ПРАВИТЕЛЬСТВО САНКТ-ПЕТЕРБУРГА

КОМИТЕТ ПО ОБРАЗОВАНИЮ

САНКТ-ПЕТЕРБУРГСКОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ

ПРОФЕССИОНАЛЬНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

“КОЛЛЕДЖ МЕТРОПОЛИТЕНА И ЖЕЛЕЗНОДОРОЖНОГО

ТРАНСПОРТА”

Тема: Разработка приложения для информационного терминала метрополитена в среде Microsoft Visual Studio и MS SQL Server

МДК 05.02 Разработка кода информационных систем

Специальность 09.02.07 “Информационные системы и программирование”

Базовая подготовка среднего профессионального образования

Выполнил обучающий гр.№302

Павлов М.С

Руководитель:

Преподаватель

Ситало Алексей Валерьевич

Санкт-Петербург

2022

**СОДЕРЖАНИЕ:**

Введение…………………………………………………………..…...

1. Разработка эскизного и технического проекта программы……..

1.1. Назначение и область применения……………………..

1.2.Технические характеристики……………………………

2. Разработка рабочего проекта……………………………………..

2.1. Разработка программы…………………………………..

2.2. Спецификация программы………………………………

2.3. Текст программы…………………………………………

2.4. Описание программы……………………………………

2.5. Тестирование программы……………………………….

Заключение……………………………………………………………

Список использованной литературы…………………………………

**Введение.**

Постоянно растущая конкуренция, необходимость привлечения и удержания клиентов вынуждают компании проводить мероприятия по повышению качества обслуживания (удобство, скорость обслуживания, информированность клиента). Принцип повышения клиентоориентированности находит свое отражение во внедрении информационных систем и частичной автоматизации процесса обслуживания потребителя. Обусловленная этими мероприятиями потребность в оборудовании, программном обеспечении и комплексных услугах автоматизации и информатизации предприятий будет расти. Дополнительным стимулом развития рынка станут государственные программы информатизации и модернизации, запланированные на период до 2023 года и обеспечивающие стабильный спрос на оборудование, ПО и комплексные решения.

Главная цель, преследуемая при разработке и внедрении: повышение качества оказываемых пассажирам услуг. Созданный терминал имеет большой сенсорный экран, на котором удобно получать справочную информацию о нужном поезде, производить регистрацию пользователей, рассчитывать точное прибытие поезда, видеть доступные пересадки, .

Конструкция терминала обеспечивает безопасность и удобство работы с ним пассажиров и обслуживающего персонала. Устройство имеет компактные размеры в сочетании с легким весом, возможность крепиться к стене либо на специализированную напольную стойку, что делает его универсальным, при выборе места установки и монтаже.

Информационные киоски — активно развивающийся сегмент рынка устройств самообслуживания. Спрос на них после нескольких лет стагнации вырос в связи с запуском в России масштабных программ информатизации многих сфер деятельности. В частности, на всей территории России создается инфраструктура для получения государственных и муниципальных услуг, включающая интерактивные терминалы. Продолжается информатизация здравоохранения. Все больше поликлиник внедряют системы электронной записи на приём к врачам. Повсеместно идет оснащение терминалами самообслуживания арбитражных судов. Устанавливаются сенсорные киоски для ведения электронных журналов в школах.

Недавно началось внедрение систем оценки качества обслуживания, которые базируются также на сенсорных киосках. В частности, установка таких терминалов идет в большинстве региональных медицинских учреждений и во многих государственных организациях.

**1. Разработка эскизного и технического проекта программы.**

*1.1. Назначение и область применения.*

Данная программа является приложением для терминала в метро. Областью применения данной программы является информатизация общества, работающего с терминалом. Так как, поставлена задача, разработать приложение под операционную систему Windows, то использоваться программа может только в среде 64 битной операционной системы Windows 10.

***1.2. Технические характеристики. Данный пункт должен содержать следующие подпункты:***

*1.2.1. Постановка задачи*

Моё приложение позволит выполнять ежедневные рутинные операции быстрее, а точнее находить нужные поезда, точное прибытие поезда, даст возможность студентам рассчитывать свой маршрут до унив(быстрее).

В качестве инструментального средства разработки базы данных выбрана Microsoft SQL

Данное приложение предназначена для общества(находящихся в метро), она позволит находить нужный поезд гораздо быстрее,производить регистрацию пользователей, рассчитывать точное прибытие поезда.

