**TON DUC THANG UNIVERSITY**

**FACULTY OF INFORMATION TECHNOLOGY**

**SOFTWARE ENGINEERING**

**FINAL REPORT**

**Name: Lieu Dang Khoa**

**Vi Nguyen Thanh Dat**

**Supervisor: Pham Thai Ky Trung**

**HO CHI MINH CITY, 2022**

# TABLE OF CONTENTS

[**TABLE OF CONTENTS**](#_gjdgxs) **2**

[**1.INTRODUCTION**](#_suqrbs6udwmg) **4**

[**2. Project Management Plan**](#_vbwl1x46aa3z) **5**

[2.1. Project Organization](#_gqqcuee41q39) 5

[2.2. Lifecycle Model Used](#_9l6fxiysq83j) 5

[2.3. Risk Analysis](#_luslm41whhai) 5

[2.4. Hardware and Software Resource Requirements](#_ywpf1k7g39n) 8

[2.5. Deliverables and Schedule](#_9no3g5wjs84u) 9

[2.6. Monitoring, Reporting, and Controlling Mechanisms](#_622dslsgnsom) 9

[2.7. Professional Standards](#_qnheq546i76y) 9

[2.8. Evidence all the artifacts have been placed under configuration management](#_w8nlngw0ud7e) 9

[2.9. Impact of the project on individuals and organizations](#_1ovw1hu36a1f) 9

[**3. Requirement Specifications**](#_t9ogyvj14n5n) **10**

[3.1. Stakeholders for the system](#_2l77fshk4vu5) 10

[3.2. Use case model](#_ezn1szf1yodw) 10

[3.2.1. Graphical use case model](#_xepbfg2zpczo) 10

[3.2.2. Textual Description for each use case](#_y0bowafsbzx5) 10

[3.3. Functional requirements](#_wkmyy4dvh0cb) 10

[3.4. Non-functional requirements](#_867ey4zgfwci) 10

[**4. Architecture**](#_qt4bzdtsybiu) **11**

[4.1. Architectural style(s) used](#_kgvk386ixjgb) 11

[4.2. Architectural model](#_gtielt310cae) 11

[4.3. Technology, software, and hardware used](#_qirudw3etlqd) 11

[4.4. Rationale for your architectural style and model](#_7umst8fx00mo) 11

[● Saves time: UML can help developers save time by automating some of the design processes](#_wxetbsm005tq) 11

[● Enhances collaboration: UML allows different software developers to work on the same project by providing a common language](#_ar9i3i5imxtm) 11

[● Provides a better understanding of a system: UML can help them understand the system better and identify potential challenges](#_i5u5kumsmtbc) 12

[● Unifies design: UML provides a standard way to design software and systems](#_k75him7n3eme) 12

[**5. Design**](#_7sldb1g962i0) **12**

[5.2. Static model – class diagrams](#_jxc84b381y6r) 14

[5.3. Dynamic model – sequence diagrams](#_cdg9rgles245) 15

[5.4. Rationale for your detailed design model](#_dkqgv18y5204) 16

[5.5. Traceability from requirements to detailed design model](#_j6sgn8c4t78i) 16

[**6. Test Plan**](#_jcpcp2dbp60) **16**

[6.1. Requirements/specifications-based system level test cases](#_sob8pqq9dl7r) 16

[6.2. Traceability of test cases to use cases](#_v0t4fbiq01y6) 16

[6.3. Techniques used for test generation](#_xw5gkobm3miu) 16

[6.4. Assessment of the goodness of your testsuite (Which metrics were used for such](#_qj38bk4144ed) 16

[**7. Demo**](#_nz286fg5s47r) **16**

[7.1. Database](#_ni19ao8lsisq) 16

[7.2. Source code](#_fvdav7bp2tnz) 18

[7.3. Testing](#_svjlys8saol4) 63

# 1.INTRODUCTION

1.1 Purpose and Scope  
To introduce our product, what it can do, different user cases

1.2. Product Overview (including capabilities, scenarios for using the product, etc.)

Our product helps distribute Supplement Facts products to various companies (agents). Overall, the accountant has the ability to create Good Received when the company imports goods, they shall be able to create Goods Delivery Note to deliver goods to agents, accountants shall be able to view incoming/outgoing stock report, and agents can choose their preferred payment method.

1.3. Structure of the Document

1. Introduction
2. Project Management Plan
3. Requirement Specifications
4. Architecture
5. Design
6. Test Plan
7. Demo
8. References

1.4. Terms, Acronyms, and Abbreviations  
Payment method: methods to purchase goods

Payment status: check if payment has been finished

# 2. Project Management Plan

## 2.1. Project Organization

Our team consists of two members:

* Lieu Dang Khoa
* Vi Nguyen Thanh Dat

## 2.2. Lifecycle Model Used

Waterfall model

## 2.3. Risk Analysis

* **Identify the risks**
  + The risks we face are missing the deadline to submit our finished product
  + We failed to do well in our report due to numerous parts we have to do
  + We risked not designing our product look good enough
  + The database we construct might not be sufficient enough
  + Internet is quite slow in my house and this compromise the whole project
* **Analyze the risks:** We will try to determine the likelihood of each risk and their consequences
  + Missing deadline: 50%, we will get low score on our final, which will be disastrous
  + Failed to perform well in the report: 60%, our score will definitely be affected as report accounts for 30% of total points
  + Bad looking product: 50%, I am not good at designing and so do my friend, our score might be affected
  + Insufficient database: 50%, we don’t have enough data for our product to work with
  + Bad internet: 30%, slow internet connection is a problem but rarely does it affect the overall product
* **Develop a** [**risk management**](https://searchcompliance.techtarget.com/definition/risk-management) **plan:**

| Risks | Impact | Priority | Status |
| --- | --- | --- | --- |
| Missing deadline | Affects overall score | High | Open |
| Failed to perform well in the report | Affects overall score + project schedule | Medium | Open |
| Bad looking product | Affects overall score + cost | Medium | Open |
| Insufficient database | Affects overall product | Medium | Open |
| Bad internet | Affects project schedule + can compromise project | Low | Open |

* **Implement the risk management plan:**

| Risks | Measures to tackle | Priority | Status |
| --- | --- | --- | --- |
| Missing deadline | Rush everyone to finish soon, stay up late at night to do project | High | Open |
| Failed to perform well in the report | Asking for advice from other teams | Medium | Open |
| Bad looking product | Asking for advice from my classmates who are good at designing | Medium | Open |
| Insufficient database | Put more time into research a better database | Medium | Open |
| Bad internet | Go to the internet cafe | Low | Open |

## 2.4. Hardware and Software Resource Requirements

Visual Studio: Visual Studio is an Integrated Development Environment(IDE) developed by Microsoft to develop GUI(Graphical User Interface), console, Web applications, web apps, mobile apps, cloud, and web services.

Windows forms application: is a graphical class library included as part of Microsoft .NET, .NET Framework and is an event-driven application that runs on the desktop computer. A Windows forms application will normally have a collection of controls such as labels, textboxes, list boxes, etc.

ASP.NET MVC is basically a web development framework, which combines the features of MVC (Model-View-Controller) architecture, ideas and techniques from Agile development, and the best parts of the existing ASP.NET platform. It is a complete alternative to traditional ASP.NET Web Forms.2.5. Deliverables and Schedule. It can be used for building websites and web applications on .NET Framework using HTML, CSS, and JavaScript. ASP.NET MVC 5 is a web framework based on Model-View-Controller (MVC) architecture.

## 2.5. Deliverables and Schedule

Due: 12:00 14/12/2022

## 2.6. Monitoring, Reporting, and Controlling Mechanisms

## 2.7. Professional Standards

Skills: knowledge of C#, how to create a windows application, how to create a website, knowledge of Visual Studio overall

Experience: At least 6 months of experience

Behaviors: Finish the product on time, be quick and efficient

## 2.8. Evidence all the artifacts have been placed under configuration management

## 2.9. Impact of the project on individuals and organizations

(Include a description of what impact your project will have on individuals and society)

This product will help a company manage and sell supplemental fact products to other companies.

I can use this product to help sell certain products in a small scale operation, which will increase the local economy

# 3. Requirement Specifications

## 3.1. Stakeholders for the system

We fund the whole project

## 3.2. Use case model

### 3.2.1. Graphical use case model

### 3.2.2. Textual Description for each use case

## 3.3. Functional requirements

1. Accountants should be able to create Goods Received when the company imports goods.
2. Admin can log in to their account
3. The website will always be online
4. Agents shall be able to place an order items and choose their preferred payment methods
5. Accountants shall be able to create Goods Delivery Note to deliver goods to agents
6. Accountants shall be able to view incoming/outgoing stock report,

## 3.4. Non-functional requirements

1. All forms shall be inside the main form
2. There should only be one account of the administrator
3. The website will have a 99% uptime
4. There will be multiple payment options for users to choose from

## 

# **4.** **Architecture**

## 4.1. Architectural style(s) used

Data centered architecture

## 4.2. Architectural model

UML

## 4.3. Technology, software, and hardware used

Visual Studio

Microsoft SQL Server Management Studio

SQL Server

## 4.4. Rationale for your architectural style and model

I use the data centered architecture due to the advantages of this architecture style

* Repository of data is independent of clients
* Client work independent of each other
* It may be simple to add additional clients.
* Modification can be very easy

I use the UML model due to the advantages of it

### Saves time: UML can help developers save time by automating some of the design processes

### Enhances collaboration: UML allows different software developers to work on the same project by providing a common language

