КОМИТЕТ ПО ОБРАЗОВАНИЮ МИНГОРИСПОЛКОМА

УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ

«МИНСКИЙ ГОСУДАРСТВЕННЫЙ КОЛЛЕДЖ ЭЛЕКТРОНИКИ»

Специальность: 2-40 01 01

«Программное обеспечение информационных технологий»

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

РАЗРАБОТКА ИНФОРМАЦИОННОЙ СИСТЕМЫ ПО УЧЁТУ ДЕЯТЕЛЬНОСТИ СЕРВИСНОГО ОТДЕЛА ОРГАНИЗАЦИИ

|  |  |  |
| --- | --- | --- |
| Автор проекта | (А.В. Никитина | ) |
| Руководитель проекта | (C.А. Оберган | ) |

2019

СОДЕРЖАНИЕ

Введение ................................................................................................................................. 3 1 Проектирование задачи ...................................................................................................... 4 1.1 Организация данных........................................................................................................ 4

1.2 Описание программы ...................................................................................................... 4

1.3 Система помощи............................................................................................................... 9

1. Инструменты разработки..................................................................................................11
2. Реализация......................................................................................................................... 12
   1. Функции: логическая и физическая реализации.......................................................... 12
   2. Функции и элементы управления..................................................................................12
   3. Функциональное тестирование..................................................................................... 13
3. Сообщения системы ......................................................................................................... 14

Заключение........................................................................................................................... 15

Список использованных источников ................................................................................. 16

Приложение А ......................................................................................................................17

Приложение Б ...................................................................................................................... 43 Приложение В ...................................................................................................................... 46

ВВЕДЕНИЕ

Целью проекта является разработка информационной системы по учёту деятельности сервисного отдела организации по ремонту электронной техники с применением языка программирования C#.

Информационная система обеспечивает принятие заказов на ремонт техники, оперативный контроль за ходом выполнения ремонтных работ, формирует по запросу бланк заказа. Разработана база клиентская база и база устройств

C# – объектно-ориентированный язык программирования. Разработан в 1998-2001

годах в компании Microsoft как язык разработки приложений для платформы Microsoft .NET.

На сегодняшний момент язык программирования C# один из самых мощных, быстро развивающихся и востребованных языков в ИТ-отрасли. В настоящий момент на нем пишутся самые различные приложения: от небольших десктопных программ до крупных веб-порталов и веб-сервисов, обслуживающих ежедневно миллионы пользователей.

C# является объектно-ориентированным поддерживает полиморфизм, наследование, перегрузку операторов, статическую типизацию. Объектно-ориентированный подход позволяет решить задачи по построению крупных, но в тоже время гибких, масштабируемых и расширяемых приложений. И C# продолжает активно развиваться, и с каждой новой версией появляется все больше новых функциональностей.

Пояснительная записка является полным описанием готового программного средства и принципа его работы.

Пояснительная записка содержит все необходимые разделы.

В разделе «Введение» указана оценка современного состояния решаемой проблемы, исходные данные для разработки проекта, перечень решаемых в работе задач.

В разделе «Проектирования задачи» описывается организация данных, концептуальный прототип, система помощи, средства защиты и разграничения доступа к данным.

В разделе «Реализация» описывается логическая и физическая организация данных, функциональное тестирование.

В разделе «Список использованных источников» указан список литературы и других источников, которые были использованы при написании данной работы.

1. Проектирование задачи
   1. Организация данных

Данные вводятся в программу посредством соответствующим им полям ввода и кнопкам, после этого, вносятся в базу данных в соответствующие им таблицы.

* 1. Описание программы

Целью данного проекта является разработка программы информационной системы по учёту деятельности сервисного отела организации по ремонту электронной техники, для предоставления сотрудникам организации возможности: оформления заказов, добавление новых заказчиков и устройств, отслеживание статуса заказа, закрытие заказа, поиск заказов, возможность изменения статусов заказа по ходу ремонта, связь сотрудников разных подразделений, возможность ведения всех таблиц базу данных разными пользователями с разными правами доступа.

Программа содержит 4 типа учётных записей:

* Администратор;
* Сервисный работник;
* Инженер;
* Менеджер.

У каждого пользователя есть свои учётные данные, которые используются для входа, учётные данные может назначать только Администратор. Данные для авторизации Администратора будут выданы непосредственно Администратору. Так же каждый пользователь обладает собственным набором доступа к информации.

При открытии программы пользователю открывается окно «Авторизация», в котором пользователю необходимо ввести свои учётные данные. (Рисунок 1.2.1).

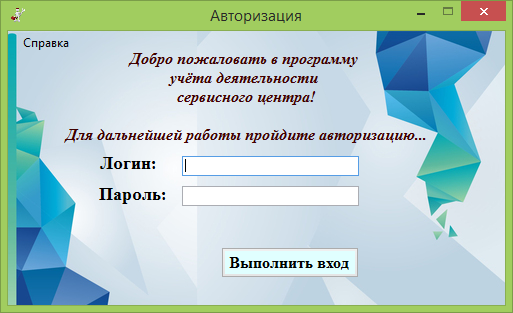


Рисунок 1.2.1 – Главная форма программы «Авторизация»

Далее пройдя авторизацию пользователь попадает в окно своей учётной записи и может выполнять предоставленные ему функции.

Первая учётная запись, которую мы рассмотрим – Учётная запись «Администратор». При запуске в окне «Администратор» Вы можете наблюдать панель выбора действий, которая включает в себя кнопки Добавления, Изменения, Удаления учетной записи и выбор параметров поиска, кроме панели, также панель с поиском, панель с фильтрацией и таблица в которой хранятся все данные необходимые для функционирования процесса авторизации. (Рисунок 1.2.2).

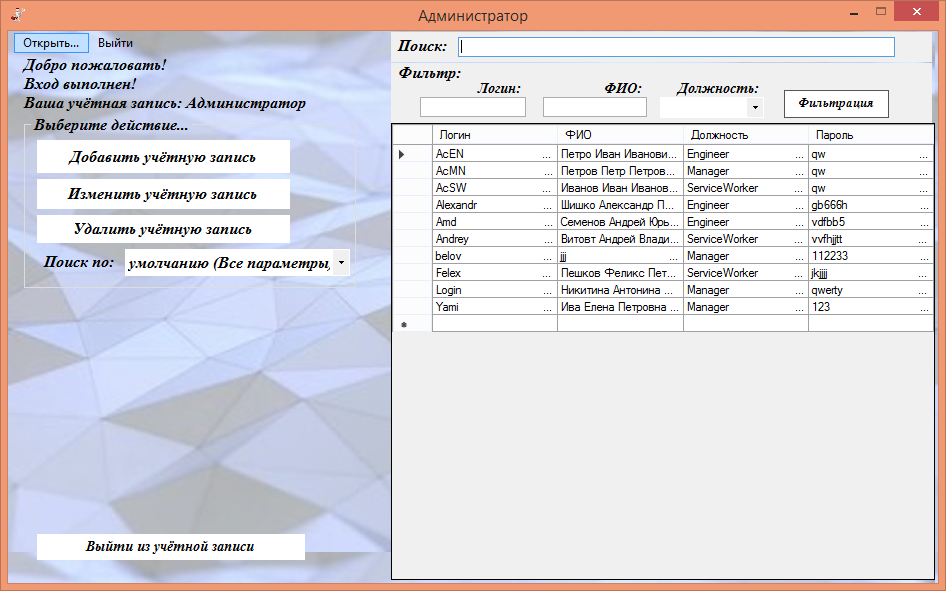


Рисунок 1.2.2 – Окно «Администратор»

На данной форме пользователь может выполнить следующие действия:

* Провести поиск (Предварительно выбрав параметр в панели выбора действий); (Рисунок 1.2.3).

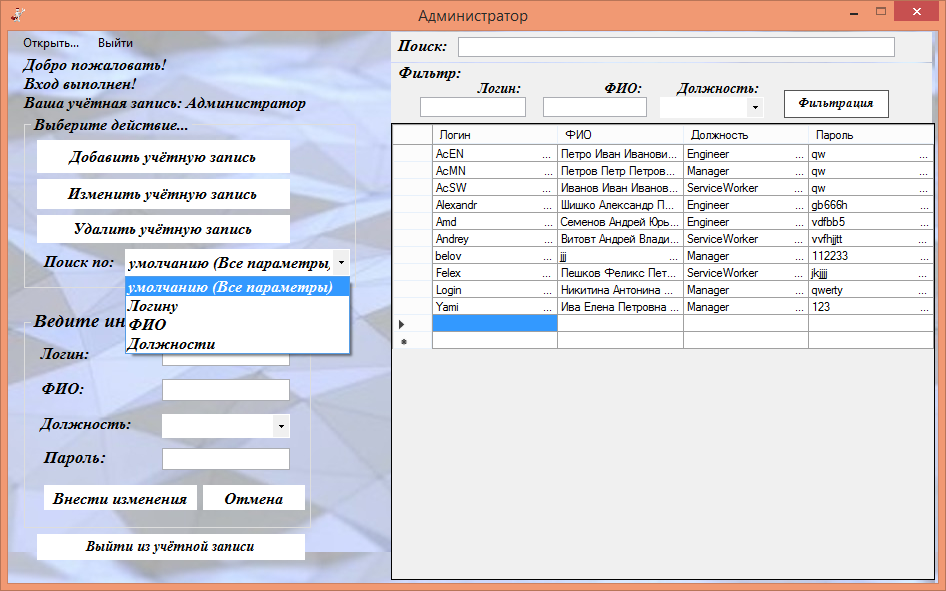


Рисунок 1.2.3 – Параметры поиска

* Провести фильтрацию учётных данных;
* Добавить или изменить учётную запись. При нажатии этих кнопок появиться панель ввода информации; (Рисунок 1.2.4).
* Удалить выделенную учётную запись;
* Открыть справку и о программе;
* Выйти из учётной записи;
* Выйти из программы;
* Просмотреть таблицу учётных данных.

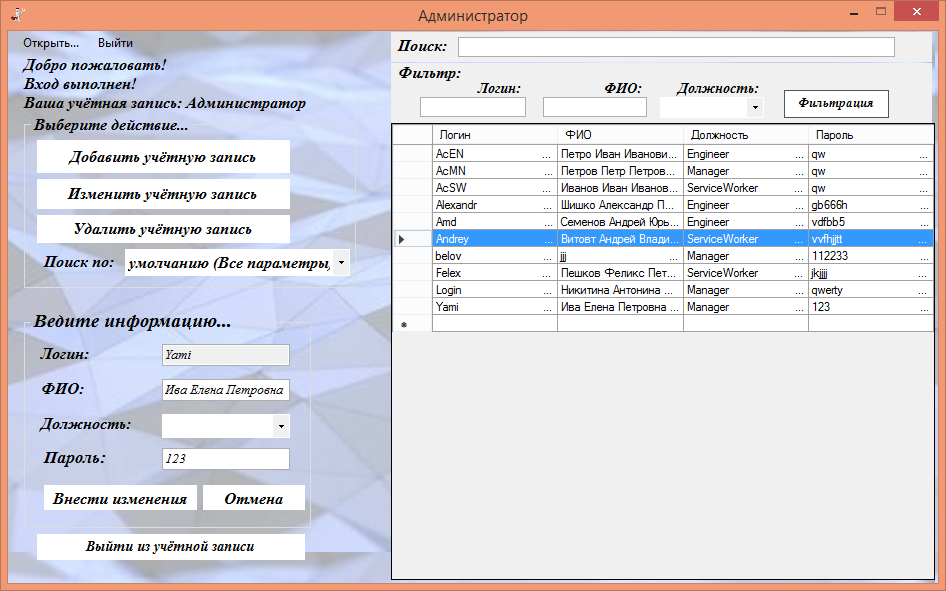


Рисунок 1.2.4 – Окно «Администратор» с панелью ввода информации

Следующая учётная запись, которую мы рассмотрим – «Сервисный работник». При запуске в окне «Сервис» Вы можете наблюдать таблицу заказов и панель выбора действий, которая включает в себя: панель закрытия акта панель, кнопки оформления заказа, просмотра таблиц: заказов, устройств и заказчиков. (Рисунок 1.2.5).

