Федеральное казенное профессиональное образовательное учреждение

«Новокузнецкий государственный гуманитарно-технический колледж-интернат»

Министерства Труда и социальной защиты Российской Федерации

Специальность: 09.02.04 «Информационные системы (по отраслям)»

КУРСОВОЙ ПРОЕКТ

по дисциплине: МДК 02.02 Управление проектами

на тему: Разработка программного продукта, реализующего функционал ежедневника

РАСЧЁТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Студент                                                                               Уртеков.С.В.

дата                           личная подпись        расшифровка подписи

Руководитель проекта                                                       Грисман С.С.

дата                           личная подпись        расшифровка подписи

Норм контроль                                                                  Михайлов Н.Н,

дата                           личная подпись         расшифровка подписи

Оценка

Новокузнецк, 2020г.

**Оглавление**

[**ВВЕДЕНИЕ** 3](#_Toc58953988)

[**1. ТЕХНИЧЕСКОЕ ЗАДАНИЕ НА РАЗРАБОТКУ ПРОГРАММНОГО ПРОДУКТА** 4](#_Toc58953989)

[**2 ОБЩАЯ ЧАСТЬ** 5](#_Toc58953990)

[**2.1. Выбор и обоснование средств реализации базы данных** 5](#_Toc58953991)

[**2.2 Выбор и обоснование средств разработки программного продукта** 6](#_Toc58953992)

[**2.3 Анализ прототипов, существующих ИС** 7](#_Toc58953993)

[**3 СПЕЦИАЛЬНАЯ ЧАСТЬ** 10](#_Toc58953994)

[**3.1 Разработка алгоритма и проектирование структуры базы данных** 10](#_Toc58953995)

[**3.2 Разработка программного продукта** 11](#_Toc58953996)

[**3.3** **Руководство пользователя** 18](#_Toc58953997)

[**Заключение** 24](#_Toc58953998)

[**Приложение** 25](#_Toc58953999)

# **ВВЕДЕНИЕ**

Целью данного курсового проекта является, разработка программного продукта, реализующего функционал ежедневника с использованием, программного языка C# и СУБД MS SQL Server

Разрабатываемый программный продукт поможет эффективно использовать рабочее, и свободное время, а также хранение, выведение на экран, запланированных событий пользователю. В современном мире, огромное количество планов, целей, задач, которые необходимо решать в этот же день, либо через годы. Запомнить всё невозможно, а данный продукт с использование информационных технологий будет особенно актуален.

Выдвигается гипотеза, что программный продукт, реализующий функционал ежедневникас использованием СУБД MS SQL Server будет полезным для использования в повседневный жизни. Используя этот программный продукт, поможет сконцентрировать ваше внимание на достижения поставленных задач и целей. В результате, личностный рост, и успехи в трудовой деятельности.

Объектом в данной работе является программный продукт, реализующего функционал ежедневника.

Предметом является возможность оптимизации процессов, связанных с обработкой информации, регистрация незарегистрированных, и авторизация в системе уже зарегистрированных пользователей.

Для достижения поставленной цели необходимо выполнить следующие задачи:

* Разработать техническое задание;
* Проанализировать прототипы;
* Разработать программный продукт
* Провести тестирование и отладить программный продукт;
* Разработать руководство пользователя

Разрабатываемое приложение сможет применяться в повседневной жизни, для достижений поставленных целей и реализации запланированных задач в своё время, используя его с максимальной эффективностью.

# **1. ТЕХНИЧЕСКОЕ ЗАДАНИЕ НА РАЗРАБОТКУ ПРОГРАММНОГО ПРОДУКТА**

**Разработка программного продукта, реализующего функционал ежедневника**

Прототипом данного программного продукта является приложение с интернет ресурса: <https://www.leadertask.ru/content/view/481> имеющий функциональную часть работы данного приложения.

**Требования к функциональным характеристикам следующие:**

1. Необходимо предусмотреть режим работы пользователя. Вход пользователя организуется путем ввода логина и пароля.
2. В режиме пользователя должна быть предусмотрена возможность зарегистрироваться, заполнить свою анкету (ФИО, адрес, дату рождения, телефон, Email). Реализовать возможность, смены пароля и редактирование данных.
3. В интерфейсе приложения должен присутствовать календарь, а так = же ячейки для занесения данных пользователем на определённый день, и временем запланированным событием.
4. Предусмотреть возможность выводить определённое сообщение пользователю о начале события в предусмотренное время.
5. Организовать кнопку при, клике на которую форма с выведенной информацией о начале события закрывается и сообщение повторяется через определенное время, либо не повторяется.

**Требования к информационному обеспечению следующие:**

К приложению должна быть предоставлена документация содержащая техническое описание программного продукта и руководство пользователя.

**Требование к лингвистическому обеспечению следующие:**

Приложение должно иметь интерфейс и документацию на русском языке.

**Требования к программному обеспечению следующие:**

Разработка программного продукта предусматривается под OS Windows не ниже версии 7.

**Требования к техническому обеспечению следующие:**

Для работы приложения требуется MS SQL Server, Visual Studio С#. Также необходимо не менее 2Гб оперативной памяти и 100Мб свободного места на жестком диске.

# **2 ОБЩАЯ ЧАСТЬ**

## **2.1. Выбор и обоснование средств реализации базы данных**

Неотъемлемой частью разработки приложения и его адекватного функционирования, хранение, а также возврата необходимых данных, работала СУБД MS SQL Server.

База данных MS SQL Server это упорядоченная информация, связанная между собой определёнными отношениями. Представлена она в виде таблиц, в которых и лежит вся эта информация.

**СУБД**Microsoft SQL Server – это система управления базами данных от компании Microsoft. Она очень популярна в корпоративном секторе, особенно в крупных компаниях. И это не просто СУБД – это целый комплекс приложений, позволяющий хранить и модифицировать данные, анализировать их, осуществлять безопасность этих данных и многое другое.

SQL сервер - 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. Также если размер вашей базы данных стремиться к гигабайту и продолжает увеличиваться, то SQL сервер единственно возможный метод обеспечения ее функционирования.

MS SQL Server не предназначен непосредственно для разработки пользовательских приложений, а выполняет функции управления базой данных. Сервер имеет средства удаленного администрирования и управления операциями, организованные на базе объектно-ориентированной распределенной среды управления.

Одним из преимуществ SQL Server является простота его применения, в частности администрирования. SQL Server Enterprise Manager, входящий в состав всех редакций Microsoft SQL Server (за исключением MSDE), представляет собой полнофункциональное и достаточно простое средство для администрирования этой СУБД.

При написании приложения был выбран язык программирования C#.

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

При проектировании приложения были использованы такие программные средства как MS SQL Server, Visual Studio, MS Excel.

Фундамент любого программного продукта формируется с помощью база данных. В нашем случаи, наилучшим вариантом стал SQL Server. Он полностью отвечает всем требованиям.

При выполнении данной работы была выбрана интегрированная среда разработки Visual Studio (IntegratedDevelopmentEnvironment - IDE)

* Легко просматривайте структуру объектов в коде с помощью обзора внутренних объектов в обозревателе решений и быстро находите файлы в своем решении.
* Щелкните ссылку на код или нажмите клавишу F1, выделив ошибку, чтобы выполнить поиск содержимого в Интернете, которое поможет устранить выявленные проблемы.
* Visual Studio помогает при написании кода, вне зависимости от используемого языка, от C#, VB и C++ до JavaScript и Python, предоставляя помощь в реальном времени.
* Более высокая скорость разработки. Многие из функциональных возможностей Visual Studio направлены на то, чтобы помогать разработчику делать свою работу как можно быстрее.
* Удобные функции, вроде функции IntelliSense (которая умеет перехватывать ошибки и предлагать правильные варианты).
* Функции поиска и замены (которая позволяет отыскивать ключевые слова как в одном файле, так и во всем проекте).
* Функции автоматического добавления и удаления комментариев (которая может временно скрывать блоки кода), позволяют разработчику работать быстро и эффективно.

