# **ОБЩАЯ ЧАСТЬ**

## **2.1 Выбор и обоснование средств реализации БД**

Основным требованием работы является наличие интерфейса приложения, разрабатываемого MsSQL Server.

SQLсервер - программа, которая предназначена для хранения базы данных и обеспечения доступа к этим данным из других программ. В SQL Server включена поддержка языка XML и протокола HTTP, средства повышения быстродействия и доступности, позволяющие распределить нагрузку и обеспечить бесперебойную работу. Сложный доступ к данным используется для надежности их хранения.  SQL сервер позволяет резервное копирование в любой момент рабочего дня без отключения пользователей.

MsSQL Server обладает большим количеством преимуществ:

1. СУБД MsSQL Server является одной из самых быстрых баз данных среди имеющихся на современном рынке.

2. СУБД MsSQL Server является высокопроизводительной и относительно простой в использовании СУБД, которую значительно проще инсталлировать и администрировать, чем многие другие большие системы.

3. СУБД MsSQL Server распространяется бесплатно для домашнего использования.

4. MsSQL Server предназначена для работы в сети, может быть доступна через Интернет, но при этом снабжена развитой системой защиты от несанкционированного доступа.

5. MsSQL Server отлично работает как под управлением разных версий UNIX, так и под управлением других систем: Windows и OS/2. При этом система работает как на мощных серверах, так и на домашних ПК.

При написании приложения был выбран язык программирования C#, так как именно он больше всех подходит начинающим программистам для учебных разработок.

В разрабатываемом приложении необходимо добавлять, редактировать, удалять и делать отчёты по блюдам, поэтому выгоднее использовать MsSQL Server, потому что он, кроме своих преимуществ, предоставляет эту возможность совершенно бесплатно. Это действительно одна из самых мощных, быстрых и надежных СУБД.

## **2.2. Выбор и обоснование средств разработки программного продукта**

При проектировании приложения была использована программа C#.

C# является наиболее распространенной средой программирования с широкими возможностями. Кроме того, она для индивидуальных разработчиков бесплатна.

Главным преимуществом C# является производительность. Обеспечивает возможность создания разнообразных приложений на основе одного набора навыков [22].

C# — объектно-ориентированный язык программирования.

Основные достоинства С#:

* C# создавался параллельно с каркасом Framework .Net и в полной мере учитывает все его возможности — как FCL, так и CLR;
* C# является полностью объектно-ориентированным языком, где даже типы, встроенные в язык, представлены классами;
* C# является мощным объектным языком с возможностями наследования и универсализации;
* C# является наследником языков C/C++, сохраняя лучшие черты этих популярных языков программирования. Общий с этими языками синтаксис, знакомые операторы языка облегчают переход программистов от С++ к C#;
* сохранив основные черты своего великого родителя, язык стал проще и надежнее. Простота и надежность, главным образом, связаны с тем, что на C# хотя и допускаются, но не поощряются такие опасные свойства С++ как указатели, адресация, разыменование, адресная арифметика;
* благодаря каркасу Framework .Net, ставшему надстройкой над операционной системой, программисты C# получают те же преимущества работы с виртуальной машиной, что и программисты Java. Эффективность кода даже повышается, поскольку исполнительная среда CLR (Common Language Runtime - общеязыковая исполняющая среда) представляет собой компилятор промежуточного языка, в то время как виртуальная Java-машина является интерпретатором байт-кода;
* мощная библиотека каркаса поддерживает удобство построения различных типов приложений на C#, позволяя легко строить Web-службы, другие виды компонентов, достаточно просто сохранять и получать информацию из базы данных и других хранилищ данных;
* реализация, сочетающая построение надежного и эффективного кода, является немаловажным фактором, способствующим успеху C#.

## **2.3 Анализ прототипов, существующих программных продуктов**

При реализации программного продукта, разработчик анализирует и систематизирует данные, просматривая аналогичные приложения. На данном этапе происходит весь сбор информации, выявление достоинств и недостатков, которые имеются у прототипов. После проведённой работы, он продумывает план создания приложения, куда будут включены плюсы, а также исключены ошибки прототипов.

Схема анализа прототипов включает в себя:

* дизайн приложения и общее впечатление от него;
* доступность контактной информации для пользователей;
* структура программного продукта и удобство использования;
* навигация в приложении;
* объем и актуальность представленной информации в данном программном продукте;
* удобство использования панели администратора данного приложения.

При разработки информационной системы для формирования «Ежедневного меню» прототипами стали Web-сайты столовых «Все свои» и «Время есть».

Сайт «Все свои», на который можно перейти по ссылке <https://vsesvoispb.ru/menju-na-nedelju/> представлено на рисунке 1.

