***Final Report***

Software Engineering 1

CIS – 375

Professor: Kiumi Akingbehin

Due Date: August 20th, 2018

**The Debuggers:**

**Marco Seman**

**Martin Zanaj**

**Jason Lu**

**Maricruz Zamora**

**Mohammed AlMuzel**

**Audrey White**

**Table of Contents**

[﷟HYPERLINK "bookmark://\_Toc522545971" ￼ 3](#_Toc522545971)

[﷟HYPERLINK "bookmark://\_Toc522545972" ￼ 3](#_Toc522545972)

[﷟HYPERLINK "bookmark://\_Toc522545973" ￼ 3](#_Toc522545973)

[﷟HYPERLINK "bookmark://\_Toc522545974" ￼ 3](#_Toc522545974)

[﷟HYPERLINK "bookmark://\_Toc522545975" ￼ 3](#_Toc522545975)

[﷟HYPERLINK "bookmark://\_Toc522545976" ￼ 3](#_Toc522545976)

[﷟HYPERLINK "bookmark://\_Toc522545977" ￼ 6](#_Toc522545977)

[﷟HYPERLINK "bookmark://\_Toc522545978" ￼ 11](#_Toc522545978)

[﷟HYPERLINK "bookmark://\_Toc522545979" ￼ 11](#_Toc522545979)

# **Software Requirement Specification:**

## **Summary\statement:**

In our Software Requirement Specification, we focused on analyzing the different functionalities and requirements of the program in relation to the target audience through the user interface. To better understand how the data would be connected to all functionalities, we created a data flow diagram. This made it easier to understand where there would be a need to use the data and what specific data we needed for each functionality. We created Data Models for each use case to layout the interfaces the user would experience. The Sequence Diagrams, State Diagram, Communication Diagrams were helpful for seeing the step by step actions within each use case and how the actors interact, along with the UML Diagram.

# **Design Document:**

## **Summary\statement:**

In our Design Documentation, the main focus was generating an idea for the design of our demo prototype. We decided on using a graphical user interface and came up with a system architectural design which represented the interface of our software in its entirety. Our Decomposition Diagram was helpful along the way where we broke down every element to obtain a better understanding of how deep and important a specific element is. While working on the design document, we’ve completed the majority of our prototype and delivered a quality description on each interface shown. Our group has made sure that the software created is consistent with the previous documentations submitted.

# **Test Plan:**

## **Summary of test result:**

In our test plan documentation, we’ve completed a total of five test cases, all of which consisted on testing our Graphical User Interface. In those five test cases, our group decided to use four variations of test types which include: Black Box, Boundary, Top-Bottom, and Unit Testing. The test cases completed mainly consisted on button event handlers and user entry. The results of our test cases have all been successful, and the outcome results has come out as expected.

