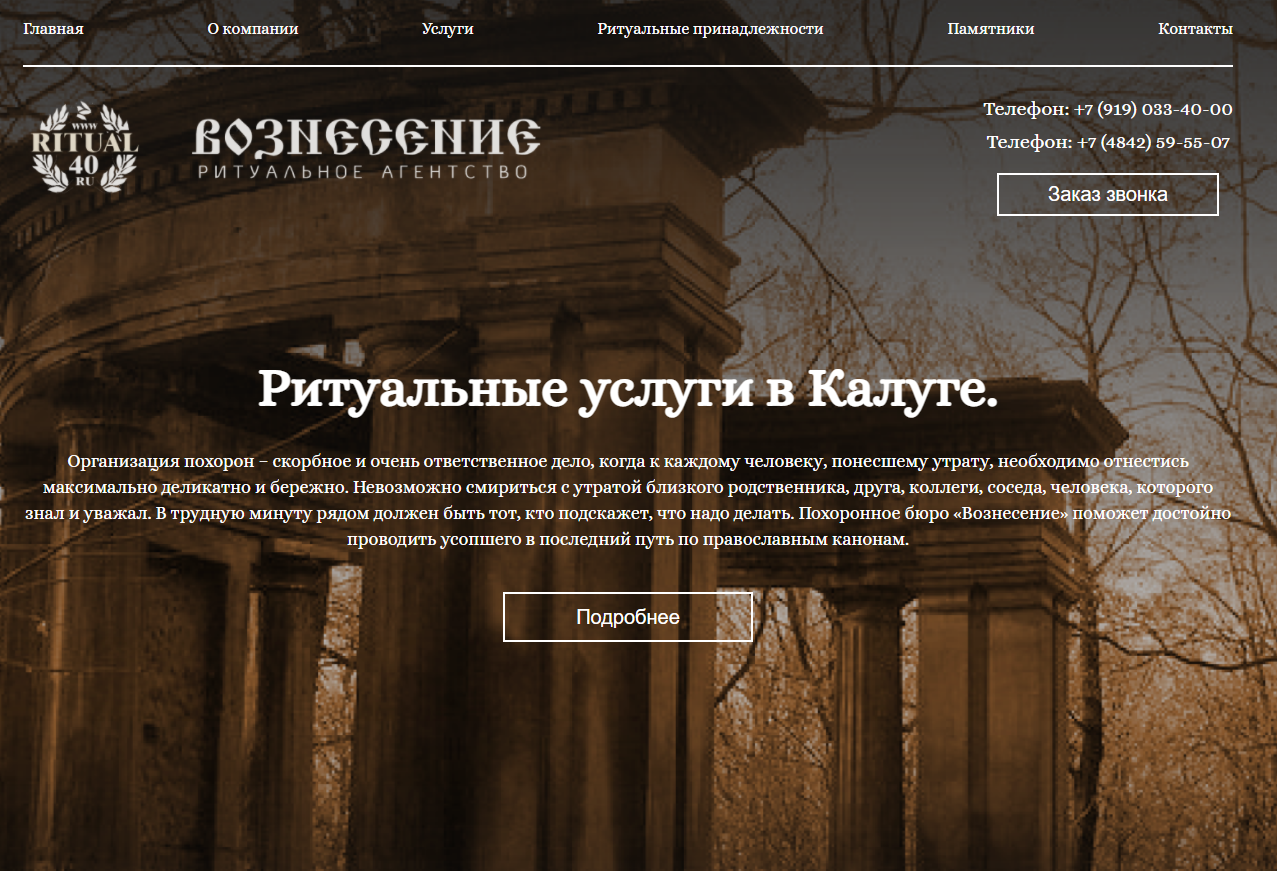


Рисунок 1 – Домашняя страница сайта «ritual40.ru»

Плюсы:

* Удобный и красивый интерфейс.
* Вся информация о компании на домашней странице.

Минусы:

* В услугах не перечисляются все возможности которая компания может предоставить.
* 3 номера телефона, один ин которых не обслуживатеся, на домашней странице, пользователь может с лёгкостью запутаться.

2. Похоронный дом «Стелла» (Рисунок 2)

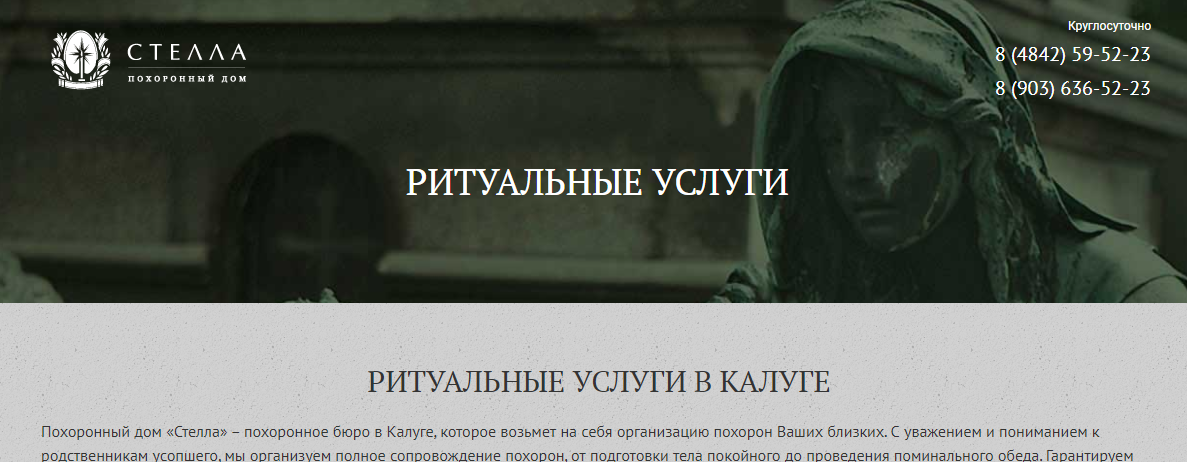


Рисунок 2 – Домашняя страница сайта «stella-ritual.ru»

Плюсы:

* Понятный интерфейс.

Минусы:

* Услуги компании представлены комплектами услуг, за фиксированную плату, что не дает пользователю гибко настроить ритуальную процессию.
* Большое количество номеров телефона, что путает пользователя

# 2.4 Перечень задач, подлежащих решению в процессе разработки.

* Реализация взаимодействия приложения с базой данных;
* Реализация методов записи пациентов на прием в сетку расписания в базе данных; добавления специальностей или специалистов; ввод карточек пациентов;
* Разработка пользовательского интерфейса программы, соответствующего заданным требованиям.

