LONDON METROPOLITAN UNIVERSITY

Module Code & Module Title

CS6004NT Application Development

Assessment Weightage & Type

30% Individual

Year and Semester 2020-21

Autumn, Year Long

Student Name: Girija Tamang

London Met ID:18030995

College ID: np05cp4s190007

Assignment Due Date: 1-22-2021

Assignment Submission Date: 1-22-2021

I confirm that I understand my coursework needs to be submitted online via Google Classroom under the relevant module page before the deadline in order for my assignment to be accepted and marked. I am fully aware that late submissions will be treated as non-submission and a marks of zero will be awarded.

Table of Contents

Table of Figures

1. Introduction

The project scenario is to use the .NET platform to create a Feedback Desktop application in C#. This program was created for a Vintage Hotel to collect customer reviews. The software is designed as a simple and user-friendly Interface, and each user who interacts with the application will have a simple and comfortable experience.

This application allows the admin user to enter various criteria to get reviews and to rate the services. It also enables customers to provide their valuable feedback from where an overall service feedback and rating report is produced. The key goals of this application are to preserve and analyze customer reviews. When inserting data into the Interface, the software handles errors encountered and displays different dialogue boxes that direct you to fill in the correct details. This application will help to maintain records of user feedback and produce relevant information.

2. User Manual

The following is the detailed details for running the program along with the proper screenshot:

To run Program

1. Open the Microsoft Visual Studio.

2. Go to File and choose the Project/Solution option or press CTRL+SHIFT+O to open a dialog box and the project named as Admin Login.sln must be opened.

2.1 Welcome Page.

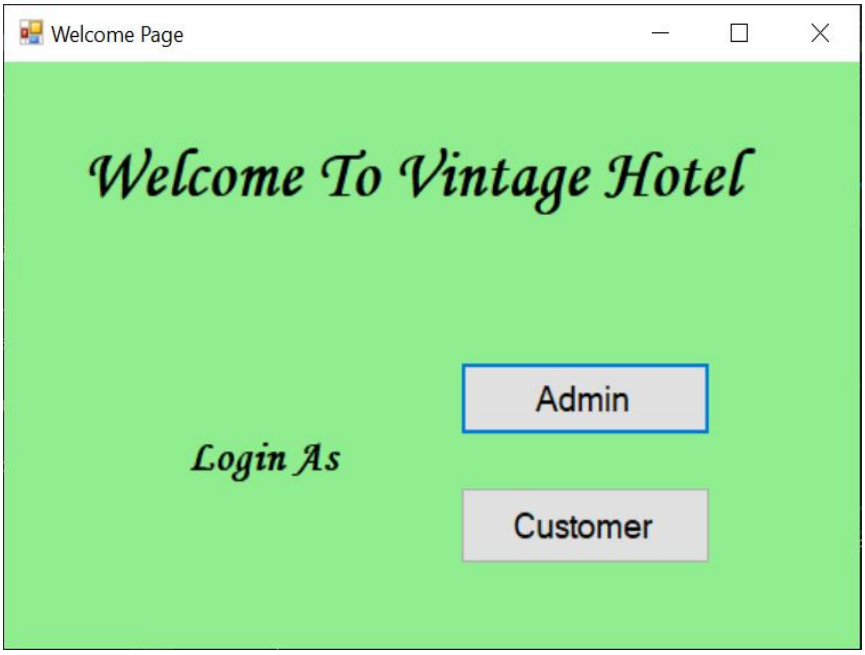


Figure 1: Welcome page.

After the successful run of the application, welcome page open where we can login as Admin and Customer. If we choose Admin, it opens the admin login form and if we choose customer it opens a feedback form.

2.2 Admin Login Screen

After choosing Admin, Admin Login form opens.

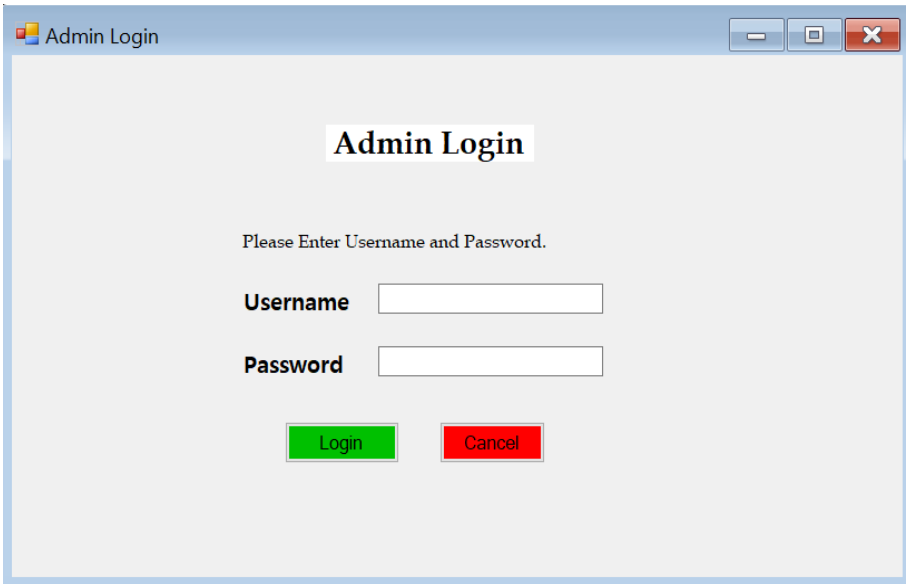


Figure 2: Admin login

The username and password for admin login is admin and admin respectively. If the wrong username and password is entered it will display an error message as shown below.

