VAASAN AMMATTIKORKEAKOULU
VASA YRKESHÖGSKOLA
UNIVERSITY OF APPLIED SCIENCES

Afonne Joseph

WEB BASED ERROR REPORTING MANAGEMENT APPLICATION

Technology and Communication
2010

FOREWORD

It is said that fours walls of an educational institution is not only a place to develop academically but also to shape one’s life and future. During my sojourn at the Vaasa University of Applied Science, I had picked lots in terms of academics and also to build a rounded out personality despite the challenges faced.

I would thank my supervisor Dr. Ghodrat Moghadampour for his support in giving his time and advice for this project work. He has been a lot of support to the students who have chosen Software Engineering as their major.

I am most grateful to my wife Sanna Afonne whose love has been a driving force for me during my pursuit of the bachelor degree. Your patience and understanding have paid off.

In Vaasa, November, 2010
Joseph Afonne

VAASAN AMMATTIKORKEAKOULU

UNIVERSITY OF APPLIED SCIENCES

Degree Programme of Information Technology

ABSTRACT

Author	Joseph Afonne
Title	Web Based Error Reporting Management Application
Year	2010
Language	English
Pages	67
Name of Supervisor	Ghodrad Moghadampour

After product deployment, customers start using the product and naturally they might encounter various problems. Recording problems and their solutions can help to solve the same problems in the future. Therefore, helpdesk personnel play a vital role in serving customers efficiently, particularly when they have problems. Such efficiency does not come about by chance, for it is the efficient management of such help process, which leads to efficiency. This is also true with the management of users' error reports.

This project work aimed at developing a web application for managing customers' error reports at UpCode Ltd., which is a software company in Vaasa. When users of a product encounter problems, they normally contact the company for solutions. The company will typically first of all check if the problem had been encountered and solved before. If that is the case, the solution is made available to the error reporting user. Otherwise, the error report is registered and a time frame for developing a solution is defined. The record will be then updated once a solution has been found for the problem.

This application was developed using ASP.NET technology and C# as the programming language for the server-side. All the necessary data was stored in MySQL database and retrieved for display when needed. It also used other technologies such as JavaScript, XML and Cascading Style Sheet (CSS). These technologies work together to deliver dynamic web pages from the server to the client browser. The main functionalities implemented by this application include: login, search, register solution, view solutions list, register error, view/edit error list, change password, view user records, register new customer, view customer list, send email to new application user, register new application users, view/edit application users records among others.

Keywords Product Deployment, Efficient Management, Error Reporting

Contents

1 INTRODUCTION

Web applications are on the increase today, as there is an increasing need to connect and use applications running in remote computers through the web- either internet or intranet. As more and more companies are expanding and opening up new branches the world over, there is need to have central information database for the branches to have access to.

After product deployment, customers start using the product and naturally they might encounter various problems. Recording problems and their solutions can help a great deal solving the same problems in the future. This is especially true when the same problems come from different customers. A great deal of time is saved by not solving the same problems, for which there are solutions developed, over again.

Against this backdrop, this thesis topic was born. This project work, Web Based Error Reporting Management Application, was designed for UpCode Ltd., a software company in Vaasa. When they are contacted by the users who use their products with problems, they want to respond immediately to the problems. The first thing they do is to search the database to see if there has been such a problem in the past and if there is a solution for it. If this is the case, then the user gets the solution, otherwise the problem is registered and time line is defined for the solution. Upon finding a solution to the problem, the user is got back to. By using this application, those in technical support or helpdesk can respond efficiently to the problems of the users.

This project was developed with ASP.NET, which is the most complete platform for web development that has ever been put together. Besides, ASP.NET incorporates the .NET platform and the uses all the full-fledged object-oriented programming features of the platform. The programming language used to program the ASP.NET pages was C# programming language, which is a powerful language used by many developers. The Visual Studio 2005, which was used for this project, offers a good IDE (Interface Development Environment) for developing web pages. In addition to ASP.NET technology, XML technology which was also used for the project is a good means for data exchange in different platforms as it is platform-independent and language-independent. All the data are stored in MySQL database which uses the SQL language to perform all the data manipulation operations. [1]