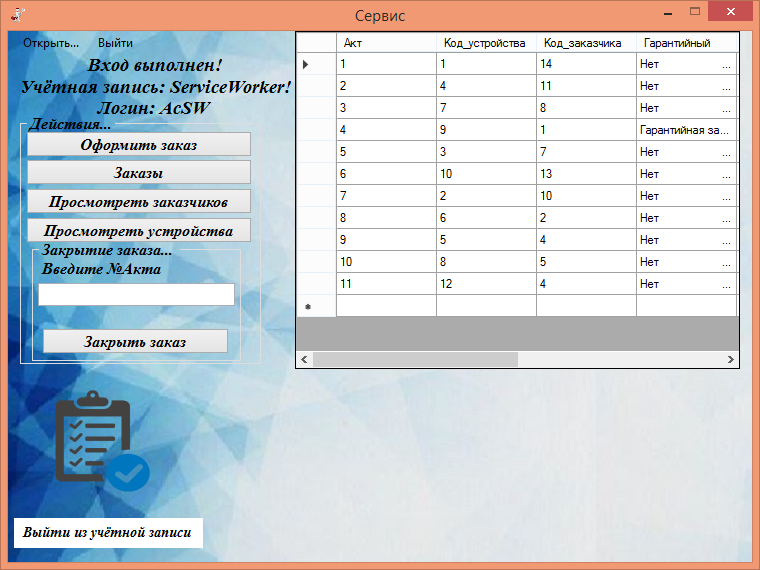


Рисунок 1.2.5 – Окно «Сервис» при запуске

На данной форме пользователь может выполнить следующие действия:

* Просмотр таблиц: Заказы, Устройства, Заказчики;
* Закрытие выполненного заказа;
* Оформление заказа: При оформлении заказа на панели появляется панель ввода информации. (Рисунок 1.2.6). При выборе устройства и заказчика появятся панели создания/изменения информации по устройствам и заказчикам. (Рисунок 1.2.7).

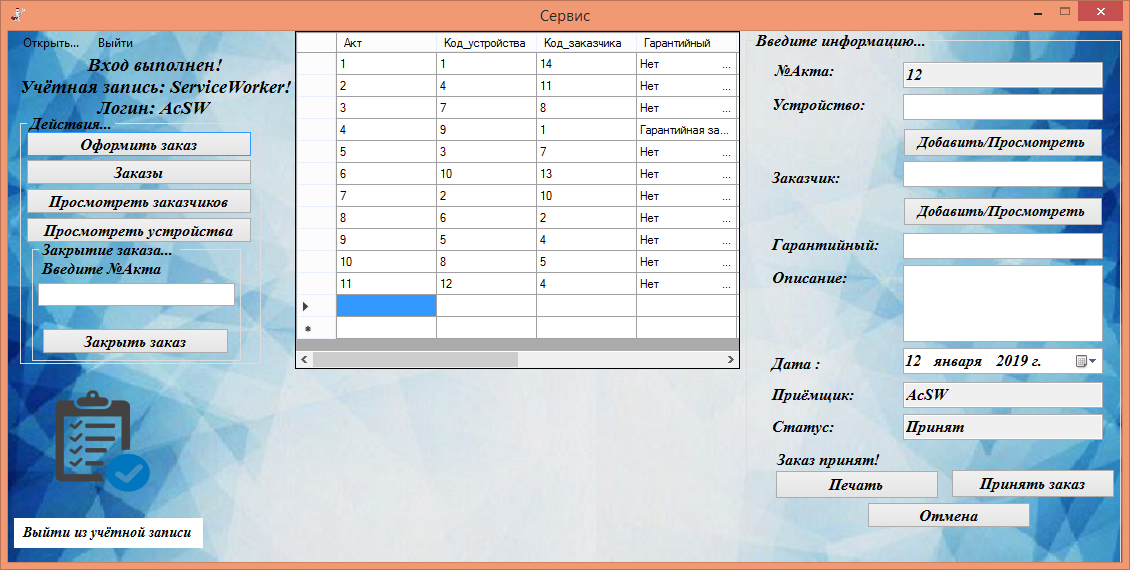


Рисунок 1.2.6 – Окно «Сервис» при оформлении заказов

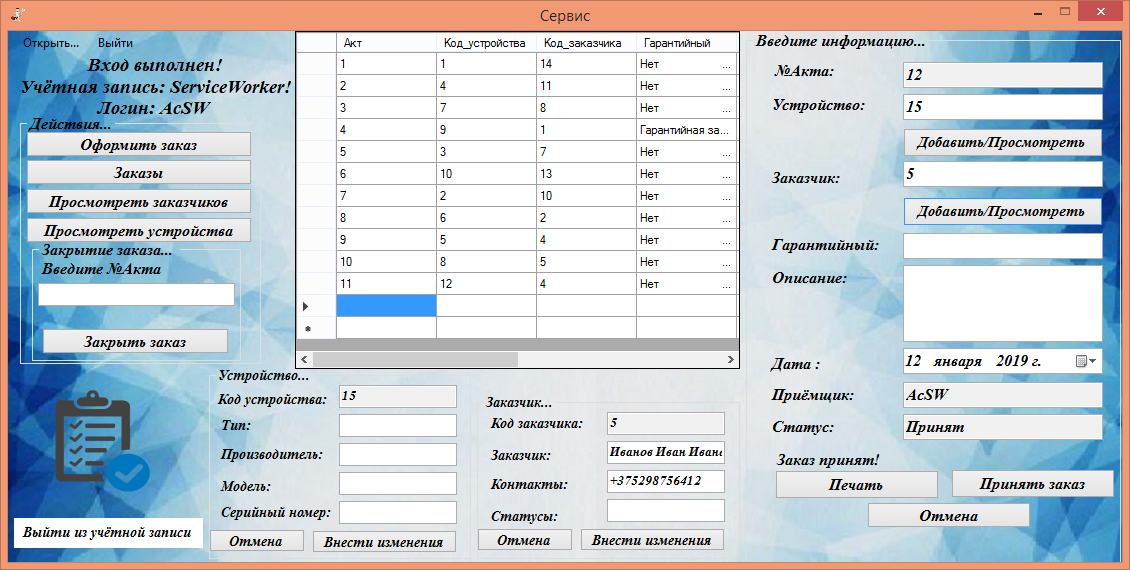


Рисунок 1.2.7 – Окно «Сервис» при оформлении заказа и добавлении или изменении устройств и заказчиков

Следующая учётная запись, которую мы рассмотрим – «Инженер». При запуске в окне «Инженер» Вы можете наблюдать панель изменения статусов заказа, Таблицы: Заказы и Работы и поиск по данным таблицам, также кнопки добавить работу в справочник и удалить работу. (Рисунок 1.2.8).

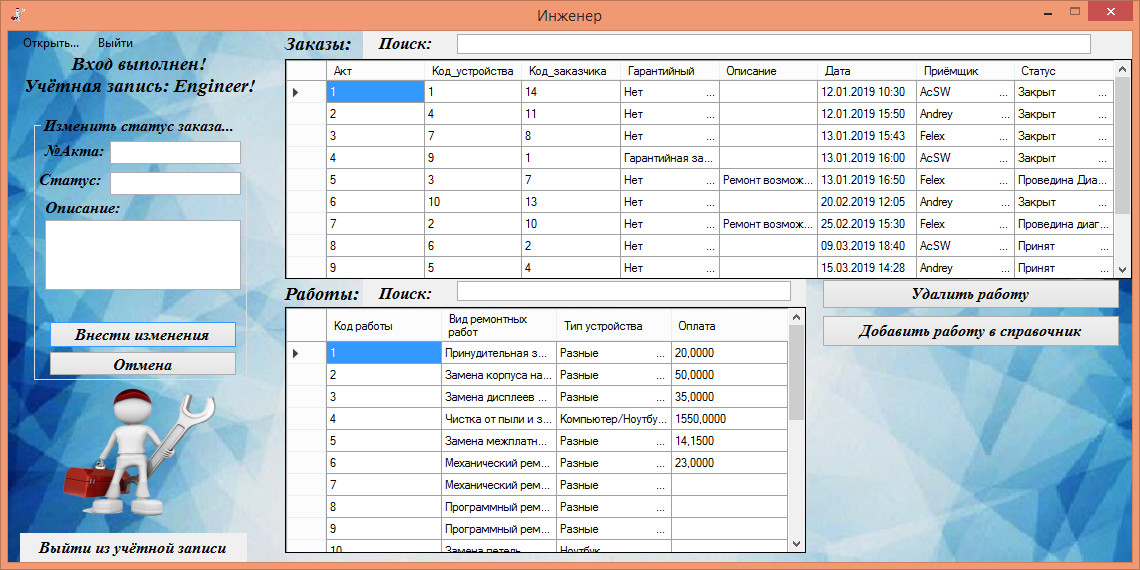


Рисунок 1.2.8 – Окно «Инженер» при запуске

На данной форме пользователь может выполнить следующие действия:

* Изменить статус заказа;
* Просмотреть таблицы: Заказы, Работы;
* Провести поиск по таблицам: Заказы, Работы;
* Удалить выбранную работу из справочника;
* Добить работу в справочник: При выполнении данной операции на форме появиться панель добавления работы; (Рисунок 1.2.9).



Рисунок 1.2.9 – Окно «Инженер» при добавлении работ

Последняя учётная запись, которую мы рассмотрим – «Менеджер». При запуске в окне «Менеджер» Вы можете наблюдать таблицы: Заказы и Работы и поиск по данным таблицам, панель установления оплаты и фильтр. (Рисунок 1.2.10).

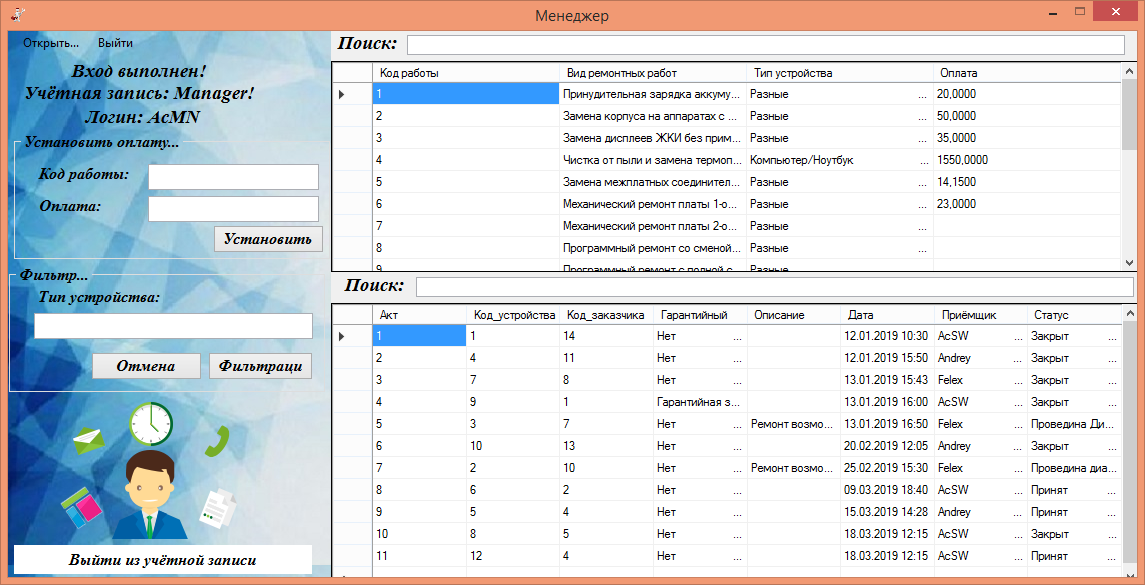


Рисунок 1.2.10 – Окно «Менеджер» при добавлении работ

На данной форме пользователь может выполнить следующие действия:

* Установить оплату за работу;
* Просмотреть таблицы: Заказы, Работы;
* Провести поиск по таблицам: Заказы, Работы;
* Фильтр работ по типам устройств.
  1. Система помощи

Система помощи в программе представлена в виде всплывающих подсказок и сообщениях об успешном выполнении. (Рисунок 1.3.1). В программе также присутствует контекстное меню(Рисунок 1.3.2), в котором есть справка (Рисунок 1.3.3), который будет полезен пользователю, чтобы разобраться с работой программы

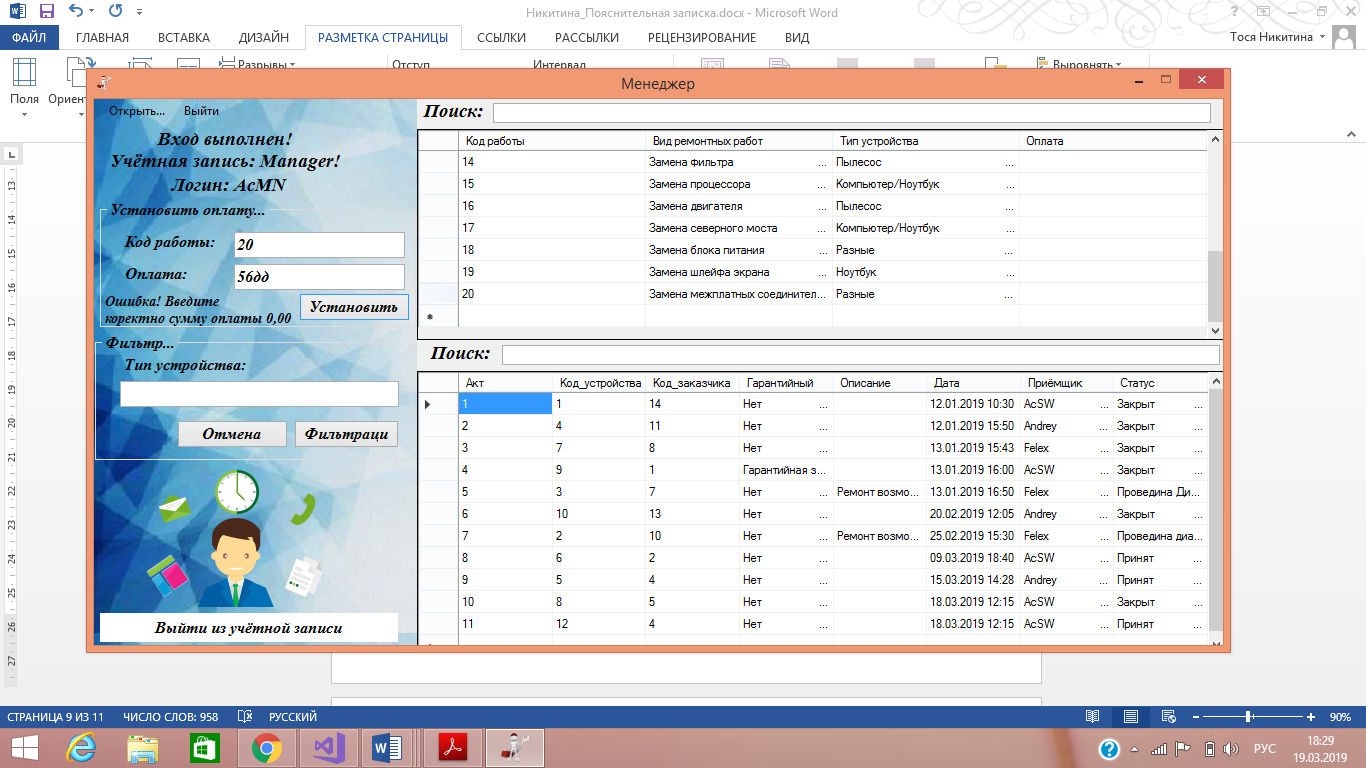
.

Рисунок 1.3.1 – Подсказка при некорректном вводе

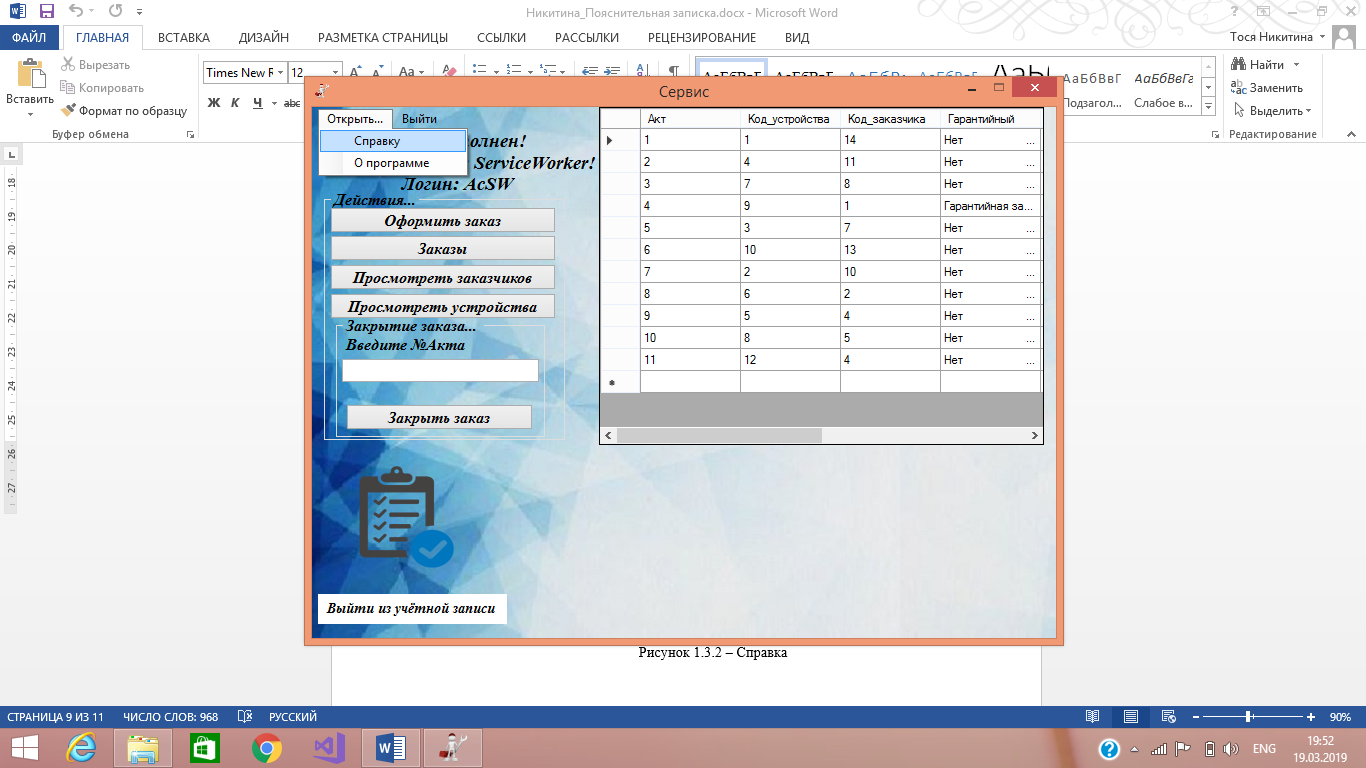
.

Рисунок 1.3.2 – Контекстное меню

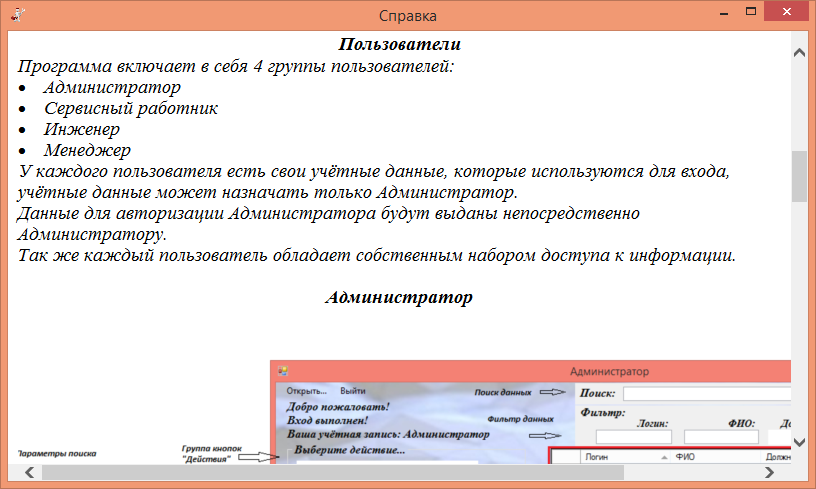


Рисунок 1.3.2 – Окно «Справка»

2 Инструменты разработки

Среда разработки: Visual Studio 2017.

Microsoft Visual Studio – это новая разработка компании Microsoft, позволяющая создавать приложения, работающие на платформе .NET. Особенность этой платформы заключается в широком наборе сервисов, которые доступны в различных языках программирования. При этом сервисы реализуются в виде промежуточного кода, который не зависит от базовой архитектуры. Едва ли не главной целью создания такой платформы было оснащение разработчиков специальными сервисно-ориентированными приложениями, которые могли бы работать на любой платформе, начиная от персонального компьютера и заканчивая мобильным устройством.

Microsoft Visual Studio объединяет в себе огромное количество функций, позволяющих осуществлять разработки для Windows всех версий. В Visual Studio реализуется новая среда разработчика, благодаря которой создавать приложения стало проще. Microsoft Visual Studio – это обновленная и упрощенная программная среда, для которой характерна высокая производительность, причем она не зависит от особенностей оборудования.

Каждая новая версия программы состоит из новейших инструментов и технологий, позволяющих разрабатывать приложения с учетом особенностей и положительных моментов современных платформ. Например, Visual Studio 2017 может поддерживать более ранние версии, в том числе Windows ХР и Windows Server 2003. При этом разработчикам открыта дорога к созданию новых и модернизации уже существующих приложений, предназначенных для ранних версий ОС Windows.

Visual Studio 2017 представляет собой передовую программу, которая дает возможность любым по размеру командам осуществлять проектирование и создание привлекательных приложений. Благодаря инструментам гибкого планирования можно внедрять методы последовательной разработки и применяться гибкие методологии в темпе, удобном для пользователя.

С помощью расширенных средств моделирования, обнаружения и проектирования можно максимально полно описать систему, которая позволит наиболее удачно реализовать конкретную концепцию архитектуры.

1. Реализация
   1. Функции: логическая и физическая реализации

Программа разработана для упрощения введения учёта деятельности сервисного центра организации. При запуске программы пользователь проходит авторизацию и попадает на свою учётную запись со своими функциональными возможностями описанными в пункте 1.2 Описание программы.

Каждая учётная запись имеет возможность изменить, либо добавить записи в доступные ей таблицы базу данных. В случае успешного добавления пользователь увидит текстовую подсказку об успешном внесении изменений (Рисунок 3.1.1).