*1.2.2. Создание UML диаграммы*

Процесс выполнения данной программы можно представить в виде диаграммы изображенной на рис. 1.

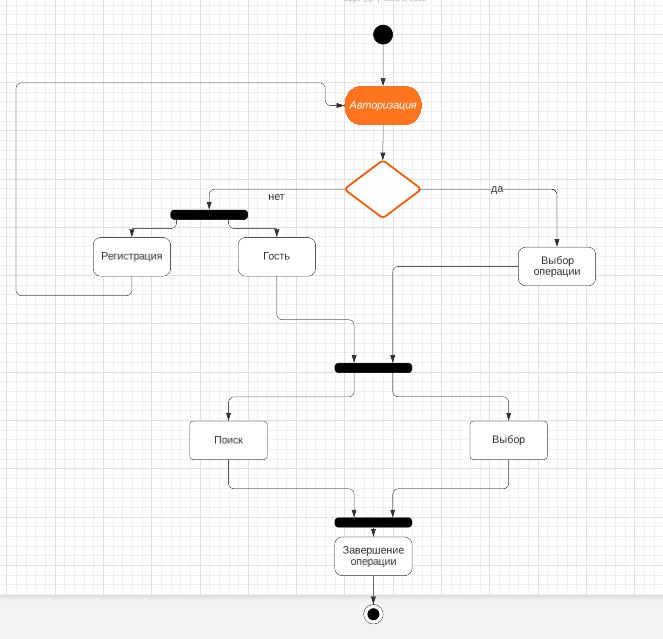


Рис. 1. Диаграмма активности

*1.2.3. Организация входных и выходных данных.*

В программе были использованы три типа входных данных:

* Текстовые

Существует два способа ввода этих данных в программу:

- прямая загрузка в режиме разработки форм;

- программная загрузка данных из файла в соответствующий компонент в режиме работы программы.

Выходные данные:

представляют собой графическую информацию, выводимую на дисплей ЭВМ.

*1.2.4. Выбор состава технических и программных средств.*

Исходя из целей курсовой работы, требуется создать программу, работающую под ОС Windows. В связи с тем, что разработанный алгоритм подразумевает использование графических возможностей, звука, анимации и параллельных процессов взаимодействия с пользователем, было принято решение использовать интегрированную среду разработки программ MS VISUAL STUDIO 2019. Данная среда разработки позволяет достаточно быстро создавать приложения для Windows.

Технические средства, достаточные для оптимальной работы программы:

- Windows 7/8/10

- Видео память не менее 2Mb. Ускоритель графики не требуется.

- Оперативной памяти не менее 16 Mb.

- Процессор Intel Xeon e2620 и выше.

- Объем свободного места на жестком диске не менее 1 gb.

***2. Разработка программы.***

***2.1. Разработка программы***

Требования к графическому интерфейсу:

- Содержать привычные и понятные пользователю пункты меню, соответствующие функциям обработки.

- Ориентироваться на пользователя, который общается с программой на внешнем уровне взаимодействия.

- Удовлетворять правилу "шести" – в одну линейку меню включать не более 6 понятий, каждое из которых содержит не более 6 опций.

При создании программы в системе Visual Studio, к ним предъявляются следующие требования:

- многооконное (не менее 2-х окон);

- наличие меню;

- защита от некорректного ввода данных;

- обработка исключительных ситуаций;

- наличие сведений о программе;

- наличие сведений об авторе.

Для разработки приложения информационный терминал используется среда визуального программирования Visual Studio. В данной среде был создан проект с названием WindowsFormsApp1 . Данный проект состоит из Базы данных и четырёх форм: Form1: (см. рис. 2)

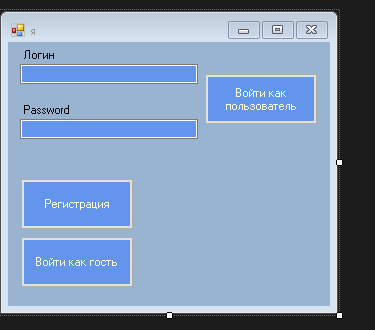


Рис. 2. Form1. Главная форма

Главная форма представляет собой стандартное окно класса TForm и содержит следующие необходимые для функционирования программы компоненты, расположенные ниже, в порядке их размещения:

