**Введение**

Современный отдел сбыта – это весьма сложный объект как технически, так и с точки зрения управления, работу отдела сбыта любого предприятия можно сравнить с работой оптового склада. Потребность в таких отделах существует на всех этапах движения товаров, начиная от источников сырья и заканчивая конечными потребителями.

Отделы сбыта представляют собой подразделения оптовых баз, предназначаемых для хранения торговых запасов, комплектования товарного ассортимента, они являются подразделением не только для оптовых баз, но и для предприятий оптовой торговли, а также являются материально технической базой розничной торговли.

Функции отдела сбыта:

- получение товара от поставщиков и осуществление контроля за их количеством и качеством.

- обеспечение надлежащих усилий их хранения.

- сортировка и подготовка товаров к продаже.

- комплектование заказов покупателей.

- товароснабжение различных торговых предприятий.

Процессы, протекающие внутри отдела сбыта технически сложны и требуют трудовых и финансовых ресурсов. Чтобы упростить учет товара и контроль за его движением, необходимо навести порядок и тщательно систематизировать всю сопряженную информацию. Ускорить выполнение бизнес-задач помогают специализированные оборудование и программное обеспечение.

Любой бизнес, малый или большой, нуждается в понятном для пользователя интерфейсе программного обеспечения. Для записи, выборки и обработки данных, хранящихся в компьютерной базе данных, необходима система управления базой данных, каковой и является ПО MySQL.

Целью выполнения курсового проекта является развитие навыков реализации приложения с использованием прикладного интерфейса; среды разработки, получение опыта разработки сетевых программных средств.

В ходе выполнения курсового проекта у студента должно формироваться представление об этапах проектирования и создания сетевых программных средств.

В ходе достижения цели решаются следующие задачи:

– развитие логического и алгоритмического мышления;

– выработка умения формулировать суждения и выводы, логически последовательно и доказательно их излагать;

– овладение навыками самостоятельной исследовательской работы;

– закрепление использования технологии сетевого программирования на стороне клиента и сервера.

**1 Описание предметной области**

* 1. **Общие положения**

Для создания базы данных для приложения отдела сбыта необходимо описать рабочую модель его работы.

База данных позволяет пользователю ознакомиться с наличием товара на складе, поставщиками, клиентами и совершёнными сделками. Добавлять, удалять и изменять необходимую информацию, тем самым ведя учёт движения товара на складе в режиме реального времени.

База данных отдела сбыта – это экономически выгодно, так как её наличие значительно упрощает работу экономистов и логистов, можно проследить движение товара на каждом его этапе, так же полностью контролировать и документировать все процессы, происходящие в пределах отдела.

Целью работы является проектирование и реализация базы данных для отдела сбыта металлургического завода средствами MySQL в среде VisualStudio.

Создание базы данных решает следующие задачи:

– структурирование информации о поставщиках;

– структурирование информации о покупателях;

– структурирование информации о товарах на складе;

С помощью базы данных можно:

– узнавать информацию о поставщиках (код поставщика, имя, телефон);

– узнавать информацию о покупателях (код покупателя, имя, телефон);

– узнавать информацию о товарах на складе (код товара, название товара, количество, цена продажи за единицу товара);

– научиться формировать логические связи в ПО MySQL между таблицами;

– самостоятельно разработать формы для заполнения;

– разработать удобный для клиента интерфейс для работы с базой данных;

– освоить навыки программирования в среде VisualStudio на языке программирования C#.

* 1. **Сведения из теории**

MySQL — это быстрая, надежная, открыто распространяемая СУБД. MySQL, как и многие другие СУБД, функционирует по модели "клиент/сервер". Под этим подразумевается сетевая архитектура, в которой компьютеры играют роли клиентов либо серверов.

Клиентская программа MySQL представляет собой утилиту командной строки. Эта программа подключается к серверу по сети. Команды, выполняемые сервером, обычно связаны с чтением и записью данных на жестком диске. MySQL взаимодействует с базой данных на языке, называемом SQL (Structured Query Language — язык структурированных запросов).

Windows Forms — это технология пользовательского интерфейса для .NET, представляющая собой набор управляемых библиотек, которые упрощают выполнение стандартных задач, таких как чтение из файловой системы и запись в нее.