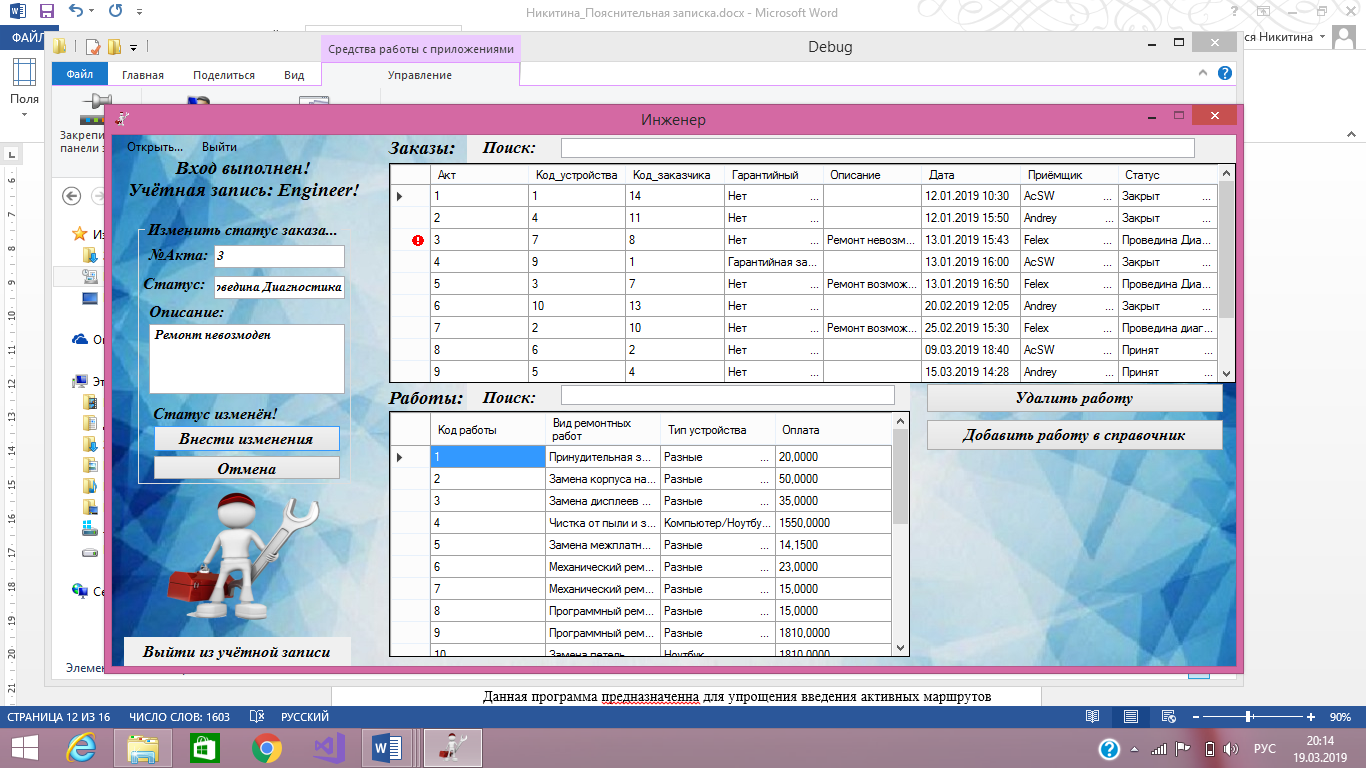


Рисунок 3.1.1 – Статус изменен

В случае некорректных данных в текстовом поле подсказки пользователь увидит подсказку

Следующего вида (Рисунок 3.1.2).

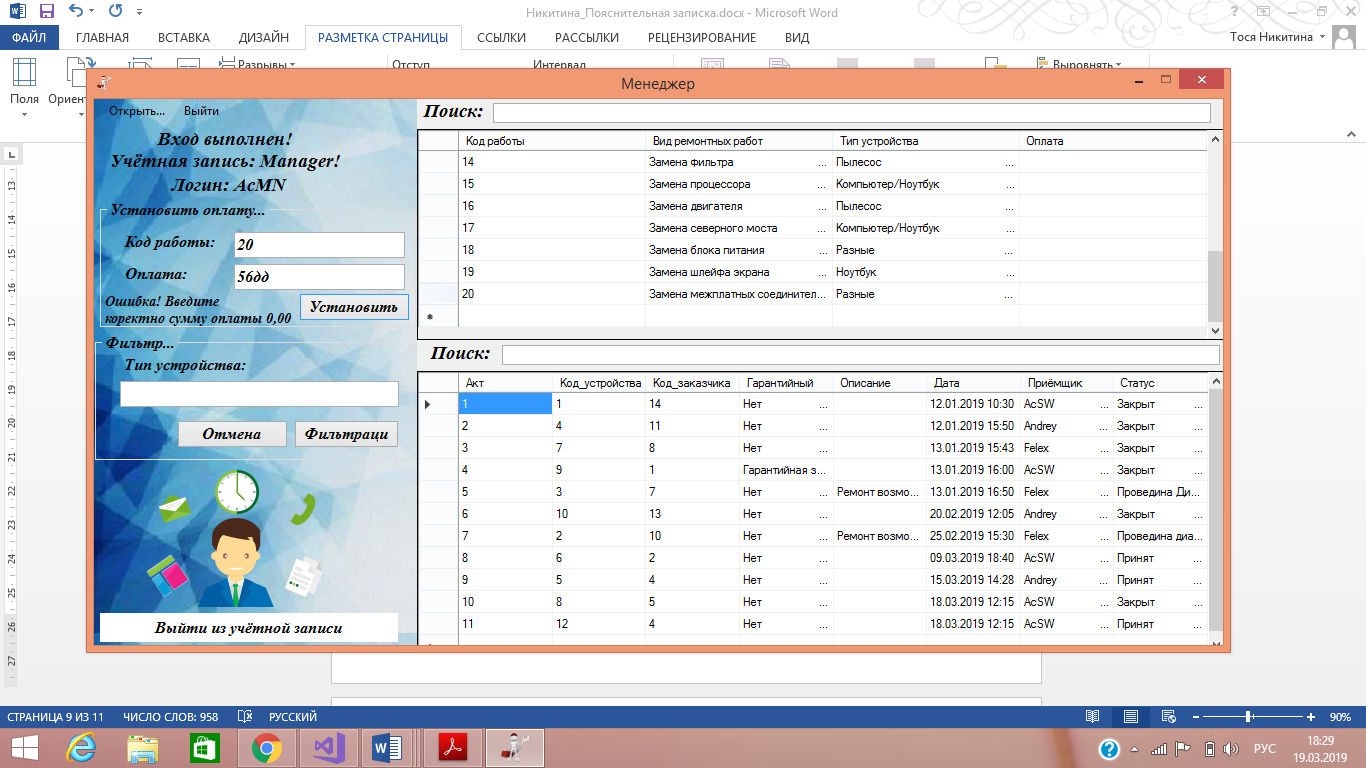


Рисунок 3.1.2 – Ошибка

На главной форме также можно получить справку о работе с программой с помощью пункта главного меню «Справка», краткую информацию о самой программе и её разработчике с помощью пункта главного меню «О программе».

* 1. Функции и элементы управления

Данная программа предназначена для упрощения введения учёта деятельности сервисного центра. На форме присутствуют следующие элементы управления: кнопки, переключатели, контекстное меню, база данных.

* 1. Функциональное тестирование

В ходе функционального тестирования на этапе разработке были выявлены недостатки и отсутствие некоторых функционально важных элементов.

Функциональное тестирование выявило слабые места в программе, что помогло устранить и доработать программу. Устранение и доработка функциональности программы была произведена в рамках технического задания, с добавлением дополнительного функционала, не противоречащего с требованиями.

В ходе тестирования были обнаружены программные ошибки и ошибки в интерфейсе.

После внесения доработок и поправок программа прошла повторное тестирование, не выявившее недочетов на данном этапе.

Из выше сказанного можно сделать вывод, что программа полностью функционирует и пригодна для решения задач.

4 Сообщения системы

В программе присутствуют системные сообщения, которые появляются при выполнении различных действий, например Заказ сформирован(Рисунок 4.1)

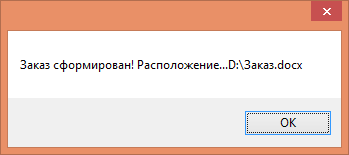


Рисунок 4.1 – Заказ сформирован

ЗАКЛЮЧЕНИЕ

В данном проекте представлена программа предназначенная автоматизированного веления информационной системы по учёту деятельности сервисного отдела организации по ремонту электронной техники. Данная программа упрощает работу сервисного центра за счёт грамотного взаимодействия между пользователями системы и разграниченными для них функциями. Программа имеет привлекательный и простой интерфейс и в то же время не перегружена функциональными возможностями.

В ходе написания данной программы были получены знания работы с языком C#. Кроме того, были выявлены и устранены ошибки и недостатки программы. Проведено функциональное и полное тестирование, как на корректных и некорректных данных в ходе которых программа приобрела законченный вид полностью работающего продукта.

Программа является полноценным программным продуктом и может быть доработана и усовершенствована дополнительными функциональными возможностями или расширением области изучения команд и языков.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Методические указания по выполнению проектов Специальность: 2-40 01 01 «Программное обеспечение информационных технологий».
2. Албахари, Джозеф, Албахари, Бен – C# 6.0. Справочник. Полное описание языка – Вильямс, 2016 – 1040 с.
3. Зиборов, В. В. – Visual C# 2012 на примерах – БХВ-Петербург, 2013 – 475 с.
4. Дейтел, Пол, Дейтел, Харви – Как программировать на Visual C# 2012 – Спб.:Питер, 2015 – 858 с.
5. Хабрахабр [Электронный ресурс]. – Электронные данные. – Режим доступа: http://habrahabr.ru.
6. MSDN MicroSoft [Электронный ресурс]. – Электронные данные. – Режим доступа: http://msdn.microsoft.com.
7. 3. Шилдт Г. – С#. Учебный курс.

ПРИЛОЖЕНИЕ А

(обязательное)

Текст программы

Листинг файла Program.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace ServiceCenter

{

static class Program

{

/// <summary>

/// Главная точка входа для приложения.

/// </summary>

[STAThread]

static void Main()

{

Application.EnableVisualStyles();

Application.SetCompatibleTextRenderingDefault(false);

Application.Run(new Form1());

}

}

}

Листинг файла Form1.cs

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.SqlClient;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace ServiceCenter

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

string s;

public string log;

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

{

}

private void textBox1\_KeyDown(object sender, KeyEventArgs e)

{

if (e.KeyCode == Keys.Enter)

{

textBox2.Focus();

e.SuppressKeyPress = true;

}

}

private void textBox2\_KeyDown(object sender, KeyEventArgs e)

{

if (e.KeyCode == Keys.Enter)

{

button1.PerformClick();

e.SuppressKeyPress = true;

}

}

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

{

if ((textBox1.Text == "admin") && (textBox2.Text == "admin"))

{

Hide();

Form2 f = new Form2();

f.Show();

}

else if ((textBox1.Text == "") && (textBox2.Text == ""))

{

label4.Visible = true;

label4.Text = "Ошибка! Введите данные!";

}

else

{string path= Application.StartupPath + @"\Database1.mdf";

string myConnectionString = @"Data Source = (LocalDB)\MSSQLLocalDB; AttachDbFilename = '"+path+"'; Integrated Security = True";

string mySelectQuery = "SELECT Должность, Логин FROM [Authorization] WHERE Логин='" + textBox1.Text + "'AND Пароль='" + textBox2.Text + "' ";

SqlConnection myConnection = new SqlConnection(myConnectionString);

SqlCommand myCommand = new SqlCommand(mySelectQuery, myConnection);

try

{

myConnection.Open();

SqlDataReader myReader = myCommand.ExecuteReader();

if (myReader.HasRows == false) { label4.Visible = true; label4.Text = "Вход не выполнен! Повторите попытк!"; };

while (myReader.Read()) { s = myReader.GetString(0); s = s.Replace(" ", string.Empty); log = myReader.GetString(1); }

if (s != "")