# 2.5 Обоснование выбора инструментов и платформы для разработки.

Средой проектирования, позволяющими реализовать подобное программное обеспечение, была выбрана Microsoft Visual Studio 2022. Среда разработки Visual Studio представляет собой полный набор средств разработки для создания приложений .NET 6, настольных приложений и мобильных приложений. В этих языках используются функциональные возможности платформы .NET всех предыдущих версий, которая позволяет получить доступ к ключевым технологиям, упрощающим разработку веб-приложений ASP и XML (веб-службы). Можно создать обычные приложения Microsoft Windows и приложения с архитектурой "клиент-сервер" с помощью использования конструкторов в Visual Studio.

Преимуществами выбранной среды программирования являются удобство интерфейса и сравнительно невысокие требование к системе.

В качестве БД было выбрано PostgreSQL. Это популярная свободная объектно-реляционная система управления базами данных.

PostgreSQL базируется на языке SQL и поддерживает многочисленные возможности:

* Поддержку БД неограниченного размера;
* Мощные и надёжные механизмы транзакций и репликации;
* Расширяемую систему встроенных языков программирования и поддержку загрузки C-совместимых модулей;
* Наследование;
* Легкая расширяемость.

Платформой для разработки была выбрана Windows 10. Это доминирующая и наиболее распространённая операционная система, которая имеет наибольшую поддержку работы с различным ПО.

Windows предоставляет четыре основные платформы приложений, каждая из которых имеет свои преимущества:

* + Универсальная платформа Windows (UWP). На этой платформе предоставляется система общих типов, интерфейсы API и модель приложений для всех устройств под управлением Windows 10. Приложения UWP могут быть нативными или управляемыми.
  + WPF и Windows Forms. Эти платформы на базе .NET предоставляют систему общих типов, интерфейсы API и модель приложений для управляемых приложений.

В качестве языка программирования для реализации проекта был выбран С#. C# — это объектно- и компонентно-ориентированный \_ язык программирования. C# предоставляет языковые конструкции для непосредственной поддержки такой концепции работы. Программы C# выполняются в .NET, виртуальной системе выполнения, вызывающей общеязыковую среду выполнения (CLR) и набор библиотек классов. Благодаря этому C# подходит для создания и применения программных компонентов.

Данные платформы были выбраны в связи с прошлым опытом их использования.

# 3. Проектно-конструкторская часть

# 3.1 Разработка структуры приложения

Программа создана как Win32-приложение, проект WPF.

Все компоненты разрабатываются на Windows 10 в среде разработки Visual Studio 2022, используя C# в качестве языка программирования.

Системой управления базой данных была выбрана PostgreSQL.

# 3.2 Разработка алгоритмов обработки информации.

Алгоритм подключения базы данных к программе с помощью EntityFramework:

internal class DBcontexts : DbContext

{

public DbSet<Customer> Customer { get; set; }

public DbSet<Cemetery> Cemetery { get; set; }

public DbSet<Grave> Grave { get; set; }

public DbSet<Coffin> Coffin { get; set; }

public DbSet<FuneralProcession> FuneralProcession { get; set; }

public DBcontexts()

{

Database.EnsureCreated();

}

protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)

{

optionsBuilder.UseNpgsql("Host=localhost;Port=5432;Database=curswork;Username=postgres;Password=\*\*\*\*\*\*\*\*");

}

}

Алгоритм работы изменения данных клиента (Рисунок 3). С начало администратор с помощью combobox’a выбирает клиента для изменения:

using (DBcontexts db = new DBcontexts())

{

var humanNames = db.Customer.Select(p => p.Id.ToString() + " " + p.FirstName + " " + p.LastName).ToList();

CustomerSelector.ItemsSource = humanNames;

}

После чего происходит обновление текстовых полей в форме:

private void CustomerSelector\_SelectionChanged(object sender, RoutedEventArgs e)

{

Customer customer;

int customerID;

using (DBcontexts db = new DBcontexts())

{

customerID = int.Parse(CustomerSelector.SelectedItem.ToString().Split(' ')[0]);

customer = db.Customer.Where(p => p.Id == customerID).FirstOrDefault();

}

nameInput.Text = customer.FirstName;

lastNameInput.Text = customer.LastName;

PatronInput.Text = customer.Patronymic;

phoneInput.Text = customer.PhoneNumber;

cityInput.Text = customer.City;

}

Далее идет запись изменённых(или нет) параметров. По такому принципу записываются все остальные сущности в базе данных:

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

{

Customer customer;

int customerID;

using (DBcontexts db = new DBcontexts())

{

customerID = int.Parse(CustomerSelector.SelectedItem.ToString().Split(' ')[0]);

customer = db.Customer.Where(p => p.Id == customerID).FirstOrDefault();

customer.FirstName = nameInput.Text;

customer.LastName = lastNameInput.Text;

customer.Patronymic = PatronInput.Text;

customer.PhoneNumber = phoneInput.Text;

customer.City = cityInput.Text;

db.SaveChanges();

}

MessageBox.Show("Клиент успешно отредактикрован!");

Close();

}

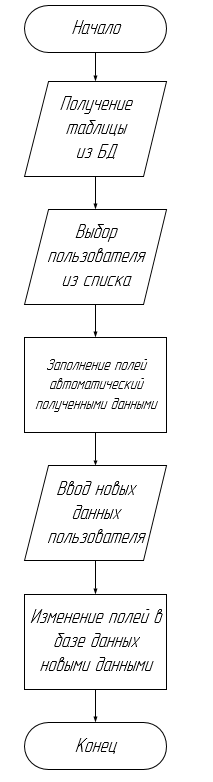


Рисунок 3 - Блок схема алгоритма изменения параметров клиента

Алгоритм удаление клиента из базы данных. Изначально мы с помощью combobox’a выбираем пользователя, а затем ищем в базе данных данного пользователя и удаляем:

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

{

using (DBcontexts db = new DBcontexts())

{

int CutomerId = int.Parse(CustomerSelector.SelectedItem.ToString().Split(' ')[0]);

Customer customer = db.Customer.Where(p => p.Id == CutomerId).FirstOrDefault();

db.Customer.Remove(customer);

db.SaveChanges();

}

MessageBox.Show("Клиент успешно удален!");

Close();