С помощью среды разработки, такой как VisualStudio, можно создавать интеллектуальные клиентские приложения Windows Forms, которые отображают информацию, запрашивают ввод пользователя и взаимодействуют с удаленными компьютерами по сети.

В WindowsForms форма — это визуальная поверхность, на которой выводится информация для пользователя. Обычно приложение Windows Forms строится путем добавления элементов управления в формы и создания кода для реагирования на действия пользователя, такие как щелчки мыши или нажатия клавиш. Элемент управления — это отдельный элемент пользовательского интерфейса, предназначенный для отображения или ввода данных.

При выполнении пользователем какого-либо действия с формой или одним из ее элементов управления создается событие. Приложение реагирует на эти события, как задано в коде, и обрабатывает события при их возникновении.

C# — объектно-ориентированный, ориентированный на компоненты язык программирования. C# предоставляет языковые конструкции для непосредственной поддержки такой концепции работы. Благодаря этому C# подходит для создания и применения программных компонентов. С момента создания язык C# обогатился функциями для поддержки новых рабочих нагрузок и современными рекомендациями по разработке ПО. В основном C# — объектно-ориентированный язык. Вы определяете типы и их поведение. Вот лишь несколько функций языка C#, которые позволяют создавать надежные и устойчивые приложения.

Сборка мусора автоматически освобождает память, занятую недостижимыми неиспользуемыми объектами. Типы, допускающие значение null, обеспечивают защиту от переменных, которые не ссылаются на выделенные объекты. Обработка исключений предоставляет структурированный и расширяемый подход к обнаружению ошибок и восстановлению после них. Лямбда-выражения поддерживают приемы функционального программирования. Синтаксис LINQ создает общий шаблон для работы с данными из любого источника. Поддержка языков для асинхронных операций предоставляет синтаксис для создания распределенных систем. В C# имеется Единая система типов. Все типы C#, включая типы-примитивы, такие как int и double, наследуют от одного корневого типа object. Все типы используют общий набор операций, а значения любого типа можно хранить, передавать и обрабатывать схожим образом. Более того, C# поддерживает как определяемые пользователями ссылочные типы, так и типы значений. C# позволяет динамически выделять объекты и хранить упрощенные структуры в стеке. C# поддерживает универсальные методы и типы, обеспечивающие повышенную безопасность типов и производительность. C# предоставляет итераторы, которые позволяют разработчикам классов коллекций определять пользовательские варианты поведения для клиентского кода.

* 1. **Постановка задачи**

База данных должна хранить данные о продукции предприятия, информацию о покупателях. Система должна формировать договора поставки. В процессе договорной кампании с покупателем составляются договора на поставку готовых изделий. Договор состоит из двух частей: общей части, включающей в себя реквизиты покупателя и поставщика, предмет поставки и т.д., и спецификации в которой приводятся сведения о товарах и сроках поставки. При отгрузке изделий покупателю товар сопровождается накладной.

База данных отдела(factoryDB) сбыта будет состоять из трёх таблиц, каждая из которых выполняет свою определенную функцию для работы приложения (рисунок 1). В базе данных имеются следующие таблицы: информация о покупателях, информация о поставщиках, информация о товарах на складе.

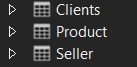


Рисунок 1 – Сущности базы данных factoryDB

Таблицы «Сlients» и «Seller» содержат информацию о покупателях и поставщиках. В таблице описаны следующие поля:

- код поставщика/покупателя

- имя поставщика/покупателя

- телефон

Ключевым полем является поле код поставщика/покупателя. Данное поле имеет целочисленный тип и предназначено для манипуляции полями внутри базы данных.

Поле «имя поставщика/покупателя» имеет строковый тип данных и предоставляет информацию об имени поставщика/покупателя.

Таблица «Product» содержит информацию о товарах на складе. В таблице описаны следующие поля:

- код товара

- название товара

- количество товара на складе

- стоимость за единицу товара

Ключевым полем является поле код товара. Данное поле имеет целочисленный тип и предназначено для манипуляции полями внутри базы данных.