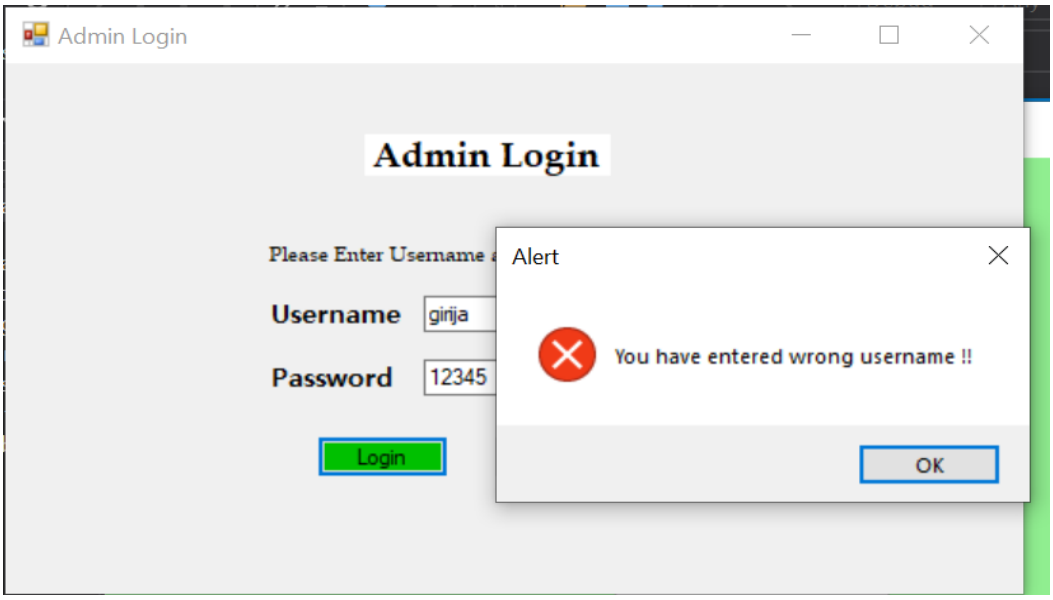


Figure 3: Error message displayed if username is incorrect.

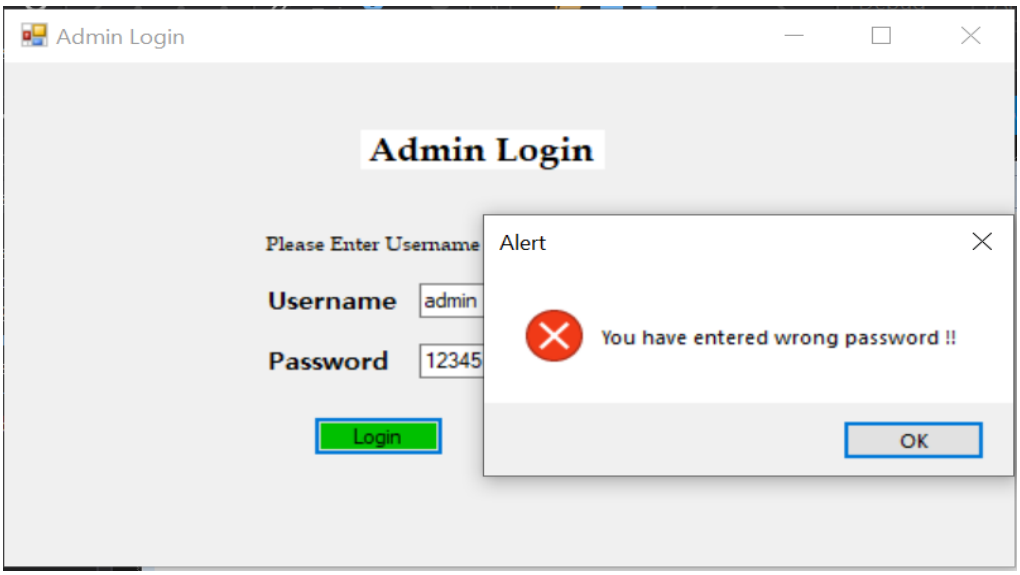


Figure 4: Error message displayed if username is incorrect.

When the correct username and password is entered, a successful message is shown as below.

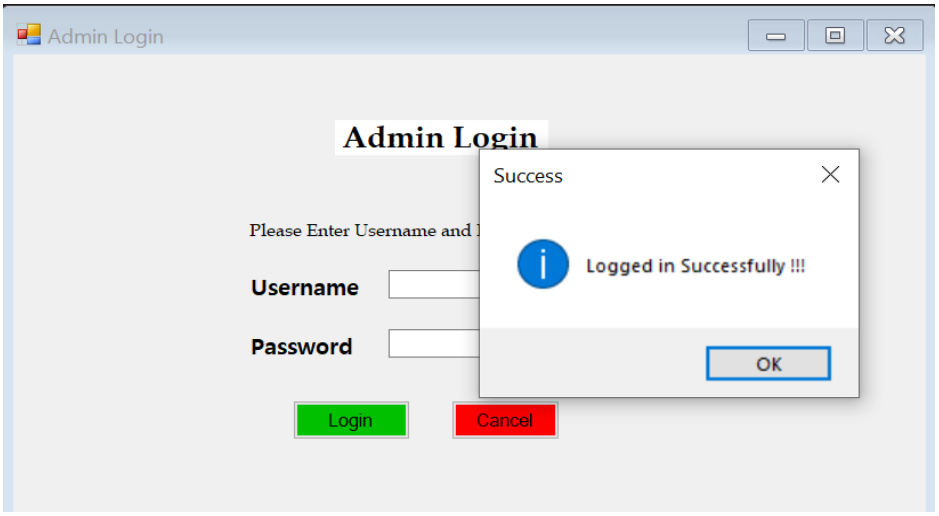


Figure 5: Admin Login successful message

2.3 Admin Dashboard

After logging into the system with correct credentials, Admin Dashboard form opens.