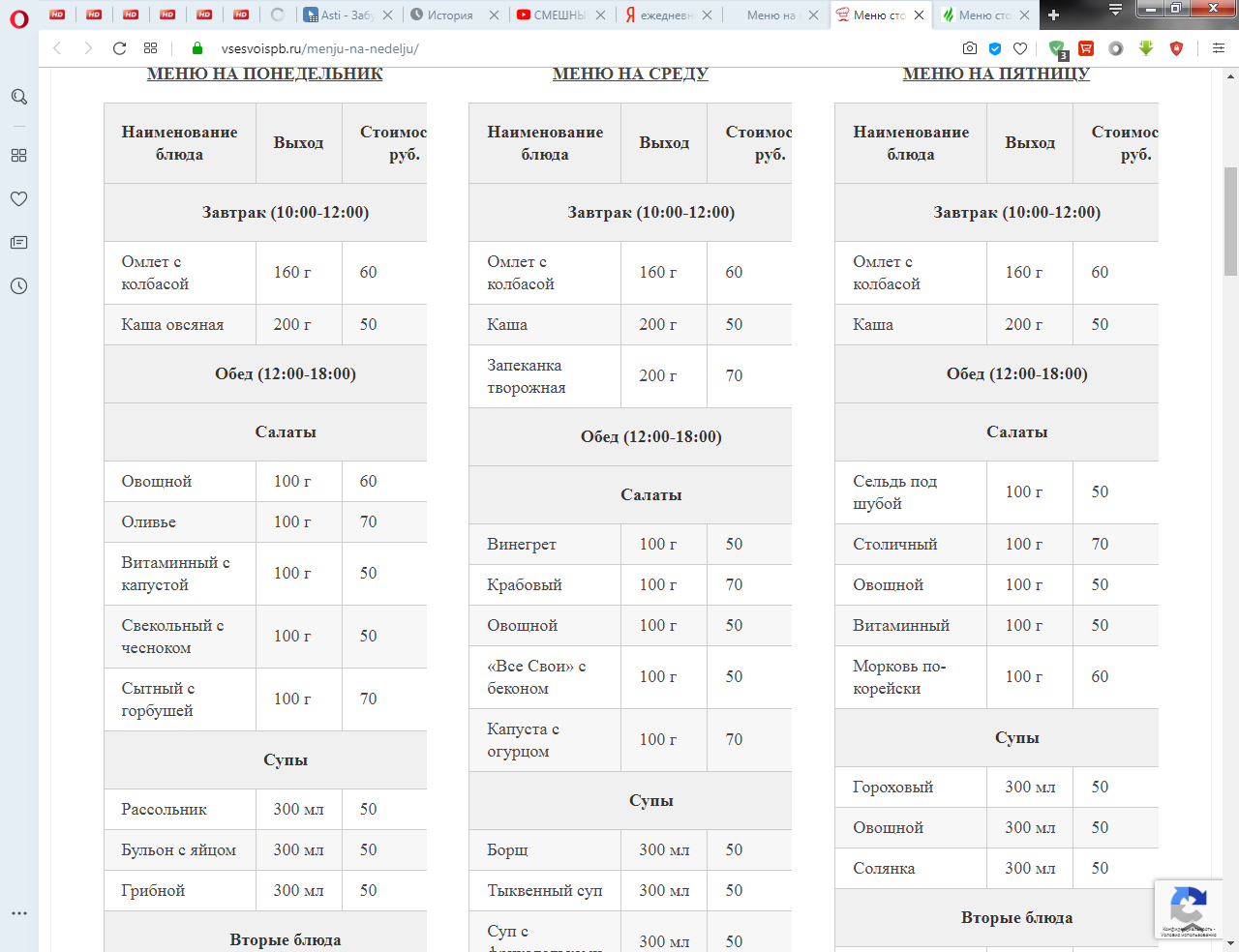


Рисунок 1 - «Все свои»

Простой дизайн и навигация. Не реализована в нём калорийность, состав и фото блюд.

Сайт «Время есть», которым можно воспользоваться, при переходе по ссылке <http://gottime.ru/menu/> представлено на рисунке 2.