1.1 UpCode Ltd.

Sture Udd has owned and run an independent Print Company, UPC Print, since 1978. This company has of today been one of the Finland’s leading printing companies. Sture has always been an inventor and in the 90’s, he realized that despite the enormous expansion of the online and the internet, the mobile phone held the key to the future of cross-media and print communication. Over the next few years and with a substantial investment in R&D, by early 2005 the first article was released in the Finnish press about UPC's Mobile Technology. [5]

The UpCode brand name and logo are used globally to maximize further investment on research and development of mobile applications and intelligent solutions. UpCode has engaged representatives worldwide and has individuals and an assortment of people as independent representatives. It has offices around the globe including: Helsinki, London, China, Argentina, Dubai and Rwanda. So, in all UPC Communication Center has employed 100 people in Finland and 50 abroad. [5]

UpCode Ltd. has developed lots of products and services ranging from mobile applications to server solutions. These are developed in-house. UpCode program is available for download using the mobile phone in any country and can be used for example to buy ticket for a theatre. The following are the products and areas they are used:

- UpTrack (Logistics)
- UpControl (Security, reporting)

- UpCheck (Digital signature, authentication)
- UpCard (ID, payments)
- UpCare (Healthcare)
- UpChannel (Marketing)
- UpStore (Customization, training)
- UpWire (Payment)
- UpSkill (Education)
- UpTour (Tourism)
- UpFair (Events)
- UpGame (Games, entertainment)

Services offered by UpCode include:

- Complete solutions
- Service
- Implementation support
- Consulting
- Training
- Analyzing and optimization
- Helpdesk
- Technical Academy
- Business school
- Installation Support
- Technical support
- Business analysis

The following are solutions developed by UpCode:

Complete, including server based content

- Web-based
- Locally based

Complete, including mobile based content

- Content in a single mobile phone
- Content in multiple mobile phones

Supporting software

UpCode reader

- 2D, 1D, OCR, COLOR, PROPERIETY CODE

Device recognition system

UpCode Code Creator

- Web-based
- Computer software
- Branding

UpCode Site Creator

- Mobile Websites

UpCode Administrator

- Tracking, statistics, linking, content

UpCode SMS

UpCode MMS decoding

Company clients [5]

1.2 Overview of technologies used

This project used a number of technologies that work together to achieve the final result. They are the following:

1.2.1 Active Server Pages Dot Net (ASP.NET)

ASP.NET is a powerful server-side technology that is geared toward creating dynamic web pages, web applications and web services. This is one of the technologies built into Microsoft.Net Framework. This is the successor for Microsoft’s Active Server Pages (ASP). Meanwhile, ASP.NET does not do all the work but passes some work to .NET Framework to do some processing. ASP.NET is built on the Common Language Runtime (CLR), which allows programmers to write ASP.NET code using any of the languages supported by .NET. This project uses C#, a powerful and the most popular programming language in the .NET Framework to create dynamic web pages. The API classes bundled in this platform are what perform all the wonders in creating web pages that have dynamic content. ASP.NET was born because of the need to solve problems about ASP with regard to separation of presentation and content. In ASP, there was no such separation. Since it is built on (Common Language Runtime) CLR, it offers an object-oriented programming environment and garbage collection, desirable features. [1]

The building blocks for creating web pages in .NET are called web forms. Their file names end with '.aspx' extension. These files contain (X)HTML markup, server-side web controls and user controls that developers place on pages to produce static and dynamic web pages. .NET Framework encourages separation of the code-behind logic on a different file from the presentation. The code-behind files are named with 'aspx.cs' and differentiated from pages files 'aspx'. This separation of the logic from the presentation has the advantage of making debugging easy. Also, it makes it easy to write codes that respond to events, like the page load. [3]

1.2.2 JavaScript

This is a client side scripting language that affects only the browser- the browser and content. Almost all browsers support JavaScript. When the server sends message (usually in the form of HTML) to the browser, JavaScript module is sent too. Then the browser will execute the module. Each browser has different ways of handling JavaScript and at the end achieves the same result. This project work uses JavaScript technology to produce user-browser interaction. This brings life to HTML and makes web pages dynamic. Instead of static HTML documents, JavaScript can turn them into dynamic applications. With JavaScript, web pages can be manipulated once they have been rendered by the browser and made to interact with the user through forms and controls, and create visual effects. [4]

There are two ways to embed JavaScript code in a web page. They are:

- Embed the code directly in an event attribute in the HTML code.