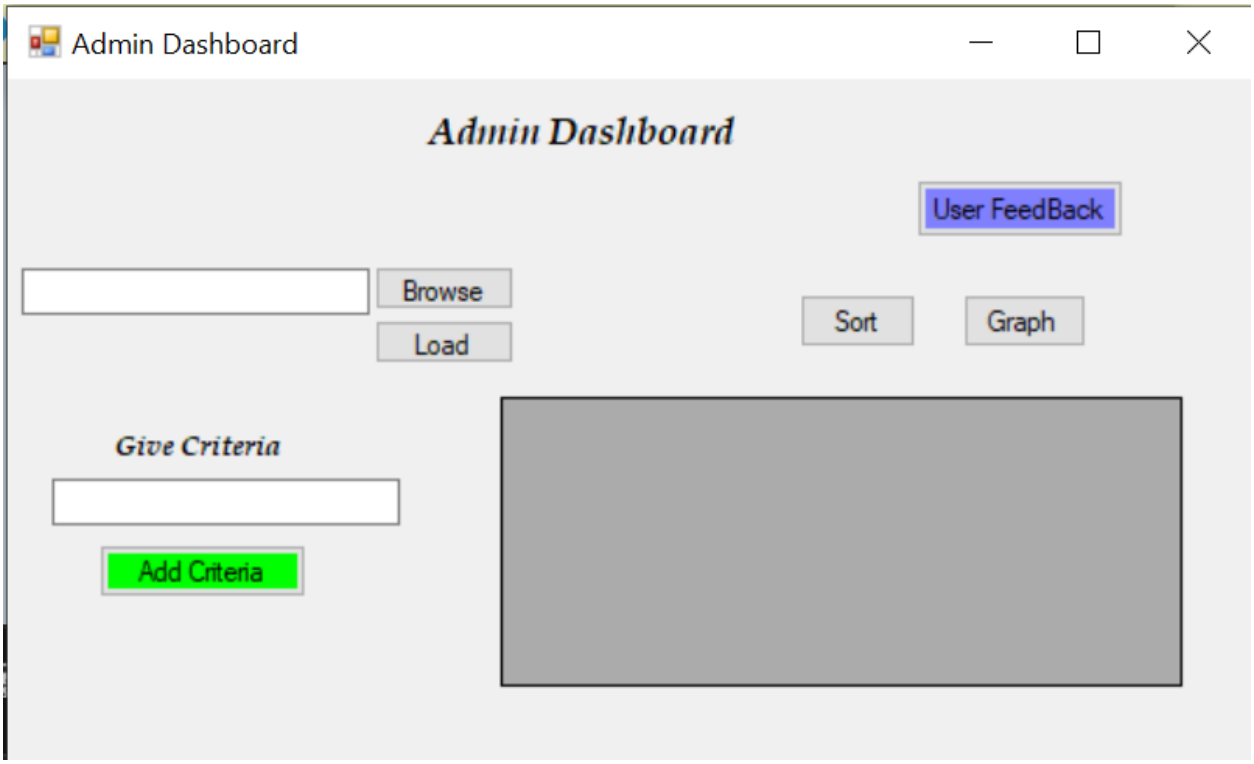


Figure 6: Admin dashboard.

The user will have many button options in form to click for. These buttons perform different tasks which are described as follows:

Add Criteria: This button allows admin to give criteria for giving feedback and to rate the services.

User Feedback: This button allows the admin to visit the user feedback page.

Browse: This button allows admin to import csv file location from computer.

Load: This Button load all the information of the csv file in the data grid table.

Sort: This button is used for sorting the data in ascending order based on rating of criteria.

Graph: This button is used to view the chart of the total data of criteria.

2.4. User Feedback

After successfully adding criteria by admin in data grid of the feedback form. Users can give their feedback.

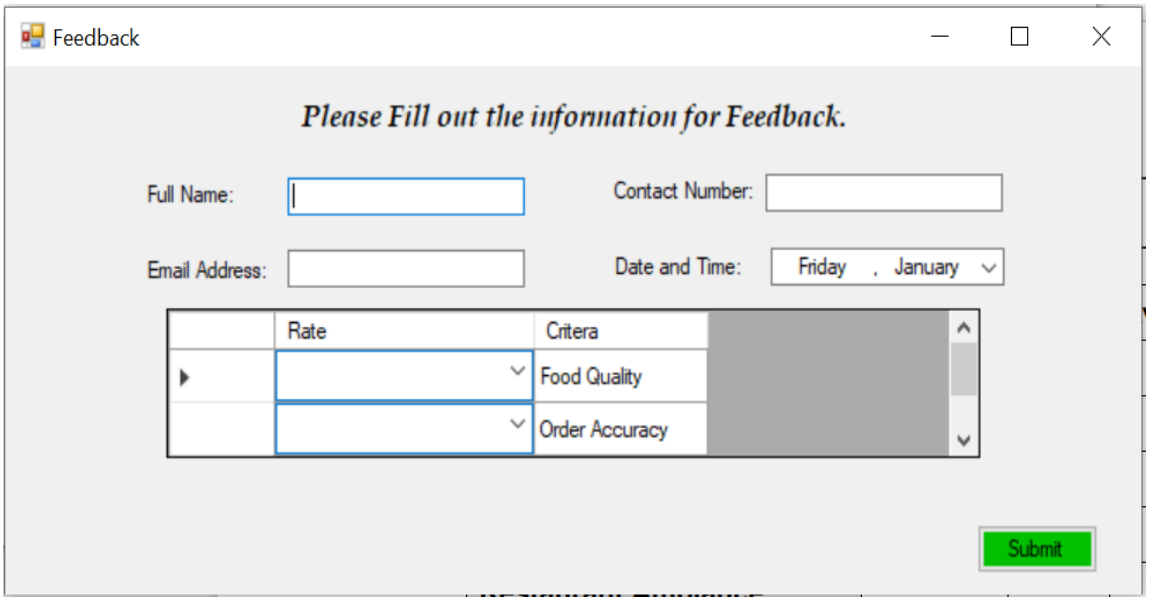


Figure 7: User feedback form.

This form opens where user's login in the system as customers. Here is only one submit button which is used for recording the feedback of the customer. It records the feedback of customer in a csv file.