Microsoft Excel - это прикладная программа, которая предназначена для создания электронных таблиц и автоматизированной обработки табличных данных (выполнения сложных вычислений, построения диаграмм, статистической обработки данных, обработки данных в списках, решение задач оптимизации). Microsoft Excel входит в состав Microsoft Office и на сегодняшний день Excel является одним из наиболее популярных приложений в мире.

## **2.3 Анализ прототипов, существующих ИС**

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;
* структура программного продукта и удобство использования;
* навигация в приложении;

объем и актуальность представленной информации в данном программном продукте;

* удобство использования;

При проектировании приложения реализующего функционал ежедневника прототипом стали Web приложение «Онлайн органайзер планировщик ежедневник»

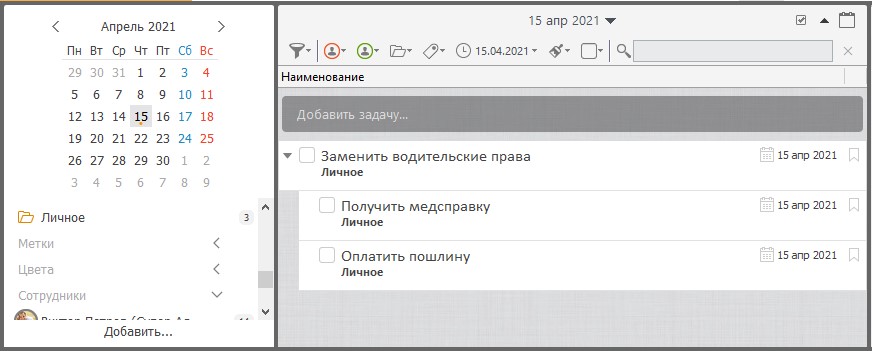
Ознакомится и скачать это приложение можно по ссылке: <https://www.leadertask.ru/content/view/481> Очень удобная возможность, состояние списка дел на любой день в будущем. Есть задачи, которые нужно выполнить через продолжительный период времени. Например, обновить водительское удостоверение. Программа должна предоставлять возможность выбрать любой день, даже в следующем году, и составить в нём список задач, представлено на рисунке 1

Рисунок 1 – «интерфейс приложения».

Прикрепление файлов: фотографий, документов, медиа файлов. Такая возможность пригодиться для того, чтобы в одном месте всю информацию задаче. Рисунок 2.

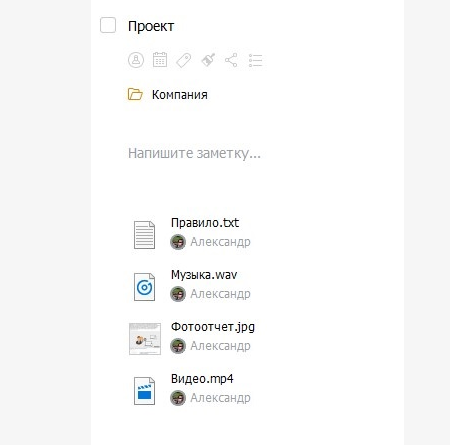


 Рисунок 2 – «прикреплённые файлы к определённым событиям».

Составление почасового расписания, если в программе реализована временная школа, как в бумажном ежедневнике, то вы сможете создать наглядное почасовое расписание на день. Представлено на рисунке 3.

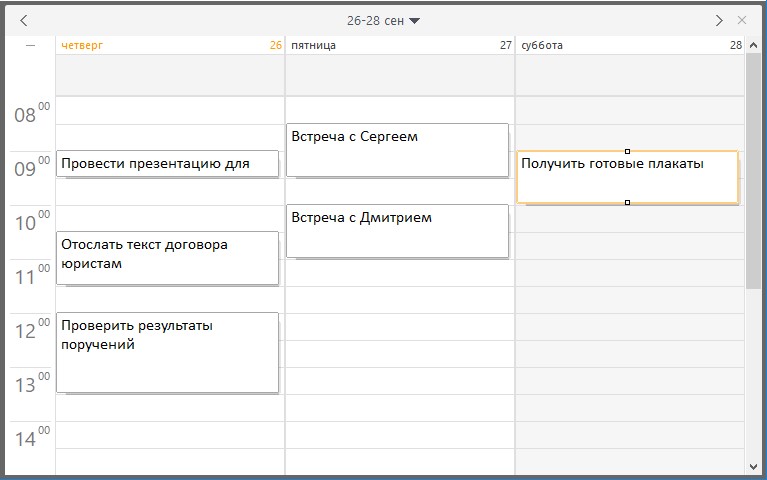


Рисунок 3 – «почасовое расписание на день».

Некоторые дела должны быть привязаны к конкретному времени. Яркий пример – встречи. Программа не позволит вам опоздать, если за несколько минут до начала покажет напоминание.

# **3 СПЕЦИАЛЬНАЯ ЧАСТЬ**

## **3.1 Разработка алгоритма и проектирование структуры базы данных**

База данных программного приложения реализующего функционал ежедневника сформирована на платформе MS. SQL Server. Графическая и физическая модель таблиц, базы данных, применялся режим конструктора SQL Server, где и были сформированы свойства полей всех таблиц базы данных программного приложения.

Данные представляемого электронного продукта «Ежедневник» расположены в трёх таблицах, связанных между собой, обладающие особой структурой:

* «Category» (Категории)
* «Everybay» (События)
* «Users» (Пользователи)

Таблица «Category» имеет связь с таблицей «Everybay» предназначена для хранения и отображения для какой категории относится планируемое событие. Ключевым полем является «id\_categ» (Код категрии), тип данных int.Так же имеются поле «name\_cated» (Наименование категории) – nchar(10).

Таблица «Everybay» имеет ключевое поле «1d\_ever» (Уникальный номер события), тип данных int. Связь с таблицей «Users» осуществляется по полю Users» «id\_user» (Идентификатор пользователя), тип данных int. Так же существует связь с таблицей «Category» по полю «id\_categ» (Код категории), тип данных int, является, внешнем ключом. Следующие строки в таблице «Everybay»: «task» (Задачи) – nchar(100), «Data» (Дата назначения задачи) – datetime, «status» (Флаг, да\нет) – bit.

Третья таблица«Users» (Данные о пользователе),  ключевое поле «Id\_User» (Идентификатор пользователя) – int, следующие поля таблицы: «Phone» (Номер телефона пользователя) – nchar(50), «Lastname» (Фамилия ) - , «Name» (Имя) – nchar(50) , «Surname» (Отчество) – nchar(50), «Email» (Электронный адрес) – nchar(50) , «Password» (Пароль) -  nchar(50).

Хранение личной информации, обеспечение целостность данных, достигается с помощью функционала базы данных. СУБД автоматически, с помощью ключевых полей и связей, один ко многим, достигается возможность каскадного обновления и удаления информации. Диаграмма база данных, со связями, рисунок 4, представлен ниже по тексту.

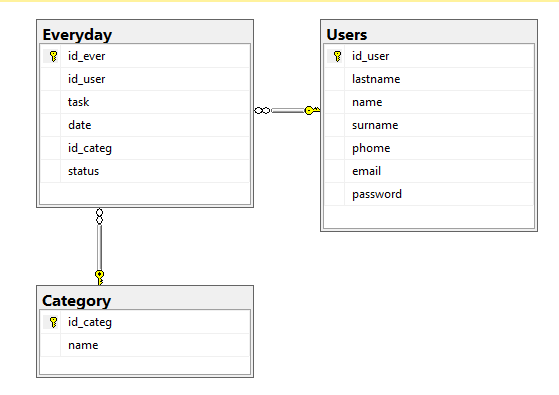


Рисунок 4 – «Схема базы данных со связями, один ко многим».