}

Алгоритм вывода всех пользователей (так же работает для вывода всех заказов):

using (DBcontexts db = new DBcontexts())

{

var customers = db.Customer.ToList();

dataIDGrid.ItemsSource = customers;

}

Алгоритм вывода всех заказов на сегодня (Рисунок 4)::

using (DBcontexts db = new DBcontexts())

{

DateTime dateTime = DateTime.Now;

var info = db.FuneralProcession.Where(p => p.Date == dateTime.ToShortDateString()).ToList();

dataIDGrid.ItemsSource = info;

}

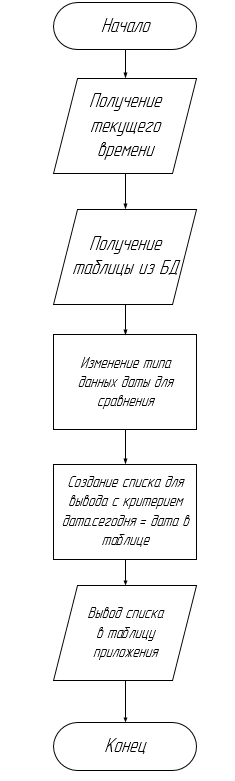


Рисунок 4 - Блок схема алгоритма вывода всех заказов на текущий день

Алгоритм вывода всех актуальных заказов (Рисунок 5):

using (DBcontexts db = new DBcontexts())

{

var Fp = db.FuneralProcession.ToList();

for (int i = 0; i < Fp.Count;)

{

if (DateTime.Parse(Fp[i].Date).AddHours(23) < DateTime.Now)

{

Fp.RemoveAt(i);

} else

{

i++;

}

}

dataIDGrid.ItemsSource = Fp;

}

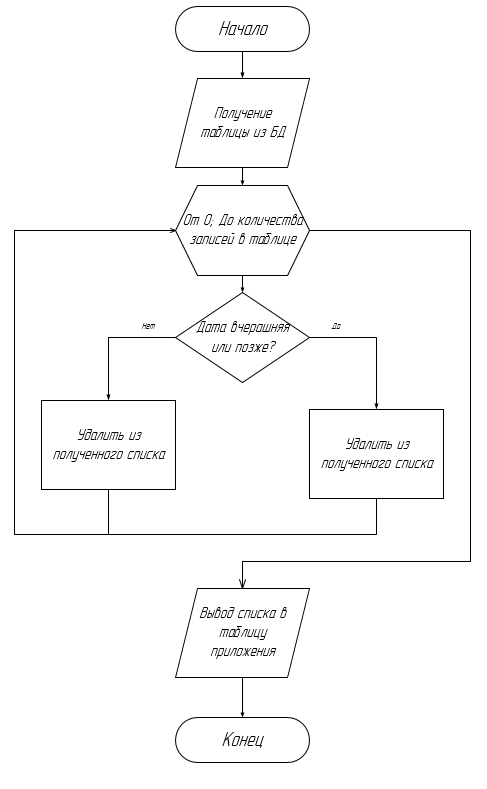


Рисунок 5 - Блок схема алгоритма вывода всех актуальных заказов

# 3.2 Разработка архитектуры приложения

Архитектурной конструкцией приложения является ASP.NET MVC 6 EntityFramework (Рисунок 6)

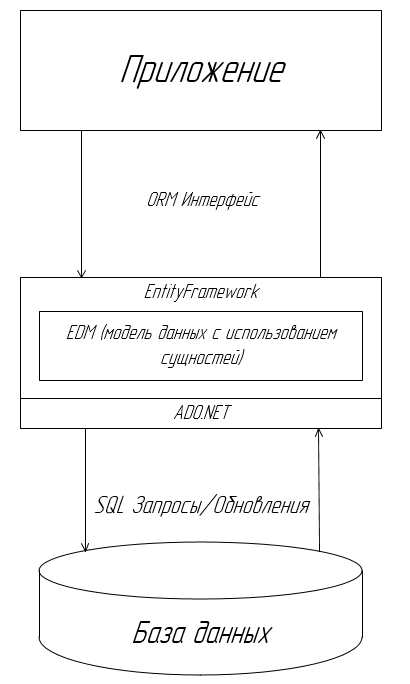


Рисунок 6 - Архитектура приложения

База данных хранит исходные данные о клиентах, заказах и так далее, заполненные администратором. Запросы к базе данных формируются в EntityFramework. После получения данных из базы данных EntityFramework передает ответ на клиент

ADO.NET — это набор классов, предоставляющих службы доступа к данным программистам, которые используют платформу .NET Framework. ADO.NET имеет богатый набор компонентов для создания распределенных приложений, совместно использующих данные. Это неотъемлемая часть платформы .NET Framework, которая предоставляет доступ к реляционным данным, XML-данным и данным приложений.

Модель EDM — это набор основных понятий, которые описывают структуру данных независимо от формы хранения. Модель EDM заимствует свойства модели «сущность-связь», описанной Питером Ченом в 1976 г., более того, она строится на модели «сущность-связь» и расширяет возможности ее традиционного использования.

ORM — технология программирования, которая связывает [базы данных](https://ru.wikipedia.org/wiki/%D0%91%D0%B0%D0%B7%D1%8B_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85) с концепциями [объектно-ориентированных языков программирования](https://ru.wikipedia.org/wiki/%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5).

# 3.4 Разработка графического интерфейса взаимодействия пользователя с системой.

Графический интерфейс был сделан с помощью языка XAML который используется в WPF, выбор был сделан за счет того что производительность у WPF выше чем у WinForm, где используется GDI/GDI+, за счёт использования аппаратного ускорителя графики через DirectX.

В основе WPF лежит векторная система визуализации, не зависящая от разрешения устройства вывода и созданная с учётом возможностей современного графического оборудования.

Возможности этого модуля расширяются с помощью комплексного набора функций разработки приложений, которые включают в себя язык XAML, элементы управления, привязку к данным, макет, двумерную и трехмерную графику, анимацию, стили, шаблоны, документы, мультимедиа, текст и типографические функции. WPF является частью .NET, поэтому вы можете создавать приложения, включающие другие элементы .NET API.