- Add a <script> tag that contains the JavaScript code. In this way, the code is either run automatically when the page loads or create a JavaScript function that will be called in response to a client-side event. [4]

JavaScript supports a wide range of rich set of client-side events. It is easy to learn and implement. [4]

1.2.3 Cascading Style Sheets (CSS)

CSS is an acronym for cascading style sheets. It is a powerful browser technology that provides styling of the dynamic web pages to give them pleasant look. It helps to improve the document by separating data and presentation. That means that CSS is used for formatting information of HTML document. Formatting information includes fonts, background, colors, layout, and other aspects of the appearance of the HTML document and not the content of the document. HTML document format by CSS has the advantage of one style sheet being used for many pages instead of having different style sheets for different documents. In this project, style sheet is used for the formatting of menus and data tables. [4]

Dynamic Hypertext Markup Language (DHTML) uses JavaScript to manipulate CSS to dynamically change the appearance or looks of tags and position of elements in an HTML document. For example, if a page has several buttons that users can click to change the appearance of the page based on their personal preference. Styles in CSS are created using name/pairs separated by a colon. This is called property. Each property can be formatted using different values. There are two types of CSS styles: inline styles and document-level style sheets. Inline styles determine the appearance of the individual tags in an HTML document. This is defined using the STYLE attribute together with a string that contains the name/value pairs. On the other hand, document-level style sheets determine global formatting for HTML tags. This is defined by creating document-level sheets within a set of <STYLE>…<STYLE> tag pairs. This is placed in the <HEAD> section of a document. [4]

1.2.4 Extensible Markup Language (XML)

XML stands for extensible Markup Language. It is a primary way to transport data in and around .NET applications. This technology is popular today in web applications because it is platform and application-independent. This means that it does not matter whether the two applications are written in different languages or use different operating system; XML makes it possible for them to exchange data. Data is carried as pure texts. This application, Web Based Error Reporting Management Application, uses some XML technology for data exchange. For example the menu is built using X system to show to different hierarchy. [1]

Microsoft's .NET Framework uses XML heavily and gives ASP.NET applications a rich set of features for manipulating XML data. The benefits of using XML in modern application include the following:

- Adoption: XML is common. It is used by many companies to store data. Whenever data needs to be shared, the first consideration is XML.
- Extensibility and flexibility: XML does not impose rules about data semantics. This makes it easy to fit in to any type of data and is cheapest to implement.
- Related standards and tools: XML has got tools (example, parsers) and the surrounding standards (such as XML schema, XPath, and XSLT). These help in creating and processing of XML documents. Because of this, programmers in any language have ready-made components for reading XML, verifying that XML is valid, verifying XML against set of rules (schema), searching XML, and transforming one format of XML into another. [1]

1.2.5 MySQL

MySQL is a relational database management system (RDMS) that is designed to run as a server providing multi-user access to a number of databases it contains. MySQL runs on many different platforms system platforms including Microsoft Windows. This application uses MySQL data source to store data. There is need to store and retrieve data for this system. MySQL is based on SQL technology and it is easy to use. It makes creating, manipulating and reorganizing data to meet the different needs of the users. .NET Framework has API classes to access the stored data. This is called ADO.NET. For .NET Framework to communicate with MySQL database, the MySQL connector driver must be installed. It is added to the project. [1]

2 WEB BASED ERROR REPORTING MANAGEMENT APPLICATION

Web Based Error Reporting Management Application helps to keep a log of customers/users errors and their solutions. When a report for a problem is made, it is helpful and time-saving to search the database to see if such problem has been previously reported and solved. Otherwise it is registered and time frame for the solution defined for the error.

The application provides two different levels of users with their respective privileges and functions, namely:

1. Personnel
2. Administrator

The personnel are the users of the Web Based Error Reporting Management Application while the administrator is the personnel with additional functionalities. The table below is the table of functionalities of the application, description and priority. 5 is the highest value of priority and 3 the lowest.