3. Architecture of Software

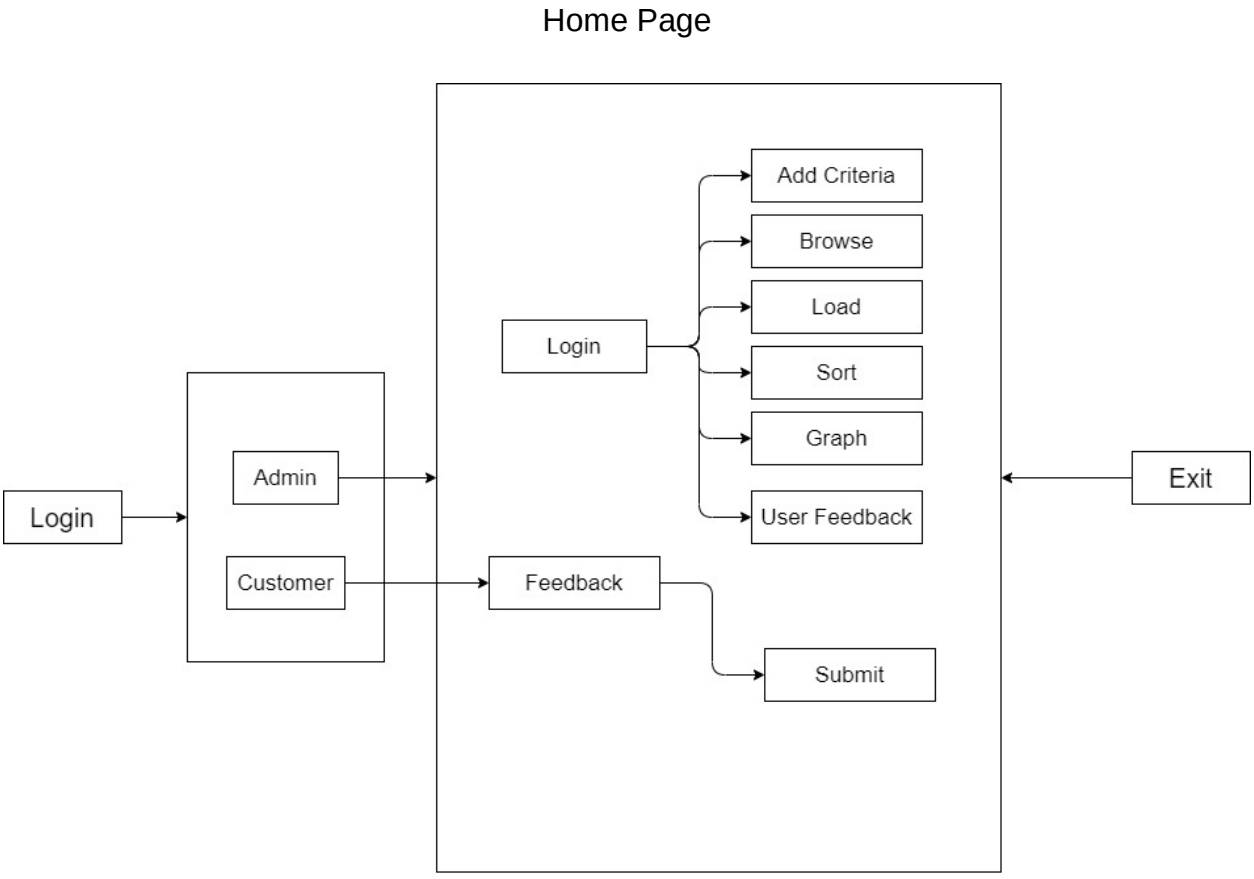


Figure 8: System Architecture.

The above figure represents the architecture of the developed system. At first, the user needs to choose login as Admin or Customer. If the user chose admin, the user needs to provide correct credentials for login in the system. After logging into the system with correct credentials, the system will display the admin dashboard form. From this form users can give criteria, browse csv file location, and import data in a data grid table, sorts the data, generate graphs and open user feedback form. If a user login as a customer, the user will direct to a feedback form from where the user can provide their valuable feedback.

3.1 Class Description

Welcome.cs This is the main page of the desktop application. After running the application, Welcome.cs class is called, and the welcome form is displayed. The form contains two buttons one as Admin and another as Customer. If the user selects a customer, it opens a feedback form and if the user selects the admin button, it opens the admin login page.

Class	**Description**
Form1.cs (Admin Login)	In the respective text boxes, the user must enter the username and the password. If the text boxes for entering the username and password contain wrong information, an error message box will be displayed.
Form2.cs (Admin Dashboard)	This class is called when admin login in the system successfully. After providing criteria in the related text field and clicking the add criteria button, criteria are added to DataGrid view of user form. When the browse button is clicked admin can import feedback data from the computer, after that load button is clicked, feedback data will open in DataGrid of the admin dashboard. When the user feedback button is clicked, it opens a feedback form.
Form 4.cs (User Feedback)	This class is called when a customer login into the system. The users fill the data and click on the submit button all the information is saved in a csv file.

3.2 Method Description

Methods	Description
Welcome.cs	
private void admin_Click(object sender, EventArgs e)	When the admin button is triggered, it opens admin login from.
private void customer_Click(object sender, EventArgs e)	When the customer button is triggered, it opens the customer feedback form.
Form1.cs (Admin login)	
private void login_Click(object sender, EventArgs e)	When the login button is triggered, it checks the username and password given by admin. if wrong information is given, it shows an error message otherwise it opens admin form.
private void cancel_Click(object sender, EventArgs e)	When the cancel button is triggered, the admin login app window is closed.
Form2.cs (Admin Dashboard)	
private void Criteria_Click(object sender, EventArgs e)	When the Criteria button is triggered, it checks the empty data and if there is criteria data given it provides data to user feedback DataGrid.
private void Feedback_Click(object sender, EventArgs e)	Opens customer feedback form, when the user feedback button is clicked.
private void browse_Click(object sender, EventArgs e)	When the browse button is clicked, it helps to choose a csv file from the computer to import feedback data.
Form4.cs (Feedback Form)	
private void submit_Click(object sender, EventArgs e)	When this button is triggered, it saves the feedback information given by the customer in csv file format.

3.3 Functionality

The developed system can be used to maintain and track customer feedback. The system will be provided with the prescribed criteria by the admin to provide feedback and rate the services. The user can log in to the system to provide feedback. Developed system is a desktop-based application that is lite software and can run within a minimum of 2GB of RAM.