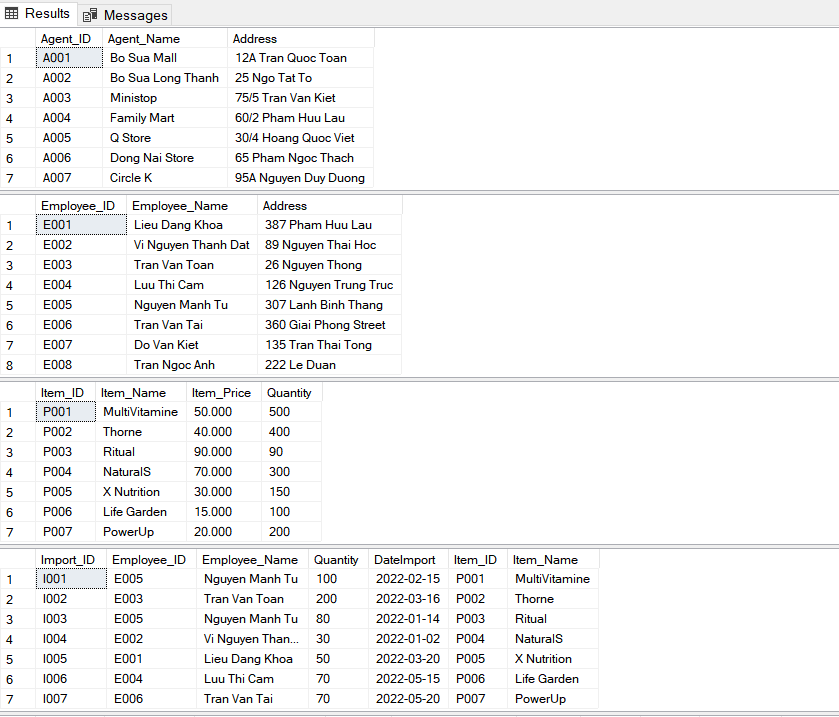
### Provides a better understanding of a system: UML can help them understand the system better and identify potential challenges

### Unifies design: UML provides a standard way to design software and systems

# 5. Design

5.1. Database design

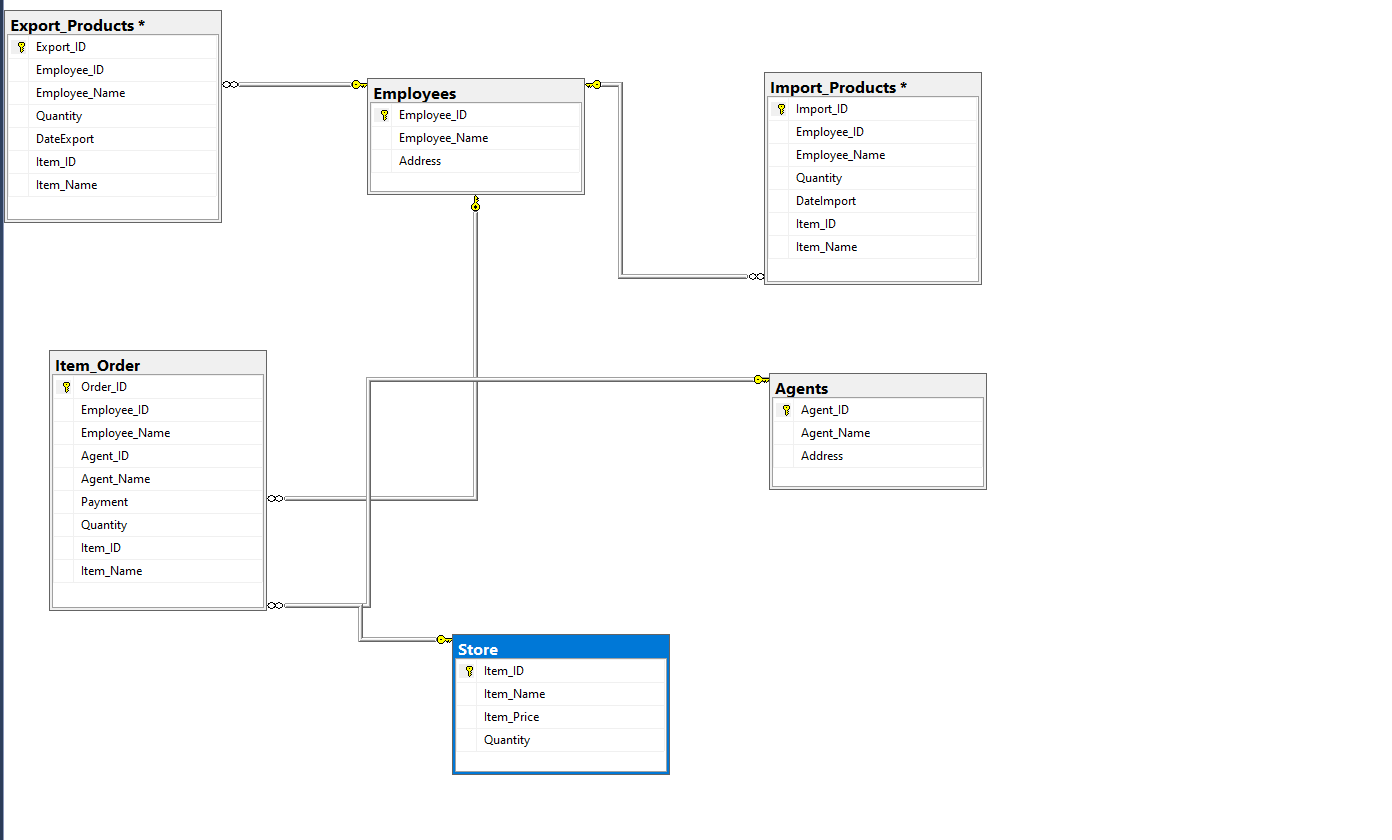
Database used:



