CSV MANAGER

Project Documentation

Project Title: CSV Manager

Date: 14th December, 2018

Project Justification: This project is basically managing the .CSV Database file of Gardo Limited.

Project Characteristics and Requirements:

1. An easy to use Graphical User interface.
2. All features should be calculated when the option is selected, not at initial load. This will allow the data sets to be updated and the calculations rerun without reopening the application. GL's technical team would like the UI to continue being responsive during these calculations.
3. Have the flexibility to point the application to a certain folder on a PC to find the .csv files.
4. List all stores
5. Allow the Finances team to find the following data:
    a. The total sales in all the stores
    b. The total sales for a single store
    c. The total sales in a week for all stores
    d. The total sales in a week for a single store
    e. The total monthly sales for all stores
    f. The total monthly sales for a single store
    g. Given tax rates for taxes such a 2% sales tax, 17% VAT and 1% NHIL, for each of the above, you need to display the Sales totals BEFORE and AFTER TAX, as well as THE VARIOUS TAX AMOUNTS.
6. A Graphical User Interface using WinForms or WPF

Project Management Deliverables: Project plan, Work breakdown structure etc.,

Product-related deliverables: Research report, design documentation, software code etc.,

Project success Criteria: The goal is to develop an efficient and a convenient software application for Gardo limited to help promote the companies manage organization.

I. Statement of work

The CSV Management System should manage the companies CSV database and any other CSV file from a directory on the computer. The managing can be categorized by displaying the data of a store sales, location, and cost within a week or year. The sales should also be categorising by before and after tax charges are applied.

II. Software Design Document

The CSV management system is being developed for the Gardo Limited. The important modules that are going to implement in the proposed system.

- Display the location of the stores
- Display the data for a single stores cost
- Calculate the cost sum before and after tax.
- Log in, depends upon user name and password.

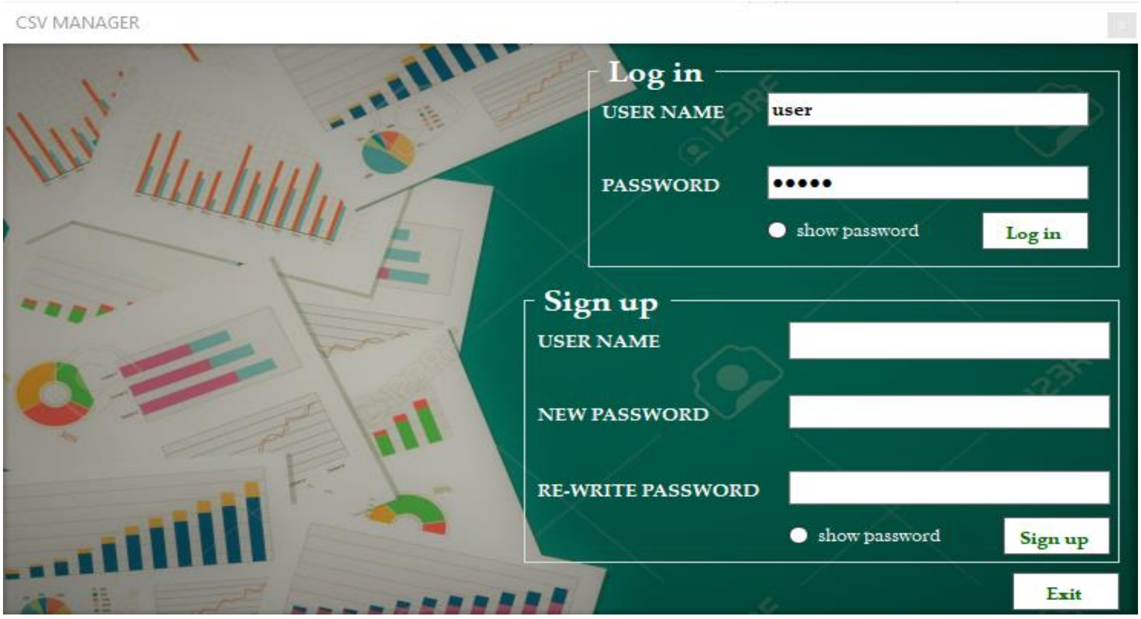
I. Constraints

I to follow the rules specified in the team contract. I am proceeding to develop an appropriate system; however, the design might slightly change to meet the client requirements.