{

switch (s)

{

case "ServiceWorker":

label4.Text = "ServiceWorker";

Hide();

Form5 f = new Form5(log);

f.Show();

break;

case "Engineer":

label4.Text = "Engineer";

Hide();

Form4 f1 = new Form4(log);

f1.Show();

break;

case "Manager":

label4.Text = "Manager";

Hide();

Form3 f2 = new Form3(log);

f2.Show();

break;

default:

label4.Text = "Вход не выполнен! Повторите попытку!";

break;

}

}

}

catch

{

label4.Text = "Вход не выполнен! Повторите попытку!";

}

finally { myConnection.Close(); }

}

}

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

{

Form7 f7 = new Form7();

f7.ShowDialog();

}

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

{

AboutBox1 ab = new AboutBox1();

ab.ShowDialog();

}

private void Form1\_FormClosing(object sender, FormClosingEventArgs e)

{

Application.Exit();

}

}

}

Листинг файла Form2.cs

using System;

using System.Collections.Generic;

using System.Data;

using System.Data.SqlClient;

using System.Windows.Forms;

namespace ServiceCenter

{

public partial class Form2 : Form

{

bool d=false;

bool iz = false;

public Form2()

{

InitializeComponent();

}

int znach = 0;

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

{

if (d==true)

{

string path = Application.StartupPath + @"\Database1.mdf";

string myConnectionString = @"Data Source = (LocalDB)\MSSQLLocalDB; AttachDbFilename = '" + path + "'; Integrated Security = True";

SqlConnection connect = new SqlConnection(myConnectionString);

string sql = "INSERT INTO [Authorization] (Логин, ФИО, Должность, Пароль) VALUES (@Логин, @ФИО, @Должность,@Пароль );";

SqlCommand cmd\_SQL = new SqlCommand(sql, connect);

cmd\_SQL.Parameters.AddWithValue("@Логин", логинTextBox.Text);

cmd\_SQL.Parameters.AddWithValue("@ФИО", фИОTextBox.Text);

cmd\_SQL.Parameters.AddWithValue("@Должность", должностьComboBox.Text);

cmd\_SQL.Parameters.AddWithValue("@Пароль", парольTextBox.Text);

try

{

connect.Open();

int n = cmd\_SQL.ExecuteNonQuery();

label8.Text = "Запись успешно добавлена!";

label8.Visible = true;

}

catch (SqlException ex)

{

label8.Text = "Ошибка! Запись не добавлена введите данные корректно! ";

label8.Visible = true;

}

finally

{

connect.Close();

this.authorizationTableAdapter.Fill(this.database1DataSet.Authorization);

}

}

if (iz == true)

{

string path = Application.StartupPath + @"\Database1.mdf";

string myConnectionString = @"Data Source = (LocalDB)\MSSQLLocalDB; AttachDbFilename = '" + path + "'; Integrated Security = True";

SqlConnection connect = new SqlConnection(myConnectionString);

string sql = "Update [Authorization] set ФИО=@ФИО, Должность=@Должность, Пароль=@Пароль WHERE Логин=@Логин;";

SqlCommand cmd\_SQL = new SqlCommand(sql, connect);

cmd\_SQL.Parameters.AddWithValue("@Логин", логинTextBox.Text);

cmd\_SQL.Parameters.AddWithValue("@ФИО", фИОTextBox.Text);

cmd\_SQL.Parameters.AddWithValue("@Должность", должностьComboBox.Text);

cmd\_SQL.Parameters.AddWithValue("@Пароль", парольTextBox.Text);

try

{

connect.Open();

int n = cmd\_SQL.ExecuteNonQuery();

label8.Text = "Запись успешно измененина!";

label8.Visible = true;

}

catch (SqlException ex)

{

label8.Text = "Ошибка! Запись не измененина";

label8.Visible = true;

}

finally

{

connect.Close();

this.authorizationTableAdapter.Fill(this.database1DataSet.Authorization);

}

}

логинTextBox.ReadOnly = false;

groupBox1.Visible = false;

Validate();

authorizationBindingSource.EndEdit();

authorizationTableAdapter.Update(database1DataSet.Authorization);

}

private void Form2\_Load(object sender, EventArgs e)

{

// TODO: данная строка кода позволяет загрузить данные в таблицу "database1DataSet.Table". При необходимости она может быть перемещена или удалена.

this.authorizationTableAdapter.Fill(this.database1DataSet.Authorization);

dataGridView1.Rows[0].Selected = false;

логинTextBox.TabStop = false;

comboBox1.Text = comboBox1.Items[0].ToString();

Validate();

authorizationBindingSource.EndEdit();

должностьComboBox.Text = dataGridView1.Rows[dataGridView1.CurrentRow.Index].Cells[2].ToString();

}

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

{

}

private void textBox1\_TextChanged(object sender, EventArgs e)

{

if ((znach == 0) && (textBox1.Text != ""))

{

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

{

dataGridView1.Rows[i].Selected = false;

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

if (dataGridView1.Rows[i].Cells[j].Value != null)

if (dataGridView1.Rows[i].Cells[j].Value.ToString().Contains(textBox1.Text))

{

dataGridView1.Rows[i].Selected = true;

}

}

}

else if (textBox1.Text != "") Search(znach);

else if (textBox1.Text == "") dataGridView1.ClearSelection();

}

private void Search(int znach)

{

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

{

dataGridView1.Rows[i].Selected = false;

int j = znach - 1;

if (dataGridView1.Rows[i].Cells[j].Value != null)

if (dataGridView1.Rows[i].Cells[j].Value.ToString().Contains(textBox1.Text))

{

dataGridView1.Rows[i].Selected = true;

if (textBox1.Text == "") dataGridView1.ClearSelection();

}

}

}

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

{

authorizationBindingSource.AddNew();

логинTextBox.ReadOnly = false;

groupBox1.Visible = true; d = true;

}

private void button4\_Click(object sender, EventArgs e)

{

логинTextBox.ReadOnly = true;

должностьComboBox.Text = dataGridView1.Rows[dataGridView1.CurrentRow.Index].Cells[2].Value.ToString();

groupBox1.Visible = true;

iz = true;

}

private void button5\_Click(object sender, EventArgs e)

{

string path = Application.StartupPath + @"\Database1.mdf";

string myConnectionString = @"Data Source = (LocalDB)\MSSQLLocalDB; AttachDbFilename = '" + path + "'; Integrated Security = True";

SqlConnection connect = new SqlConnection(myConnectionString);

string sql = "Delete from [Authorization] WHERE Логин=@Логин;";

SqlCommand cmd\_SQL = new SqlCommand(sql, connect);

cmd\_SQL.Parameters.AddWithValue("@Логин", dataGridView1.Rows[dataGridView1.CurrentRow.Index].Cells[0].Value.ToString());

try

{

connect.Open();

int n = cmd\_SQL.ExecuteNonQuery();

label8.Text = "Запись успешно Удалена!";

label8.Visible = true;

}

catch (SqlException ex)

{

label8.Text = "Ошибка! Запись не удалена";

label8.Visible = true;

}

finally

{

connect.Close();

}

authorizationBindingSource.RemoveCurrent();

Validate();

authorizationBindingSource.EndEdit();

authorizationTableAdapter.Update(database1DataSet.Authorization);

}

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

{

switch (comboBox1.SelectedIndex)

{

case 0: znach = 0;

break;

case 1:

znach = 1;

break;

case 2:

znach = 2;

break;

case 3:

znach = 3;

break;

default:

znach = 0;

break;

}

}

private void button6\_Click(object sender, EventArgs e)

{

List<string> filterParts = new List<string>();

if (textBox2.Text != "")

filterParts.Add("Логин = '" + textBox2.Text + "'");

if (textBox3.Text != "")

filterParts.Add("ФИО = '" + textBox3.Text + "'");

if (comboBox2.SelectedIndex >= 0)

filterParts.Add("Должность = '" + comboBox2.Text + "'");

string filter = string.Join(" AND ", filterParts);

authorizationBindingSource.Filter = filter;

}

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

{

switch (comboBox2.SelectedIndex)

{

case 0:

comboBox2.Text = "Manager";

break;

case 1:

comboBox2.Text = "Engineer";

break;

case 2:

comboBox2.Text = "ServiceWorker";

break;

default:

comboBox2.Text = "";

break;

}

}

private void textBox2\_TextChanged(object sender, EventArgs e)

{

Search(1);

}

private void textBox3\_TextChanged(object sender, EventArgs e)

{

Search(2);

}

private void button2\_Click\_1(object sender, EventArgs e)

{

Form1 frm = new Form1();

frm.Show(); this.Hide();

}

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

{

dataGridView1.Rows[dataGridView1.RowCount - 2].Cells[0].Value = " ";

authorizationBindingSource.RemoveAt(dataGridView1.RowCount-2);

Validate();

authorizationBindingSource.EndEdit();

authorizationTableAdapter.Update(database1DataSet.Authorization);

groupBox1.Visible = false;

}

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

{

Form7 f7 = new Form7();

f7.ShowDialog();

}

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

{

AboutBox1 ab = new AboutBox1();

ab.ShowDialog();

}

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

{

Form1 frm = new Form1();

frm.Show(); this.Close();

}

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

{

Application.Exit();

}

private void Form2\_FormClosing(object sender, FormClosingEventArgs e)

{

Application.Exit();

}

}

}

Листинг файла Form3.cs

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.SqlClient;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace ServiceCenter

{

public partial class Form3 : Form

{

string log;

public Form3(string l)

{

InitializeComponent();

log = l;

}

private void Form3\_Load(object sender, EventArgs e)

{

// TODO: данная строка кода позволяет загрузить данные в таблицу "database1DataSet.Reception". При необходимости она может быть перемещена или удалена.

this.receptionTableAdapter.Fill(this.database1DataSet.Reception);

// TODO: данная строка кода позволяет загрузить данные в таблицу "database1DataSet.Work". При необходимости она может быть перемещена или удалена.

this.workTableAdapter.Fill(this.database1DataSet.Work);

label3.Text = "Логин: " + log;

}

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

{

this.Validate();

this.workBindingSource.EndEdit();

this.tableAdapterManager.UpdateAll(this.database1DataSet);

}

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

{

List<string> filterParts = new List<string>();

if (textBox4.Text != "")

filterParts.Add("[Тип устройства] = '" + textBox4.Text + "'");

string filter = string.Join(" AND ", filterParts);

workBindingSource.Filter = filter;

}

private void button4\_Click(object sender, EventArgs e)

{

List<string> filterParts = new List<string>();

string filter = string.Join(" AND ", filterParts);

workBindingSource.Filter = filter;

}

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

{

try

{

string path = Application.StartupPath + @"\Database1.mdf";

string myConnectionString = @"Data Source = (LocalDB)\MSSQLLocalDB; AttachDbFilename = '" + path + "'; Integrated Security = True";

SqlConnection connect = new SqlConnection(myConnectionString);

string sql = "Update [Work] set [Оплата]=@Оплата WHERE[Код работы]=@Код;";

SqlCommand cmd\_SQL = new SqlCommand(sql, connect);

cmd\_SQL.Parameters.AddWithValue("@Оплата", textBox7.Text);

cmd\_SQL.Parameters.AddWithValue("@Код", textBox6.Text);

try

{

connect.Open();

int n = cmd\_SQL.ExecuteNonQuery();

label9.Visible = true;

label9.Text = "Установлена";

}

catch (SqlException ex)

{

label9.Text = "Ошибка! Введите \nсуществующий Код работы";

label9.Visible = true;

}

finally

{

connect.Close();

this.workTableAdapter.Fill(this.database1DataSet.Work);

bool b=false;

if (textBox6.Text != "")