Table 2-1 Application task functions and priorities, highest = 5 and lowest = 3

Task #	Task	Description	Prerequisite	Priority
1	Login	Logs user in to use the application	None	5
2	Report error	Allows customers to report errors	1	5
3	View/edit error list	Allows for viewing/editing of registered errors	1	5

4	Register solution	Allows for registering of solution of customers' error	1	5
5	View/edit solution list	Allows for viewing/editing of registered solutions	1	5
6	Search error/solution	Allows searching for a previously reported error or solution	1	5
7	Register new application users	Allows the administrator to register new users of the application	1	4
8	View/edit application users list	Allows the administrator to view/edit the list of all application users	1	4
9	Change password	Allows application users to change their password	1	4
10	Display/edit application user record	Allows application users to display/edit own record	1	4
11	Send email to new application users	Allows administrator to send email to new application users	1, 7	4
12	Register new products	Allows for registering new products	1	3
13	View/edit product list	Allows for viewing/editing of the products	1	3

14	Register new product platform	Allows for registering new product platforms	1	3
15	View/edit product platform list	Allows for viewing/editing of the product platforms	1	3

From the Table 2-1 above, we can see that the functions that have the highest value of priority are the ones that have to do with errors, solutions and search. They include:

- Register error
- View/edit errors
- Register solution
- View/edit solutions
- Search for error/solution

Some other functions have the second highest priority of 4. But other functions are nice to have, hence their low priority of 3. These functions are rarely used. They are mainly the functions that deal with products and product platforms.

The next diagram shows the use case diagram for the Web Based Error Reporting Management Application. It shows the functionalities of the system. A close look at the diagram identifies two actors - administrator and personnel.