II. System Architecture: The Three Tier Architecture:

To develop software for Library management system the architecture that we are implementing is Three Tier Architecture. The User Interface Layer, Application Logic Layer and Data Storage Layer together called as Three Tier Architecture.

- User Interface Layer: The User Interface Layer will consist of everything shown directly to the client. This Layer will communicate only with the Application Logic Layer. This Layer will also be responsible for initial validation of any user inputted data. Any validation that requires communication to the Data Storage Layer will however be completed at the Logic Layer.



CSV MANAGER

Log in

USER NAME: user

PASSWORD: •••••

○ show password

Sign up

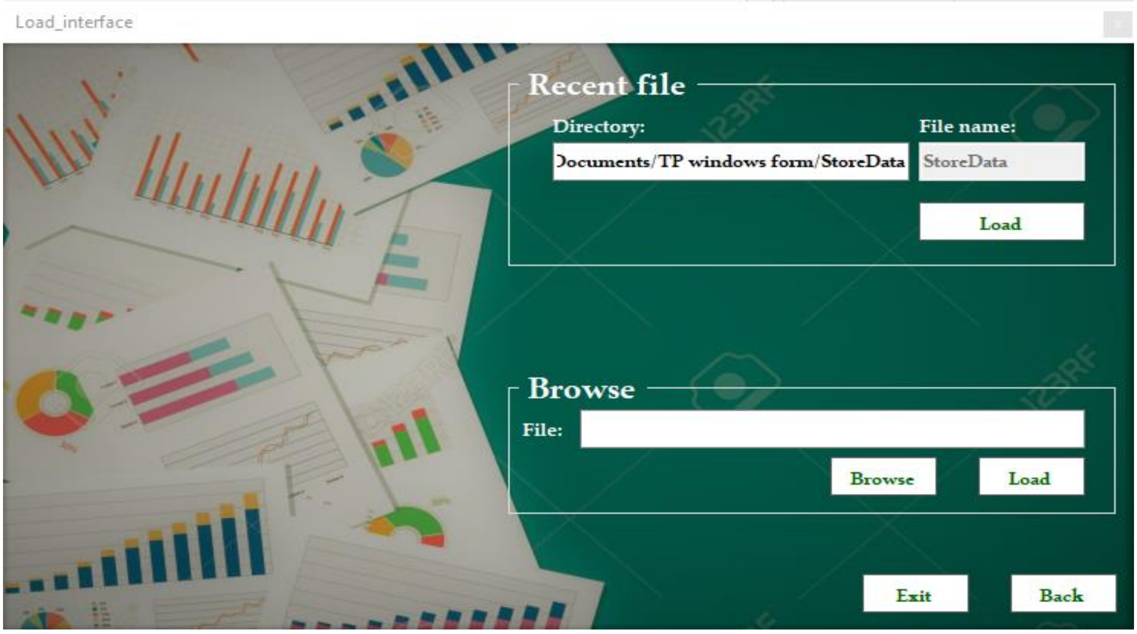
USER NAME

NEW PASSWORD

RE-WRITE PASSWORD

○ show password

The screenshot above represents the Log in interface which serves as a security barrier for users accessing the application. The Log in button is having a default “User name” and “Password”. Users can also Sign up using personal information like “User name” and “Password”.