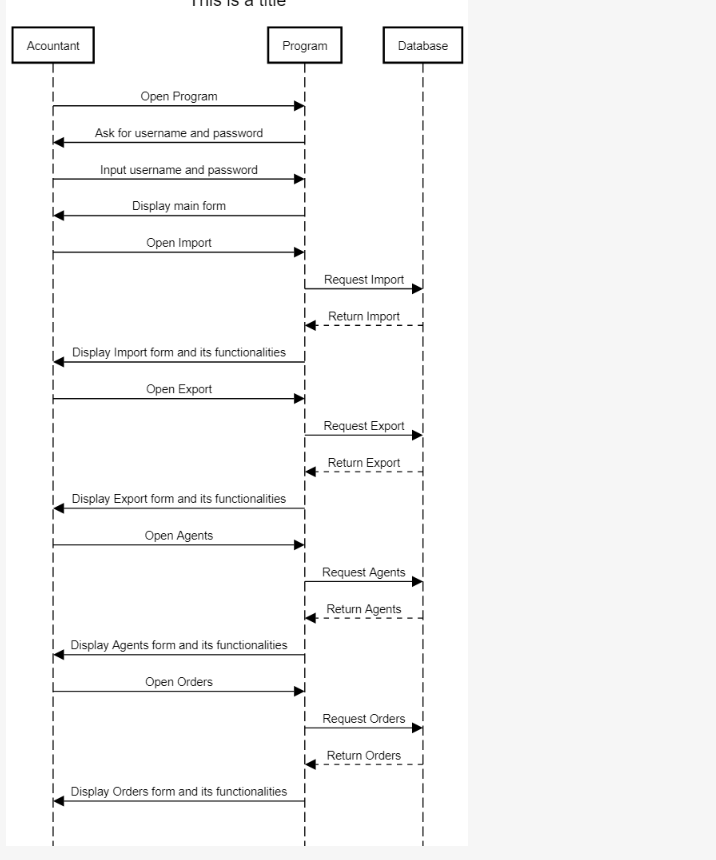
## 

## 

## 5.2. Static model – class diagrams



## 5.3. Dynamic model – sequence diagrams



## 5.4. Rationale for your detailed design model

## 5.5. Traceability from requirements to detailed design model

# 6. Test Plan

## 6.1. Requirements/specifications-based system level test cases

Visual Studio application, SQL Server, internet connected computer.

## 6.2. Traceability of test cases to use cases

## 6.3. Techniques used for test generation

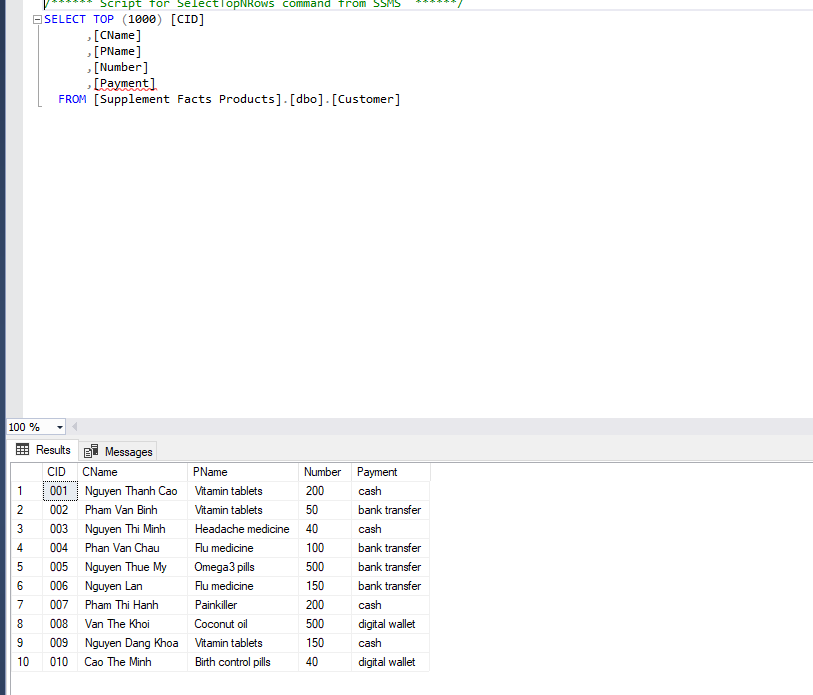
## 6.4. Assessment of the goodness of your testsuite (Which metrics were used for such

assessment?)