Для реализации базы данных будет использоваться среда разработки VisualStudio, в ней мы используем язык программирования C#, а также интерфейс программирования приложений с графическим интерфейсом - WindowsForms

Запросы к базам данных выполняются с помощью SQL-запросов. SQL — декларативный язык программирования, применяемый для создания, модификации и управления данными в реляционной базе данных, управляемой соответствующей системой управления базами данных. Подключение к БД реализуется внутренними средствами VisualStudio.

**2 Технологии разработки приложения**

**2.1 Алгоритм решения**

Работа программы начинается с момента её запуска. После запуска производятся действия, записанные в виде автоматических событий. В самом начале происходит подключение к базе данных, далее посредством обращения к подключенной БД происходят необходимые манипуляции с данными(вывод в DataGridView, добавление, удаление и редактирование данных).

**2.2 Макет приложения**

Созданное приложение состоит из 4 вкладок, которые представлены на рисунках 2,3,4,5.

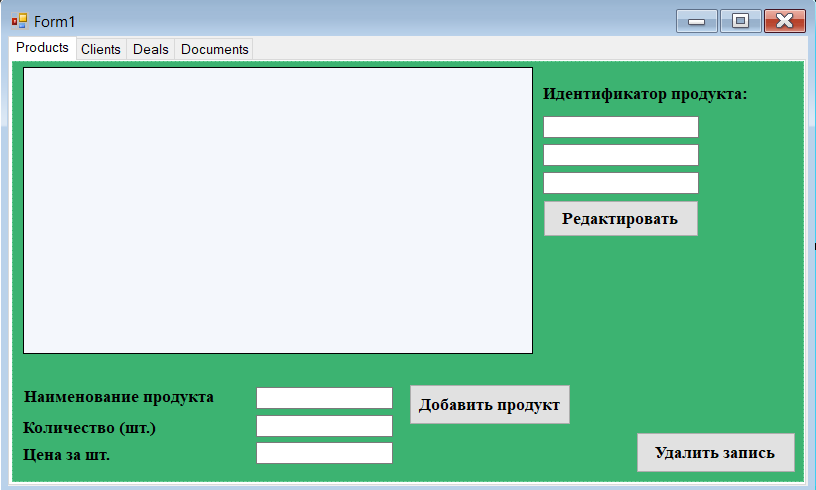


Рисунок 2 – Вкладка Products

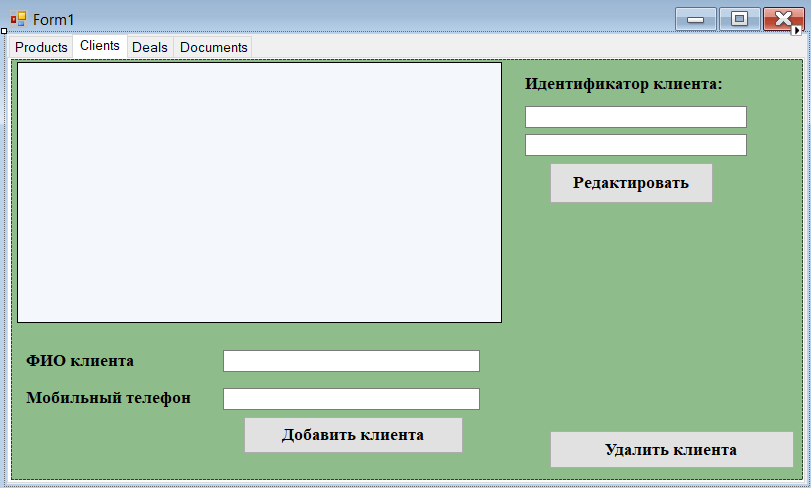


Рисунок 3 – Вкладка Clients

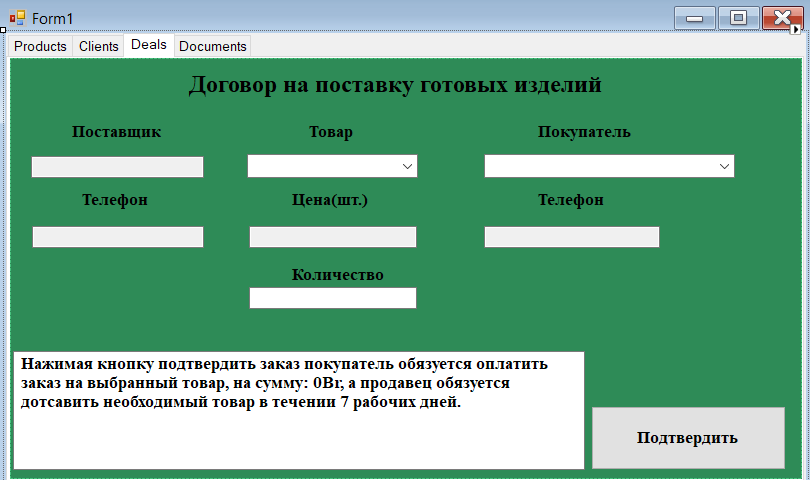


Рисунок 4 – Вкладка Deals

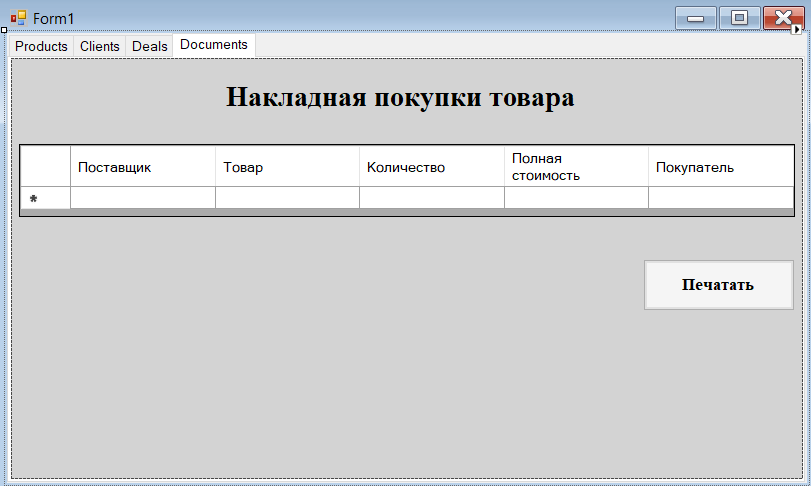


Рисунок 5 – Вкладка Documents

**2.3 Описание программы**

Данная программа работает посредством запросов к базе данных и последующим выводом информации в пользовательскую форму.

Вначале необходимо совершить подключение к базе данных.

Код подключения представлен ниже на листинге 1:

private void Form1\_Load(object sender, EventArgs e)

{