Меню приложения встречающее администратора выглядит так: (Рисунок 7)

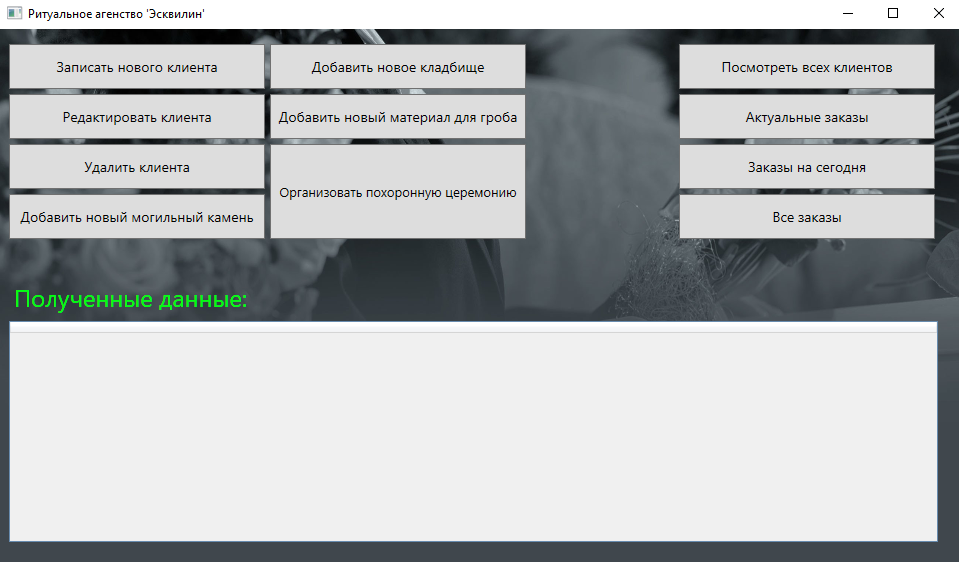
****

Рисунок 7 - Главное окно приложения

Также в приложении присутствуют другие окна (Рисунок 8), позволяющие удалять/изменять/добавлять пользователя, добавлять различную номенклатуру и организовывать похороненную церемонию.

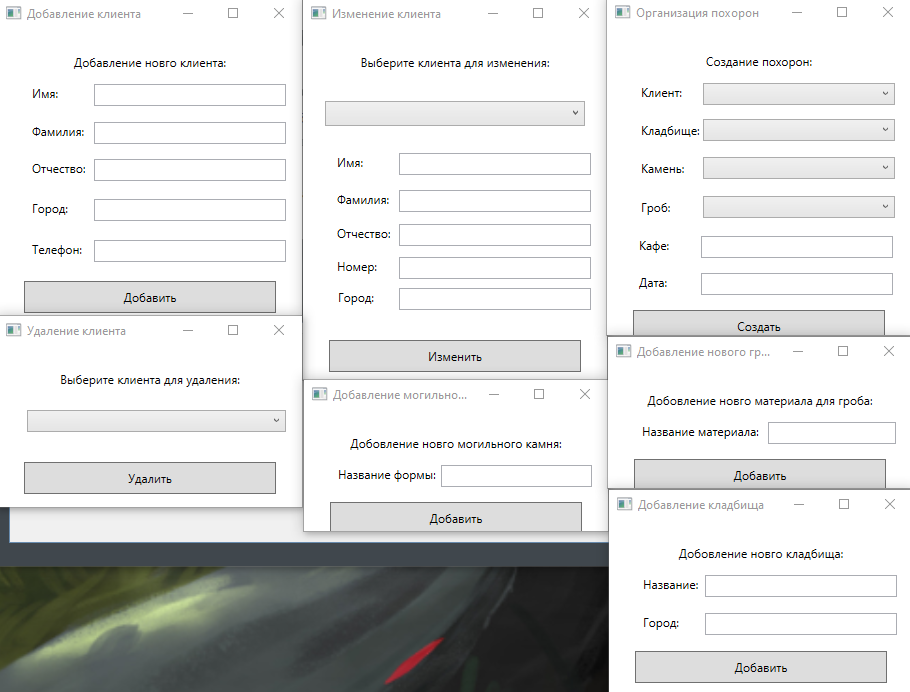


Рисунок 8 - Все второстепенное окна приложения позволяющие работать с базой данных.

Use case диаграмма использования интерфейса программы (Рисунок 9).

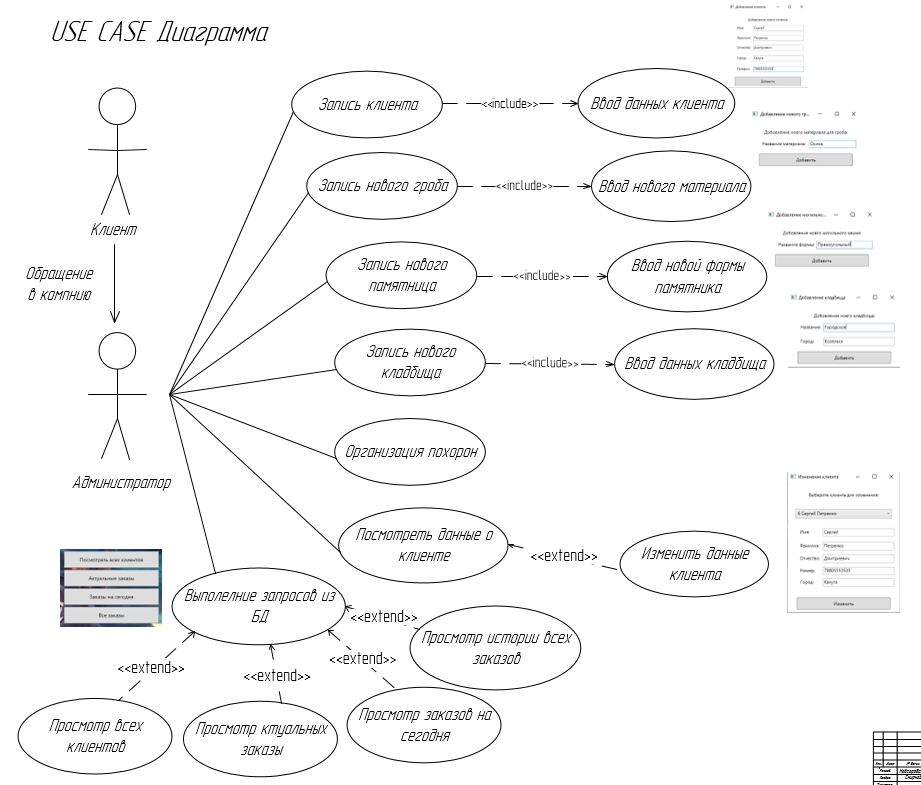


Рисунок 9 - Диаграмма вариантов использования.