Load_interface

Recent file

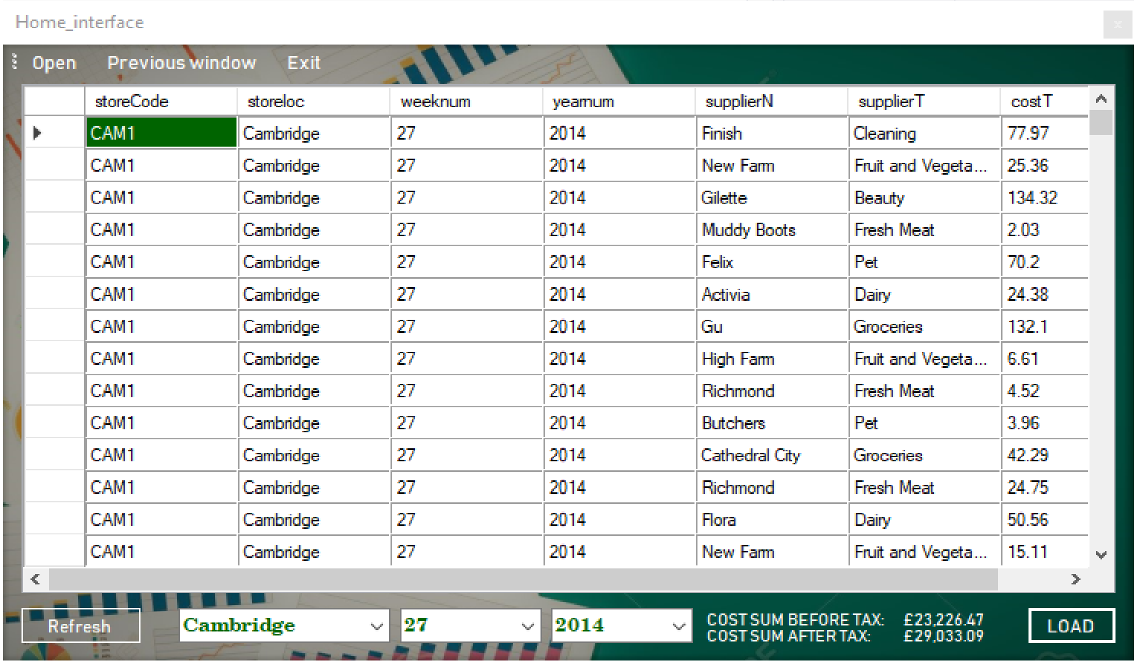
Directory: Documents/TP windows form/StoreData

File name: StoreData

Browse

File:

The window above named “Load_interface” allows users to have access to Gardo Limited Database files and also open .csv file directory on the computer of their choice.



	storeCode	storeloc	weeknum	yearnum	supplierN	supplierT	costT
▶	CAM1	Cambridge	27	2014	Finish	Cleaning	77.97
	CAM1	Cambridge	27	2014	New Farm	Fruit and Vegeta...	25.36
	CAM1	Cambridge	27	2014	Gilette	Beauty	134.32
	CAM1	Cambridge	27	2014	Muddy Boots	Fresh Meat	2.03
	CAM1	Cambridge	27	2014	Felix	Pet	70.2
	CAM1	Cambridge	27	2014	Activia	Dairy	24.38
	CAM1	Cambridge	27	2014	Gu	Groceries	132.1
	CAM1	Cambridge	27	2014	High Farm	Fruit and Vegeta...	6.61
	CAM1	Cambridge	27	2014	Richmond	Fresh Meat	4.52
	CAM1	Cambridge	27	2014	Butchers	Pet	3.96
	CAM1	Cambridge	27	2014	Cathedral City	Groceries	42.29
	CAM1	Cambridge	27	2014	Richmond	Fresh Meat	24.75
	CAM1	Cambridge	27	2014	Flora	Dairy	50.56
	CAM1	Cambridge	27	2014	New Farm	Fruit and Vegeta...	15.11

Cambridge 27 2014

COST SUM BEFORE TAX: £23,226.47
COST SUM AFTER TAX: £29,033.09

The window above named “Home interface” is the home interface of CSV manager which allows user to display, calculate and Load the database of Gardo Limited.

- Application Logic Layer: The Business logic for our project is Visual Basic. It will be responsible for providing a secure connection to the Data Storage Layer and formatting information received from the Database directory for presentation to the User Interface Layer. Scripts that return user specific data, such as advisory stream selection will be run on the Data Storage Layer.

i. **Step 1:** First setup the login code to give user access to the “Load interface”. The following code is typed into the “CSV Manager” log in button. The code set a condition for the login requirement to access Load interface form.

Code

```
string admin, password;
            admin = txtusername.ToString();
            password = txtpassword.UseSystemPasswordChar.ToString();
            if (txtusername.Text == "user" && txtpassword.Text == "11111")
            {
                this.Hide();
                Load_interface form2 = new Load_interface();
                form2.ShowDialog();
            }
            else
            {
                MessageBox.Show("Incorrect User name / Password");

            }
```

ii. **Step 2:** Then double clicked on the “Sign up” button in the “CSV Manager” window above to type a condition for the sign up requirements to access Load interface form.