{

for (int i = 0; i < workDataGridView.RowCount; i++)

{

workDataGridView.Rows[i].Selected = false;

int j = 0;

if (workDataGridView.Rows[i].Cells[j].Value != null)

if (workDataGridView.Rows[i].Cells[j].Value.ToString().Contains(textBox6.Text))

{

b = true;

break;

}

}

if (b == false) { label9.Text = "Ошибка! Введите \nсуществующий Код работы"; }

}

}

}

catch

{

label9.Text = "Ошибка! Введите \nкоректно сумму оплаты 0,00";

}

}

private void workDataGridView\_DataError(object sender, DataGridViewDataErrorEventArgs e)

{

label9.Text = "Ошибка! Введите \nкоректно сумму оплаты 0,00";

e.ThrowException = false;

}

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

{

Form1 frm = new Form1();

frm.Show(); this.Close();

}

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

{

Form7 f7 = new Form7();

f7.ShowDialog();

}

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

{

AboutBox1 ab = new AboutBox1();

ab.ShowDialog();

}

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

{

Form1 frm = new Form1();

frm.Show(); this.Close();

}

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

{

Application.Exit();

}

private void Form3\_FormClosing(object sender, FormClosingEventArgs e)

{

}

}

}

Листинг файла Form4.cs

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.SqlClient;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace ServiceCenter

{

public partial class Form4 : Form

{

public Form4(string log)

{

InitializeComponent();

}

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

{

this.Validate();

this.receptionBindingSource.EndEdit();

this.tableAdapterManager.UpdateAll(this.database1DataSet);

}

private void Form4\_Load(object sender, EventArgs e)

{

// TODO: данная строка кода позволяет загрузить данные в таблицу "database1DataSet.Work". При необходимости она может быть перемещена или удалена.

this.workTableAdapter.Fill(this.database1DataSet.Work);

// TODO: данная строка кода позволяет загрузить данные в таблицу "database1DataSet.Reception". При необходимости она может быть перемещена или удалена.

this.receptionTableAdapter.Fill(this.database1DataSet.Reception);

}

private void textBox4\_TextChanged(object sender, EventArgs e)

{

if (textBox4.Text != "")

{

for (int i = 0; i < receptionDataGridView.RowCount; i++)

{

receptionDataGridView.Rows[i].Selected = false;

for (int j = 0; j < receptionDataGridView.ColumnCount; j++)

if (receptionDataGridView.Rows[i].Cells[j].Value != null)

if (receptionDataGridView.Rows[i].Cells[j].Value.ToString().Contains(textBox4.Text))

{

receptionDataGridView.Rows[i].Selected = true;

}

}

}

if (textBox1.Text == "") receptionDataGridView.ClearSelection();

}

private void button12\_Click(object sender, EventArgs e)

{

string path = Application.StartupPath + @"\Database1.mdf";

string myConnectionString = @"Data Source = (LocalDB)\MSSQLLocalDB; AttachDbFilename = '" + path + "'; Integrated Security = True";

SqlConnection connect = new SqlConnection(myConnectionString); string sql = "Update [Reception] set Статус=@Статус, Описание=@Описание WHERE Акт=@Акт;";

SqlCommand cmd\_SQL = new SqlCommand(sql, connect);

cmd\_SQL.Parameters.AddWithValue("@Статус", textBox2.Text);

cmd\_SQL.Parameters.AddWithValue("@Описание", textBox3.Text);

cmd\_SQL.Parameters.AddWithValue("@Акт", textBox1.Text);

try

{

connect.Open();

int n = cmd\_SQL.ExecuteNonQuery();

label5.Text = "Статус изменён!";

label5.Visible = true;

}

catch (SqlException ex)

{

label5.Text = "Ошибка! Статус не изменён!";

label5.Visible = true;

}

finally

{

connect.Close();

this.receptionTableAdapter.Fill(this.database1DataSet.Reception);

}

}

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

{

SqlConnection connect = new SqlConnection(@"Data Source = (LocalDB)\MSSQLLocalDB; AttachDbFilename = C:\Users\tosya\_\source\repos\ServiceCenter\ServiceCenter\Database1.mdf; Integrated Security = True");

string sql = "Delete from [Work] WHERE[Код работы] = @Код\_работы;";

SqlCommand cmd\_SQL = new SqlCommand(sql, connect);

cmd\_SQL.Parameters.AddWithValue("@Код\_работы", workDataGridView.Rows[workDataGridView.CurrentRow.Index].Cells[0].Value.ToString());

try

{

connect.Open();

int n = cmd\_SQL.ExecuteNonQuery();

label11.Text = "Запись успешно Удалена!";

label11.Visible = true;

}

catch (SqlException ex)

{

label11.Text = "Ошибка! Запись не удалена";

label11.Visible = true;

}

finally

{

connect.Close();

}

workBindingSource.RemoveCurrent();

Validate();

workBindingSource.EndEdit();

workTableAdapter.Update(database1DataSet.Work);

}

private void button4\_Click(object sender, EventArgs e)

{

workBindingSource.AddNew();

groupBox1.Visible = true;

}

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

{

string path = Application.StartupPath + @"\Database1.mdf";

string myConnectionString = @"Data Source = (LocalDB)\MSSQLLocalDB; AttachDbFilename = '" + path + "'; Integrated Security = True";

SqlConnection connect = new SqlConnection(myConnectionString); string sql = "INSERT INTO [Work] ([Код работы], [Вид ремонтных работ], [Тип устройства]) VALUES (@Код, @Вид, @Тип);";

SqlCommand cmd\_SQL = new SqlCommand(sql, connect);

cmd\_SQL.Parameters.AddWithValue("@Код", код\_работыTextBox.Text);

cmd\_SQL.Parameters.AddWithValue("@Вид", вид\_ремонтных\_работTextBox.Text);

cmd\_SQL.Parameters.AddWithValue("@Тип", тип\_устройстваTextBox.Text);

try

{

connect.Open();

int n = cmd\_SQL.ExecuteNonQuery();

label11.Text = "Запись успешно добавлена!";

label11.Visible = true;

}

catch (SqlException ex)

{

label11.Text = "Ошибка! Запись не добавлена введите данные корректно! ";

label11.Visible = true;

}

finally

{

connect.Close();

}

Validate();

workBindingSource.EndEdit();

workTableAdapter.Update(database1DataSet.Work);

}

private void button5\_Click(object sender, EventArgs e)

{

workDataGridView.Rows[workDataGridView.RowCount - 2].Cells[0].Value = 2147483640;

workBindingSource.RemoveAt(workDataGridView.RowCount - 2);

Validate();

workBindingSource.EndEdit();

workTableAdapter.Update(database1DataSet.Work);

groupBox1.Visible = false;

}

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

{

textBox1.Text = "";

textBox2.Text = "";

textBox3.Text = "";

}

private void textBox5\_TextChanged(object sender, EventArgs e)

{

if (textBox5.Text != "")

{

for (int i = 0; i < workDataGridView.RowCount; i++)

{

workDataGridView.Rows[i].Selected = false;

for (int j = 0; j < workDataGridView.ColumnCount; j++)

if (workDataGridView.Rows[i].Cells[j].Value != null)

if (workDataGridView.Rows[i].Cells[j].Value.ToString().Contains(textBox5.Text))

{

workDataGridView.Rows[i].Selected = true;

if (textBox1.Text == "") workDataGridView.ClearSelection();

}

}

}

if (textBox1.Text == "") workDataGridView.ClearSelection();

}

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

{

Form1 frm = new Form1();

frm.Show(); this.Hide();

}

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

{

Form7 f7 = new Form7();

f7.ShowDialog();

}

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

{

AboutBox1 ab = new AboutBox1();

ab.ShowDialog();

}

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

{

Form1 frm = new Form1();

frm.Show(); this.Close();

}

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

{

Application.Exit();

}

private void Form4\_FormClosing(object sender, FormClosingEventArgs e)

{

Application.Exit();

}

}

}

Листинг файла Form5.cs

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.SqlClient;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using Word = Microsoft.Office.Interop.Word;

namespace ServiceCenter

{

public partial class Form5 : Form

{

string log;

public Form5(string l)

{

InitializeComponent();

log = l; }

private readonly string TeamplateFileName = Application.StartupPath + @"\template.docx";

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

{

this.Validate();

this.receptionBindingSource.EndEdit();

this.tableAdapterManager.UpdateAll(this.database1DataSet);

приёмщикTextBox.Text = log;

}

private void Form5\_Load(object sender, EventArgs e)

{

// TODO: данная строка кода позволяет загрузить данные в таблицу "database1DataSet.Clients". При необходимости она может быть перемещена или удалена.

this.clientsTableAdapter.Fill(this.database1DataSet.Clients);

// TODO: данная строка кода позволяет загрузить данные в таблицу "database1DataSet.Device". При необходимости она может быть перемещена или удалена.

this.deviceTableAdapter.Fill(this.database1DataSet.Device);

// TODO: данная строка кода позволяет загрузить данные в таблицу "database1DataSet.Clients". При необходимости она может быть перемещена или удалена.

this.clientsTableAdapter.Fill(this.database1DataSet.Clients);

// TODO: данная строка кода позволяет загрузить данные в таблицу "database1DataSet.Reception". При необходимости она может быть перемещена или удалена.

this.receptionTableAdapter.Fill(this.database1DataSet.Reception);

Form1 fr1 = new Form1();

label3.Text = "Логин: " + log;

deviceDataGridView.ClearSelection();

groupBox3.Visible = false;

groupBox2.Visible = false;

groupBox1.Visible = false;

receptionDataGridView.Size = new Size(445, 338);

receptionDataGridView.Location = new System.Drawing.Point(287, 0);

receptionDataGridView.Visible = true;

receptionDataGridView.Rows[0].Selected = false;

deviceDataGridView.Rows[0].Selected = false;

clientsDataGridView.Rows[0].Selected = false;

this.Width = 760;

this.Height = 570;

}

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

{

this.Width = 1130;

this.Height = 570;

receptionBindingSource.AddNew();

groupBox1.Visible = true;

актTextBox.Text = Convert.ToString(receptionDataGridView.RowCount - 1);

приёмщикTextBox.Text = log;

статусTextBox.Text = "Принят";

}

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

{

this.Width = 760;

this.Height = 570;

groupBox2.Visible = false;

groupBox3.Visible = true;

string path = Application.StartupPath + @"\Database1.mdf";

string myConnectionString = @"Data Source = (LocalDB)\MSSQLLocalDB; AttachDbFilename = '" + path + "'; Integrated Security = True";

SqlConnection connect = new SqlConnection(myConnectionString); string sql = "INSERT INTO [Reception] ([Акт], [Код\_устройства], [Код\_заказчика], [Гарантийный], [Описание], [Дата], [Приёмщик], [Статус]) VALUES (@Акт, @Код\_устройства, @Код\_заказчика,@Гарантийный, @Описание, @Дата,@Приёмщик, @Статус);";

SqlCommand cmd\_SQL = new SqlCommand(sql, connect);

cmd\_SQL.Parameters.AddWithValue("@Акт", актTextBox.Text);

cmd\_SQL.Parameters.AddWithValue("@Код\_устройства", код\_устройстваTextBox.Text);

cmd\_SQL.Parameters.AddWithValue("@Код\_заказчика", код\_заказчикаTextBox.Text);

cmd\_SQL.Parameters.AddWithValue("@Гарантийный", гарантийныйTextBox.Text);

cmd\_SQL.Parameters.AddWithValue("@Описание", описаниеTextBox.Text);

cmd\_SQL.Parameters.AddWithValue("@Дата", датаDateTimePicker.Value.ToString());

cmd\_SQL.Parameters.AddWithValue("@Приёмщик", приёмщикTextBox.Text);

cmd\_SQL.Parameters.AddWithValue("@Статус", статусTextBox.Text);

try

{

connect.Open();

int n = cmd\_SQL.ExecuteNonQuery();

label4.Text = "Заказ успешно оформоен!";

label4.Visible = true;