sqlConnection = new SqlConnection(ConfigurationManager.ConnectionStrings["DB"].ConnectionString);

sqlConnection.Open();

if(sqlConnection.State != ConnectionState.Open)

{

MessageBox.Show("DB NOT connected");

}

ShowProd("Product");

ShowClients();

ShowSeller();

ShowComboProd();

ShowComboClients();

}

Листинг 1 – Подключение БД

**2.4 Результаты работы программы**

При запуске программы, появляется её интерфейс. С его помощью осуществляется переход на другие вкладки. Код вкладки «Products» представлен на листинге 2:

private void button1\_Click(object sender, EventArgs e)

{

SqlCommand command = new SqlCommand(

$"INSERT INTO [Product] (Name,Amount,Cost) VALUES (N'{textBox1.Text}','{textBox2.Text}','{textBox3.Text}')",sqlConnection);

// MessageBox.Show("Добалена " + command.ExecuteNonQuery().ToString() + " Строка");

command.ExecuteNonQuery();

ShowProd("Product");

textBox1.Text = "";

textBox2.Text = "";

textBox3.Text = "";

}

private void listBox1\_SelectedIndexChanged(object sender, EventArgs e)

{

}

private void button2\_Click(object sender, EventArgs e)

{

DelProd();

}

private void dataGridView1\_MouseClick(object sender, MouseEventArgs e)

{

int id = Convert.ToInt32(dataGridView1.CurrentRow.Cells[0].Value.ToString());

label4.Text = "Идентификатор продукта: " + id.ToString();

textBox4.Text = dataGridView1.CurrentRow.Cells[1].Value.ToString();

textBox5.Text = dataGridView1.CurrentRow.Cells[2].Value.ToString();

textBox6.Text = dataGridView1.CurrentRow.Cells[3].Value.ToString();

/\* SqlCommand command = new SqlCommand($"SELECT \* FROM Product WHERE id={id}", sqlConnection);

textBox4.Text = command.ExecuteScalar().ToString();