# 4. Проектно-технологическая часть

# 4.1 Проектирование начального и тестового наполнения базы данных. Порядок развертывания системы

Всего в базе данных 5 таблиц (Рисунок 10):

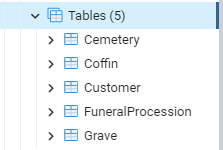


Рисунок 10 - Таблицы базы данных

Проектирование наполнения базы данных следующие:

1. Таблица Cemetery (Рисунок 11)

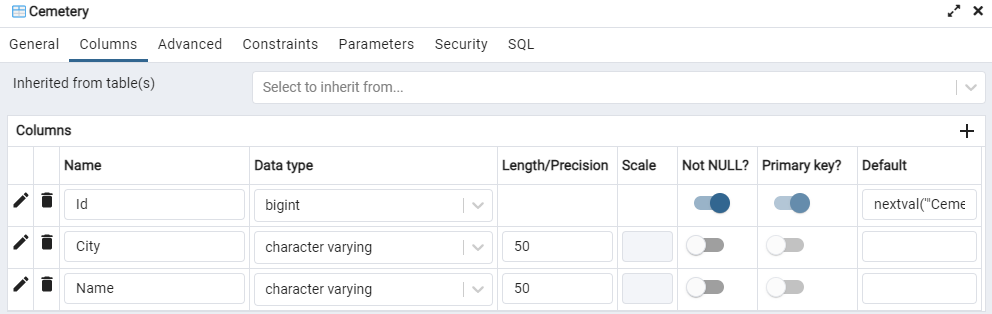


Рисунок 11 - Название колонок таблицы «Cemetery»

Заполненная данными таблица Cemetery (Рисунок 12)



Рисунок 12 - Таблица «Cemetery» с данными

1. Таблица Coffin (Рисунок 13)

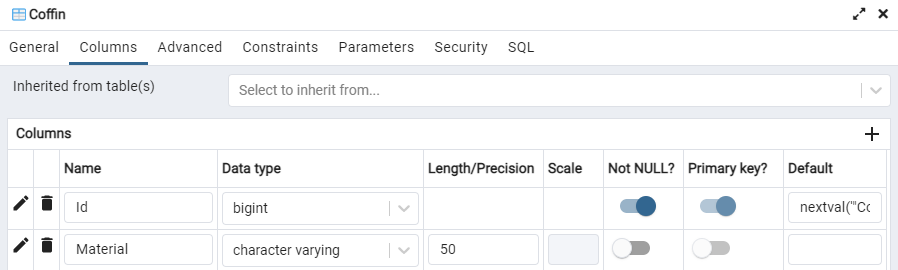


Рисунок 13 - Название колонок таблицы «Coffin»

Заполненная данными таблица Coffin (Рисунок 14)

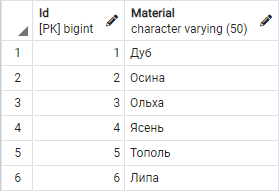


Рисунок 14 - Таблица «Coffin» с данными

1. Таблица Customer (Рисунок 15)

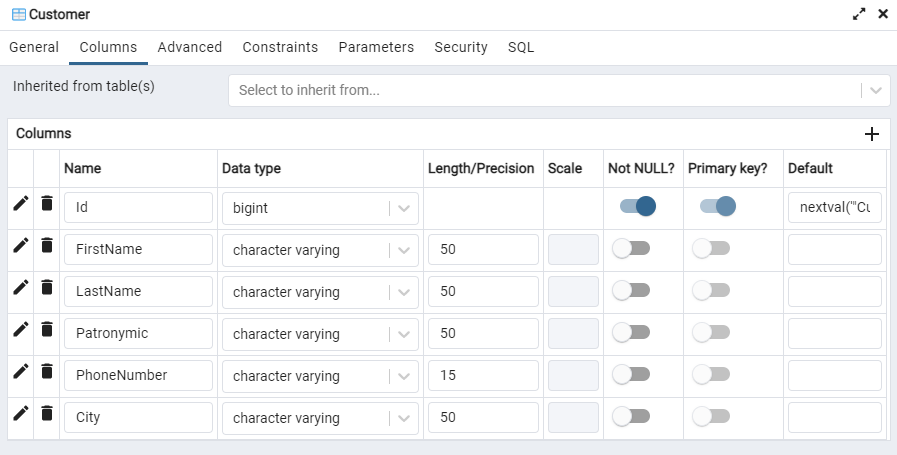


Рисунок 15 - Название колонок таблицы «Customer»

Заполненная данными таблица Customer (Рисунок 15)

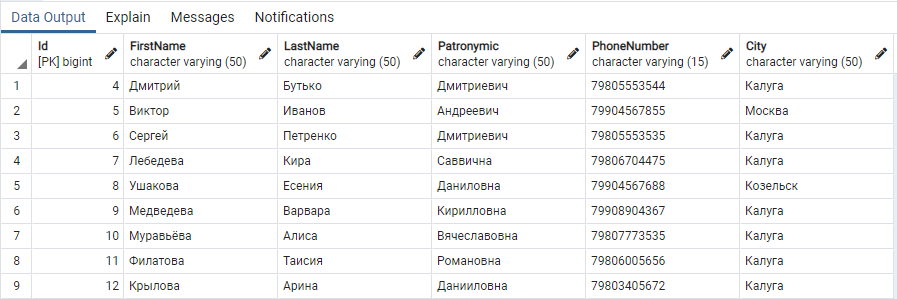


Рисунок 15 - Таблица «Customer» с данными

1. Таблица FuneralProcession (Рисунок 16)

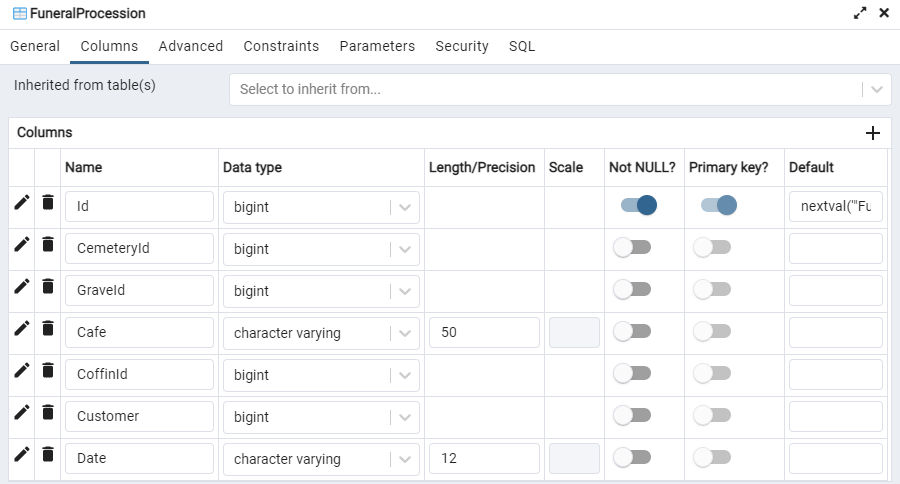


Рисунок 16 - Название колонок таблицы «FuneralProcession»