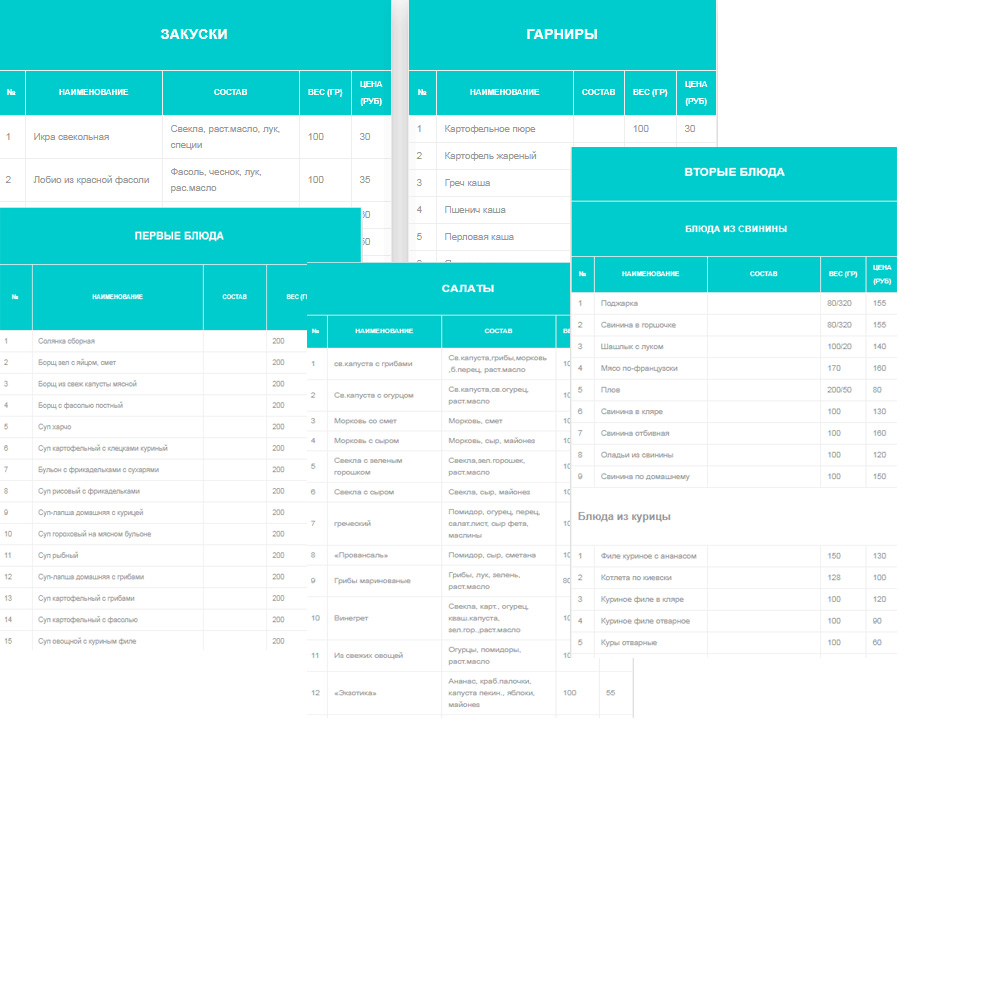


Рисунок 2 - «Время есть»

В дизайне данного сайта много нагромождений. При этом, сам сайт простой с удобной навигацией. Не реализована в нём калорийность и фото блюд.

# **3 СПЕЦИАЛЬНАЯ ЧАСТЬ**

## **3.1 Разработка алгоритма и проектирование структуры БД**

База данных «Bludo» была создана на основании задания, представленного в приложения А.

Модель базы данных «Bludo» создана в дизайнере MsSQL Server.

При создании таблиц базы данных использовался режим конструктора MsSQL Server, где были установлены свойства полей всех таблиц данной базы данных.

База данных «Bludo» состоит из четырёх таблиц:

Для создания таблиц использовался режим конструктора, позволяющий установить свойства полей таблицы.

Модель базы данных «Bludo» состоит из четырёх таблиц, каждая из которых имеет свою структуру, необходимую для установления связи между всеми таблицами:

* «Admin» (Администратор);
* «Bludo» (Блюда);
* «Menu» (Меню);
* «Type» (Тип);

Таблица «admin» предназначена для входа в систему в режиме администратора при вводе логина и пароля.

Таблица «Bludo» (Блюда) имеет связь с таблицей «Menu» (Меню).

Таблица «Type» (Тип), имеет связь с таблицей «Menu» (Меню).

Целостность данных *–* это значит, что данные будут надёжно хранится в базе данных. СУБД автоматически поддерживает целостность данных с помощью установленных связей между таблицами. Таблицы базы данных связаны связями «один ко многим»; для каждой из связей установлена поддержка целостности данных, каскадные обновление и удаление. Схема связей таблиц представлена в соответствии с рисунком 3.