Code

```
string name;
            string password_1, password_2;

            name = txt_new.ToString();
            password_1 = txt_np.ToString();
            password_2 = txt_rnp.ToString();
            if (password_1 == password_2)
            {
                this.Hide();
                Load_interface form2 = new Load_interface();
                form2.ShowDialog();
            }
            else
            {
                MessageBox.Show("Password don't match");
            }
```

iii. **Step 3:** Following code was typed in `radioButton1_CheckedChanged to hide and show user password input.`

Code

```
private void radioButton1_CheckedChanged(object sender, EventArgs e)
        {
            rbnhide_1.Visible = true;
            txtpassword.UseSystemPasswordChar = false;
            rbnshow1.Visible = false;
        }

private void rbnshow2_CheckedChanged(object sender, EventArgs e)
        {
            txt_np.UseSystemPasswordChar = false;
            txt_rnp.UseSystemPasswordChar = false;
            rbnhide_2.Visible = true;
            rbnshow2.Visible = false;
        }

private void rbnhide_1_CheckedChanged(object sender, EventArgs e)
        {
            txtpassword.UseSystemPasswordChar = true;
            rbnshow1.Visible = true;
            rbnhide_1.Visible = false;
        }

        private void rbnhide_2_CheckedChanged(object sender, EventArgs e)
        {
            txt_np.UseSystemPasswordChar = true;
            txt_rnp.UseSystemPasswordChar = true;
            rbnshow2.Visible = true;
            rbnhide_2.Visible = false;
        }
```

iv. **Step 4:** The following code goes to the “Load button” for Load_interface.cs to access the “Home_interface”.

Code

```
private void btnl1_Click_1(object sender, EventArgs e)
        {
            this.Hide();
            Home_interface form3 = new Home_interface();
            form3.ShowDialog();
        }
```

v. **Step 5:** The following code goes to the “Browser” for Load_interface.cs to browse a directory on your computer for a .csv file.

Code

```
private void button2_Click(object sender, EventArgs e)
        {
            OpenFileDialog openFileDialog = new OpenFileDialog();
            // To list only csv files, we need to add this filter
            openFileDialog.Filter = "|*.csv"; //Only show .csv file in the
browser directory
            DialogResult result = openFileDialog.ShowDialog();

            if (result == DialogResult.OK)
            {
                tbCSVPath.Text = openFileDialog.FileName;
            }
        }
```

vi. **Step 6:** The following code goes to the “Load” in Browse Group Box for Load_interface.cs to Load .csv file in Datagridview. A bug was identified and fixed using the System.NullReferenceException.

Code

```
private void button1_Click(object sender, EventArgs e)
        {
            btnrefresh.Visible = true;
            dgCSVData.Visible = true;
            try
            {
                dgCSVData.DataSource =
GetDataTableFromCSVFile(tbCSVPath.Text);
            }
            catch (Exception ex)
            {
                MessageBox.Show(ex.Message, "Import CSV File",
MessageBoxButtons.OK,
  MessageBoxIcon.Error);
            }
        }
```

vii. **Step 7:** The following code goes to the “Home_interface” Open/Store Code. The data is rerieved and displayed on the list box using modulo.

Code

```
private void storeToolStripMenuItem_Click(object sender, EventArgs e)
        {
            string filePath1 =
Environment.GetFolderPath(Environment.SpecialFolder.MyDocuments);
            string theFile = filePath1 + @"\StoreData\StoreCodes.csv";
            Array myArray = File.ReadAllLines(theFile);
            lb.Visible = true;
            dgvData.Visible = false;
            foreach (String v in myArray)
            {
                int x = 0;
                foreach (String vm in v.Split(',').AsParallel())
                {

                    if (x % 2 == 1)
                    {
                        lb.Items.Add(vm);
                    }
                    x++;
                }
            }
        }
```

viii. **Step 8:** The following code goes to the “Home_interface” Open/Total sales/Single sales. A Performance Profiling and Evaluation using Parallel.Foreach to access Data.

Code

```
private void storeToolStripMenuItem_Click(object sender, EventArgs e)
        {
            string filePath1 =
Environment.GetFolderPath(Environment.SpecialFolder.MyDocuments);
            string theFile = filePath1 + @"\StoreData\StoreCodes.csv";
            Array myArray = File.ReadAllLines(theFile);
            lb.Visible = true;
            dgvData.Visible = false;
            foreach (String v in myArray)
            {
                int x = 0;
                foreach (String vm in v.Split(',').AsParallel())
                {

                    if (x % 2 == 1)
                    {
                        lb.Items.Add(vm);
                    }
                    x++;
                }
            }
        }
```

- Data Storage Layer: The Data Storage Layer will record all information required by the Logic and User Interface Layer. This data will include Store details for the Cost, and general user and admin user information.