\*/

// MessageBox.Show(SqlDataAdapter);

}

private void button3\_Click(object sender, EventArgs e)

{

int id = Convert.ToInt32(dataGridView1.CurrentRow.Cells[0].Value.ToString());

SqlCommand command = new SqlCommand(

$"UPDATE Product SET Name=N'{textBox4.Text}',Amount=N'{textBox5.Text}',Cost='{textBox6.Text}' WHERE id={id}", sqlConnection);

// MessageBox.Show("Добалена " + command.ExecuteNonQuery().ToString() + " Строка");

command.ExecuteNonQuery();

ShowProd("Product");

textBox4.Text = "";

textBox5.Text = "";

textBox6.Text = "";

}

private void tabPage1\_MouseClick(object sender, MouseEventArgs e)

{

textBox1.Text = "";

textBox2.Text = "";

textBox3.Text = "";

textBox4.Text = "";

textBox5.Text = "";

textBox6.Text = "";

}

Листинг 2 – Код вкладки «Products»

Аналогичный код имеет вкладка: «Clients».

Код добавления сделок представлен на листинге 3:

private void comboBox1\_SelectedIndexChanged(object sender, EventArgs e)

{

string selectedState = comboBox1.SelectedItem.ToString();

// MessageBox.Show(selectedState);

SqlDataReader dr2 = null;

SqlCommand cmdSel = new SqlCommand($"SELECT \* FROM Product WHERE Name=N'{selectedState}'", sqlConnection);

dr2 = cmdSel.ExecuteReader();

while (dr2.Read())

{

textBox13.Text= dr2["Cost"].ToString();

}

cmdSel.Dispose();

cmdSel = null;

dr2.Close();

}

private void comboBox2\_SelectedIndexChanged(object sender, EventArgs e)

{

string selectedState = comboBox2.SelectedItem.ToString();

int value;

int.TryParse(string.Join("", selectedState.Where(c => char.IsDigit(c))), out value);

// MessageBox.Show(value.ToString());

SqlDataReader dr2 = null;

SqlCommand cmdSel = new SqlCommand($"SELECT \* FROM Clients WHERE Id='{value}'", sqlConnection);

dr2 = cmdSel.ExecuteReader();

while (dr2.Read())

{

textBox14.Text = dr2["Tel"].ToString();

}

cmdSel.Dispose();

cmdSel = null;

dr2.Close();

}

private void textBox15\_TextChanged(object sender, EventArgs e)

{

// MessageBox.Show(textBox13.Text);

try

{

double a = Convert.ToDouble(textBox13.Text, System.Globalization.CultureInfo.InvariantCulture);

double b = Convert.ToDouble(textBox15.Text, System.Globalization.CultureInfo.InvariantCulture);

textBox16.Text = "Нажимая кнопку подтвердить заказ покупатель обязуется оплатить заказ на выбранный товар, на сумму: " + a \* b + "Br, " +

" а продавец обязуется дотсавить необходимый товар в течении 7 рабочих дней.";

}

catch (Exception)

{

textBox16.Text = "Нажимая кнопку подтвердить заказ покупатель обязуется оплатить заказ на выбранный товар, на сумму: " + 0 + "Br, " +

" а продавец обязуется дотсавить необходимый товар в течении 7 рабочих дней.";

}

}

Листинг 3 – Добавление сделок

Добавление товаров и клиентов реализовано с помощью выпадающего списка.

Код создания документов представлен на листинге 4:

private void button7\_Click(object sender, EventArgs e)

{

String post = textBox11.Text;

String tovar = comboBox1.Text;

double prise = 0;

try

{

double a = Convert.ToDouble(textBox13.Text, System.Globalization.CultureInfo.InvariantCulture);

double b = Convert.ToDouble(textBox15.Text, System.Globalization.CultureInfo.InvariantCulture);

prise = a \* b;

}

catch (Exception)

{

prise = 0;

}

String amount = textBox15.Text;

String pokup = comboBox2.Text;

dataGridView3.Rows[0].Cells[0].Value = post;

dataGridView3.Rows[0].Cells[1].Value = tovar;

dataGridView3.Rows[0].Cells[2].Value = amount;

dataGridView3.Rows[0].Cells[3].Value = prise;

dataGridView3.Rows[0].Cells[4].Value = pokup;

tabControl1.SelectedTab = tabControl1.TabPages["TabPage4"];