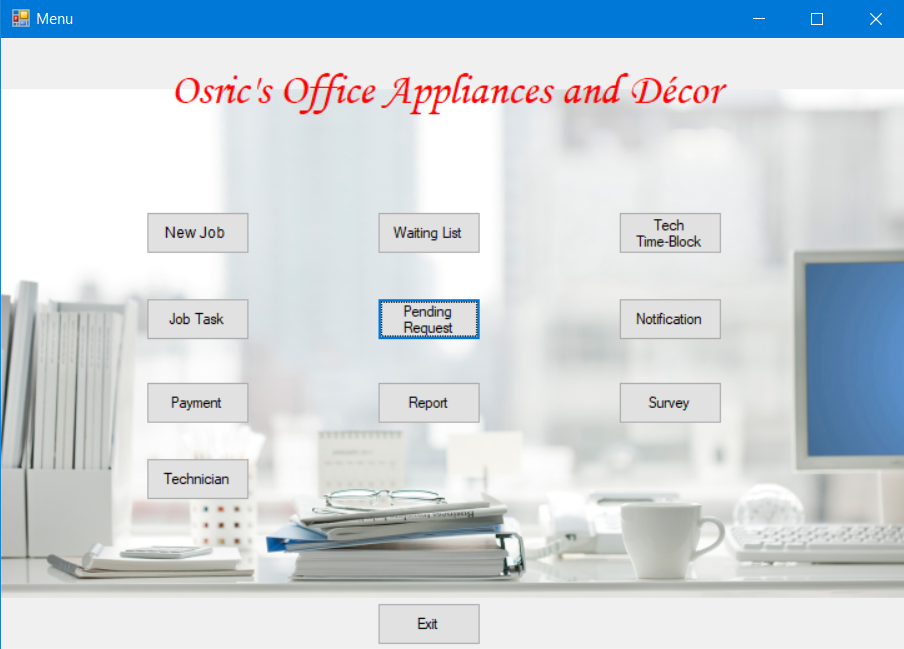
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test # | Test Type | Input | Expected Output and/or Action | Actual Output and/or Action | Pass/Fail Decision |
| 1 | Black Box Testing | Button Event Handler. The user will select exit to return to the main menu | Every interface should have the ability to go back to the main menu screen7f | We had the ability to return to the main menu | Pass |
| 2 | Boundary Testing | Zip Code  Name  Phone Number  Email | Email Invalid (if no @ entered)  Does not allow user to enter more than 15 characters for name, no more than 5 for Zip code, and no more than 10 for phone number.  Error: Missing Input (for blank boxes) | Zip Code–only allows 5 charc.  Name-only allows 15 charc.  Phone Number- only allows 10 char.  Email-no message for not entering @ | Failed |
| 3 | Boundary Testing | Correcting failed in test case 2 : Email | Email Invalid (if no @ entered) | Error message “\*Error:INVALID EMAIL” | Pass |
| 4 | Top-Bottom Testing | Button Event Handler, and Login Access | Check and see if every function is accessible by first logging in, and then checking if all the button event handlers are working as anticipated. | All button event handlers take you to the appropriate window | Pass |
| 5 | Unit Testing | User login access code  mp000000 | (Attempting incorrect user login)  Authorization Fail! Try Again. | Displays error message “Authorization Fail! Try Again.” | Pass |
| 6 | Unit Testing | User login access code  m0001 | The user is logged in | The main menu appears, giving the user the option | Pass |

## **Test Cases:**

**First Test Case – Black Box:**

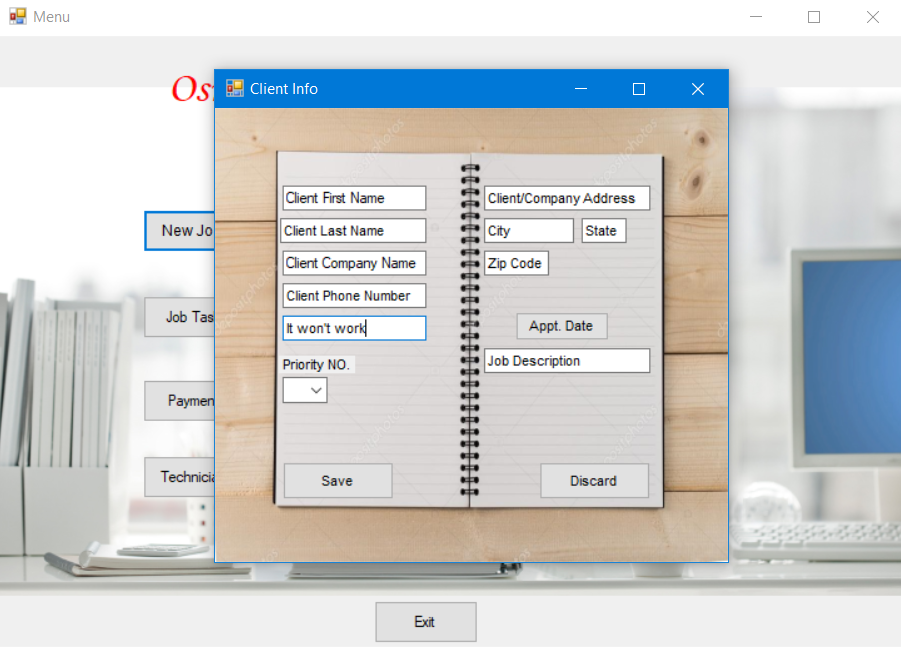
* First test is black box, which will get as an input button event handler. So, the user will be able to select exit to return to the main menu. The expected action from every interface is the ability to go back to the main menu screen, which worked as expected (Pass).