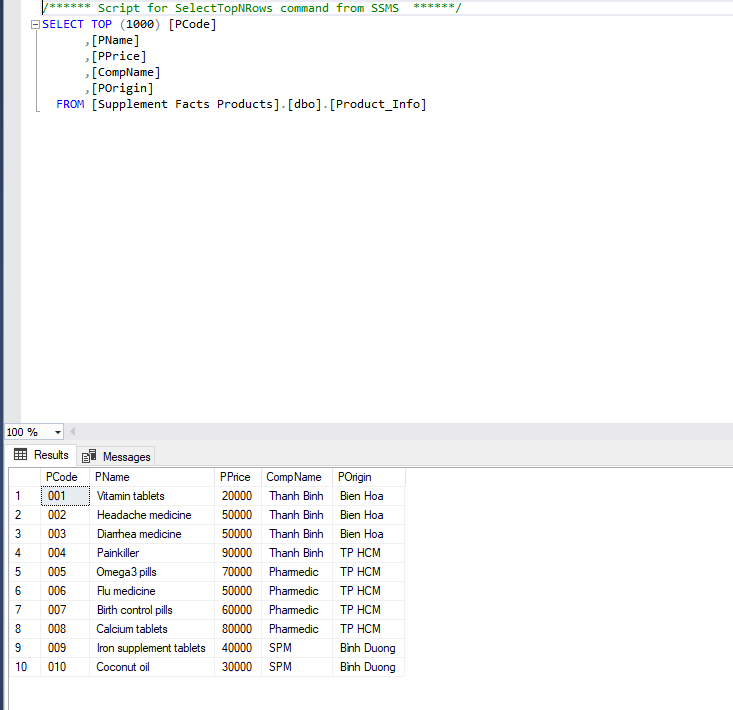
# 7. Demo

## 7.1. Database

Customer



Product\_Info



## 7.2. Source code

C# Form:

1. frmMain

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 Final

{

public partial class frmMain : Form

{

public frmMain()

{

InitializeComponent();

}

public void frmLogin()

{

Form frm = new frmLogin();

frm.ShowDialog();

}

/\*

public void xemdanhmuc(int intDanhMuc)

{

Form frm = new frmXemDanhMuc();

frm.Text = intDanhMuc.ToString();

frm.ShowDialog();

}\*/

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

{

frmLogin();

}

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

{

frmLogin();

}

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

{

DialogResult answer;

answer = MessageBox.Show("Are you sure?", "Exit", MessageBoxButtons.YesNo, MessageBoxIcon.Question);

if (answer == DialogResult.Yes) Application.Exit();

}

private void show\_panel(Form form)

{

this.pContainer.Controls.Clear();

this.pContainer.Visible = true;

form.Dock = DockStyle.Fill;

form.TopLevel = false;

form.TopMost = true;

form.FormBorderStyle = FormBorderStyle.None;

this.pContainer.Controls.Add(form);

form.Show();

}

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

{

Form f = new Import();

show\_panel(f);

}

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

{

Form f = new Export();

show\_panel(f);

}

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

{

Form f = new Report();

show\_panel(f);

}

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

{

this.pContainer.Hide();

}

}

}

1. frmLogin

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 static System.Windows.Forms.VisualStyles.VisualStyleElement;

namespace Final

{

public partial class frmLogin : Form

{

public frmLogin()

{

InitializeComponent();

}

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

{

}

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

{

}

private void label2\_Click(object sender, EventArgs e)

{

}

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

{

}

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

{

if ((txtUsername.Text == "Admin") && (txtPassword.Text == "123"))

this.Close();

else

{

MessageBox.Show("Wrong password!!!", "Annoucement", MessageBoxButtons.OKCancel, MessageBoxIcon.Warning);

txtUsername.Focus();

}

}

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

{

DialogResult answer;

answer = MessageBox.Show("Are you sure ?", "Answer", MessageBoxButtons.YesNo, MessageBoxIcon.Question);

if (answer == DialogResult.Yes)

Application.Exit();

}

}

}

1. Import

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Configuration;

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 Final

{

public partial class Import : Form

{

String strConn = ConfigurationManager.ConnectionStrings["MyConn"].ConnectionString;

public Import()

{

InitializeComponent();

}

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

{

int index = e.RowIndex;

if (index < 0 || index >= dataGridView1.RowCount)

return;

try

{

DataGridViewRow row = dataGridView1.Rows[index];

String Pcode = Convert.ToString(row.Cells[0].Value);

String PName = Convert.ToString(row.Cells[1].Value);

String Price = Convert.ToString(row.Cells[2].Value);

String CompName = Convert.ToString(row.Cells[3].Value);

String Origin = Convert.ToString(row.Cells[4].Value);

txtPCode.Text = Pcode;

txtPName.Text = PName;

txtPrice.Text = Price;

txtCompName.Text = CompName;

txtOrigin.Text = Origin;

}

catch (Exception ex)

{

throw new Exception("Error:" + ex.Message);

}

}

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

{

SqlConnection conn = new SqlConnection(Program.strConn);

conn.Open();

SqlCommand cmd = new SqlCommand("SELECT \* FROM Product\_info ", conn);

SqlDataAdapter adapter = new SqlDataAdapter(cmd);

DataTable dt = new DataTable();

adapter.Fill(dt);

if (dt.Rows.Count > 0)

{

dataGridView1.DataSource = dt;

}

else {

MessageBox.Show("No data");

}

adapter.Dispose();