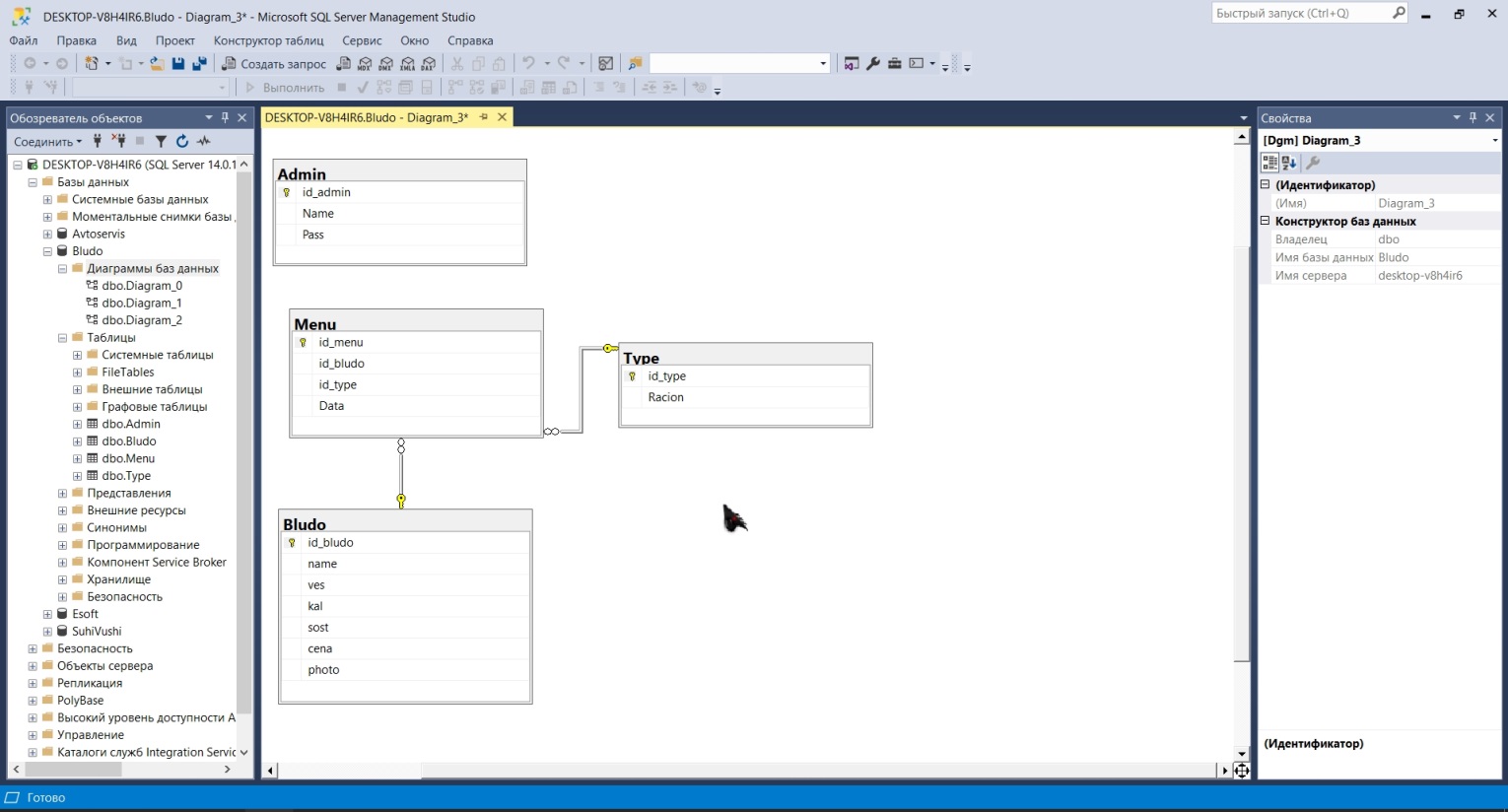


Рисунок 3 - «Схема связей таблиц»

## **3.2 Разработка программного продукта**

При разработке данного программного продукта в нём было реализовано несколько подсистем. Первая из них – это система авторизации. Данный механизм реализован посредством программного кода C#, причём вывод происходит при вызове диалогового окна. Код файла на Form 1, благодаря которому происходит процесс представлен на кнопке «Вход».

Кнопка «Вход».

string Name = textBox1.Text.Trim();

string Pass = textBox2.Text.Trim();

if (Name == "" || Pass == "")

{

MessageBox.Show("Вы не ввели данные");

}

else

{

SqlConnection connect = new SqlConnection(Program.st\_connect);

connect.Open();

string s = $"select \* from Admin where Name ='{Name}' and Pass='" +   
 Pass + "'";

SqlCommand command = new SqlCommand(s, connect);

SqlDataAdapter dataAdapter = new SqlDataAdapter(command);

DataTable data = new DataTable();

dataAdapter.Fill(data);

if (data.Rows.Count > 0)

{

Program.id\_admin = (int)data.Rows[0]["id\_admin"];

if (Program.id\_admin == (int)data.Rows[0]["id\_admin"])

{

Form2 f = new Form2();

f.Show();

}

}

connect.Close();

}

Стоит отметить, что в данном файле, как и во всей системе происходит работа с базой данных. После ввода логина и пароля происходит отправка запроса на выборку по логину (Login) и паролю (password) после чего возвращает результат выполнения. Далее проверяется количество строк результата. Если строк не ноль, то функция возвращает положительный ответ, в противном случае – отрицательный. В программе предусмотрен вывод предупреждения в случае ошибочных данных. Всё это также представлено на кнопке «Вход».

Вторая из подсистем – это управление блюдами. Отвечает она за открытия кнопок/форм для удаления, добавления, редактирования блюд, создания отчётов, принудительного обновления базы и составления меню на сегодня. Коды программы представлены на кнопках/формах: «Удаление», «Добавление», «Редактирование», «Создать отчёт по блюдам», «Создать меню на сегодня», «Обновить базу».

Кнопка «Удаление».

int k = dataGridView1.CurrentRow.Index;

int id\_bludo = Convert.ToInt32(dataGridView1[0, k].Value);

SqlConnection conn = new SqlConnection(Program.st\_connect);

conn.Open();

string s = "DELETE FROM Bludo WHERE id\_bludo=" +

id\_bludo.ToString();

SqlCommand comm = new SqlCommand(s, conn);

comm.ExecuteScalar();

conn.Close();

MessageBox.Show("Блюдо удалено");

Форма «Добавление».

if (textBox1.Text.Trim() == "" ||

textBox2.Text.Trim() == "" ||

textBox3.Text.Trim() == ""

)

{

MessageBox.Show("не все поля заполнены!");

}

else

{

SqlConnection conn = new SqlConnection(Program.st\_connect);

conn.Open();

string s = "select\* from Bludo where name='" + textBox1.Text.Trim() + "'";

SqlCommand comm = new SqlCommand(s, conn);

SqlDataReader read = comm.ExecuteReader();

if (read.HasRows)

{

MessageBox.Show("Такое блюдо уже существует");

}

else

{

read.Close();

string s1 = "insert into Bludo values " +

"('" + textBox1.Text.Trim() + "'," +

"" + textBox2.Text.Trim() + "," +

"" + textBox3.Text.Trim() + "," +

"'" + textBox4.Text.Trim() + "'," +

"'" + textBox5.Text.Trim() + "'," +

"'" + label6.Text.Trim() + "')";

SqlCommand comm1 = new SqlCommand(s1, conn);

comm1.ExecuteScalar();

this.Close();

}

conn.Close();

}

if (openFileDialog1.ShowDialog() == DialogResult.OK)

{

label6.Text = Path.GetFileName(openFileDialog1.FileName);

File.Copy(openFileDialog1.FileName,

AppDomain.CurrentDomain.BaseDirectory + label6.Text, true);

pictureBox1.Image = Image.FromFile(openFileDialog1.FileName);

}

Форма «Редактирование».