Заполненная данными таблица FuneralProcession (Рисунок 17)



Рисунок 17 - Таблица «FuneralProcession» с данными

1. Таблица Grave (Рисунок 18)

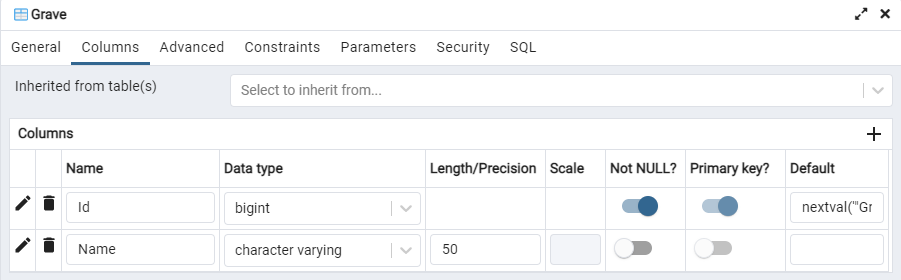


Рисунок 18 - Название колонок таблицы «Grave»

Заполненная данными таблица Grave (Рисунок 19)

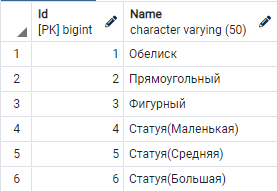


Рисунок 19 - Таблица «Grave» с данными

# 4.2 Руководство пользователя

Только запустив программу, администратор увидит главное меню программы (Рисунок 20). На нём представлен весь функционал программы.

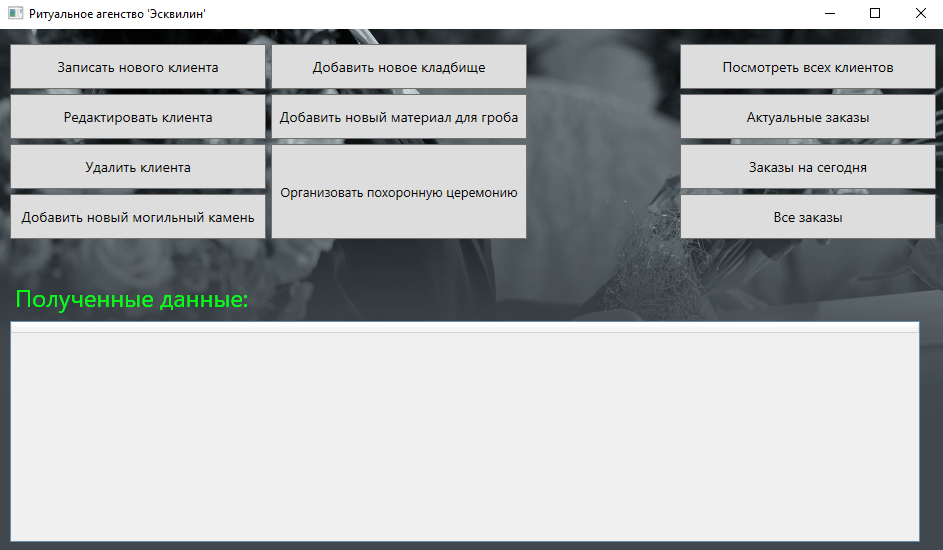


Рисунок 20 - Главное меню

Для записи нового клиента необходимо нажать на кнопку «Записать нового клиента» , далее откроется новое окно с формой для заполнения (Рисунок 21). Необходимо заполнить все поля и нажать на кнопку «Добавить»

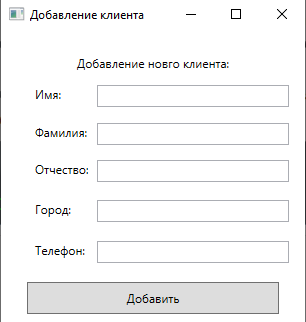


Рисунок 21 - Форма для заполнения данных о новом клиенте

Для изменения параметров определенного клиента, необходимо нажать на кнопку «Редактировать клиента» (Рисунок 22), где программа предложит выбрать клиента для изменения, а так же автоматический впишет их в поля для удобства (Рисунок 23), далее осталось только изменить один из параметров.

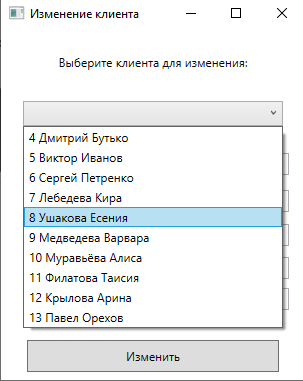


Рисунок 22 - Выбор клиента для изменения

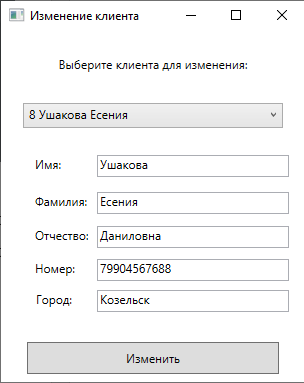


Рисунок 23 - Автоматический вставленные данные

Для удаления клиента из базы данных необходимо нажать на кнопку «Удалить клиента», где программа предложит из выбранного списка выбрать клиента для удаления (Рисунок 24), далее остается нажать на кнопку удалить (Рисунок 25).