4. Bubble Sort Algorithm

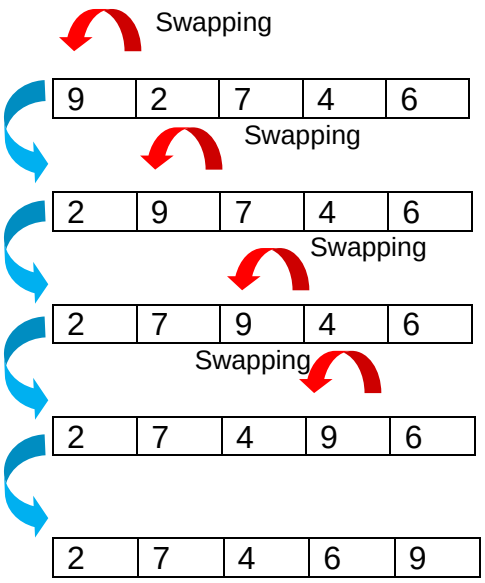
Bubble Sort is the easiest sorting algorithm that is primarily based on the concept of again and again comparing pairs of adjacent elements of a list and then swapping their positions if they exist in the incorrect order. Every pass through the list places the next greatest value at its proper position and "bubbles up" each item to the place where it belongs (Gupta, 2019).

Let the array be [9, 2, 7, 4, 6]. We know that 9 must not be on the left of 2 and so, we swap them and get [2, 9, 7, 4, 6].Next, we see that 9 need to once more no longer be on the left of 7 so we swap 7 and 9 then we get [2, 7, 9, 4, 6].we repeat this for 9 and 4 and as a result for 9 and 6 to get [2 7,4,6,9].

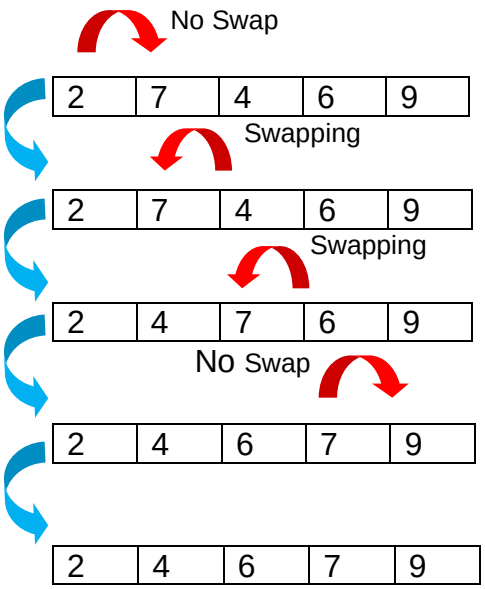
As can be seen – after one “pass” over the array, the greatest factor (9 in this case) has reached its correct position – extreme right. Let us strive to repeat this process. (2,7) is in right order. However (7,4) is a wrong order. Therefore, we swap 7 and 4 to get [2,4,7,6,9]. Now again (7,6) is improper so we do swap again and get [2,4,6,7,9]. As we see the array is sorted. In this way bubble sort exactly works (Sehgal, 2018).

A figure is well worth a thousand words so, renowned this figure for higher understanding of working of bubble sort.

First Pass



Second Pass



The array is sorted.

5. Conclusion and Reflection

5.1 Reflection on experience

This coursework is about the development and implementation of feedback/rating system in the C# desktop application. It was the first time that the C# programming language was used by me to develop a feedback application. In the Application Development Module, I acquired knowledge of the C # programming language which helps me in developing this project. However, the development of feedback systems in Microsoft Visual Studios 2019 is really a difficult task.

During system development I got a chance to learn many new things like serialization, importing and exporting CSV files containing feedback information. As I learned about creating a desktop application using features allowed by the .NET framework in the visual studio, this coursework offered a very fruitful work experience and really helped me in gaining knowledge of file handling..

5.2 Problems Experience

Different issues were encountered before the end of the project. I faced a problem while importing and exporting the csv files. When using the try and catch method for handling the exception, calling variables and making instances, to identify and correct the errors, I face problems. Many syntax errors, run-time errors and logical errors were faced while developing the program. Generating the chart and doing sorting was very difficult for me.

5.3 Solution taken for solving the problem.

With the assistance of numerous websites and video guides, the issues were solved. I have done a lot of research and study for tackling the problems that occurred while developing the project. Many errors were solved after data collection and research from the internet and from the friend and lecturer's useful guidance.

5.4 Conclusion

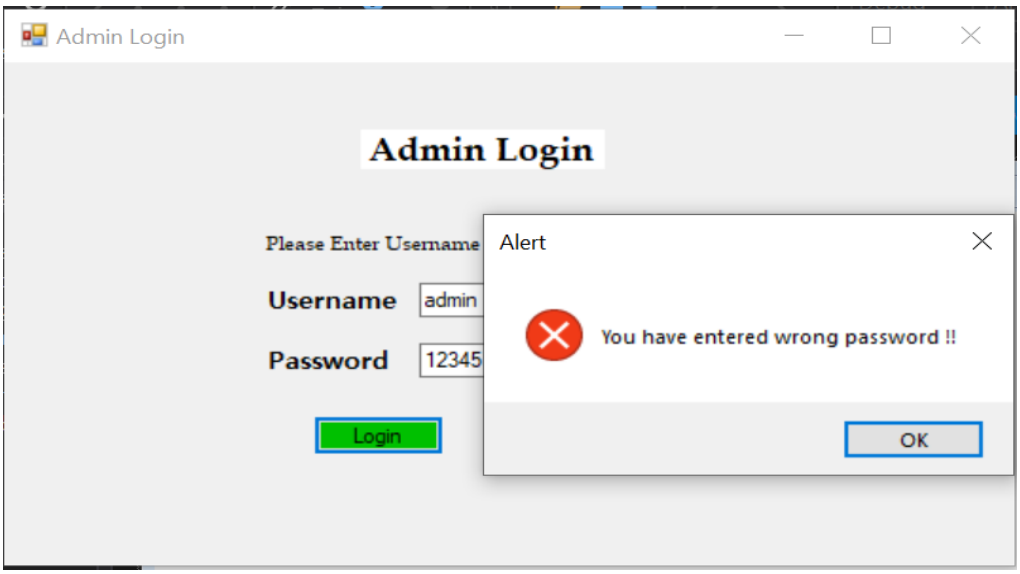
This article deals with the application development work which has been assigned to design and implement the feedback/rating framework in the C# programming language. I learned about using the process of try and catch to manage the exception, naming variables and cases, finding, and correcting the error, checking the program after completion, and systematically writing a report using the correct format. Various problems were solved with doing a lot of research and with the guidance of friends and module leaders. I can now create a desktop application that is easier to deal with and user-friendly on a .NET platform that can handle the exceptions very well.