SqlConnection conn = new SqlConnection(Program.st\_connect);

conn.Open();

string s1 = "update Bludo set " +

"name='" + textBox1.Text.Trim() + "', " +

"ves='" + textBox2.Text.Trim() + "', " +

"kal='" + textBox3.Text.Trim() + "', " +

"sost='" + textBox4.Text.Trim() + "', " +

"cena='" + textBox5.Text.Trim() + "', " +

"photo='" + label6.Text.Trim() + "' " +

"where id\_bludo=" + Edit\_ID;

SqlCommand comm1 = new SqlCommand(s1, conn);

comm1.ExecuteScalar();

this.Close();

conn.Close();  
 if (openFileDialog1.ShowDialog() == DialogResult.OK)

{

label6.Text = Path.GetFileName(openFileDialog1.FileName);

File.Copy(openFileDialog1.FileName,  
 AppDomain.CurrentDomain.BaseDirectory + label6.Text, true);

pictureBox1.Image = Image.FromFile(openFileDialog1.FileName);

}

Форма «Создать отчёт по блюдам».

1)SqlConnection conn = new SqlConnection(Program.st\_connect);

conn.Open();

int id = Convert.ToInt32(comboBox1.SelectedValue);

string s = "select Type.Racion, Bludo.name, Bludo.ves, Bludo.kal,

Bludo.sost, Bludo.cena, Bludo.photo, Menu.Data " +

"From dbo.Menu inner join dbo.Type on Menu.id\_type = Type.id\_type

inner join Bludo on Menu.id\_bludo = Bludo.id\_bludo "+

"where Bludo.id\_bludo= "+id;

SqlDataAdapter adap = new SqlDataAdapter(s, conn);

DataSet ds = new DataSet();

adap.Fill(ds);

dataGridView1.DataSource = ds.Tables[0];

2)Microsoft.Office.Interop.Excel.Application myExcel = new

Microsoft.Office.Interop.Excel.Application();

myExcel.Application.Workbooks.Add(Type.Missing);

myExcel.Columns.ColumnWidth = 15;

for (int i = 0; i < dataGridView1.ColumnCount; i++)

{

myExcel.Cells[1, i + 1] = dataGridView1.Columns[i].HeaderText;

}

for (int i = 0; i < dataGridView1.ColumnCount; i++)

{

for (int j = 0; j < dataGridView1.RowCount - 1; j++)

{

myExcel.Cells[j + 2, i + 1] = dataGridView1[i, j].Value.ToString();

}

}

myExcel.Visible = true;

Форма «Создать меню на сегодня».

SqlConnection conn = new SqlConnection(Program.st\_connect);

conn.Open();

string s = "select Type.Racion, Bludo.name, Bludo.ves, Bludo.kal,

Bludo.sost, Bludo.cena, Bludo.photo, Menu.Data " +

"From dbo.Menu inner join dbo.Type on Menu.id\_type = Type.id\_type inner

join Bludo on Menu.id\_bludo = Bludo.id\_bludo " +

"where Menu.Data= " + "'" + dateTimePicker1.Value + "'";

SqlDataAdapter adap = new SqlDataAdapter(s, conn);

DataSet ds = new DataSet();

adap.Fill(ds);

dataGridView2.DataSource = ds.Tables[0];

**3.3 Руководство пользователя**

Для того, чтобы зайти в данное приложение, необходимо запустить C#, открыть проект «Menu». На рисунке 4 представлено как это сделать.