**3.2 Разработка программного продукта**

Разрабатывая данный, «Ежедневник», программный продукт, реализовано несколько подсистем. Ниже представлен исходный код подсистем.

Авторизация пользователя – первая подсистема, осуществляет проверку на наличие пользователя в базе данных, сверяет идентичность пароля и номера телефона, с данными в базе. Код файла программы представлен на листе 1.

Листинг 1 - авторизация

string s = "select \* from Users where Phone like '%" + maskedTextBox1.Text.Trim()

+ "%' and Password like '%" + textBox\_Password.Text.Trim() + "%'";

Program.connecti.Open();

SqlCommand sql2 = new SqlCommand(s, Program.connecti);

SqlDataReader reader = sql2.ExecuteReader();

if (!reader.HasRows)

{

MessageBox.Show("Неверный логин или пароль");

maskedTextBox1.Focus();

Program.connecti.Close();

}

else

{

reader.Read();

Program.login\_user = (Convert.ToString(reader["Phone"])).Trim();

Program.Users\_id = Convert.ToInt32(reader.GetValue(0));

Program.connecti.Close();

В случаи соответствия введённого номера телефона и пароля с данными в базе данных, подсистема открывает третью форму – «Личный кабинет пользователя». Код файла программы представлен на листе 2.

Листинг – открывает 3 форму, связывает с SQL Server

string s1 = "select \* from Users where Phone like '%" + Program.login\_user+" %' ";

Program.connecti.Open();

SqlCommand sql1 = new SqlCommand(s1, Program.connecti);

SqlDataReader reader1 = sql1.ExecuteReader();

reader1.Read();

this.Visible=false;

Form3 f = new Form3();

f.label2.Text = reader1.GetValue(2).ToString().Trim().Trim() + " " + reader1.GetValue(3).ToString().Trim() + " " + reader1.GetValue(4).ToString().Trim();

f.label3.Text = reader1.GetValue(1).ToString();

f.label5.Text = reader1.GetValue(5).ToString();

Program.connecti.Close();

f.Show();

Кнопка «Регистрация», реализует открытие второй формы, связывает текст-боксы с данными SQL Server, где предоставляется возможность зарегистрироваться новому пользователю. Код файла на листинги 3 -

Листинг 4– регистрация пользователя

Form2 fo2 = new Form2();

if (fo2.ShowDialog() == DialogResult.OK)

{

Program.connecti.Open();

string s2 = "insert into Users(Phone, " +

"Lastname, Name, Surname, Email, Password) values('" +

fo2.maskedTextBox1.Text

+ "', '" + fo2.textBox1.Text.Trim()

+ "', '" + fo2.textBox2.Text.Trim()

+ "', '" + fo2.textBox3.Text.Trim()

+ "', '" + fo2.textBox4.Text.Trim()

+ "', '" + fo2.textBox5.Text.Trim() + "') ";

SqlCommand sql2 = new SqlCommand(s2, Program.connecti);

sql2.ExecuteNonQuery();

MessageBox.Show("Вы успешно зарегистрировались!");

Program.connecti.Close();

Вторая форма «Регистрация пользователя», новый пользователь заносит свои личные данные и создаёт индивидуальный код доступа. При клике на кнопку «Регистрация» срабатывает код файла, который проверяет наличие записей в строках ввода текста и подтверждения пароля. Сверяет занесённые данные с зарегистрированными пользователями в базе данных, не допускает двойников. Код файла представлен на листинг 5

Листинг 5 – проверяет заполнение полей, идентичность, пароля и подтверждение, уникальности номера телефона и пароля

if (this.DialogResult == DialogResult.OK)

{

int k = 0;

try

{

if (textBox1.Text.Trim() == "")

{

throw new Exception

("Строка Фамилия, не может быть пустой!");

}

k++;

if (textBox2.Text.Trim() == "")

{

throw new Exception

("Строка Имя, не может быть пустой!");

}

k++;

if (textBox3.Text.Trim() == "")

{

throw new Exception

("Строка Отчество, не может быть пустой!");

}

k++;

if (maskedTextBox1.Text.Trim() == "")

{

throw new Exception

("Строка Номер телефона, не может быть пустой!");

}

k++;

if (textBox4.Text.Trim() == "")

{

throw new Exception

("Строка Email, не может быть пустой!");

}

else if (textBox5.Text.Trim() == "")

{

throw new Exception

("Строка Пароль, не может быть пустой!");

}

k++;

if (textBox2.Text.Trim() == "")

{

throw new Exception

("Подтвердите пароль!");

}

if (textBox5.Text.Trim() != textBox6.Text.Trim()){

throw new Exception

("Пароли не совпадают!!");

}

Program.connecti.Open();

string s = "select \* from Users where Phone='" + maskedTextBox1.Text.Trim() + "' ";

SqlCommand sql = new SqlCommand(s, Program.connecti);

SqlDataReader reader = sql.ExecuteReader();

if (reader.HasRows)

{

Program.connecti.Close();

throw new Exception

("Данный пользователь уже зарегистрирован");

}

Program.connecti.Close();

}

catch (Exception E)

{

if (k == 0)

{

textBox1.Focus();

}

else if (k == 1)

{

textBox2.Focus();

}

else if (k == 2)

{

textBox3.Focus();

}

else if (k == 3)

{

maskedTextBox1.Focus();

}

else if (k == 4)

{

textBox4.Focus();

}

else if (k == 5)

{

textBox5.Focus();

}

else if (k == 6)

{

textBox6.Focus();

}

MessageBox.Show(E.Message, "Ошибка!");

e.Cancel= true;

}

Форма 3 – «Личный кабинет», включает в себя основной функционал приложения «Ежедневник», возможности программного продукта реализованы с помощью файловых кодов:

Редактировать данные – кнопка на форме, активирует программный код, с помощью которого редактируются личные данные пользовав в базе данных SQL Server. Код файла представлен на листинги 6.

Листинг 6 – редактор личных данных пользователя

string s2 = "select \* from Users where Phone like '%" + Program.login\_user + "% '";

Program.connecti.Open();

SqlCommand sql2 = new SqlCommand(s2, Program.connecti);

SqlDataReader reader = sql2.ExecuteReader();

reader.Read();

Form5 f1 = new Form5();

f1.textBox4.Text = reader.GetValue(2).ToString().Trim();

f1.textBox5.Text = reader.GetValue(3).ToString().Trim();

f1.textBox5.Text = reader.GetValue(4).ToString().Trim();

f1.maskedTextBox1.Text = reader.GetValue(1).ToString().Trim();

f1.textBox6.Text = reader.GetValue(5).ToString().Trim();

Program.connecti.Close();

if (f1.ShowDialog() == DialogResult.OK)

{

string s3 = "select \* from Users where Phone='" + f1.maskedTextBox1.Text.Trim() + "'";

Program.connecti.Open();

SqlCommand sql3 = new SqlCommand(s3, Program.connecti);

SqlDataReader reader1 = sql3.ExecuteReader();

if(reader1.HasRows && f1.maskedTextBox1.Text.Trim()!=Program.login\_user)

{

MessageBox.Show("Данный номер телефона уже имеется в БД");

Program.connecti.Close();

}

else

{

Program.connecti.Close();

string s = "update Users set Lastname='" + f1.textBox4.Text.Trim() +

"', Name='"+f1.textBox5.Text.Trim() +

"', Surname='" +f1.textBox6.Text.Trim() +

"',Email='"+f1.textBox1.Text.Trim() + "', Phone='"+f1.maskedTextBox1.Text.Trim() +

"' where Phone='" + Program.login\_user.Trim() + "'";

Program.connecti.Open();

SqlCommand sql = new SqlCommand(s, Program.connecti);

sql.ExecuteNonQuery();

Program.connecti.Close();

MessageBox.Show("Изменения сохранены!");

С помощью следящего программного кода, реализована возможность поменять пароль, кнопка «Сменить пароль». Код файла представлен, листинг 6.