}

private void printDocument1\_PrintPage(object sender, System.Drawing.Printing.PrintPageEventArgs e)

{

Bitmap bmp = new Bitmap(dataGridView1.Size.Width + 10, dataGridView1.Size.Height + 10);

dataGridView1.DrawToBitmap(bmp, dataGridView1.Bounds);

e.Graphics.DrawImage(bmp, 0, 0);

}

private void button8\_Click(object sender, EventArgs e)

{

printDocument1.Print();

}

}

Листинг 4 – Создание документов

**3 Руководство пользователя**

Разработанная программа предназначена для наглядного отображения данных обо всех «единицах», фигурирующих в отделе сбыта, а также для автоматизации создания сделок и печати накладных для покупателей.

База данных интегрирована в среду разработки VisualStudio. После успешного подключения можно читать, записывать, удалять и редактировать данные через пользовательский интерфейс, разработанный с помощью интерфейса программирования приложений - WindowsForms.

Пользователю для получения доступа ко всем данным необходимо лишь перейти по одной из вкладок, например, если необходима информация о продуктах, перейти на вкладку «Products», рисунок 6.

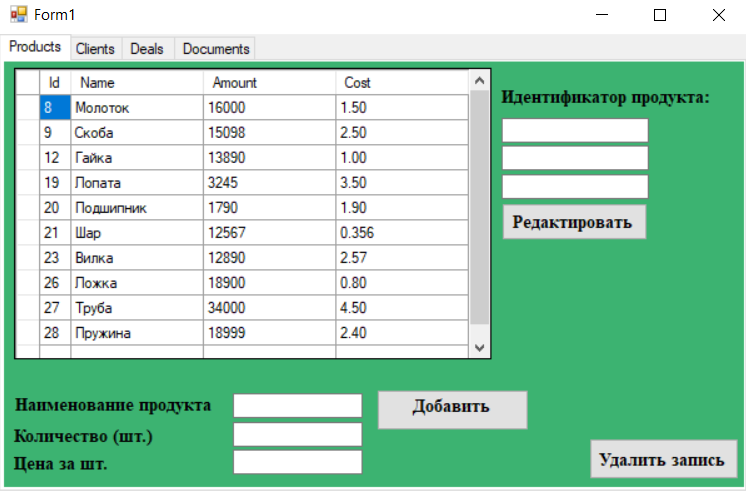


Рисунок 6 – Ассортимент выпускаемой продукции

Если требуется узнать о клиентах, переходим на вкладку «Clients», рисунок 7.