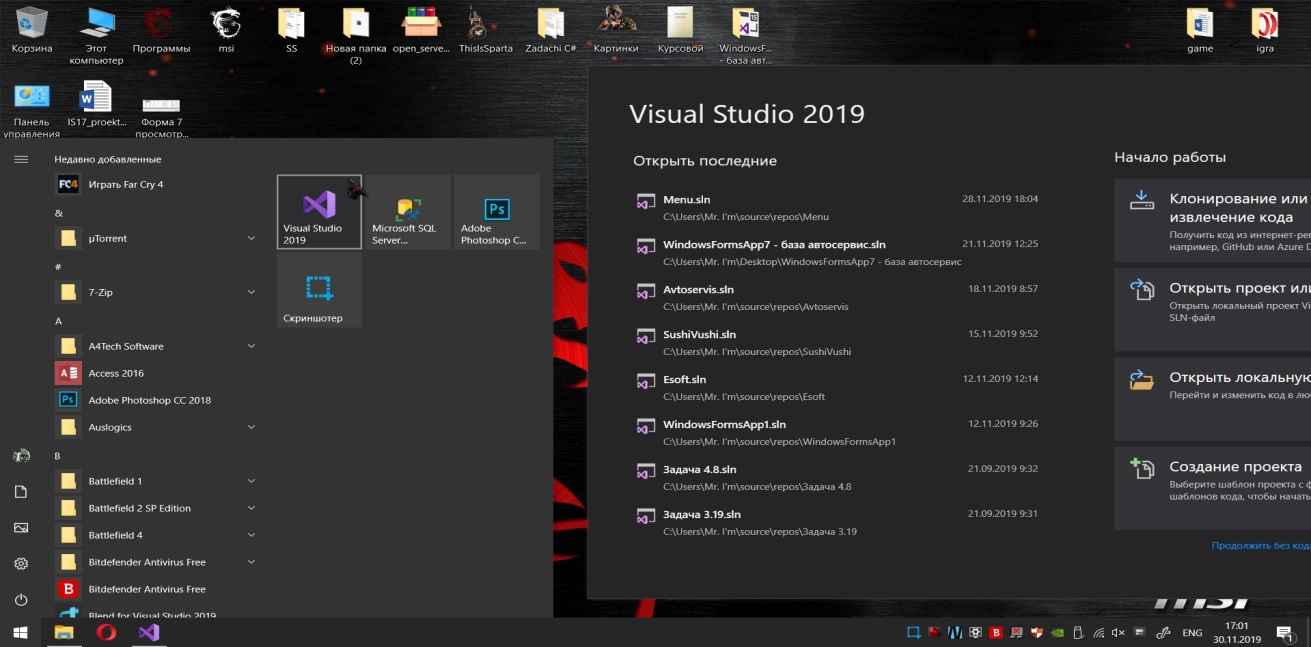


Рисунок 4 - Запуск проекта «Menu» в C#

Далее открывается стартовая форма курсового проекта «Ежедневное меню», где предлагается войти в систему, либо посмотреть меню на сегодня.

После входа в систему открывается форма управления блюдами где можно просматривать блюда и работать с ними (удалять, редактировать и т.д.). Форма управления блюдами представлено на рисунке 5.

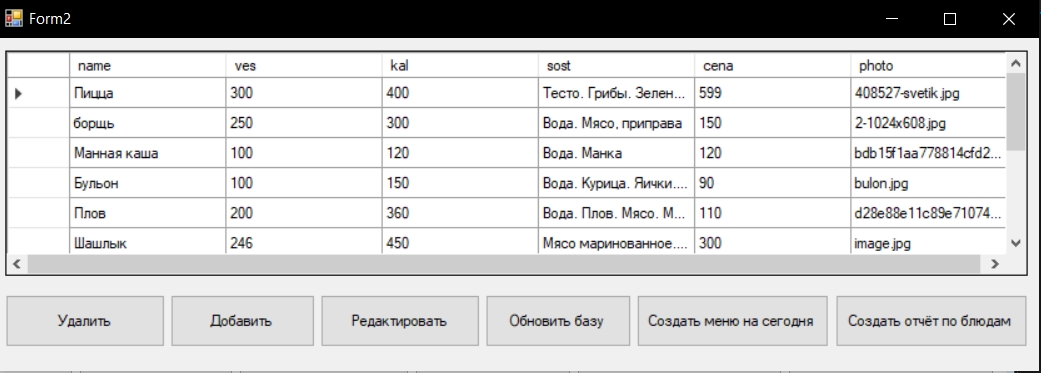


Рисунок 5 - форма управления блюдами

При нажатии на кнопку «Удалить» из таблицы удалиться выбранное блюдо, об этом проинформирует открывшийся окно с подтверждением. На рисунке 6 показано как выглядит окно подтверждения удаления.

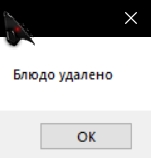


Рисунок 6 - подтверждения удаления

При нажатии на кнопку «добавить» откроется Form 3 позволяющая добавлять новый блюда. Если не все поля будут заполнены, то при добавлении появится окно «не все поля заполнены». Так же при добавлении форма проверяет на наличие существующих блюд, дабы избежать дублирования. При нажатии на кнопку «Добавить блюдо» форма закроется, указывая на то что всё успешно добавлено. На рисунке 7 показано как добавить блюдо.

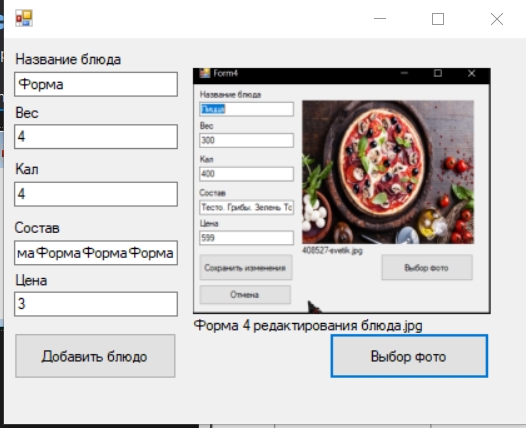


Рисунок 7 - добавление блюда

Нажав на кнопку «Редактировать» откроется Form 4, на которой происходит редактирование. На ней присутствует информация о блюде, которую можно изменить. При необходимости редактирование можно отменить, нажав соответствующую кнопку. При нажатии на кнопку «Сохранить изменения» форма закроется, указывая на то, что всё успешно изменено. На рисунке 8 показано как выглядит форма редактирования.