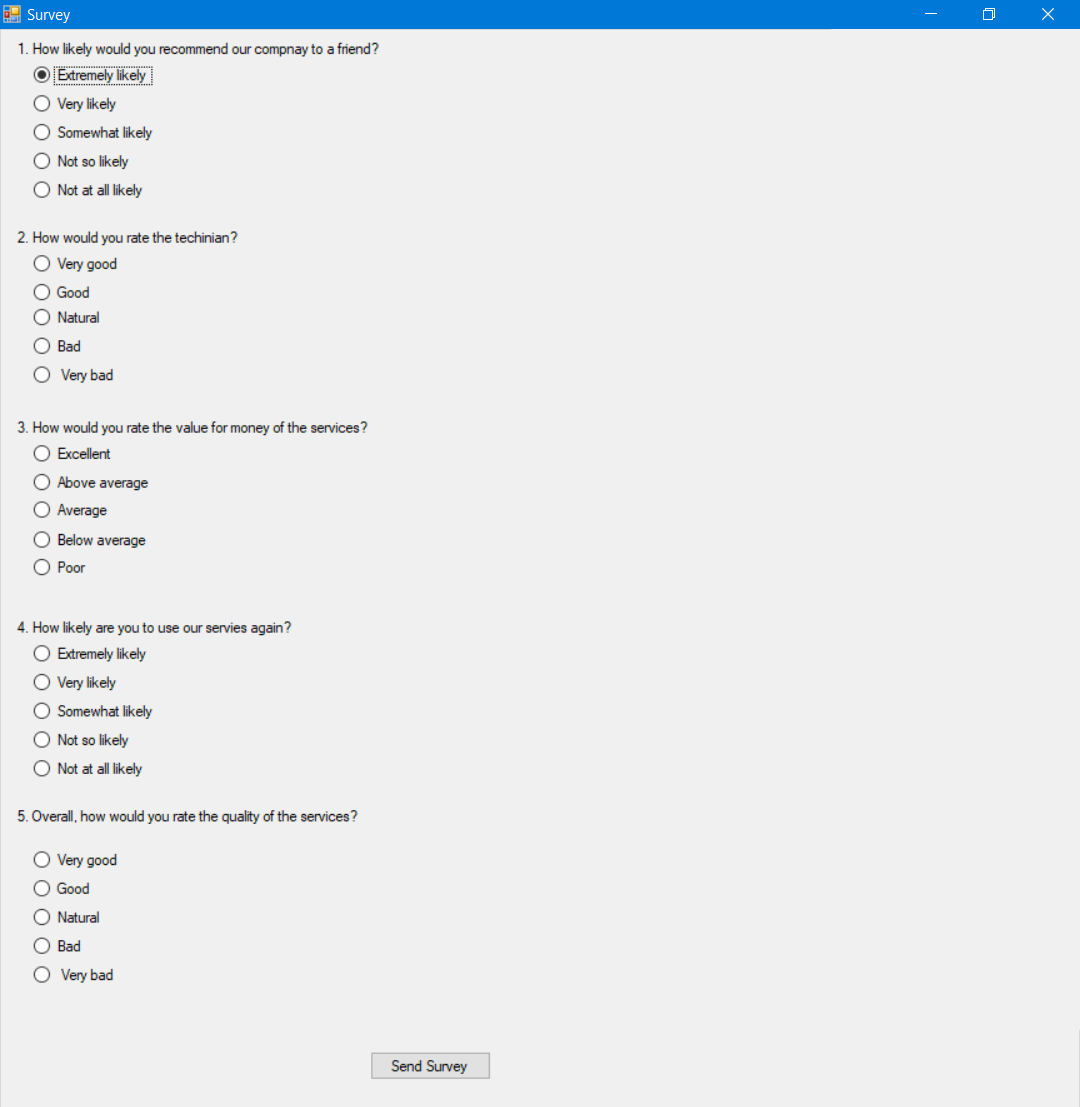


**Second Test Case – Boundary Test:**

* The second test will be boundary test, which will get as an input: the state, zip code, name, phone number. The expected action when entering invalid info is to get error message. Like when there is no “@” in email filed. Also, the application won’t accept more than 15 characters for name and no more 5 digits for the zip code, and no more 10 digits for the phone number. Which worked as expected (Pass), except for the email field and also there is no error message appeared.