Листинг 7

Form4 f = new Form4();

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

{

string s = "select Password from Users Where Phone='" + Program.login\_user + "'";

Program.connecti.Open();

SqlCommand sql1 = new SqlCommand(s, Program.connecti);

SqlDataReader reader1 = sql1.ExecuteReader();

reader1.Read();

string old\_pass = reader1.GetValue(0).ToString().Trim();

Program.connecti.Close();

if (old\_pass != f.textBox1.Text.Trim())

{

MessageBox.Show("Старый пароль введён не верно! ");

}

else

{

if(f.textBox2.Text.Trim() != f.textBox3.Text.Trim())

{

MessageBox.Show(" Пароли не совпадают ");

}

else

{

string s1 = "update Users set Password= '" + f.textBox2.Text.Trim() + "' where Phone = '" + Program.login\_user + "'";

Program.connecti.Open();

SqlCommand sql = new SqlCommand(s1, Program.connecti);

/\*SqlDataReader reader = sql.ExecuteReader();

reader1.Read();\*/

sql.ExecuteNonQuery();

Program.connecti.Close();

MessageBox.Show(" Пароль успешно изменён! ");

Program.login\_user = "";

this.Close();

Чтобы это приложение имело смысл, в базу данных необходимо вносить информацию о событии. Кнопка «Добавить событие» запускает работу программного кода. Код файла представлен листинг 7.

Листинг 8

Form6 f = new Form6();

f.Text = "Добавить События";

//this.everybayTableAdapter.Fill(this.diary1DataSet.Everybay);

f.categoryTableAdapter1.Fill(f.diary1DataSet3.Category);

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

{

string id\_ka = f.comboBox1.SelectedValue.ToString();

string sb = f.textBox2.Text.Trim();

string da = f.dateTimePicker1.Value.ToString().Trim() ;

//string st = f.comboBox1.SelectedValue.ToString();//

string s = "insert into Everybay(id\_user,task,id\_categ,  Data, STATUS ) values("+ Program.Users\_id+",'" + sb + "','" + id\_ka + "','" + da + "',0)";

Program.connecti.Open();

SqlCommand sql = new SqlCommand(s, Program.connecti);

sql.ExecuteNonQuery();

Program.connecti.Close();

load();

MessageBox.Show("Запись успешно добавлена!!");

Так же реализована в приложении «Ежедневник» интересная опция формирование и визуализация событий, за определённый промежуток времени. На форме 3 установлены два элемента «DateTime d1, DateTime d2» - старт с какой даты, финиш по какую дату выбирать события для визуализации. Кнопка «Сформировать за период» активирует код на выборку данных из SQL Server за определённый промежуток времени и выводит результат на форму 3 «dataGridView2» Код файла представлен, листинг 7.

Листинг 8

DateTime d1 = dateTimePicker1.Value;

DateTime d2 = dateTimePicker2.Value;

dataGridView2.RowCount = 1;

string s4 = "select id\_User, Lastname, Name, Surname  from Users where Phone = '"

+ Program.login\_user.Trim() + "'";

Program.connecti.Open();

SqlCommand sql3 = new SqlCommand(s4, Program.connecti);

SqlDataReader reader3 = sql3.ExecuteReader();

reader3.Read();

string id = reader3.GetValue(0).ToString().Trim();

string l = reader3.GetValue(1).ToString().Trim();

string n = reader3.GetValue(2).ToString().Trim();

string su = reader3.GetValue(3).ToString().Trim();

this.Text = l + "" + n + "" + su;

reader3.Close();

Program.connecti.Close();

string s5 = "select \* from [dbo].[Everybay] t1 " +

"inner join[dbo].[Category] t2" +

"on t1.[id\_categ] = t2.[id\_categ]" +

" where id\_user = "+id+" and t1.Data >= '"+ d1 + "' and t1.Data <= '"

+ d2 + "'";

Program.connecti.Open();

SqlCommand sql4 = new SqlCommand(s5, Program.connecti);

SqlDataReader reader4 = sql4.ExecuteReader();

while (reader4.Read())

{

string[] str = new string[6];

str[0] = reader4.GetValue(0).ToString().Trim();

str[1] = reader4.GetValue(3).ToString().Trim();

str[2] = reader4.GetValue(4).ToString().Trim();

str[3] = reader4.GetValue(7).ToString().Trim();

str[4] = reader4.GetValue(.ToString().Trim();

str[5] = reader4.GetValue(6).ToString().Trim();

dataGridView2.Rows.Add(str);

 }

reader4.Close();

Program.connecti.Close();

Реализована возможность: на форме 3 кнопка «Редактировать события» редактировать уже созданные цели, а также просмотр ближайших событий, при клике одноимённой кнопки, к актуальной дате, автоматический вывод формы «Горячие событие» на дисплей по времени запланированной цели, кнопка «Выгрузка событий в Excel», присутствует checkBox1- флаг «ОК уведомление» разрешает либо запрещает автоматическое появление формы «Горячие событие», checkBox1 на форме «Редактор событий » определяет статус события – true/false, имеется кнопка «Удалить событие» удаляет выделенную цель, кнопка «Выход» закрывает форму 3 «Личный кабинет» и открывается 1 форма «Авторизация».