}

catch (SqlException ex)

{

label4.Text = "Ошибка! Запись не добавлена введите данные корректно! ";

label4.Visible = true;

}

finally

{

connect.Close();

}

Validate();

receptionBindingSource.EndEdit();

receptionTableAdapter.Update(database1DataSet.Reception);

groupBox1.Visible = false;

groupBox2.Visible = false;

groupBox3.Visible = false;

}

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

{

bool b = false;

if (код\_заказчикаTextBox.Text != "")

{

for (int i = 0; i < clientsDataGridView.RowCount; i++)

{

clientsDataGridView.Rows[i].Selected = false;

int j = 0;

if (clientsDataGridView.Rows[i].Cells[j].Value != null)

if ((clientsDataGridView.Rows[i].Cells[j].Value.ToString().Contains(код\_заказчикаTextBox.Text)) && (b != true))

{

clientsDataGridView.CurrentCell = clientsDataGridView.Rows[i].Cells[j];

groupBox2.Visible = true; b = true;

button13.Visible = false; button5.Visible = false;

}

}

}

if (b == false)

{

clientsBindingSource.AddNew();

код\_заказчикаTextBox1.ReadOnly = false;

код\_заказчикаTextBox1.Text = Convert.ToString(clientsDataGridView.RowCount - 1);

код\_заказчикаTextBox.Text = код\_заказчикаTextBox1.Text;

код\_заказчикаTextBox1.ReadOnly = true;

groupBox2.Visible = true;

}

else if (код\_заказчикаTextBox.Text == "")

{

clientsBindingSource.AddNew();

код\_заказчикаTextBox1.ReadOnly = false;

код\_заказчикаTextBox1.Text = Convert.ToString(clientsDataGridView.RowCount - 1);

код\_заказчикаTextBox.Text = код\_заказчикаTextBox1.Text;

код\_заказчикаTextBox1.ReadOnly = true;

groupBox2.Visible = true;

}

}

static bool Search()

{ bool b = false;

return b;

}

private void button4\_Click(object sender, EventArgs e)

{

bool b = false;

groupBox3.Visible = true;

if (код\_устройстваTextBox.Text != "")

{

for (int i = 0; i < deviceDataGridView.RowCount; i++)

{

deviceDataGridView.Rows[i].Selected = false;

int j = 0;

if (deviceDataGridView.Rows[i].Cells[j].Value != null)

if ((deviceDataGridView.Rows[i].Cells[j].Value.ToString().Contains(код\_устройстваTextBox.Text)) && (b != true))

{

deviceDataGridView.CurrentCell = deviceDataGridView.Rows[i].Cells[j];

groupBox3.Visible = true; b = true;

button14.Visible = false; button6.Visible = false;

}

}

}

if (b == false)

{

deviceBindingSource.AddNew();

код\_устройстваTextBox1.ReadOnly = false;

код\_устройстваTextBox1.Text = Convert.ToString(deviceDataGridView.RowCount - 1);

код\_устройстваTextBox.Text = код\_устройстваTextBox1.Text;

код\_устройстваTextBox1.ReadOnly = true;

groupBox3.Visible = true;

}

}

private void button5\_Click(object sender, EventArgs e)

{

string path = Application.StartupPath + @"\Database1.mdf";

string myConnectionString = @"Data Source = (LocalDB)\MSSQLLocalDB; AttachDbFilename = '" + path + "'; Integrated Security = True";

SqlConnection connect = new SqlConnection(myConnectionString);

string sql = "INSERT INTO [Clients] ([Код\_заказчика], [Заказчик], [Контакты], [Статусы]) VALUES (@Код\_заказчика, @Заказчик, @Контакты,@Статусы );";

SqlCommand cmd\_SQL = new SqlCommand(sql, connect);

cmd\_SQL.Parameters.AddWithValue("@Код\_заказчика", код\_заказчикаTextBox1.Text);

cmd\_SQL.Parameters.AddWithValue("@Заказчик", заказчикTextBox.Text);

cmd\_SQL.Parameters.AddWithValue("@Контакты", контактыTextBox.Text);

cmd\_SQL.Parameters.AddWithValue("@Статусы", статусыTextBox.Text);

try

{

connect.Open();

int n = cmd\_SQL.ExecuteNonQuery();

}

catch (SqlException ex)

{

}

finally

{

connect.Close();

}

Validate();

deviceBindingSource.EndEdit();

deviceTableAdapter.Update(database1DataSet.Device);

код\_устройстваTextBox.Text = код\_устройстваTextBox1.Text;

Validate();

deviceBindingSource.EndEdit();

deviceTableAdapter.Update(database1DataSet.Device);

groupBox3.Visible = false;

код\_заказчикаTextBox.Text = код\_заказчикаTextBox1.Text;

Validate();

clientsBindingSource.EndEdit();

clientsTableAdapter.Update(database1DataSet.Clients);

groupBox2.Visible = false;

}

private void button6\_Click(object sender, EventArgs e)

{

string path = Application.StartupPath + @"\Database1.mdf";

string myConnectionString = @"Data Source = (LocalDB)\MSSQLLocalDB; AttachDbFilename = '" + path + "'; Integrated Security = True";

SqlConnection connect = new SqlConnection(myConnectionString);

string sql = "INSERT INTO [Device] ([Код\_устройства], [Тип], [Производитель], [Модель], [Серийный\_номер]) VALUES (@Код\_устройства, @Тип, @Производитель,@Модель,@Серийный\_номер );";

SqlCommand cmd\_SQL = new SqlCommand(sql, connect);

cmd\_SQL.Parameters.AddWithValue("@Код\_устройства", код\_устройстваTextBox1.Text);

cmd\_SQL.Parameters.AddWithValue("@Тип", типTextBox.Text);

cmd\_SQL.Parameters.AddWithValue("@Производитель", производительTextBox.Text);

cmd\_SQL.Parameters.AddWithValue("@Модель", модельTextBox.Text);

cmd\_SQL.Parameters.AddWithValue("@Серийный\_номер", серийный\_номерTextBox.Text);

try

{

connect.Open();

int n = cmd\_SQL.ExecuteNonQuery();

}

catch (SqlException ex)

{

}

finally

{

connect.Close();

}

Validate();

deviceBindingSource.EndEdit();

deviceTableAdapter.Update(database1DataSet.Device);

код\_устройстваTextBox.Text = код\_устройстваTextBox1.Text;

Validate();

deviceBindingSource.EndEdit();

deviceTableAdapter.Update(database1DataSet.Device);

groupBox3.Visible = false;

}

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

{

clientsDataGridView.Size = new Size(445, 338);

clientsDataGridView.Location = new System.Drawing.Point(287, 0);

receptionDataGridView.Visible = false;

deviceDataGridView.Visible = false;

clientsDataGridView.Visible = true;

}

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

{

deviceDataGridView.Size = new Size(445, 338);

deviceDataGridView.Location = new System.Drawing.Point(287, 0);

receptionDataGridView.Visible = false;

clientsDataGridView.Visible = false;

deviceDataGridView.Visible = true;

}

private void button9\_Click(object sender, EventArgs e)

{

clientsDataGridView.Visible = false;

deviceDataGridView.Visible = false;

receptionDataGridView.Size = new Size(445, 338);

receptionDataGridView.Location = new System.Drawing.Point(287, 0);

receptionDataGridView.Visible = true;

}

private void button11\_Click(object sender, EventArgs e)

{

this.Height = 570;

this.Width = 760;

receptionDataGridView.Rows[receptionDataGridView.RowCount - 2].Cells[0].Value = 2147483640;

receptionBindingSource.RemoveAt(receptionDataGridView.RowCount - 2);

Validate();

receptionBindingSource.EndEdit();

groupBox1.Visible = false;

groupBox2.Visible = false;

groupBox3.Visible = false;

}

private void button10\_Click(object sender, EventArgs e)

{

try {

string akt = актTextBox.Text; var kod\_dev = код\_устройстваTextBox.Text; var kod\_zak = код\_заказчикаTextBox.Text;

string inf\_dev = серийный\_номерTextBox.Text;

string inf\_zak = заказчикTextBox.Text;

string gar = гарантийныйTextBox.Text;

string op = описаниеTextBox.Text;

string dat = датаDateTimePicker.Value.ToString();

var wordApp = new Word.Application();

wordApp.Visible = false;

var wordDocument = wordApp.Documents.Open(TeamplateFileName);

ReplaceWordStub("{akt}", akt, wordDocument);

ReplaceWordStub("{koddev}", kod\_dev, wordDocument);

ReplaceWordStub("{kodzak}", kod\_zak, wordDocument);

ReplaceWordStub("{infdev}", inf\_dev, wordDocument);

ReplaceWordStub("{inf\_ak}", inf\_zak, wordDocument);

ReplaceWordStub("{gar}", gar, wordDocument);

ReplaceWordStub("{op}", op, wordDocument);

ReplaceWordStub("{dat}", dat, wordDocument);

wordDocument.SaveAs2(FileName: @"D:\Заказ.docx");

wordApp.Visible = true;

MessageBox.Show(@"Заказ сформирован! Расположение...D:\Заказ.docx");

}

catch { MessageBox.Show("Произошла ошибка!");

}

}

private void ReplaceWordStub(string stubToReplace, string text, Word.Document wordDocument)

{

var range = wordDocument.Content;

range.Find.ClearFormatting();

range.Find.Execute(FindText: stubToReplace, ReplaceWith: text);

}

private void button13\_Click(object sender, EventArgs e)

{

clientsDataGridView.Rows[clientsDataGridView.RowCount - 2].Cells[0].Value = 2147483640;

clientsBindingSource.RemoveAt(clientsDataGridView.RowCount - 2);

Validate();

clientsBindingSource.EndEdit();

clientsTableAdapter.Update(database1DataSet.Clients);

groupBox2.Visible = false;

}

private void button14\_Click(object sender, EventArgs e)

{

deviceDataGridView.Rows[deviceDataGridView.RowCount - 2].Cells[0].Value = 2147483640;

deviceBindingSource.RemoveAt(deviceDataGridView.RowCount - 2);

Validate();

deviceBindingSource.EndEdit();

deviceTableAdapter.Update(database1DataSet.Device);

groupBox3.Visible = false;

}

private void button15\_Click(object sender, EventArgs e)

{

Form1 frm = new Form1();

frm.Show(); this.Hide();

}

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

{

Form7 f7 = new Form7();

f7.ShowDialog();

}

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

{

AboutBox1 ab = new AboutBox1();

ab.ShowDialog();

}

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

{

Form1 frm = new Form1();

frm.Show(); this.Close();

}

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

{

Application.Exit();

}

private void Form5\_FormClosing(object sender, FormClosingEventArgs e)

{

Application.Exit();

}

private void button12\_Click(object sender, EventArgs e)

{

string path = Application.StartupPath + @"\Database1.mdf";

string myConnectionString = @"Data Source = (LocalDB)\MSSQLLocalDB; AttachDbFilename = '" + path + "'; Integrated Security = True";

SqlConnection connect = new SqlConnection(myConnectionString);

string sql = "Update [Reception] set Статус=@Статус WHERE Акт=@Акт;";

SqlCommand cmd\_SQL = new SqlCommand(sql, connect);

cmd\_SQL.Parameters.AddWithValue("@Статус", "Закрыт");

cmd\_SQL.Parameters.AddWithValue("@Акт", textBox1.Text);

try

{

connect.Open();

int n = cmd\_SQL.ExecuteNonQuery();

label5.Text = "Заказ закрыт!";

label5.Visible = true;

}

catch (SqlException ex)