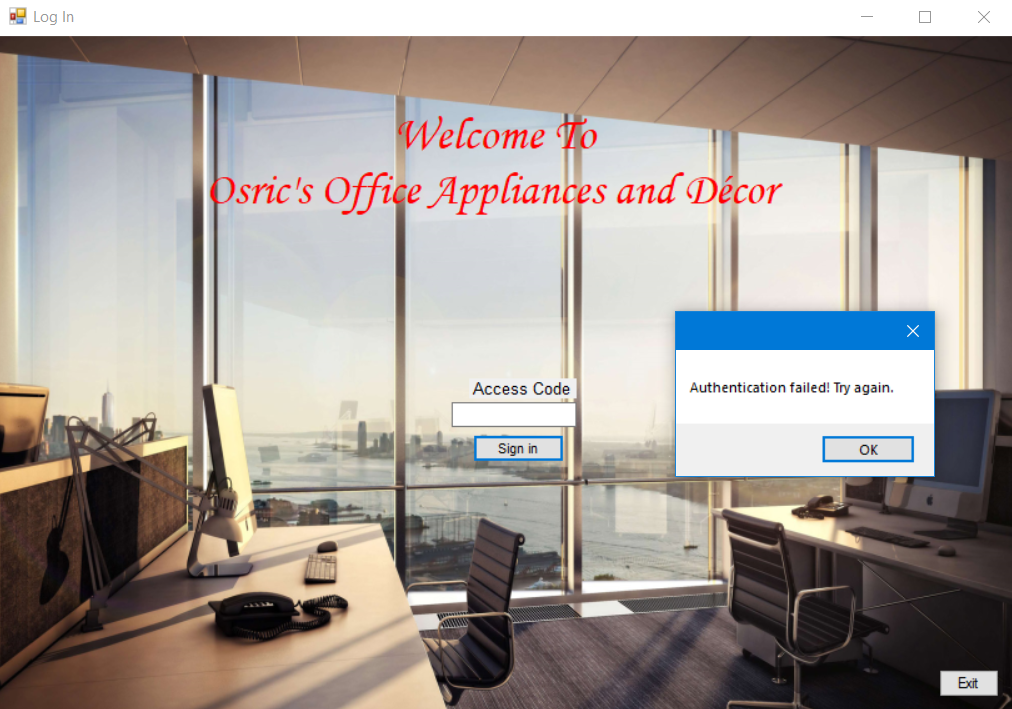


**Third Test Case – Top-Bottom Testing:**

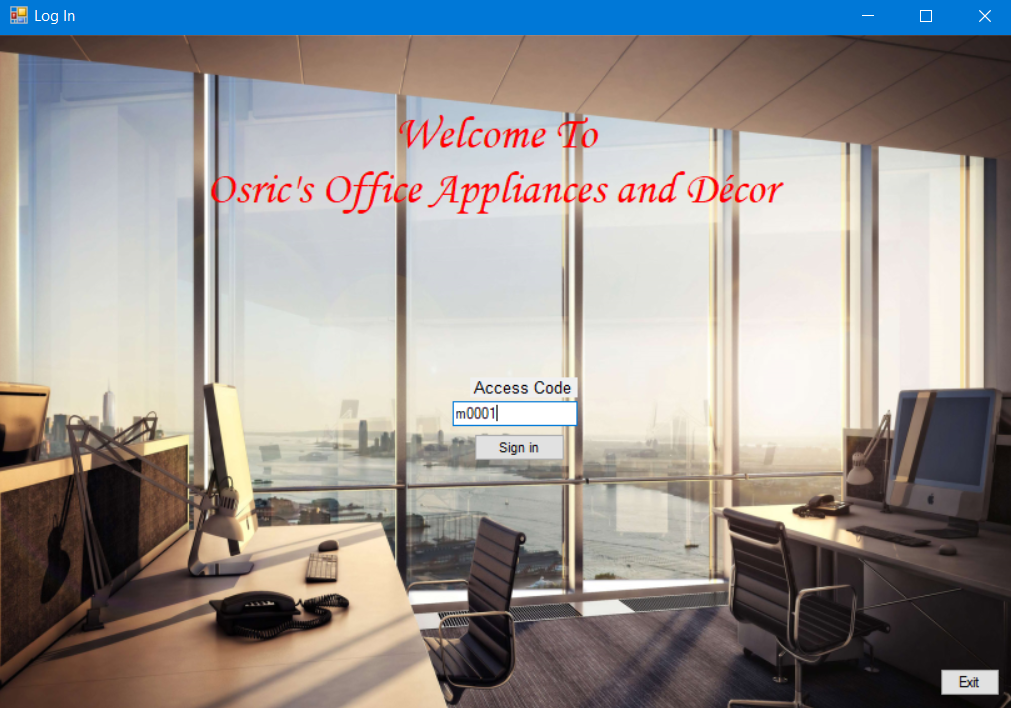
* The third test is about Top-Bottom testing, which will use button event handler, and login access as an input. The actions that should be expected on every function are accessible by first logging in, and then checking if all the button event handlers are working as anticipated, which worked as expected (Pass).