}

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

{

}

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

{

}

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

{

}

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

{

}

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

{

}

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

{

}

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

{

}

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

{

}

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

{

}

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

{

}

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

{

}

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

{

SqlConnection conn = new SqlConnection(Program.strConn);

conn.Open();

String sSQL = "INSERT INTO Product\_Info (PCode, PName, PPrice, CompName, POrigin) VALUES (@PCode, @PName, @PPrice, @CompName, @POrigin)";

SqlCommand cmd = new SqlCommand(sSQL, conn);

cmd.Parameters.Add(new SqlParameter("@PCode", txtPCode.Text));

cmd.Parameters.Add(new SqlParameter("@PName", txtPName.Text));

cmd.Parameters.Add(new SqlParameter("@PPrice", txtPrice.Text));

cmd.Parameters.Add(new SqlParameter("@CompName", txtCompName.Text));

cmd.Parameters.Add(new SqlParameter("@POrigin", txtOrigin.Text));

try

{

cmd.ExecuteNonQuery();

}

catch (Exception ex)

{

throw new Exception("Error:" + ex.Message);

}

MessageBox.Show("Save successfully!");

dataGridView1.Refresh();

}

}

}

1. Export

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Configuration;

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 Final

{

public partial class Export : Form

{

String strConn = ConfigurationManager.ConnectionStrings["MyConn"].ConnectionString;

public Export()

{

InitializeComponent();

}

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

{

int index = e.RowIndex;

if (index < 0 || index >= dataGridView1.RowCount)

return;

try

{

DataGridViewRow row = dataGridView1.Rows[index];

String CID = Convert.ToString(row.Cells[0].Value);

String CName = Convert.ToString(row.Cells[1].Value);

String PName = Convert.ToString(row.Cells[2].Value);

String Number = Convert.ToString(row.Cells[3].Value);

txtCID.Text = CID;

txtCName.Text = CName;

txtPName.Text = PName;

txtNum.Text = Number;

}

catch (Exception ex)

{

throw new Exception("Error:" + ex.Message);

}

}

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

{

SqlConnection conn = new SqlConnection(Program.strConn);

conn.Open();

SqlCommand cmd = new SqlCommand("SELECT \* FROM Customer ", conn);

SqlDataAdapter adapter = new SqlDataAdapter(cmd);

DataTable dt = new DataTable();

adapter.Fill(dt);

if (dt.Rows.Count > 0)

{

dataGridView1.DataSource = dt;

}

else

{

MessageBox.Show("No data");

}

adapter.Dispose();

}

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

{

SqlConnection conn = new SqlConnection(Program.strConn);

conn.Open();

String sSQL = "INSERT INTO Customer (CID, CName, PName, Number) VALUES (@CID, @CName, @PName, @Number)";

SqlCommand cmd = new SqlCommand(sSQL, conn);

cmd.Parameters.Add(new SqlParameter("@CID", txtCID.Text));

cmd.Parameters.Add(new SqlParameter("@CName", txtCName.Text));

cmd.Parameters.Add(new SqlParameter("@PName", txtPName.Text));

cmd.Parameters.Add(new SqlParameter("@Number", txtNum.Text));

try

{

cmd.ExecuteNonQuery();

}

catch (Exception ex)

{

throw new Exception("Error:" + ex.Message);

}

MessageBox.Show("Save successfully!");

dataGridView1.Refresh();

}

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

{

SqlConnection conn = new SqlConnection(Program.strConn);

conn.Open();

String sSQL = "UPDATE Customer SET CName = @CName, PName = @PName, Number = @Number WHERE CID = @CID";

SqlCommand cmd = new SqlCommand(sSQL, conn);

cmd.Parameters.Add(new SqlParameter("@CID", txtCID.Text));

cmd.Parameters.Add(new SqlParameter("@CName", txtCName.Text));

cmd.Parameters.Add(new SqlParameter("@PName", txtPName.Text));

cmd.Parameters.Add(new SqlParameter("@Number", txtNum.Text));

try

{

cmd.ExecuteNonQuery();

}

catch (Exception ex)

{

throw new Exception("Error:" + ex.Message);

}

MessageBox.Show("Edit successfully!");

dataGridView1.Refresh();

}

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

{

SqlConnection conn = new SqlConnection(Program.strConn);

conn.Open();

String sSQL = "DELETE FROM Customer WHERE CID = @CID";

SqlCommand cmd = new SqlCommand(sSQL, conn);

cmd.Parameters.Add(new SqlParameter("@CID", txtCID.Text));

try

{

cmd.ExecuteNonQuery();

}

catch (Exception ex)

{

throw new Exception("Error:" + ex.Message);

}

MessageBox.Show("Delete successfully!");

dataGridView1.Refresh();

}

}

}

1. Report

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 static System.Windows.Forms.VisualStyles.VisualStyleElement;

namespace Final

{

public partial class Report : Form

{

public Report()

{

InitializeComponent();

}

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

{

int index = e.RowIndex;

if (index < 0 || index >= dataGridView1.RowCount)

return;

try

{

DataGridViewRow row = dataGridView1.Rows[index];

String CID = Convert.ToString(row.Cells[0].Value);