1. Form1 – Основная форма проекта. Контейнер всех объектов.

2. TextBox – элемент ввода текста

3. Button – кнопки для активации действия.(3 кнопки)

4. Label 1,2 - для ввода логина и пароля

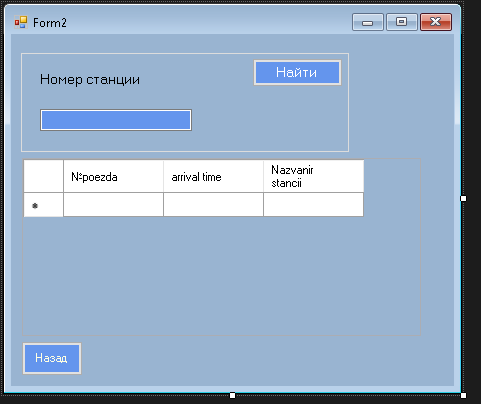


Рис. 2. Form2. Форма осн. функционала

1. Form2 – Основная форма проекта. Контейнер всех объектов.

2. TextBox – элемент ввода текста

3. Button – кнопки для активации действия.(2 кнопки)

4. Label 1 - для ввода номера станции

5. GroupBox - контейнер для размещение элементов

6. DataGridView - вывод на экран бд

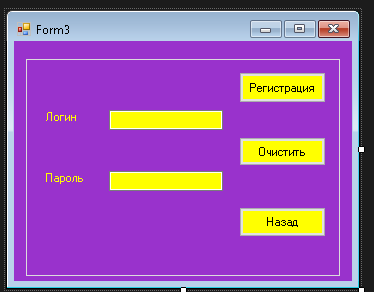


Рис. 3. Form3. Регистрация

1. Form3 – Основная форма проекта. Контейнер всех объектов.

2. TextBox 1,2 – элемент ввода текста

3. Button – кнопки для активации действия.(3 кнопки)

4. Label 1,2 - для ввода логина и пароля

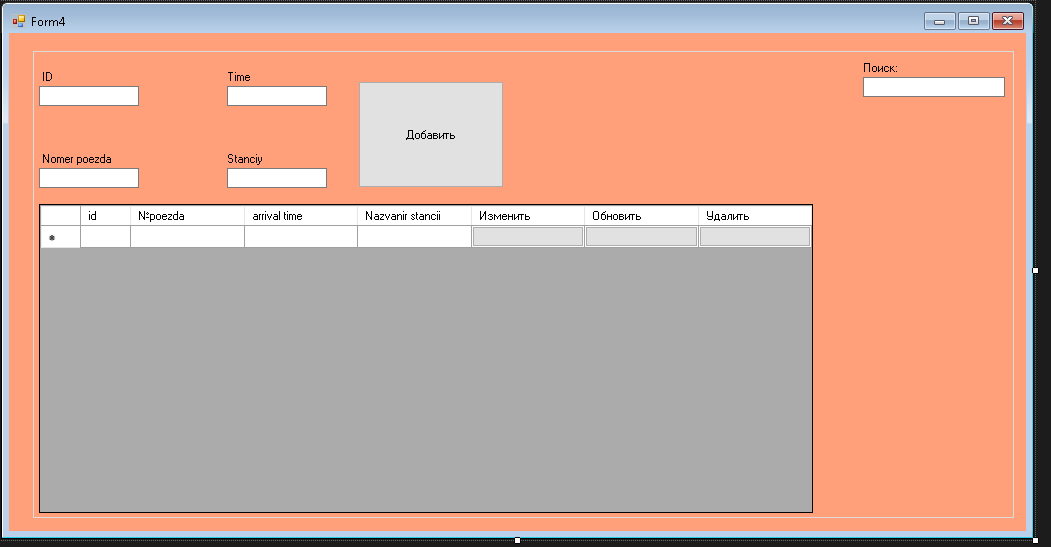


Рис. 4. Form4. Форма админа

1. Form4– Основная форма проекта. Контейнер всех объектов.

2. TextBox – элемент ввода текста

3. Button – кнопки для активации действия.(2 кнопки)

4. Label 1,2,3,4 - для ввода

5. GroupBox - контейнер для размещение элементов

6. DataGridView - вывод на экран бд

При написании программы не следует забывать о хорошем стиле программирования. После заголовка процедуры или функции записывается комментарий, содержащий поясняющий текст, а именно: назначение подпрограммы; перечень и назначении параметров; их тип.

***2.2. Спецификация программы.***

Исполнимый файл WindowsFormsApp1 программы информационного терминала имеет название соответствующее название и расположен в каталоге программы.