7.Testing

When the production process is over, the testing phase begins. Testing ensures that there are no errors or bugs that cause the software to run as a bug-free application.

Test case 1: Entering wrong username and password in admin login.

Objective: To determine whether it is possible or not to log in with an incorrect username or password.

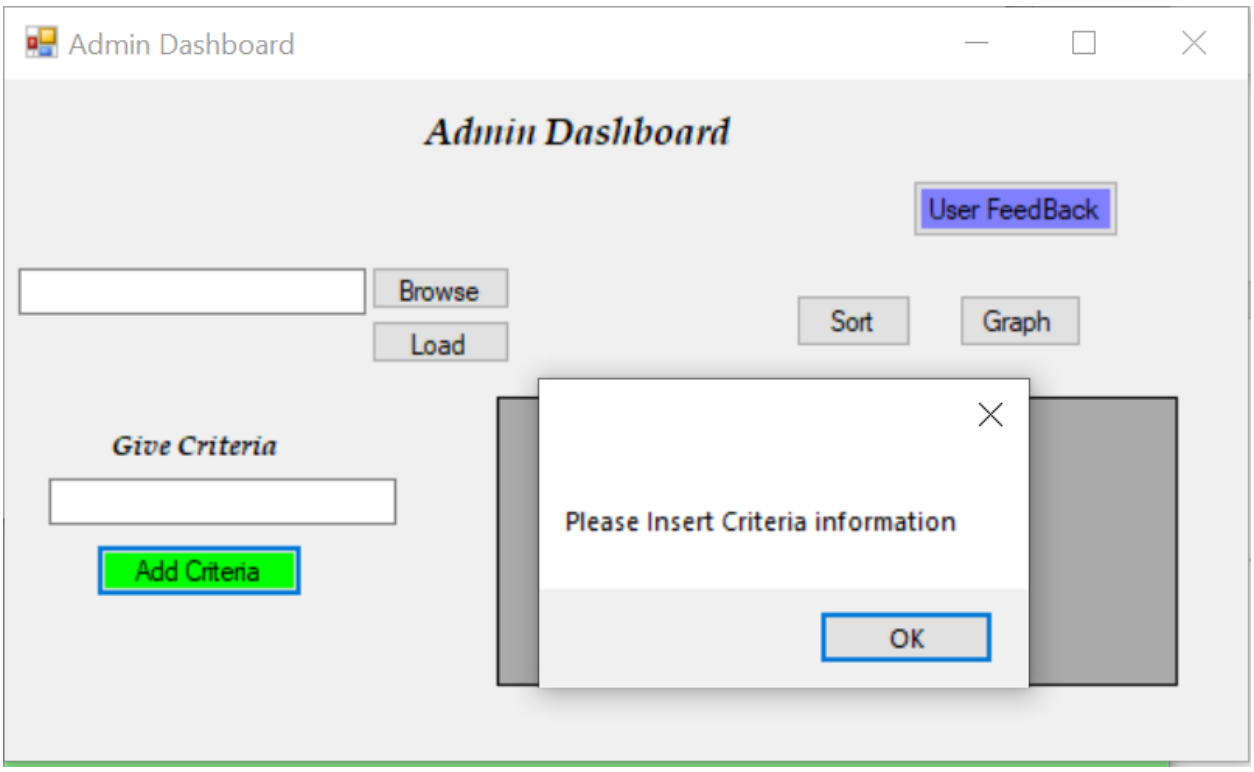


Result: It is not possible to log in with an incorrect username or password.

Conclusion: Test is successful.

Test case 2: Admin entering empty criteria for feedback.

Objective: To determine whether it is possible or not to add empty criteria in user feedback data.