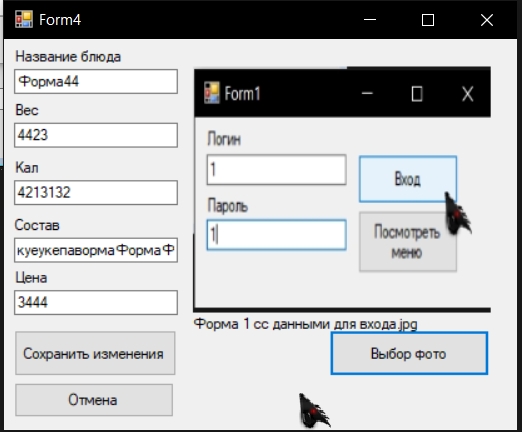


Рисунок 8 – форма редактирования

Нажав на кнопку «Создать меню на сегодня» откроется Form 5, на которой можно указать время питания, выбрать дату и блюдо. При желании можно убрать из меню неподходящее блюдо. Форма заполнения меню показана на рисунке 9.

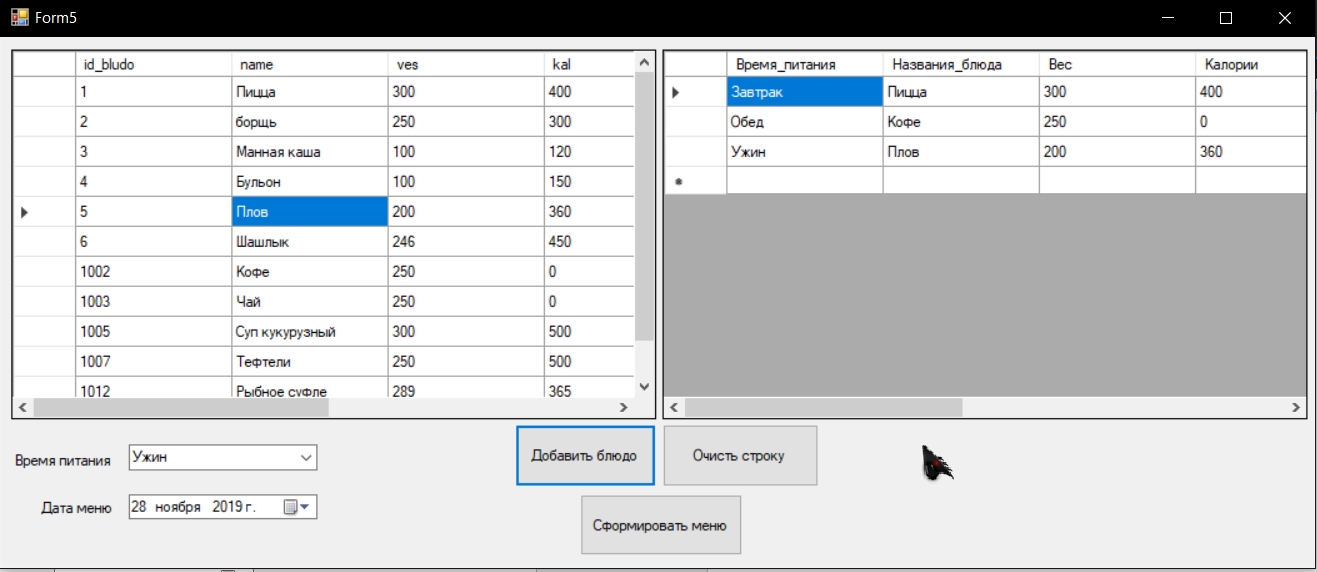


Рисунок 9 – Заполнение меню

Кликнув на кнопку «Создать отчёт по блюдам» откроется Form 6, на которой из выпадающего списка можно выбрать блюдо, и узнать когда и сколько, было сделано заказов. На рисунке 10 изображен отчёт в таблице на форме.

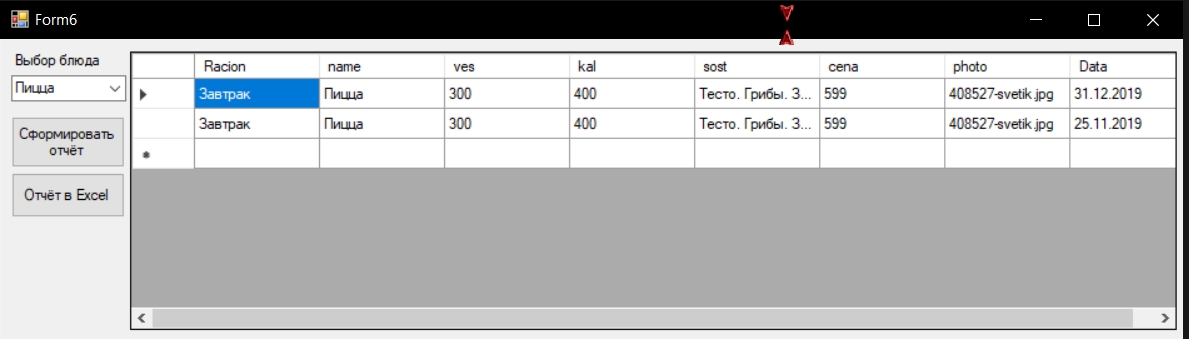


Рисунок 10 – Отчёт на форме

На рисунке 11 показан отчёт в Excel.