**Fourth Test Case – Unit Testing**

* The fourth test is about unit testing, which should accept as a user login access code “mp000000” only, which should expect to get error message “Authorization Fail! Try Again.” When attempting incorrect user login. Which worked as expected (Pass).



**Fifth Test Case – Unit Testing:**

* The last test is also about unit testing, which should get as a user login access code “m0001”. The expected action is to see the user logged in. Which actually worked since the main menu appears, which gives the user options.



# **Code:**

## **Code for Main Menu:**

#pragma once

#include "jobApproval.h"

#include "clientInfo.h"

#include "timeBlock.h"

#include "waitingList.h"

#include "jobTask.h"

#include "payment.h"

#include "report.h"

#include "survey.h"

#include "Notification.h"

#include "apptDate.h"

#include "TechnicianInfo.h"

#include "ExitProgram.h"

namespace CIS375Project1 {

using namespace System;

using namespace System::ComponentModel;

using namespace System::Collections;

using namespace System::Windows::Forms;

using namespace System::Data;

using namespace System::Drawing;

/// <summary>

/// Summary for mainMenu

/// </summary>

public ref class mainMenu : public System::Windows::Forms::Form

{

public:

mainMenu(void)

{

InitializeComponent();

//

//TODO: Add the constructor code here

//

}

protected:

/// <summary>

/// Clean up any resources being used.

/// </summary>

~mainMenu()

{

if (components)

{

delete components;

}

}

private: System::Windows::Forms::Label^ label1;

protected:

private: System::Windows::Forms::Button^ button1;

private: System::Windows::Forms::Button^ button2;

private: System::Windows::Forms::Button^ button3;

private: System::Windows::Forms::Button^ button4;

private: System::Windows::Forms::Button^ button5;

private: System::Windows::Forms::Button^ button6;

private: System::Windows::Forms::Button^ button7;

private: System::Windows::Forms::Button^ button8;

private: System::Windows::Forms::Button^ button9;

private: System::Windows::Forms::Button^ button10;

private: System::Windows::Forms::Button^ button11;

private:

/// <summary>

/// Required designer variable.

/// </summary>

System::ComponentModel::Container ^components;

#pragma region Windows Form Designer generated code

/// <summary>

/// Required method for Designer support - do not modify

/// the contents of this method with the code editor.

/// </summary>

void InitializeComponent(void)

{

System::ComponentModel::ComponentResourceManager^ resources = (gcnew System::ComponentModel::ComponentResourceManager(mainMenu::typeid));

this->label1 = (gcnew System::Windows::Forms::Label());

this->button1 = (gcnew System::Windows::Forms::Button());

this->button2 = (gcnew System::Windows::Forms::Button());

this->button3 = (gcnew System::Windows::Forms::Button());

this->button4 = (gcnew System::Windows::Forms::Button());

this->button5 = (gcnew System::Windows::Forms::Button());

this->button6 = (gcnew System::Windows::Forms::Button());

this->button7 = (gcnew System::Windows::Forms::Button());

this->button8 = (gcnew System::Windows::Forms::Button());

this->button9 = (gcnew System::Windows::Forms::Button());

this->button10 = (gcnew System::Windows::Forms::Button());

this->button11 = (gcnew System::Windows::Forms::Button());

this->SuspendLayout();