String CName = Convert.ToString(row.Cells[1].Value);

String PName = Convert.ToString(row.Cells[2].Value);

String Number = Convert.ToString(row.Cells[3].Value);

txtCID.Text = CID;

txtCName.Text = CName;

txtPName.Text = PName;

txtNum.Text = Number;

}

catch (Exception ex)

{

throw new Exception("Error:" + ex.Message);

}

}

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

{

SqlConnection conn = new SqlConnection(Program.strConn);

conn.Open();

SqlCommand cmd = new SqlCommand("SELECT TOP 5 \* FROM Customer ORDER BY Number DESC", conn);

SqlDataAdapter adapter = new SqlDataAdapter(cmd);

DataTable dt = new DataTable();

adapter.Fill(dt);

if (dt.Rows.Count > 0)

{

dataGridView1.DataSource = dt;

}

else

{

MessageBox.Show("No data");

}

adapter.Dispose();

}

}

}

In Program.cs we run file “frmMain”

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Configuration;

using System.Data.SqlClient;

namespace Final

{

internal static class Program

{

/// <summary>

/// The main entry point for the application.

/// </summary>

public static string strConn = "";

[STAThread]

static void Main()

{

strConn = ConfigurationManager.ConnectionStrings["MyConn"].ConnectionString;

Application.EnableVisualStyles();

Application.SetCompatibleTextRenderingDefault(false);

Application.Run(new frmMain());

}

}

}

Web MVC:

Views:

Customers:

@model IEnumerable<Website.Models.Customer>

@{

ViewBag.Title = "Index";

}

<h2>Click Order to purchase products:</h2>

<p>

@Html.ActionLink("Order", "Create")

</p>

<table class="table">

<tr>

<th>

@Html.DisplayNameFor(model => model.CName)

</th>

<th>

@Html.DisplayNameFor(model => model.PName)

</th>

<th>

@Html.DisplayNameFor(model => model.Number)

</th>

<th>

@Html.DisplayNameFor(model => model.Payment)

</th>

<th></th>

</tr>

@foreach (var item in Model) {

<tr>

<td>

@Html.DisplayFor(modelItem => item.CName)

</td>

<td>

@Html.DisplayFor(modelItem => item.PName)

</td>

<td>

@Html.DisplayFor(modelItem => item.Number)

</td>

<td>

@Html.DisplayFor(modelItem => item.Payment)

</td>

</tr>

}

</table>

Product\_Info:

@model IEnumerable<Website.Models.Product\_Info>

@{

ViewBag.Title = "Index";

}

<h2>Index</h2>

<table class="table">

<tr>

<th>

@Html.DisplayNameFor(model => model.PName)

</th>

<th>

@Html.DisplayNameFor(model => model.PPrice)

</th>

<th>

@Html.DisplayNameFor(model => model.CompName)

</th>

<th>

@Html.DisplayNameFor(model => model.POrigin)

</th>

<th></th>

</tr>

@foreach (var item in Model) {

<tr>

<td>

@Html.DisplayFor(modelItem => item.PName)

</td>

<td>

@Html.DisplayFor(modelItem => item.PPrice)

</td>

<td>

@Html.DisplayFor(modelItem => item.CompName)

</td>

<td>

@Html.DisplayFor(modelItem => item.POrigin)

</td>

</tr>

}

</table>

Home:

@{

ViewBag.Title = "Order";

}

<h2>Online Order:</h2>

<div>

<div class="row">

<p>

@Html.ActionLink("Availabel Products:", "Index", "Product\_Info")

</p>

<p>

@Html.ActionLink("Order now:", "Index", "Customers")

</p>

</div>

</div>

Controllers:  
 CustomerController.cs:

using System;

using System.Collections.Generic;

using System.Data;

using System.Data.Entity;

using System.Linq;

using System.Net;

using System.Web;

using System.Web.Mvc;

using Website.Models;

namespace Website.Controllers

{

public class CustomersController : Controller

{

private Conn db = new Conn();

// GET: Customers

public ActionResult Index()

{

return View(db.Customers.ToList());

}

// GET: Customers/Details/5

public ActionResult Details(string id)

{

if (id == null)

{

return new HttpStatusCodeResult(HttpStatusCode.BadRequest);

}

Customer customer = db.Customers.Find(id);

if (customer == null)

{

return HttpNotFound();

}

return View(customer);

}

// GET: Customers/Create

public ActionResult Create()

{

return View();

}

// POST: Customers/Create

// To protect from overposting attacks, enable the specific properties you want to bind to, for

// more details see https://go.microsoft.com/fwlink/?LinkId=317598.

[HttpPost]

[ValidateAntiForgeryToken]

public ActionResult Create([Bind(Include = "CID,CName,PName,Number,Payment")] Customer customer)

{

if (ModelState.IsValid)

{

db.Customers.Add(customer);

db.SaveChanges();

return RedirectToAction("Index");

}

return View(customer);

}

// GET: Customers/Edit/5

public ActionResult Edit(string id)

{

if (id == null)

{

return new HttpStatusCodeResult(HttpStatusCode.BadRequest);

}

Customer customer = db.Customers.Find(id);

if (customer == null)

{

return HttpNotFound();

}

return View(customer);

}

// POST: Customers/Edit/5

// To protect from overposting attacks, enable the specific properties you want to bind to, for

// more details see https://go.microsoft.com/fwlink/?LinkId=317598.