* 1. **Руководство пользователя**

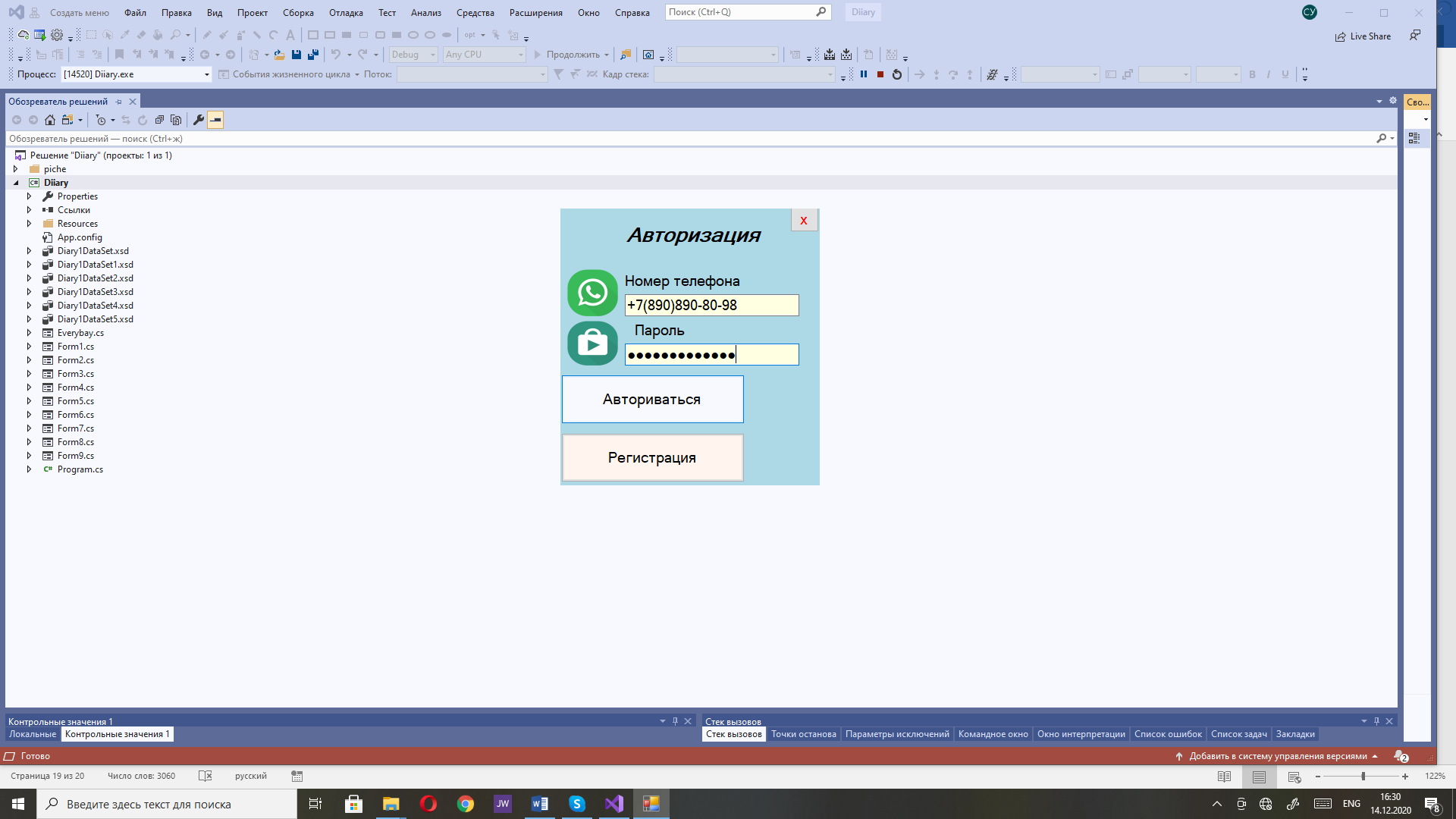
Запустив приложение, откроется форма авторизация пользователя. Введите свой номер телефона и пароль для входа в личный кабинет. Если же у вас нет пока своего аккаунта, можно создать его нажав», представлен на рисунке 5. 

Рисунок 5 – «Форма авторизации»

При клике на кнопку «Регистрация» открывается форма, где предлагается зарегистрироваться, заполнив строки своими данными: «Фамилия», «Имя», «Отчество», «Номер телефона», «email», придумать личный пароль «Пароль» и повторить его, для подтверждения. Когда все строки заполнены, пустые строки будут вызывать ошибку, кликаем на кнопку «Регистрация», поздравляю вас с успешной регистрации в системе «Ежедневник». Рисунок 6 представлен ниже по тексту.

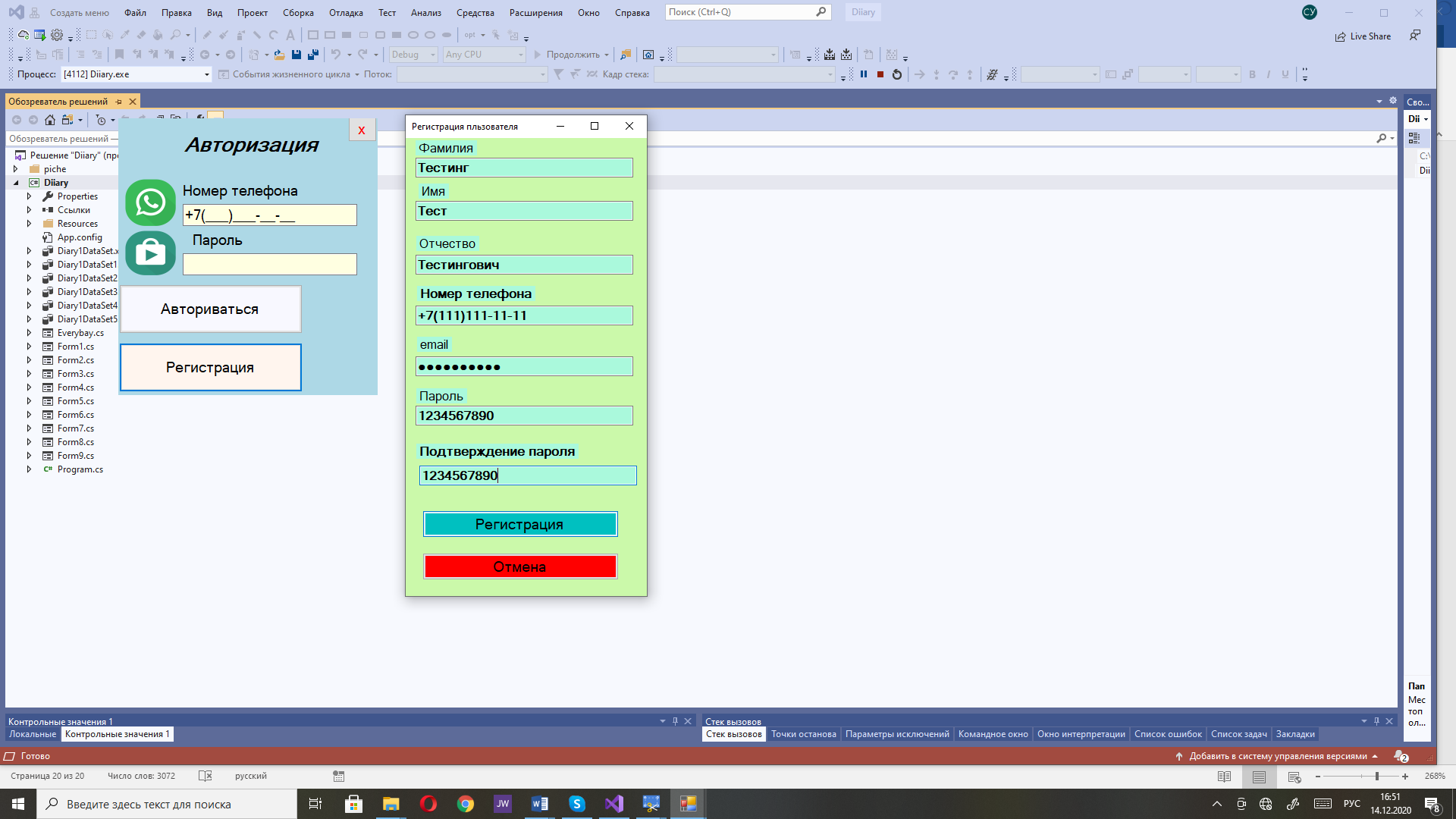


Рисунок 6 – «Регистрация»

Вернулись на форму «Авторизация», вводим, номер телефона и пароль, которые вы записывали при регистрации, в одноимённые поля формы. Кликаем на кнопку «Авторизоваться» и тогда открывается ваш личный кабинет, где вы можете манипулировать своей информацией, на своё усмотрение. Рисунок 7

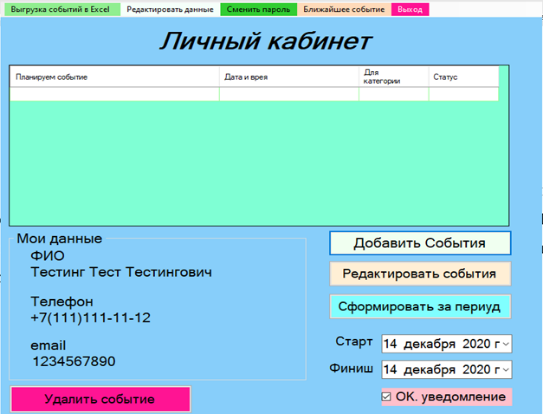


Рисунок 7 – «Личный кабинет»