***2.3. Текст программы.***

Приводится полный листинг программы вместе с комментариями.

**Главная:**

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;

using System.Data.SqlClient;

namespace WindowsFormsApp1

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void label1\_Click(object sender, EventArgs e)

{

}

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

{

Form frm = new Form3();

frm.Show();

this.Hide();

}

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

{

Class1 bd = new Class1();

var login = textBox1.Text;

var password = textBox2.Text;

SqlDataAdapter adapter = new SqlDataAdapter();//для работы с БД

DataTable table = new DataTable();//для работы с БД

//ниже объект позволяющий прописать команду для выборки данных из БД

//@ul и @up заглушки, на их место можно вписать сразу переменный, но тогда команда будет менее безопасна

SqlCommand command = new SqlCommand("select \* from reg where login = @ul and password = @up", bd.getConnection());

command.Parameters.Add("@ul", SqlDbType.VarChar).Value = login;

command.Parameters.Add("@up", SqlDbType.VarChar).Value = password;

//прописываем команду которую хотим выполнить:

adapter.SelectCommand = command;

// все полученные данные трансформируем внутрь объекта table:

adapter.Fill(table);

if (textBox1.Text == "admin")

{

Form4 frm4 = new Form4();

frm4.Show();

this.Hide();

}

else if (table.Rows.Count > 0)

{

MessageBox.Show("Добро Пожаловать!");

Form2 frm2 = new Form2();

frm2.Show();

this.Hide();

}

}

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

{

Form frm = new Form2();

frm.Show();

this.Hide();

}

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

{

}

private void pictureBox1\_Click(object sender, EventArgs e)

{

}

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

{

textBox2.PasswordChar='\*';

}

private void pictureBox1\_Click\_1(object sender, EventArgs e)

{

}

}

}

**Осн.форма:**

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 System.Text.RegularExpressions;

namespace WindowsFormsApp1

{

public partial class Form2 : Form

{

public Form2()

{

InitializeComponent();

}

private void groupBox1\_Enter(object sender, EventArgs e)

{

}

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

{

Form1 frm = new Form1();

frm.Show();

this.Hide();

}

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

{

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

this.glavnTableAdapter.Fill(this.terminalDataSet.Glavn);

}

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

{

}

// Storage for the order ID value.

private int parsedOrderID;

/// <summary>

/// Verifies that an order ID is present and contains valid characters.

/// </summary>

private bool IsCustomerNameValid()

{

// Check for input in the Order ID text box.

if (textBox1.Text == "")

{

MessageBox.Show("Please specify the Order ID.");

return false;

}

// Check for characters other than integers.

else if (Regex.IsMatch(textBox1.Text, @"^\D\*$"))

{

// Show message and clear input.

MessageBox.Show("Customer ID must contain only numbers.");

textBox1.Clear();

return false;

}

else

{

// Convert the text in the text box to an integer to send to the database.

parsedOrderID = Int32.Parse(textBox1.Text);

return true;

}

}

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

{

if (IsCustomerNameValid())

{

using (SqlConnection connection = new SqlConnection(Properties.Settings.Default.connString))

{

// Define a t-SQL query string that has a parameter for orderID.

const string sql = "SELECT \* FROM Glavn WHERE id = @id";

// Create a SqlCommand object.

using (SqlCommand sqlCommand = new SqlCommand(sql, connection))

{

// Define the @orderID parameter and set its value.

sqlCommand.Parameters.Add(new SqlParameter("@id", SqlDbType.VarChar));

sqlCommand.Parameters["@id"].Value = parsedOrderID;

try

{

connection.Open();

// Run the query by calling ExecuteReader().

using (SqlDataReader dataReader = sqlCommand.ExecuteReader())

{

// Create a data table to hold the retrieved data.