//

// label1

//

this->label1->AutoSize = true;

this->label1->BackColor = System::Drawing::Color::Transparent;

this->label1->Font = (gcnew System::Drawing::Font(L"Monotype Corsiva", 26.25F, System::Drawing::FontStyle::Italic, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(0)));

this->label1->ForeColor = System::Drawing::Color::Red;

this->label1->Location = System::Drawing::Point(128, 20);

this->label1->Name = L"label1";

this->label1->Size = System::Drawing::Size(458, 43);

this->label1->TabIndex = 0;

this->label1->Text = L"Osric\'s Office Appliances and D¨¦cor";

this->label1->Click += gcnew System::EventHandler(this, &mainMenu::label1\_Click);

//

// button1

//

this->button1->Font = (gcnew System::Drawing::Font(L"Arial", 9, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(0)));

this->button1->Location = System::Drawing::Point(116, 139);

this->button1->Name = L"button1";

this->button1->Size = System::Drawing::Size(83, 34);

this->button1->TabIndex = 1;

this->button1->Text = L"New Job ";

this->button1->UseVisualStyleBackColor = true;

this->button1->Click += gcnew System::EventHandler(this, &mainMenu::button1\_Click);

//

// button2

//

this->button2->Location = System::Drawing::Point(301, 208);

this->button2->Name = L"button2";

this->button2->Size = System::Drawing::Size(83, 34);

this->button2->TabIndex = 2;

this->button2->Text = L"Pending Request";

this->button2->UseVisualStyleBackColor = true;

this->button2->Click += gcnew System::EventHandler(this, &mainMenu::button2\_Click);

//

// button3

//

this->button3->Location = System::Drawing::Point(301, 139);

this->button3->Name = L"button3";

this->button3->Size = System::Drawing::Size(83, 34);

this->button3->TabIndex = 3;

this->button3->Text = L"Waiting List";

this->button3->UseVisualStyleBackColor = true;

this->button3->Click += gcnew System::EventHandler(this, &mainMenu::button3\_Click);

//

// button4

//

this->button4->Location = System::Drawing::Point(301, 275);

this->button4->Name = L"button4";

this->button4->Size = System::Drawing::Size(83, 34);

this->button4->TabIndex = 4;

this->button4->Text = L"Report";

this->button4->UseVisualStyleBackColor = true;

this->button4->Click += gcnew System::EventHandler(this, &mainMenu::button4\_Click);

//

// button5

//

this->button5->Location = System::Drawing::Point(116, 275);

this->button5->Name = L"button5";

this->button5->Size = System::Drawing::Size(83, 34);

this->button5->TabIndex = 5;

this->button5->Text = L"Payment";

this->button5->UseVisualStyleBackColor = true;

this->button5->Click += gcnew System::EventHandler(this, &mainMenu::button5\_Click);

//

// button6

//

this->button6->Location = System::Drawing::Point(116, 208);

this->button6->Name = L"button6";

this->button6->Size = System::Drawing::Size(83, 34);

this->button6->TabIndex = 6;

this->button6->Text = L"Job Task";

this->button6->UseVisualStyleBackColor = true;

this->button6->Click += gcnew System::EventHandler(this, &mainMenu::button6\_Click);

//

// button7

//

this->button7->Location = System::Drawing::Point(494, 275);

this->button7->Name = L"button7";

this->button7->Size = System::Drawing::Size(83, 34);

this->button7->TabIndex = 7;

this->button7->Text = L"Survey";

this->button7->UseVisualStyleBackColor = true;

this->button7->Click += gcnew System::EventHandler(this, &mainMenu::button7\_Click);

//

// button8

//

this->button8->Location = System::Drawing::Point(494, 139);

this->button8->Name = L"button8";

this->button8->Size = System::Drawing::Size(83, 34);

this->button8->TabIndex = 8;

this->button8->Text = L"Tech Time-Block";

this->button8->UseVisualStyleBackColor = true;

this->button8->Click += gcnew System::EventHandler(this, &mainMenu::button8\_Click);