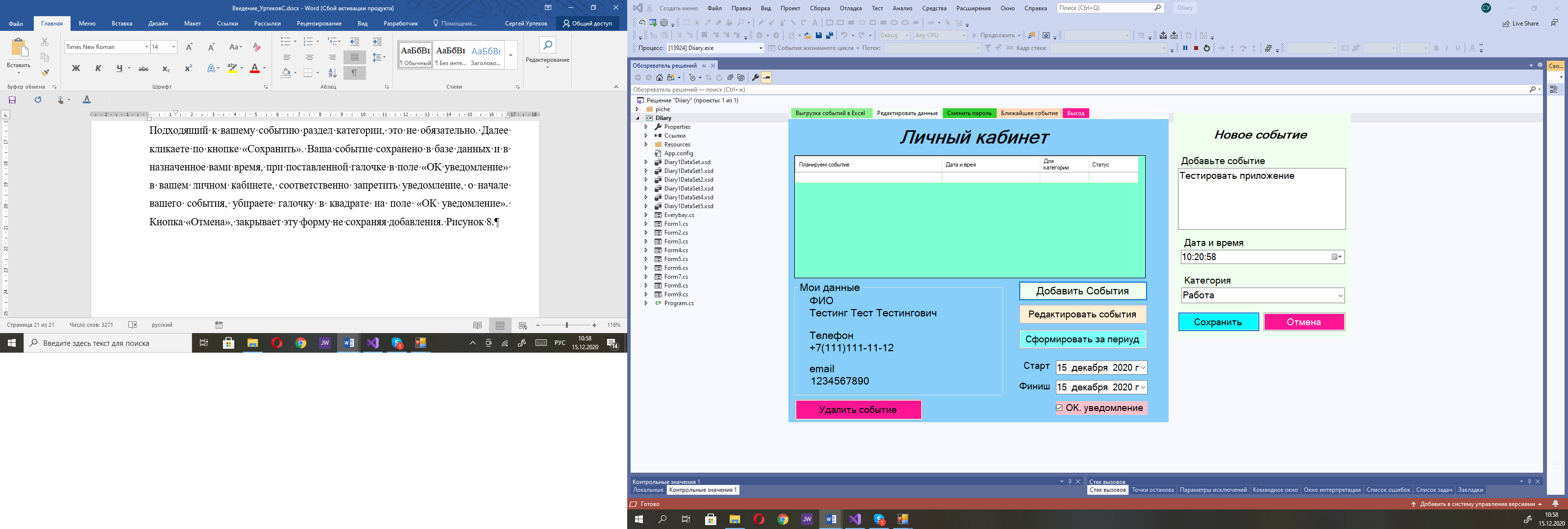
Добавляем событие, кликаем по кнопке «Добавить событие» активируется форма «Новое событие», в поле добавьте событие, записываем свои цели, напоминания, и так далее. Ниже, в строке, дата и время, назначаем на какое время и дату вы панируете своё событие. В поле, категория, можно выбрать из выпадающего списка: Семья, работа, друзья, хобби, разное, Подходящий к вашему событию раздел категории, это не обязательно. Далее кликаете по кнопке «Сохранить». Ваша событие сохранено в базе данных Кнопка «Отмена», закрывает эту форму не сохраняя добавления. Рисунок 8. 

Рисунок 8 – «Добавить новое событие»

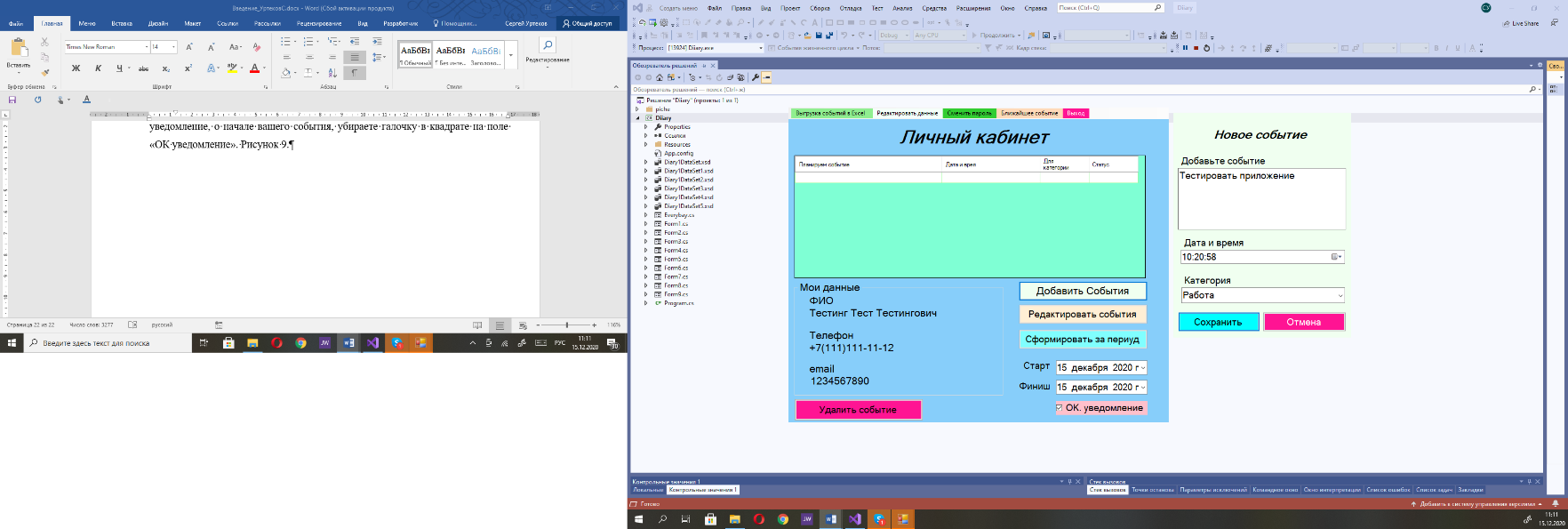
В назначенное вами время, при поставленной галочке в поле «ОК уведомление» в вашем личном кабинете, будет появляться форма, сообщающая о начале события. Соответственно запретить уведомление, о начале вашего события, убираете галочку в квадрате на поле «ОК уведомление». Рисунок 9 изображён правый нижний угол, формы «Личный кабине».

Рисунок 9 – «ОК уведомление, галочка поставлена по умолчанию».