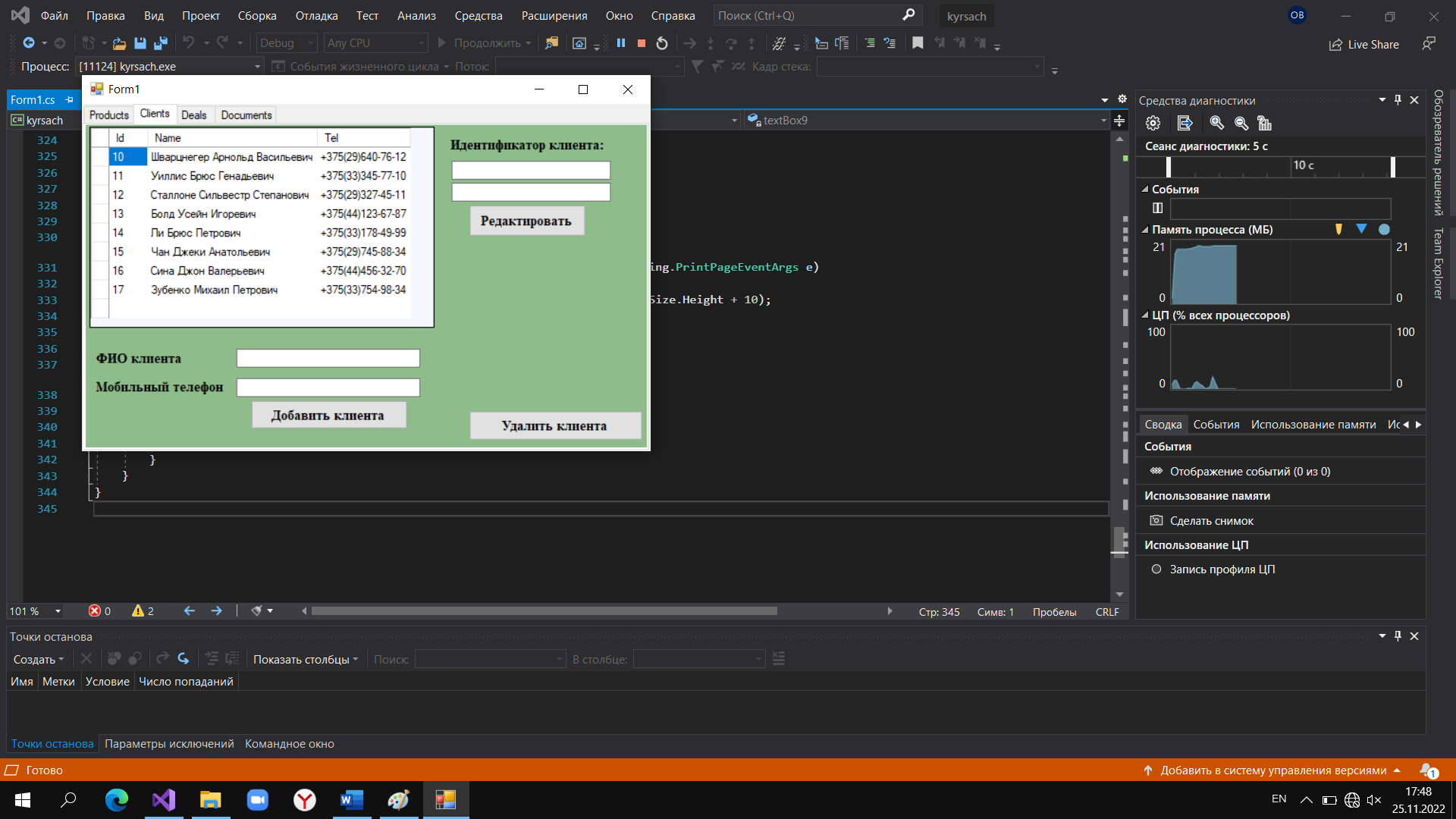


Рисунок 7 – Клиенты

Для создания договора о поставке изделий, необходимо перейти на вкладку «Deals», рисунок 8.