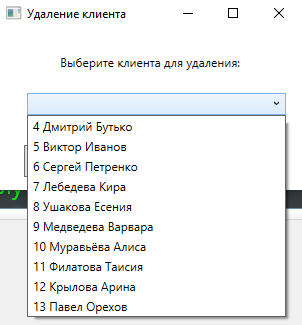
****

Рисунок 24 - Выбор удаляемого клиента

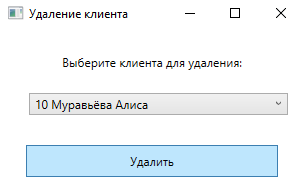


Рисунок 25 - Удаление клиента

Для добавления нового могильного камня в базу данных, необходимо нажать на кнопку «Добавить новый могильный камень», далее откроется новое окно с формой для заполнения данных (Рисунок 26), где необходимо заполнить поле «Название формы», и нажать кнопку «Добавить».

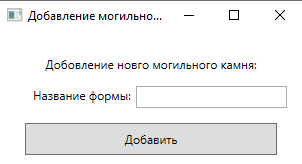


Рисунок 26 - Форма для заполнения данных о новой форме камня.

Для добавления нового кладбища в базу данных, необходимо нажать на кнопку «Добавить новое кладбище», далее откроется новое окно с формой для заполнения данных (Рисунок 27), где необходимо заполнить поля «Название» и «Город», и нажать на копку «Добавить».

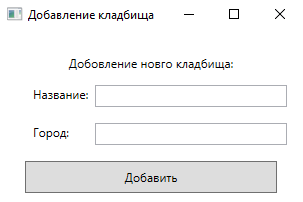
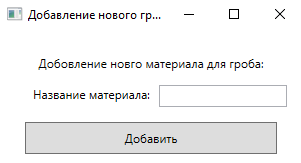


Рисунок 27 - Форма для заполнения данных о новом кладбище.

Для добавления нового материала для гроба в базу данных, необходимо нажать на кнопку «Добавить новый материал для гроба», далее откроется новое окно с формой для заполнения данных (Рисунок 28), где необходимо заполнить поле «Название материала», и нажать кнопку «Добавить».

  
Рисунок 28 - Форма для заполнения данных о новом материале для гроба.

В программе также можно сделать несколько запросов из базы данных (Рисунок 29), которые будут выводится в соответствующие таблице (Рисунок 30) подписанной как «Полученные данные»

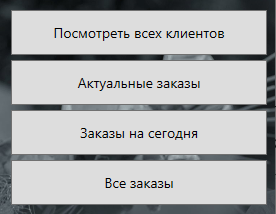


Рисунок 29 - Все запросы из базы данных

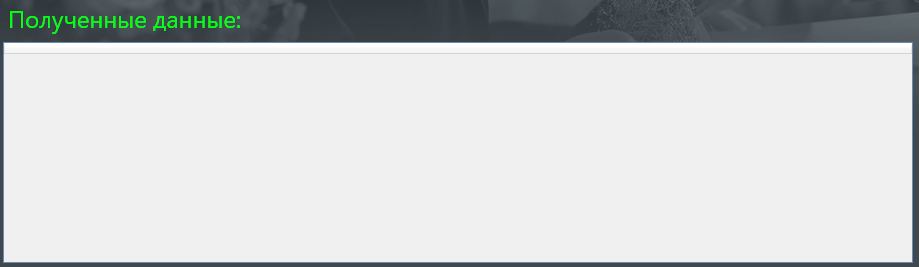


Рисунок 30 - Таблица для вывода результатов из базы данных

Поподробнее про каждую кнопку:

1. Нажав на кнопку «Посмотреть всех клиентов», в таблице приложения выведутся все данные о зарегистрированных клиентах (Рисунок 31)

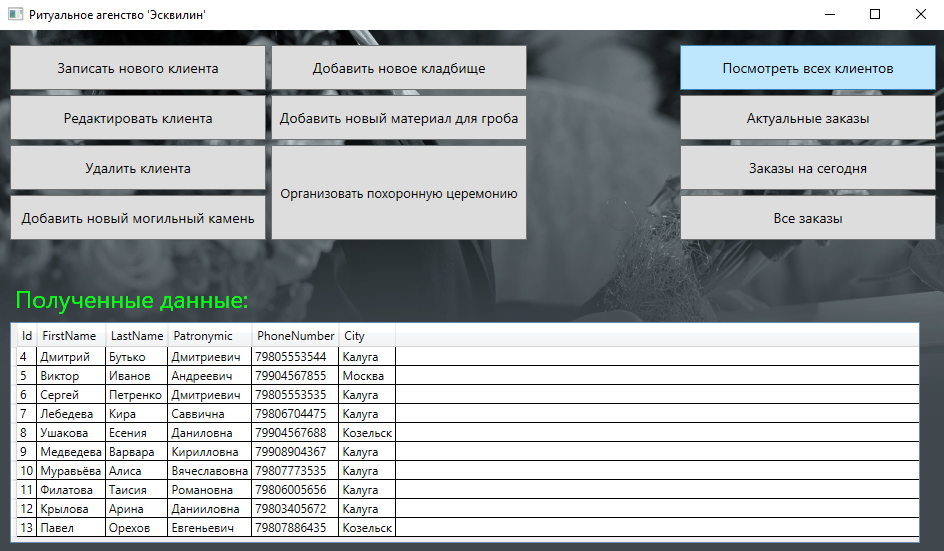


Рисунок 31 - Полученные данные запроса о выводе всех клиентов

1. Нажав на кнопку «Актуальные заказы», выведутся в таблицу только те заказы, которые должны быть выполнены сегодня или в последующие дни (Рисунок 32).

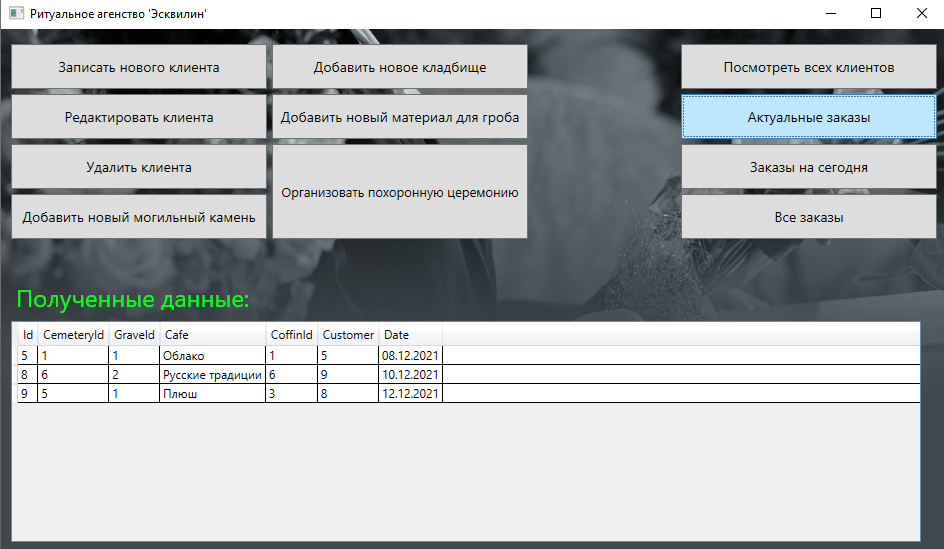


Рисунок 32 - Полученные данные запроса об актуальных заказах

1. Нажав на кнопку «Заказы на сегодня», выведутся в таблицу только те заказы, которые должны быть выполнены сегодня (Рисунок 33).