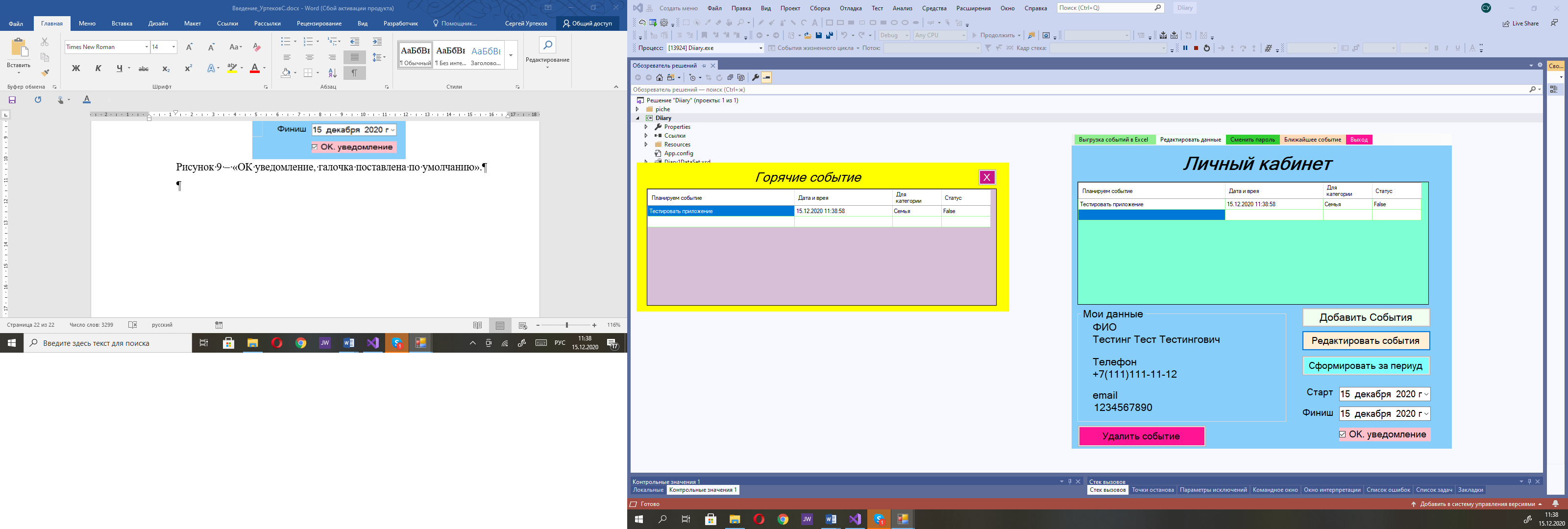
Форма уведомление «Горячие событие» напоминает, что время запланированной цели подошло. Рисунок 10. 

Рисунок 10 – «Автоматически всплывающая форма.»

Так же реализована возможность редактировать, текстовою часть, дату и время, категорию и статус, вашего события. Кликнув по кнопке «Редактировать событие» откроется одноименная форма, где и производится редактирование. Кнопка «Редактировать» подтверждает изменения, «Отмена» отменяет изменения и закрывает форму. Рисунок 11.

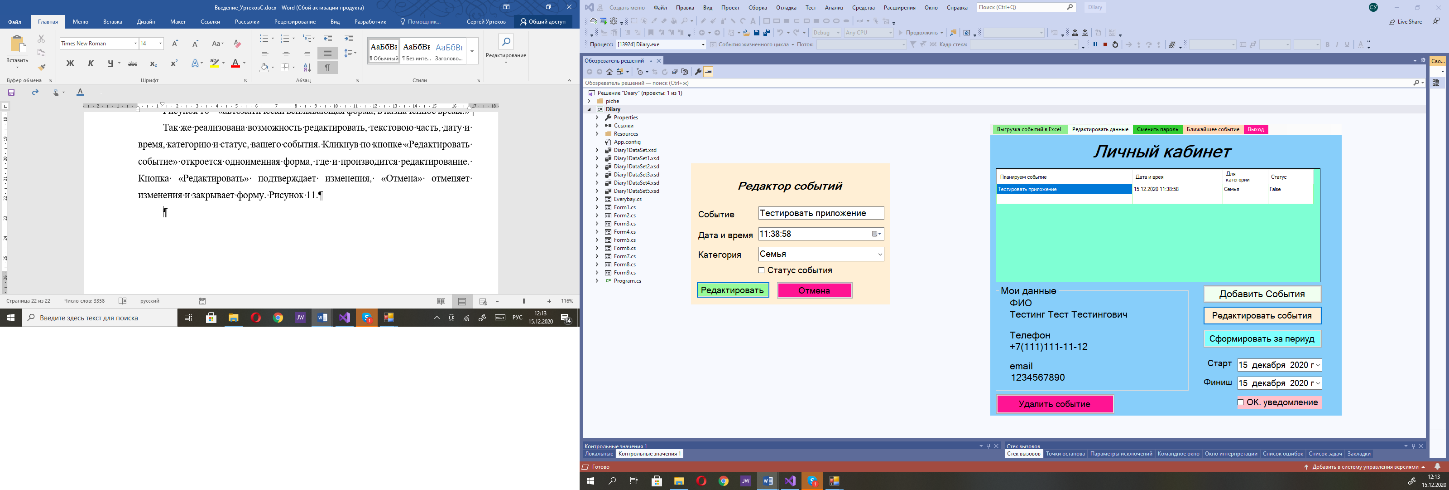


Рисунок 11- «Редактор событий»

Функция, сформировать обзор за определённый промежуток вренени, в личном кабинете, в нижнем правом углу на против слова старт, установить в поле дату, от которой вы хототе сформировать обзор прошедших событий.На против слоова фониш, до какой даты вы хотите сформировать обзор, далие кликните по кнопке «Сформировать за периуд» обновится таблица, в личном кабинете, до обзора,с сформированными вами событиями за определённый промпжцток времяни. Рисунок 12, 13.

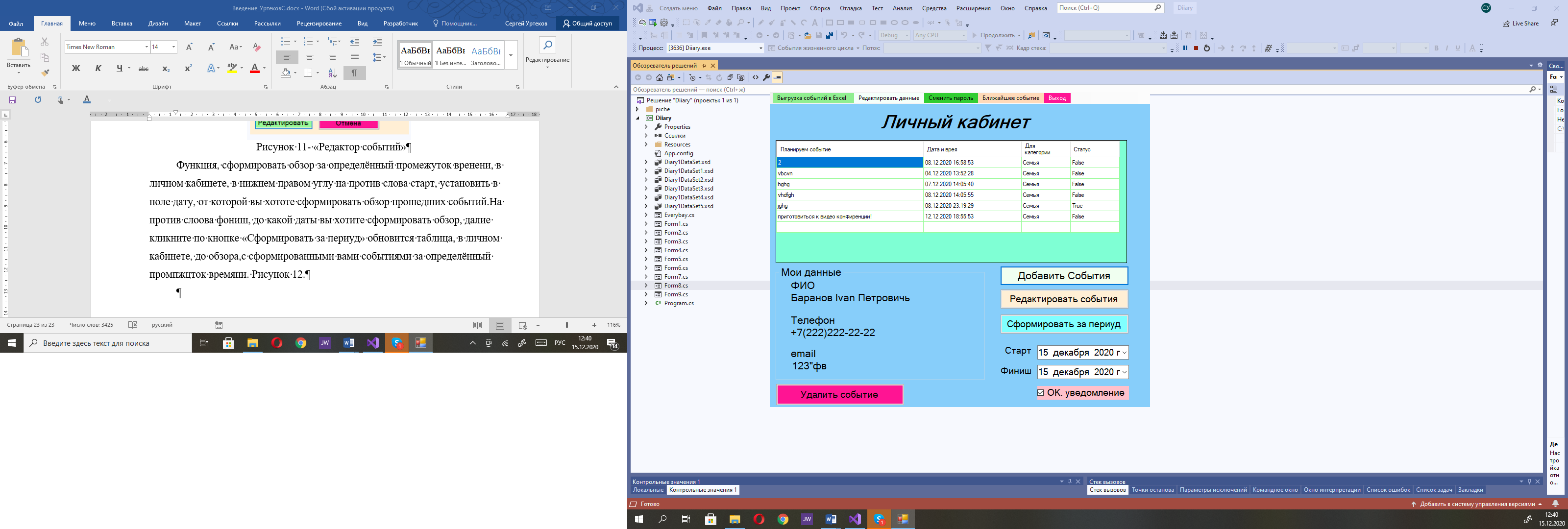
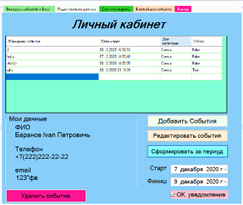


Рисунок 12, 13 – «С лева 12 рисунок сформированное событие, 13 полный список.»