//

// button9

//

this->button9->Location = System::Drawing::Point(494, 208);

this->button9->Name = L"button9";

this->button9->Size = System::Drawing::Size(83, 34);

this->button9->TabIndex = 9;

this->button9->Text = L"Notification";

this->button9->UseVisualStyleBackColor = true;

this->button9->Click += gcnew System::EventHandler(this, &mainMenu::button9\_Click);

//

// button10

//

this->button10->Location = System::Drawing::Point(116, 336);

this->button10->Name = L"button10";

this->button10->Size = System::Drawing::Size(83, 34);

this->button10->TabIndex = 10;

this->button10->Text = L"Technician";

this->button10->UseVisualStyleBackColor = true;

this->button10->Click += gcnew System::EventHandler(this, &mainMenu::button10\_Click);

//

// button11

//

this->button11->Location = System::Drawing::Point(301, 452);

this->button11->Name = L"button11";

this->button11->Size = System::Drawing::Size(83, 34);

this->button11->TabIndex = 11;

this->button11->Text = L"Exit";

this->button11->UseVisualStyleBackColor = true;

this->button11->Click += gcnew System::EventHandler(this, &mainMenu::button11\_Click);

//

// mainMenu

//

this->AutoScaleDimensions = System::Drawing::SizeF(6, 13);

this->AutoScaleMode = System::Windows::Forms::AutoScaleMode::Font;

this->BackgroundImage = (cli::safe\_cast<System::Drawing::Image^>(resources->GetObject(L"$this.BackgroundImage")));

this->BackgroundImageLayout = System::Windows::Forms::ImageLayout::Zoom;

this->ClientSize = System::Drawing::Size(724, 489);

this->Controls->Add(this->button11);

this->Controls->Add(this->button10);

this->Controls->Add(this->button9);

this->Controls->Add(this->button8);

this->Controls->Add(this->button7);

this->Controls->Add(this->button6);

this->Controls->Add(this->button5);

this->Controls->Add(this->button4);

this->Controls->Add(this->button3);

this->Controls->Add(this->button2);

this->Controls->Add(this->button1);

this->Controls->Add(this->label1);

this->Name = L"mainMenu";

this->StartPosition = System::Windows::Forms::FormStartPosition::CenterParent;

this->Text = L"Menu";

this->ResumeLayout(false);

this->PerformLayout();

}

#pragma endregion

private: System::Void button1\_Click(System::Object^ sender, System::EventArgs^ e) {

clientInfo^form = gcnew clientInfo();

form->Show();

}

private: System::Void button2\_Click(System::Object^ sender, System::EventArgs^ e) {

jobApproval ^form1 = gcnew jobApproval();

form1->Show();

}

private: System::Void button8\_Click(System::Object^ sender, System::EventArgs^ e) {

timeBlock ^form2 = gcnew timeBlock();

form2->Show();

}

private: System::Void button3\_Click(System::Object^ sender, System::EventArgs^ e) {

waitingList ^form3 = gcnew waitingList();

form3->Show();

}

private: System::Void button6\_Click(System::Object^ sender, System::EventArgs^ e) {

jobTask ^form4 = gcnew jobTask();

form4->Show();

}

private: System::Void button5\_Click(System::Object^ sender, System::EventArgs^ e) {

payment ^form5 = gcnew payment();

form5->Show();

}

private: System::Void button4\_Click(System::Object^ sender, System::EventArgs^ e) {

report ^form6 = gcnew report();

form6->Show();

}

private: System::Void button7\_Click(System::Object^ sender, System::EventArgs^ e) {

survey ^form7 = gcnew survey();

form7->Show();

}

private: System::Void button9\_Click(System::Object^ sender, System::EventArgs^ e) {

Notification ^form8 = gcnew Notification();

form8->Show();

}

private: System::Void button10\_Click(System::Object^ sender, System::EventArgs^ e) {

TechnicianInfo^form9 = gcnew TechnicianInfo();

form9->Show();

}

private: System::Void button11\_Click(System::Object^ sender, System::EventArgs^ e) {

ExitProgram ^form10 = gcnew ExitProgram();

form10->Show();

}

private: System::Void label1\_Click(System::Object^ sender, System::EventArgs^ e) {

}

};

}