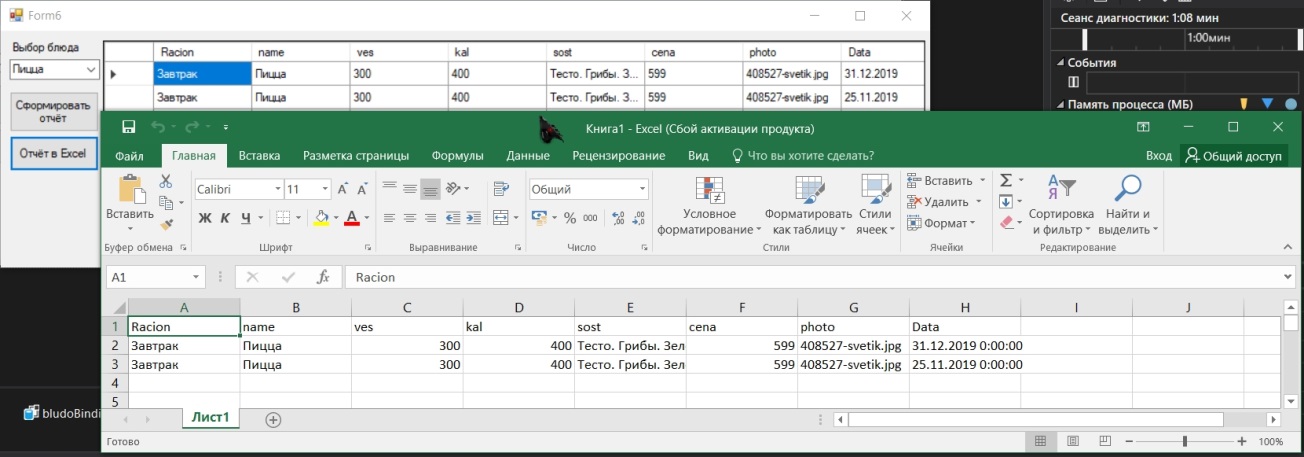


Рисунок 11 - Отчёт в Excel

Возвращаясь на стартовую форму, рассмотрим кнопку «Посмотреть меню», нажав на которую можно ознакомиться с меню на сегодня. На рисунке 12 продемонстрирована стартовая форма в режиме гость. На рисунке 13 продемонстрирован просмотр меню на сегодня.

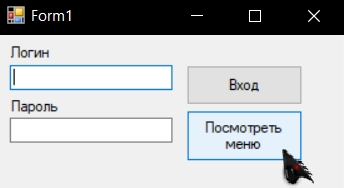


Рисунок 12 - Стартовая форма в режиме гость

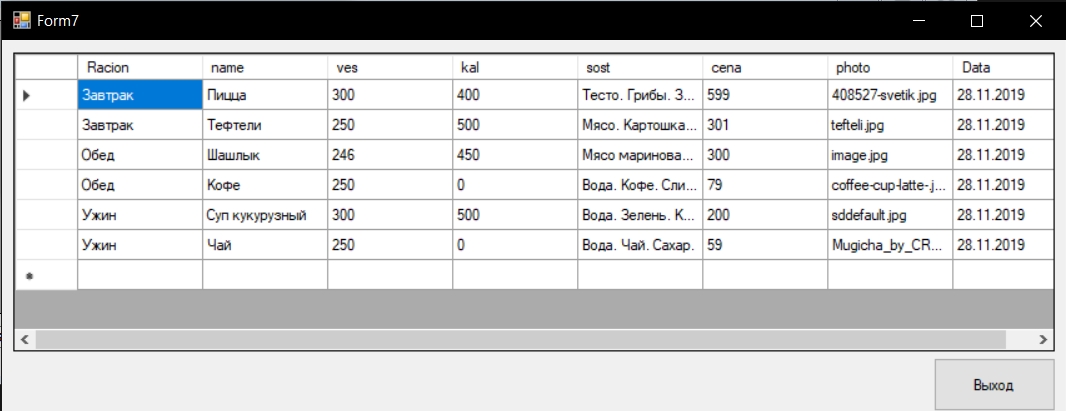


Рисунок 13 - Просмотр меню на сегодня

# **ЗАКЛЮЧЕНИЕ**

В результате курсовой работы было проведено исследование предметной области, изучена среда разработки, разработан интерфейс C#, проведено тестирование и отладка программного обеспечения, произведен сбор данных об объектах.

Разработанный проект позволит всем желающим ознакомится с ежедневным меню.

В дальнейшем планируется провести более детальную проработку интерфейса приложения.

Цель достигнута, задачи выполнены.

# **СПИСОК ЛИТЕРАТУРЫ**

1. Советов,Б.Я.. Базы данных: Учебник для СПО / Б.Я. Советов, В.В.Цехановский, В.Д. Чертовской. – 2-е изд. –М.: Издательство Юрайт, 2017 – 463 с.
2. Рыбальченко,М.В. Архитектура информационных систем: учеб. пособие для СПО / М.В. Рыбальченко. – М.: Издательство Юрайт, 2017. -91 с.
3. ГОСТ 19.701-90 ЕСПД. Схемы алгоритмов, программ, данных и систем. Обозначения условные и правила выполнения.
4. Рекомендации по оформлению самостоятельных работ №348 ФКПОУ «НГГТКИ», 2016