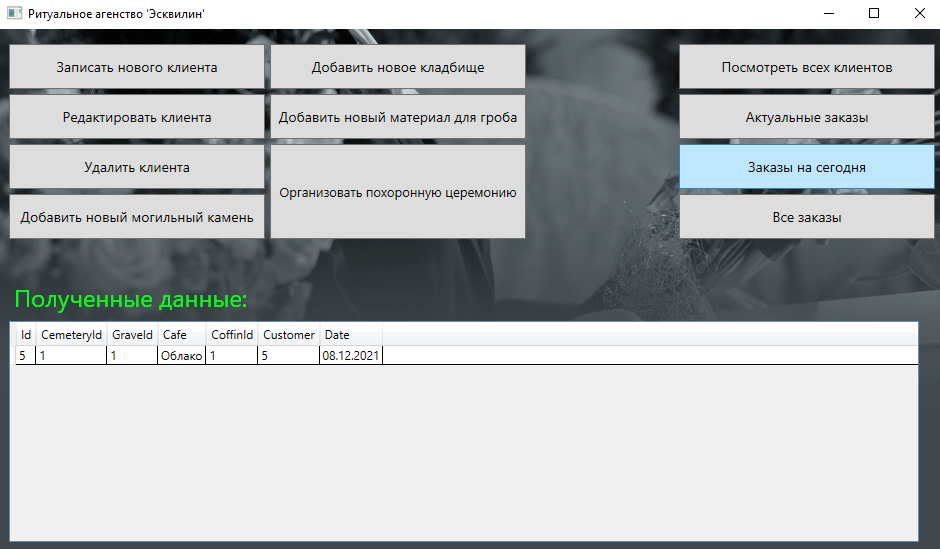


Рисунок 33 – Полученные данные запроса о заказах на сегодня

1. Нажав на кнопку «Все заказы», в таблицу выведутся все заказы, которые уже прошли и которые должны быть выполнены (Рисунок 34).

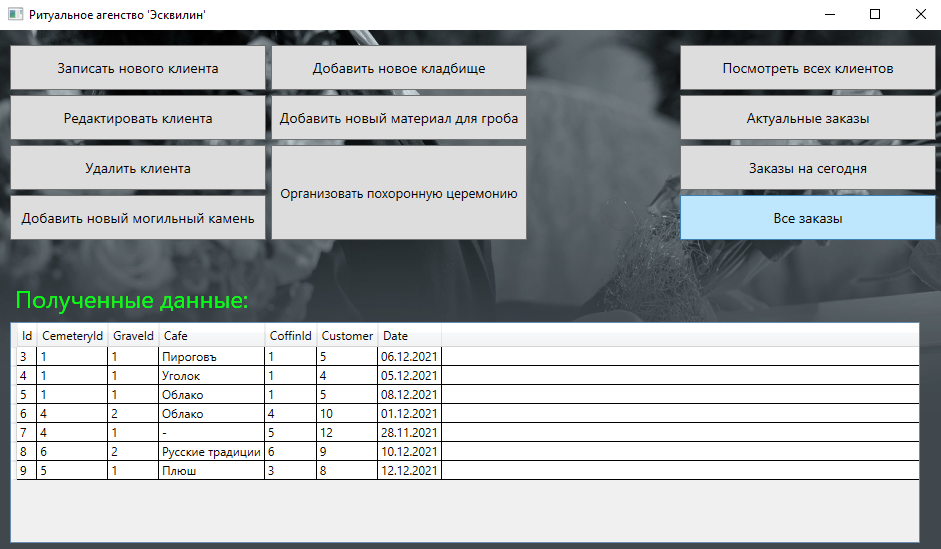


Рисунок 34 - Полученные данные запроса о всех заказах

# Заключение

В ходе выполнения курсовой работы было разработано программное обеспечение для записи клиентов на ритуальные услуги, способное записать, удалять и изменять данные находящиеся в базе данных.

Исследованы основные функции EntityFrameworkCore для работы с базой данных с помощью объектов .NET.

Работа выполнялась в несколько этапов: была выбрана архетектура, СУБД, а так же реализовано приложение.

В данной работе выполнены все поставленные задачи.

# Список литературы

1. Документация языка c# / docs.microsoft.com - https://docs.microsoft.com/ru-ru/dotnet/csharp/language-reference/, свободный.
2. Руководство по NET Core 6 режим доступа: https://metanit.com/sharp/tutorial/
3. Руководство по WPF https://metanit.com/sharp/wpf/
4. ГОСТ 34.201-90 Методические указания. Информационная технология. Комплекс стандартов на автоматизированные системы Автоматизированные системы. Требования к содержанию документов
5. ГОСТ 34.602-89. Информационная технология. Комплекс стандартов на автоматизированные системы. Виды, комплектность и обозначение документов при создании автоматизированной системы.
6. Е.В. Вершинин, Р.Б. Бобров “Методические указания к выполнению курсовой работы” Калуга КФ МГТУ им. Баумана 2021
7. Сайт ритуальных услуг «Вознесение» https://ritual40.ru
8. Сайт ритуальных услуг «Стелла» https://stella-ritual.ru
9. Шнырёв, С.Л. Базы данных: учебное пособие для вузов - М. : НИЯУ МИФИ, 2011. — 224 с. — Режим доступа: http://e.lanbook.com/book/75809
10. Ревунков, Г.И. Базы и банки данных - М.: МГТУ им. Н.Э. Баумана, 2011. — 68 с. — Режим доступа: http://e.lanbook.com/book/52425
11. Сидоров В.Н., Сломинская Е.Н., Полникова Т.В., Макарова О.Ю. Оформление графической части выпускной квалификаци-онной работы. Учебное пособие. М.: МГТУ им. Н.Э. Баумана, 2016.