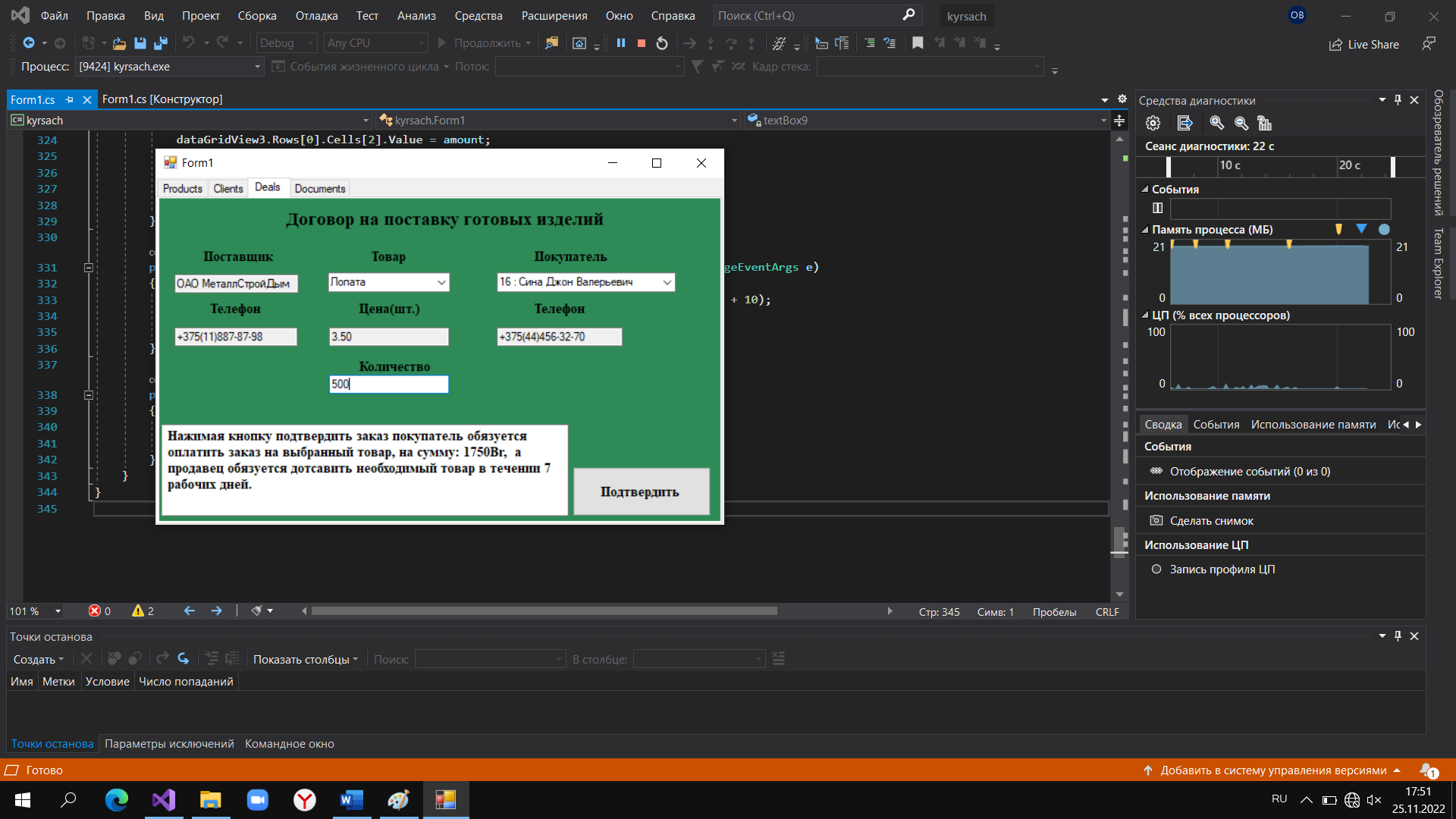


Рисунок 8 – Создание новой cделки

Для добавления информации в каждую из таблиц, предусмотрена форма добавления.

На листе графической части (1), представлена схема БД, отражающая таблицы с полями, включенные в таблицы поля, имеющие статус основного ключа и поля, имеющие статус внешнего ключа, для организации связей между таблицами. На листе (2), представлена «Диаграмма последовательностей», отражающая в динамике взаимодействие объектов информационной системы. На листе (3), представлена «Диаграмма состояний», которая представляет интерфейс пользователя и содержит список внутренних действий, выполняемых моделируемыми элементами системы при нахождении их в том или ином состоянии.

**Заключение**

Исходя из заданной цели курсового проекта, была разработана база данных «FactoryDB» с использованием программного продукта MySQL и языка программирования C#. В данной базе данных производится добавление и удаление информации о покупателях, поставщиках и товарах на заводе.

Была освоена актуальность темы, поставлены цели и задачи; дано назначение приложения.

Выбраны, обоснованы и кратко описаны средства программирования; описано требуемое оборудование и программное обеспечение.

Дано описание программы и логики, записанной в нее; представлен алгоритм решения.

Получены результаты работы программы с таблицей входных и выходных данных, а также с изображениями окон программы.

В графической части курсового проекта разработаны «Диаграмма сущность-связь», «Диаграмма последовательностей», «Диаграмма состояний».

**Список использованных источников**

1. Руководство по C#. [Электронный ресурс] – 10.11.2022– Режим доступа: https://proklondike.net/books/C#/manual.html

2. Руководство по работе с базами данных. [Электронный ресурс] – 10.11.2022. – Режим доступа: http://docplayer.ru/47935171-Rabota-s-bazami-dannyh.html

3. Основы работы с WindowsForms[Электронный ресурс] – 11.11.2022. – Режим доступа: https://web-programming.com.ua/osnovy-raboty-s-winforms/

4. СТП СМК 7.5.3 – 02.2020 Общие требования и правила оформления учебных текстовых документов. Могилев: МГУП, 2020. – 42 с