В верхней части, «Личный кабинет», кнопка «Ближайшее событие», при клике по ней, открывается одноимённая форма, где выведены ближайшие события относительно актуального времени. Рисунок 14, расположение на форме кнопки, «Ближайшее событие», рисунок 15 вывод форы с результатом.

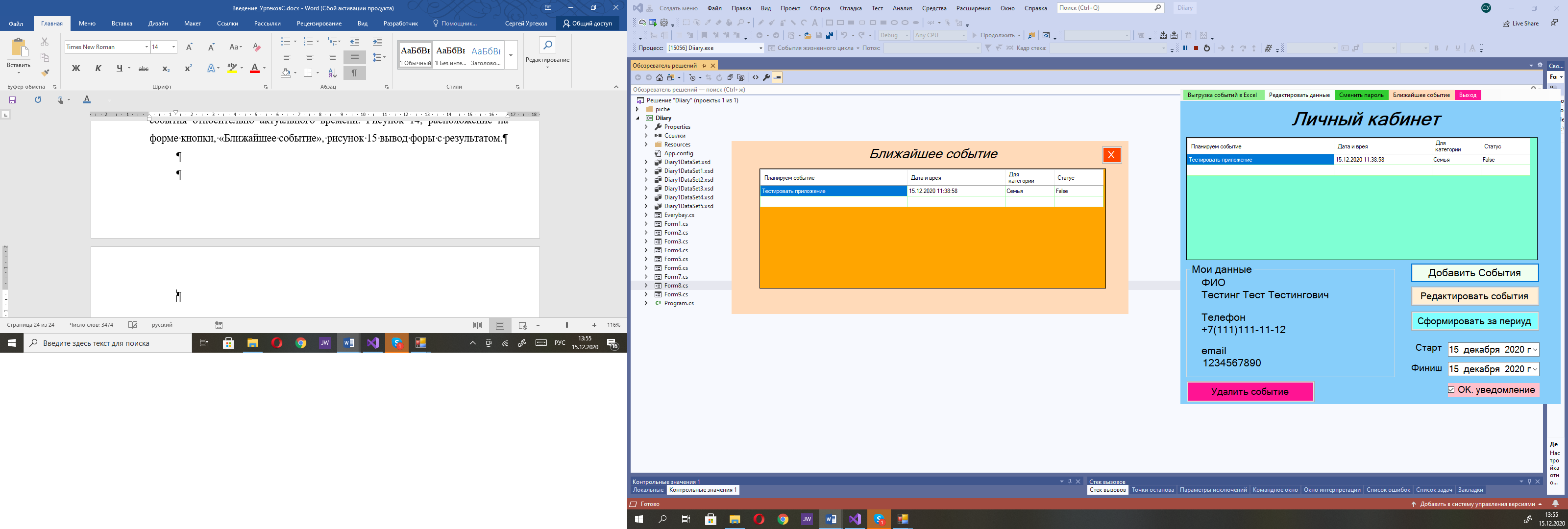


Рисунок 14 – «Расположение кнопки Ближайшее событие»

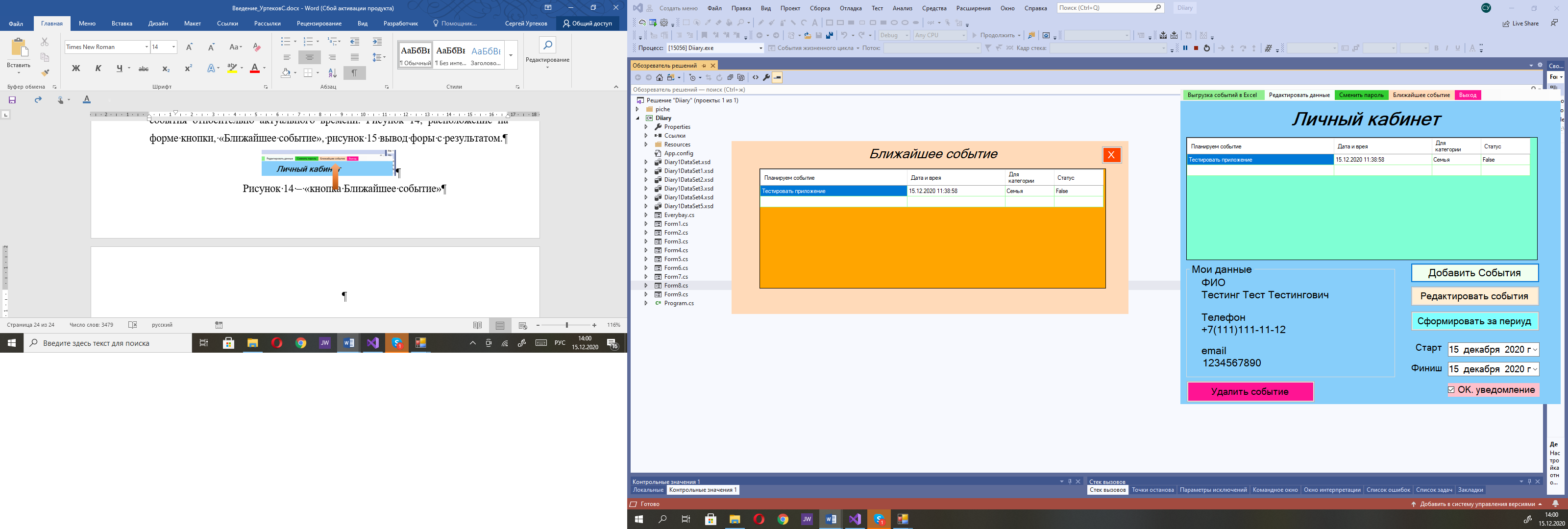


Рисунок 15 – «Форма, Ближайшее событие»

В нижнем левом углу существует кнопка удаление выделенного события. Кликнув по ней обозначенное событие удаляется. Так же в верху с права, имеется кнопка «выход» - закрывает личный кабинет и автоматически открывается первая форма «Авторизация». Как дополнение, реализована возможность, данные своих событий в Excel, при клике на кнопку «Выгрузка событий в «Excel»

# **Заключение**

В результате курсовой работы было произведено исследование предметной области, более детально изучена среда разработки, разработано детальное техническое задание, проанализированы прототипы программного продукта, разработан интерфейс проекта с ярким и привлекательным дизайном, так же стоит отметить, что разные формы в приложении оформлены в одном стиле, разработано руководство пользователя, проведено тестирование и отладка программного обеспечения.

Разработанный программный продукт, реализующий функционал приложения ежедневник поможет эффективно использовать рабочее, и свободное время, а также хранение, выведение на экран, запланированных событий пользователю. В современном мире, огромное количество планов, целей, задач, которые необходимо решать в этот же день, либо через годы. Запомнить всё невозможно, а данный продукт с использование информационных технологий будет особенно актуален

Разработка программного продукта реализующего функционал ежедневника способствовала развитию своих возможностей, по написанию файловые кодов а так же понимание логике взаимодействий форм, работы кода. Улучшились навыки создание форм, дизайна и интерфейсов форм.

В данной разработке программного продукта, возникали ошибка в синтаксисе кода, так же недочёты в разработке базы данных. Отслеживание и исправление ошибок недочётов способствовало улучшение внимание и чтение программного кода.

Разработка программного продукта реализующий функционал ежедневника предполагает усовершенствование, путём давления функционала: прикреплять фото, текстовые, видео файлы к событиям, возможность выбора и воспроизведения звукового сигнала, в назначенное время, расширения работы приложения в интернет сети.

Цель проекта достигнута, задача выполнена.

# **Приложение**