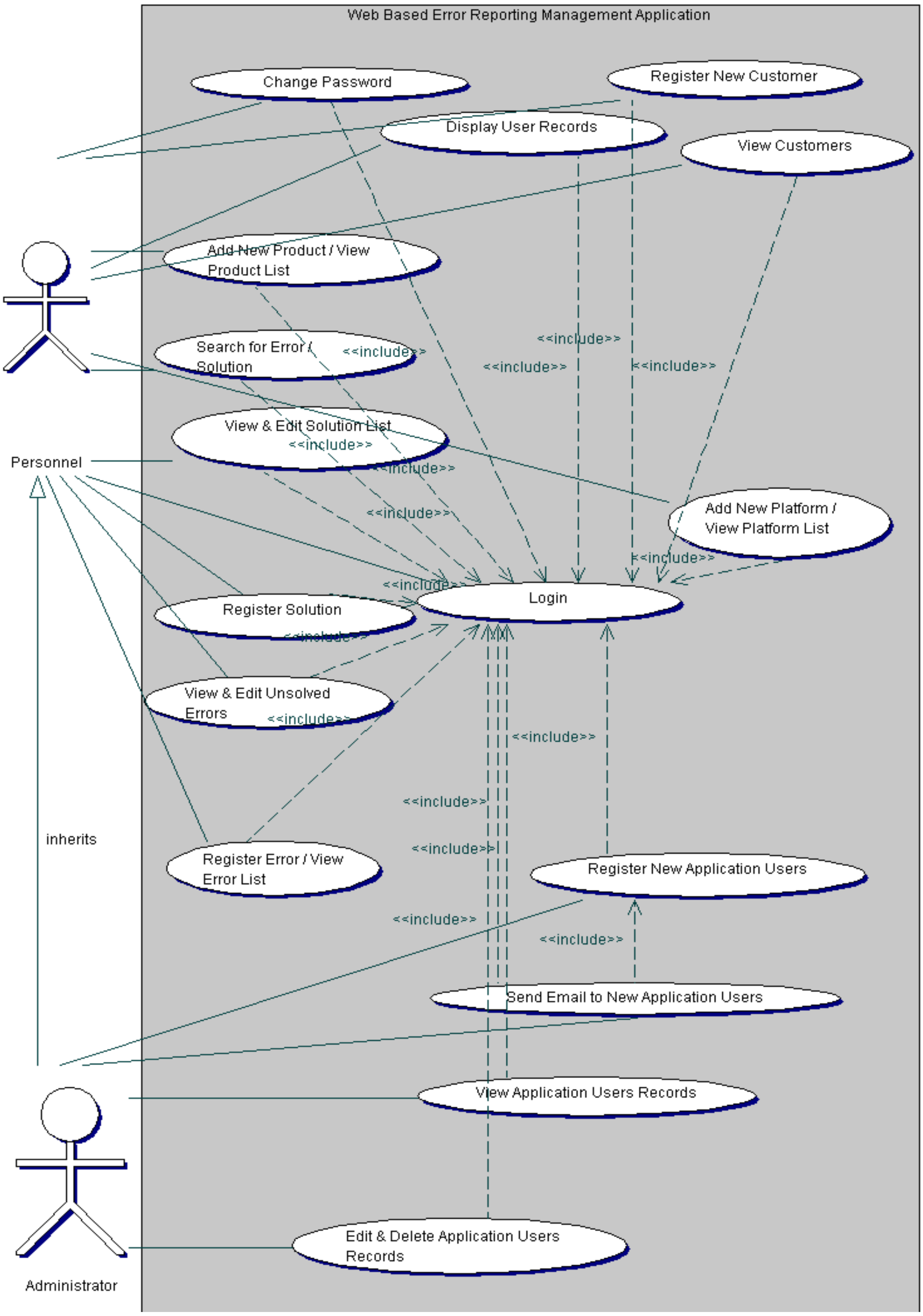

Web Based Error Reporting Management Application
Change Password
Register New Customer
Display User Records
View Customers
Add New Product / View Product List
Search for Error / Solution
View & Edit Solution List
Add New Platform / View Platform List
Login
Register Solution
View & Edit Unsolved Errors
Register Error / View Error List
Register New Application Users
Send Email to New Application Users
View Application Users Records
Edit & Delete Application Users Records
<<include>>
Personnel
inherits
Administrator

Figure 2-1 Use case diagram of Web Based Error Reporting Management Application

As shown in Figure 2-1, there is relationship between administrator and personnel actors. Administrator is personnel with additional functions. In the figure, it is shown as inherit. Administrator inherits the personnel and adds his own more functions. Functions like register new application users, send email to new users, view application users records and edit/delete application users records can only be performed by the administrator. All other functions include the login function. That means that you must log in (login function) before you can call other functions.

2.1 Personnel functionalities

The personnel or helpdesk staffs are the workers that work directly with the customers or users. These receive reports as calls, emails, or any other means from them and search the system for previously registered error and solution to the same problem and get back to the error reporters. The functionalities they handle include the following:

2.1.1 Login function

This function allows the application users, in this case the personnel, to login to the system and use the different functions allocated. Anonymous users are logged out and cannot use the system. The following diagram is the sequence diagram for this process. It outlines the steps or sequences of operation for the login function.

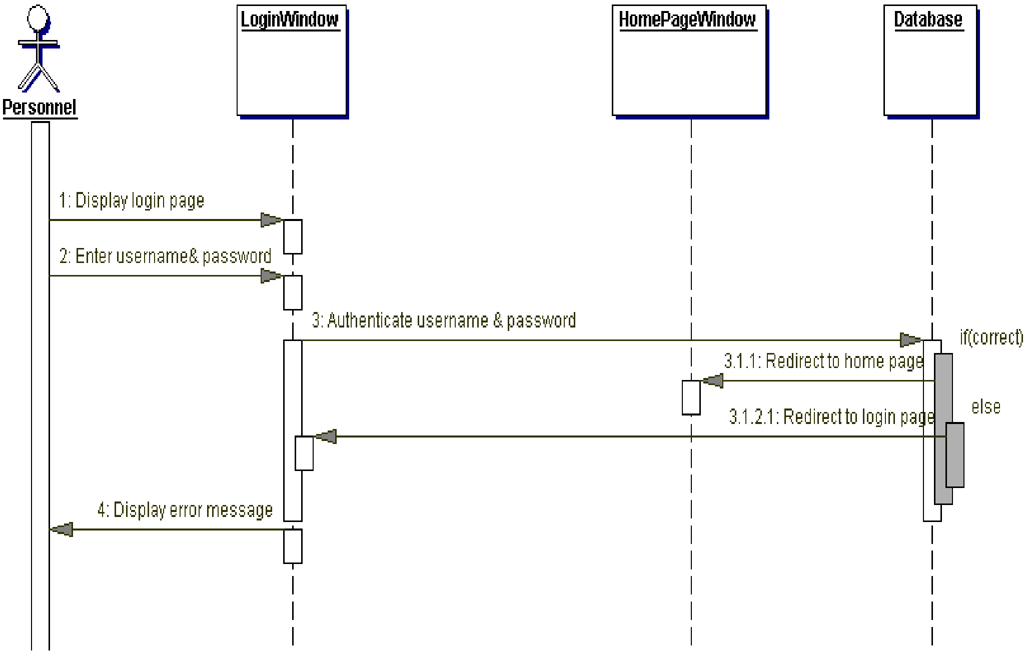

Personnel
LoginWindow
HomePageWindow
Database
1: Display login page
2: Enter username& password
3: Authenticate username & password
if(correct)
3.1.1: Redirect to home page
else
3.1.2.1: Redirect to login page
4: Display error message

Figure 2-2 Sequence diagram for login function

As shown in Figure 2-2, the user displays the login page and enters the username and password. These are authenticated against the data stored in the database. If authenticated (that means the username and password are correct) the users will be redirected to the default page (home page) otherwise he is redirected back to the login page with error message displayed. He can try again.