DataTable dataTable = new DataTable();

// Load the data from SqlDataReader into the data table.

dataTable.Load(dataReader);

// Display the data from the data table in the data grid view.

this.dataGridView1.DataSource = dataTable;

// Close the SqlDataReader.

dataReader.Close();

}

}

catch

{

MessageBox.Show("The requested order could not be loaded into the form.");

}

}

}

}

}

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

{

}

}

}

**Регистрация:**

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;

using System.Data.SqlClient;

namespace WindowsFormsApp1

{

public partial class Form3 : Form

{

public Form3()

{

InitializeComponent();

}

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

{

Class1 bd = new Class1();

var login = textBox1.Text;

var password = textBox2.Text;

string yootoo = $"insert into reg (login,password) values('{login}','{password}')";

SqlCommand command = new SqlCommand (yootoo, bd.getConnection());

bd.openConnection();

if (command.ExecuteNonQuery() == 1)

{

MessageBox.Show("Пользователь создан!");

Form1 frm = new Form1();

frm.Show();

this.Hide();

}

else

{

MessageBox.Show("Сорян пользователь не создан!(");

}

bd.closeConnection();

}

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

{

Form1 frm = new Form1();

frm.Show();

this.Hide();

}

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

{

textBox1.Clear();

textBox2.Clear();

}

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

{

}

private void groupBox1\_Enter(object sender, EventArgs e)

{

}

}

}

**Админ:**

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;

using System.Data.SqlClient;

namespace WindowsFormsApp1

{

public partial class Form4 : Form

{

public Form4()

{

InitializeComponent();

}

private DataSet dataSet = null;

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

{

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

this.glavnTableAdapter.Fill(this.terminalDataSet.Glavn);

}

private void dataGridView1\_CellContentClick(object sender, DataGridViewCellEventArgs e)

{

}

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

{

}

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

{

Class1 bd = new Class1();

(dataGridView1.DataSource as DataTable).DefaultView.RowFilter = $"id LIKE '%{textBox1.Text}%'";

}

private void label1\_Click(object sender, EventArgs e)

{

}

private void groupBox1\_Enter(object sender, EventArgs e)

{

}

private void button1\_Click\_1(object sender, EventArgs e)

{

dataGridView1.Rows.Add(id.Text, poezd.Text, time.Text, stan.Text);

id.Text = ""; // clear

poezd.Text = "";

time.Text = "";

stan.Text = "";

}

}

}

***2.4. Описание программы.***

**Общие сведения.**

Разработанное приложение является программой для работы с базой данных для приема заказов клиентов. Программа разработана в среде Visual Studio на языке C#.

**Функциональное назначение.**

Программа предназначена для работы в организации

**Используемые технические средства.**

Для стабильной работы данной программы требуются следующие технические и программные средства:

- Windows 7|8|10.

- Видео память не менее 1Mb. Ускоритель графики не требуется.

- Оперативной памяти не менее 16 Mb.

- Процессор Intel Pentium 3200 и выше.

- Объем свободного места на жестком диске не менее 400 Мб.

- Манипуляторы: клавиатура и мышь.

**Вызов и загрузка.**

Программа запускается на исполнение либо двойным щелчком левой кнопки мыши по файлу запуска программы в проводнике Windows, либо нажатием правой кнопки мыши по указанному файлу и выбором пункта меню “Открыть”.

***2.5. Тестирование программы.***

При разработке программы использовалась среда программирования. Эта среда представляет собой набор средств быстрой разработки и отладки приложений для Windows. При разработке приложения применялся визуальный редактор форм, редактор программного кода, палитра компонентов и инспектор объектов.

Основными операциями в процессе разработки являлись: создание форм, размещение компонентов, редактирование свойств этих компонентов, формирование визуального пользовательского интерфейса, написание кода программы для обработки событий, сохранение и компиляция проекта.

Помимо указанных операций применялись дополнительные операции для отладки созданного приложения. Для этого в составе С# имеется интегральный отладчик, который позволяет обнаруживать и исправлять ошибки как на стадии написания кода, так и на стадии отладки работающей программы.

Кроме стандартных средств отладки применялась методика «силовой отладки» приложения. В код программы вставлялся дополнительный код, ставящий своей целью вывод промежуточных значений во время расчетов.

При испытании программы ставилась цель проверить работоспособность программы, корректность расчетов и отсутствие логических ошибок. Также ставилась задача исключить «человеческий фактор» возникновения исключительных ситуаций и ошибок, за счет применения методике «защиты от дурака».

Результатом тестирования и отладки стало обнаружение нескольких «узких мест» в работе программы, которые могли бы ухудшить эксплуатационные характеристики программы. Все эти недостатки были исправлены или заменены наиболее оптимальными решениями.

**Заключение.**

В ходе выполнения курсовой работы, я изучил материалы прошлых работ, а также создание приложения в с подключенным к ним бд MS SQL. Благодаря изученным материалом и ПО, смог создать неполноценное, но рабочий информационный терминал в метрополитене и бд для него.

**Список использованной литературы.**

* https://www.youtube.com