{

label5.Text = "Ошибка! Введите существующий №Акта!";

label5.Visible = true;

}

finally

{

connect.Close();

if (textBox1.Text != "")

{

for (int i = 0; i < receptionDataGridView.RowCount; i++)

{

receptionDataGridView.Rows[i].Selected = false;

int j = 0;

if (receptionDataGridView.Rows[i].Cells[j].Value != null)

if (receptionDataGridView.Rows[i].Cells[j].Value.ToString().Contains(textBox1.Text))

{

receptionDataGridView.Rows[i].Cells[7].Value = "Закрыт";

Validate();

receptionBindingSource.EndEdit();

receptionTableAdapter.Update(database1DataSet.Reception);

label5.Text = "Заказ закрыт!";

}

}

}

Validate();

receptionBindingSource.EndEdit();

receptionTableAdapter.Update(database1DataSet.Reception);

}

}

}

}

Листинг файла Form7.cs

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace ServiceCenter

{

public partial class Form7 : Form

{

public Form7()

{

InitializeComponent();

}

private void Form7\_Load(object sender, EventArgs e)

{

webBrowser1.Navigate(Application.StartupPath+@"\helps.html");

}

}

}

Листинг файла AboutBox1.cs

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Drawing;

using System.Linq;

using System.Reflection;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace ServiceCenter

{

partial class AboutBox1 : Form

{

public AboutBox1()

{

InitializeComponent();

this.Text = String.Format("О программе {0}", AssemblyTitle);

this.labelProductName.Text = "Программа по учёту деятельности сервисного центра";

this.labelVersion.Text = String.Format("Версия {0}", AssemblyVersion);

this.labelCopyright.Text = "Copyright © 19.03.2019 by Никитина Антонина. All Rights Reserved.";

this.labelCompanyName.Text = "Минский Государственный Колледж Электроники";

this.textBoxDescription.Text = "Программа предназначена для учёта деятельности сервисного центра.\nУчёт деятельности включает в себя:\nПринятие заказов от клиентов\nОсуществление их рабонта / диагностики\nУстановление оплаты за ремонт\nВыдача утройства заказчику";

}

#region Методы доступа к атрибутам сборки

public string AssemblyTitle

{

get

{

object[] attributes = Assembly.GetExecutingAssembly().GetCustomAttributes(typeof(AssemblyTitleAttribute), false);

if (attributes.Length > 0)

{

AssemblyTitleAttribute titleAttribute = (AssemblyTitleAttribute)attributes[0];

if (titleAttribute.Title != "")

{

return titleAttribute.Title;

}

}

return System.IO.Path.GetFileNameWithoutExtension(Assembly.GetExecutingAssembly().CodeBase);

}

}

public string AssemblyVersion

{

get

{

return Assembly.GetExecutingAssembly().GetName().Version.ToString();

}

}

public string AssemblyDescription

{

get

{

object[] attributes = Assembly.GetExecutingAssembly().GetCustomAttributes(typeof(AssemblyDescriptionAttribute), false);

if (attributes.Length == 0)

{

return "";

}

return ((AssemblyDescriptionAttribute)attributes[0]).Description;

}

}

public string AssemblyProduct

{

get

{

object[] attributes = Assembly.GetExecutingAssembly().GetCustomAttributes(typeof(AssemblyProductAttribute), false);

if (attributes.Length == 0)

{

return "";

}

return ((AssemblyProductAttribute)attributes[0]).Product;

}

}

public string AssemblyCopyright

{

get

{

object[] attributes = Assembly.GetExecutingAssembly().GetCustomAttributes(typeof(AssemblyCopyrightAttribute), false);

if (attributes.Length == 0)

{

return "";

}

return ((AssemblyCopyrightAttribute)attributes[0]).Copyright;

}

}

public string AssemblyCompany

{

get

{

object[] attributes = Assembly.GetExecutingAssembly().GetCustomAttributes(typeof(AssemblyCompanyAttribute), false);

if (attributes.Length == 0)

{

return "";

}

return ((AssemblyCompanyAttribute)attributes[0]).Company;

}

}

#endregion

}

}

ПРИЛОЖЕНИЕ Б

(обязательное)

Экранные формы

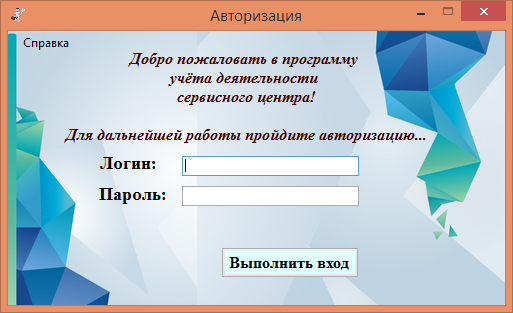


Рисунок Б.1 – Окно «Авторизация»

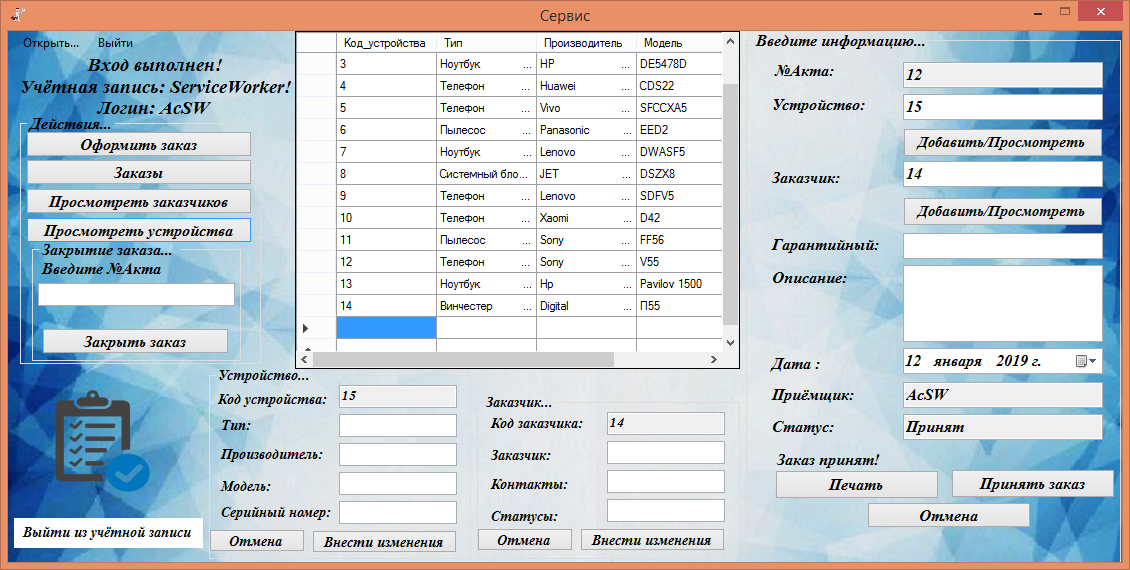


Рисунок Б.2 – Окно «Сервис»



Рисунок Б.3 – Окно «Инженер»

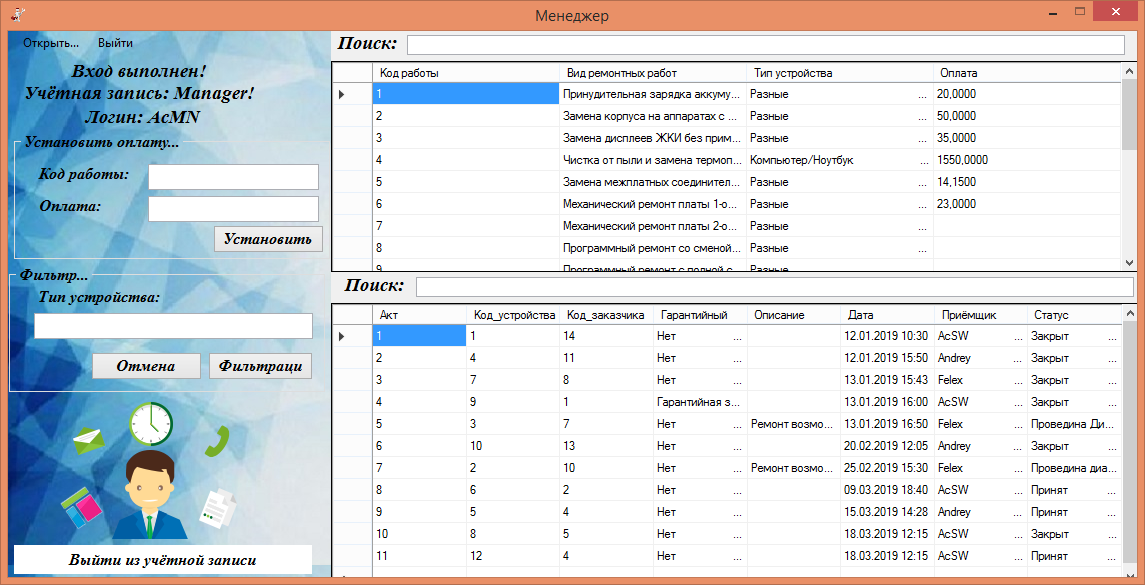


Рисунок Б.4 – Окно «Менеджер»

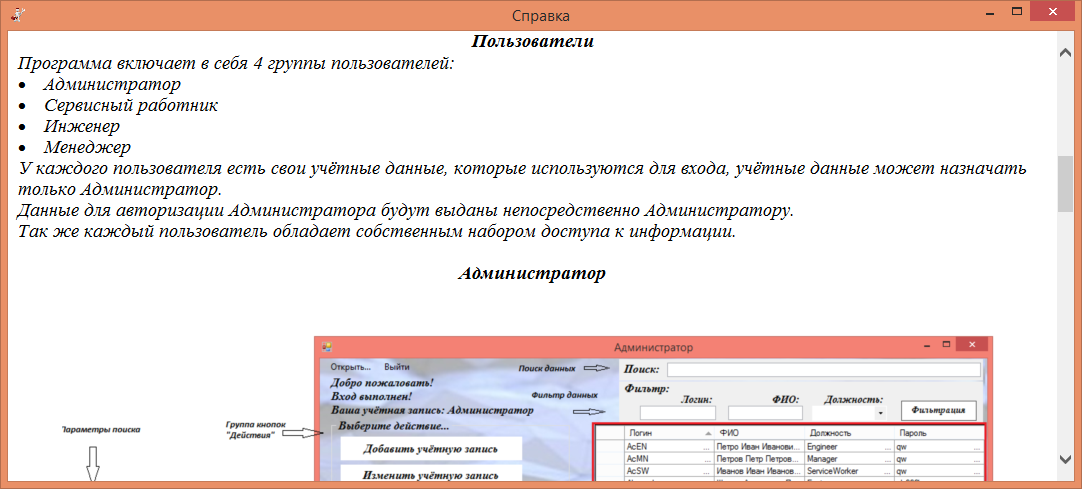


Рисунок Б.5 – Окно «Справка»

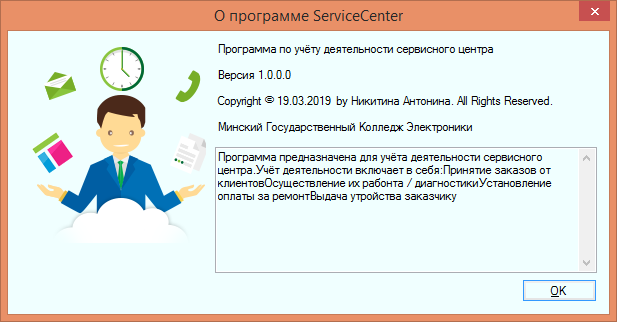


Рисунок Б.6– Окно «О программе»

ПРИЛОЖЕНИЕ

(справочное)

Диаграммы UML

(диаграмма вариантов использования)