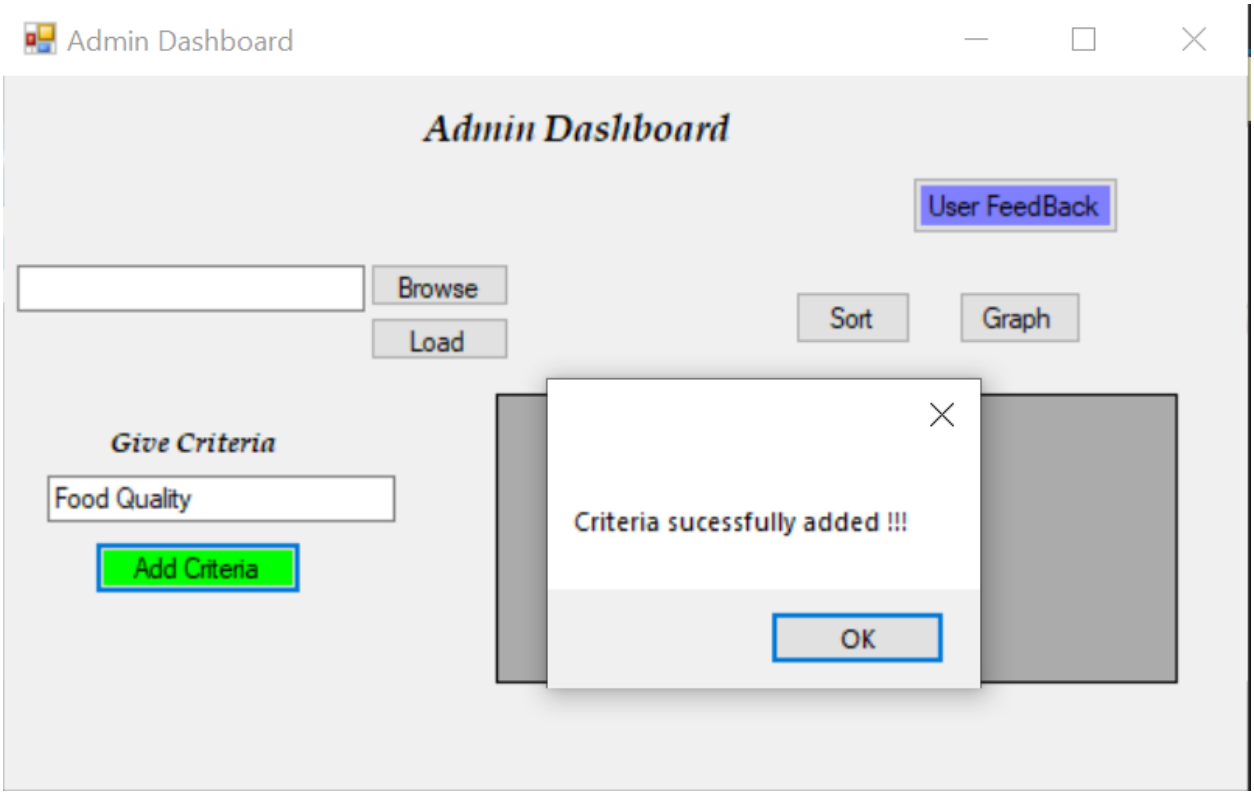


Result: It is not possible enter empty criteria.

Conclusion: Test is successful.

Test case 3: Admin entering criteria for feedback.

Objective: To add criteria in user feedback data.

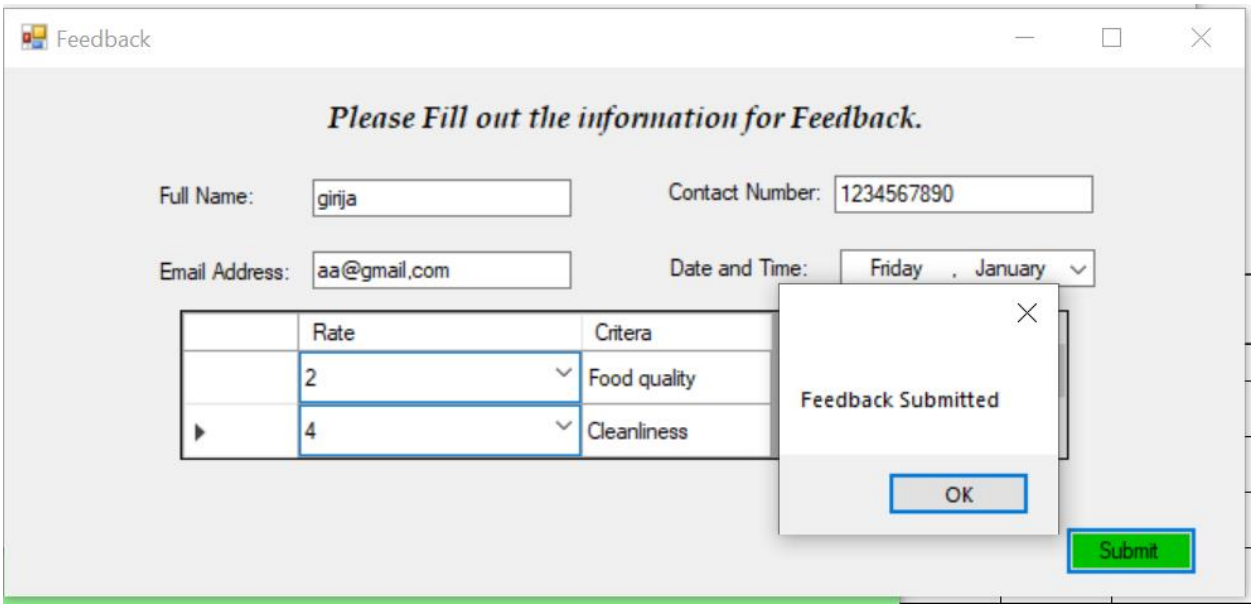


Result: Enter criteria successfully.

Conclusion: Test is successful.

Test case 4: Submitting feedback by user.

Objective: To submit user feedback.