[HttpPost]

[ValidateAntiForgeryToken]

public ActionResult Edit([Bind(Include = "CID,CName,PName,Number,Payment")] Customer customer)

{

if (ModelState.IsValid)

{

db.Entry(customer).State = EntityState.Modified;

db.SaveChanges();

return RedirectToAction("Index");

}

return View(customer);

}

// GET: Customers/Delete/5

public ActionResult Delete(string id)

{

if (id == null)

{

return new HttpStatusCodeResult(HttpStatusCode.BadRequest);

}

Customer customer = db.Customers.Find(id);

if (customer == null)

{

return HttpNotFound();

}

return View(customer);

}

// POST: Customers/Delete/5

[HttpPost, ActionName("Delete")]

[ValidateAntiForgeryToken]

public ActionResult DeleteConfirmed(string id)

{

Customer customer = db.Customers.Find(id);

db.Customers.Remove(customer);

db.SaveChanges();

return RedirectToAction("Index");

}

protected override void Dispose(bool disposing)

{

if (disposing)

{

db.Dispose();

}

base.Dispose(disposing);

}

}

}

Product\_InfoController.cs:

using System;

using System.Collections.Generic;

using System.Data;

using System.Data.Entity;

using System.Linq;

using System.Net;

using System.Web;

using System.Web.Mvc;

using Website.Models;

namespace Website.Controllers

{

public class Product\_InfoController : Controller

{

private Conn db = new Conn();

// GET: Product\_Info

public ActionResult Index()

{

return View(db.Product\_Info.ToList());

}

// GET: Product\_Info/Details/5

public ActionResult Details(string id)

{

if (id == null)

{

return new HttpStatusCodeResult(HttpStatusCode.BadRequest);

}

Product\_Info product\_Info = db.Product\_Info.Find(id);

if (product\_Info == null)

{

return HttpNotFound();

}

return View(product\_Info);

}

// GET: Product\_Info/Create

public ActionResult Create()

{

return View();

}

// POST: Product\_Info/Create

// To protect from overposting attacks, enable the specific properties you want to bind to, for

// more details see https://go.microsoft.com/fwlink/?LinkId=317598.

[HttpPost]

[ValidateAntiForgeryToken]

public ActionResult Create([Bind(Include = "PCode,PName,PPrice,CompName,POrigin")] Product\_Info product\_Info)

{

if (ModelState.IsValid)

{

db.Product\_Info.Add(product\_Info);

db.SaveChanges();

return RedirectToAction("Index");

}

return View(product\_Info);

}

// GET: Product\_Info/Edit/5

public ActionResult Edit(string id)

{

if (id == null)

{

return new HttpStatusCodeResult(HttpStatusCode.BadRequest);

}

Product\_Info product\_Info = db.Product\_Info.Find(id);

if (product\_Info == null)

{

return HttpNotFound();

}

return View(product\_Info);

}

// POST: Product\_Info/Edit/5

// To protect from overposting attacks, enable the specific properties you want to bind to, for

// more details see https://go.microsoft.com/fwlink/?LinkId=317598.

[HttpPost]

[ValidateAntiForgeryToken]

public ActionResult Edit([Bind(Include = "PCode,PName,PPrice,CompName,POrigin")] Product\_Info product\_Info)

{

if (ModelState.IsValid)

{

db.Entry(product\_Info).State = EntityState.Modified;

db.SaveChanges();

return RedirectToAction("Index");

}

return View(product\_Info);

}

// GET: Product\_Info/Delete/5

public ActionResult Delete(string id)

{

if (id == null)

{

return new HttpStatusCodeResult(HttpStatusCode.BadRequest);

}

Product\_Info product\_Info = db.Product\_Info.Find(id);

if (product\_Info == null)

{

return HttpNotFound();

}

return View(product\_Info);

}

// POST: Product\_Info/Delete/5

[HttpPost, ActionName("Delete")]

[ValidateAntiForgeryToken]

public ActionResult DeleteConfirmed(string id)

{

Product\_Info product\_Info = db.Product\_Info.Find(id);

db.Product\_Info.Remove(product\_Info);

db.SaveChanges();

return RedirectToAction("Index");

}

protected override void Dispose(bool disposing)

{

if (disposing)

{

db.Dispose();

}

base.Dispose(disposing);

}

}

}

HomeController.cs:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.Mvc;

namespace Website.Controllers

{

public class HomeController : Controller

{

public ActionResult Index()

{

return View();

}

public ActionResult About()

{

ViewBag.Message = "Your application description page.";

return View();

}

public ActionResult Contact()

{

ViewBag.Message = "Your contact page.";

return View();

}

public ActionResult Order()

{

ViewBag.Message = "Your contact page.";

return View();

}

}

}

Model: ProductDB

## 7.3. Testing

<https://youtu.be/fBAYfDXwBqE>