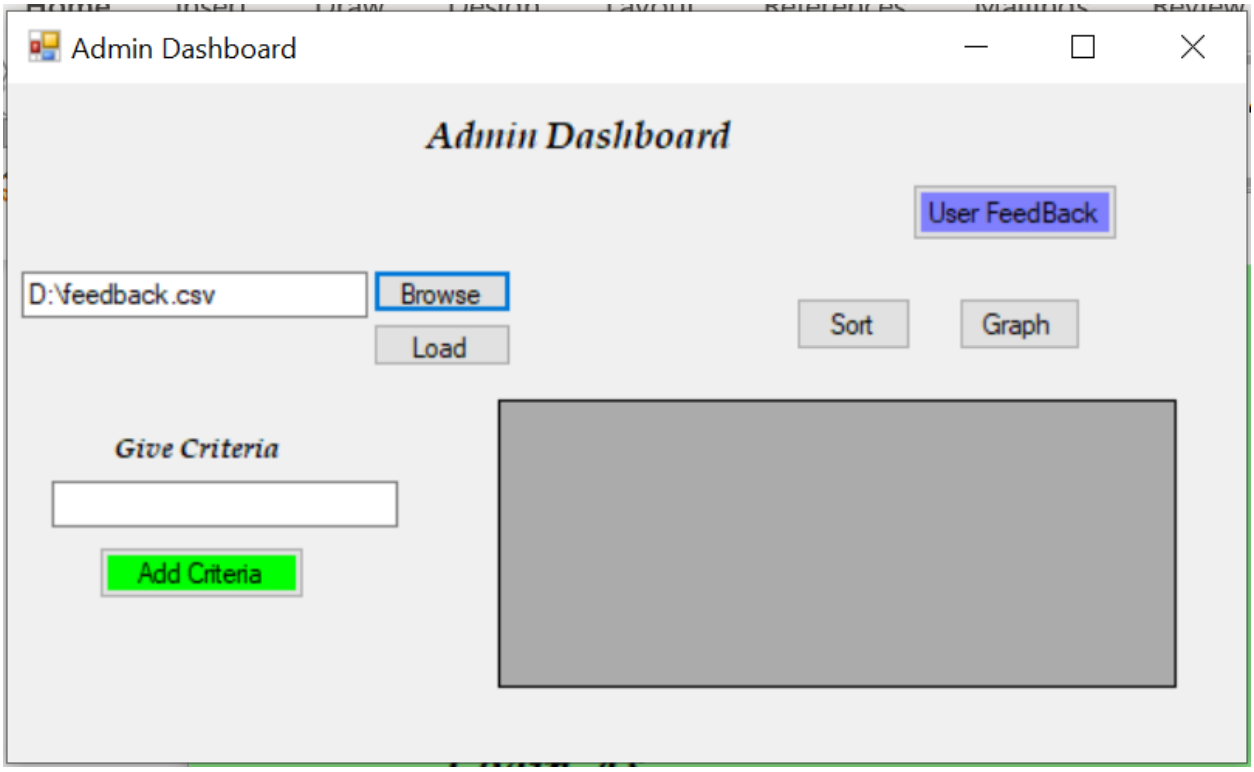


Result: Feedback successfully submitted.

Conclusion: Test is successful.

Test case 5: Importing csv file.

Objective: To import csv file which contain user feedback.

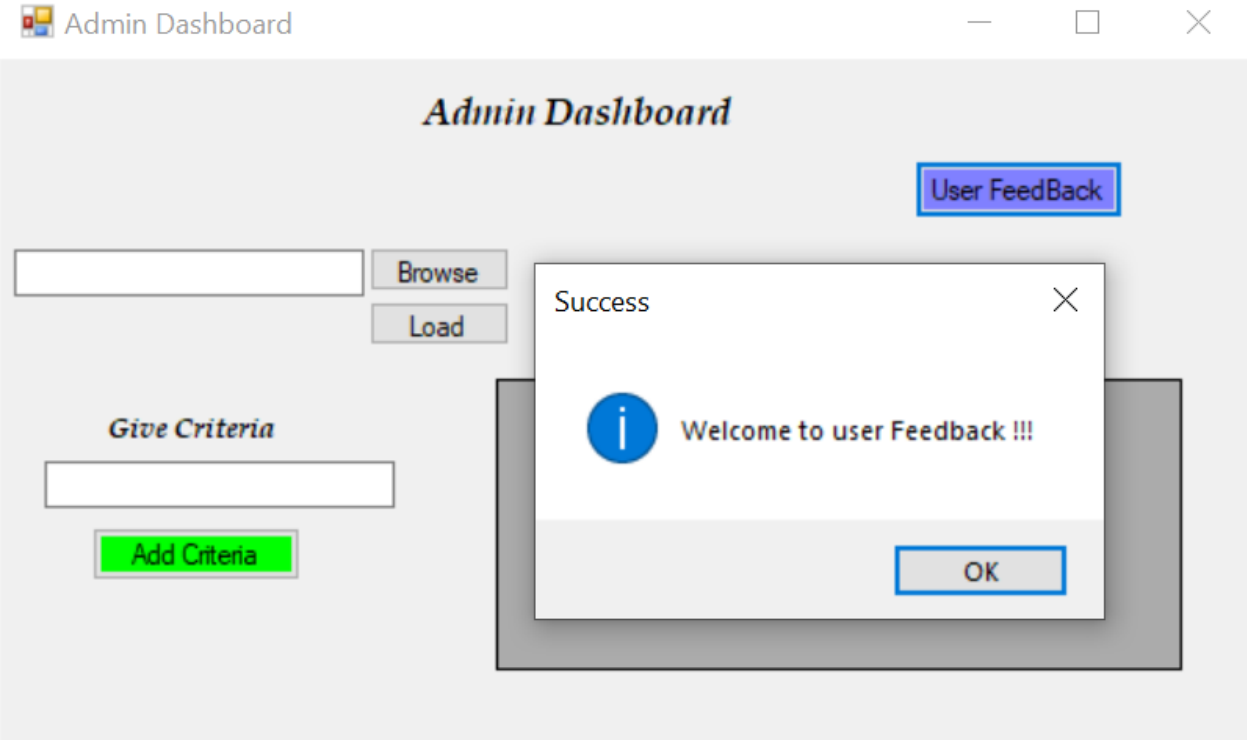


Result: Imported data successfully.

Conclusion: Test is successful.

Test case 6: Open user feedback form.

Objective: To open user feedback form from admin dashboard.



Result: User form successfully opened after clicking ok.

Conclusion: Test is successful.

8.References

Gupta, V., 2019. *Bubble Sort in Python with Example.* [Online] Available at: https://www.stechies.com/bubble-sort-python/ [Accessed 20 January 2021].

Sehgal, K., 2018. *An Introduction to Bubble Sort.* [Online] Available at: https://medium.com/karuna-sehgal/an-introduction-to-bubble-sort-d85273acfcd8 [Accessed 20 January 2021].

9. Appendix

Program.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;
using System.Windows.Forms;
namespace Admin_Login
{
    static class Program
    {
        /// <summary>
        /// The main entry point for the application.
        /// </summary>
        [STAThread]
        static void Main()
        {
            Application.EnableVisualStyles();
            Application.SetCompatibleTextRenderingDefault(false);
            Application.Run(new Welcome());
        }
```

```
    }
}
```

Welcome .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 Admin_Login
{
    public partial class Welcome : Form
    {
        public Welcome()
        {
            InitializeComponent();
        }
```

```
        private void label1_Click(object sender, EventArgs e)
        {


        }


        private void admin_Click(object sender, EventArgs e)
        {
            Form1 admin = new Form1();
            admin.Show();
        }
        private void customer_Click(object sender, EventArgs e)
        {
            Form4 userFeed = new Form4();
            userFeed.Show();
        }
        private void Welcome_Load(object sender, EventArgs e)
        {
        }
    }
}
```