2.1.2 Change Password function

The purpose of this function is to allow the user of the application to change the password. This is necessary for security purposes. It is encouraged that passwords be changed from time to time. The following diagram is the sequence diagram for change password function.

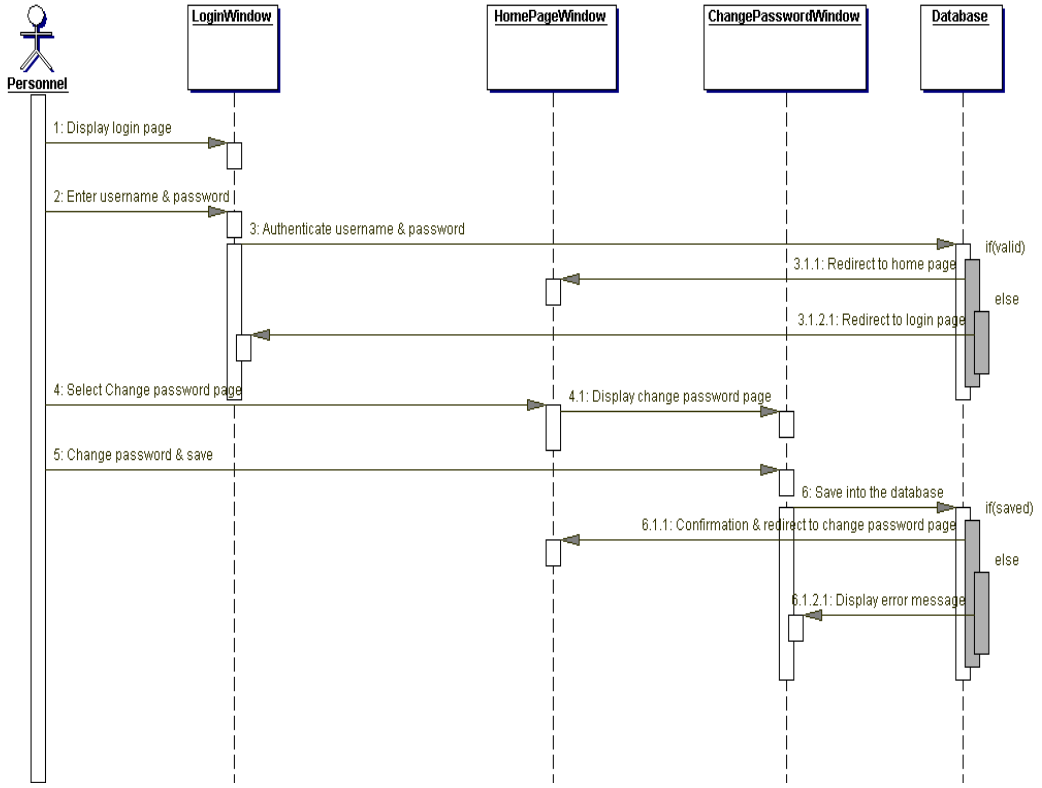

Personnel
LoginWindow
HomePageWindow
ChangePasswordWindow
Database
1: Display login page
2: Enter username & password
3: Authenticate username & password
if(valid)
3.1.1: Redirect to home page
else
3.1.2.1: Redirect to login page
4: Select Change password page
4.1: Display change password page
5: Change password & save
6: Save into the database
if(saved)
6.1.1: Confirmation & redirect to change password page
else
6.1.2.1: Display error message

Figure 2-3 Sequence diagram for changing the password

From the Figure 2-3 above, the user logs in with right username and password. He is taken to the home page. He chooses change password from the menu and is taken to the page. He enters the old password, new password and retype new password and submits. If everything is correct, the password is changed and he is redirected to the home page otherwise he get error message displayed on the screen.

2.1.3 Display user account function

This is the function that allows the user to display his account information and update the information if so desired. The diagram below shows the sequence diagram for display user account function.

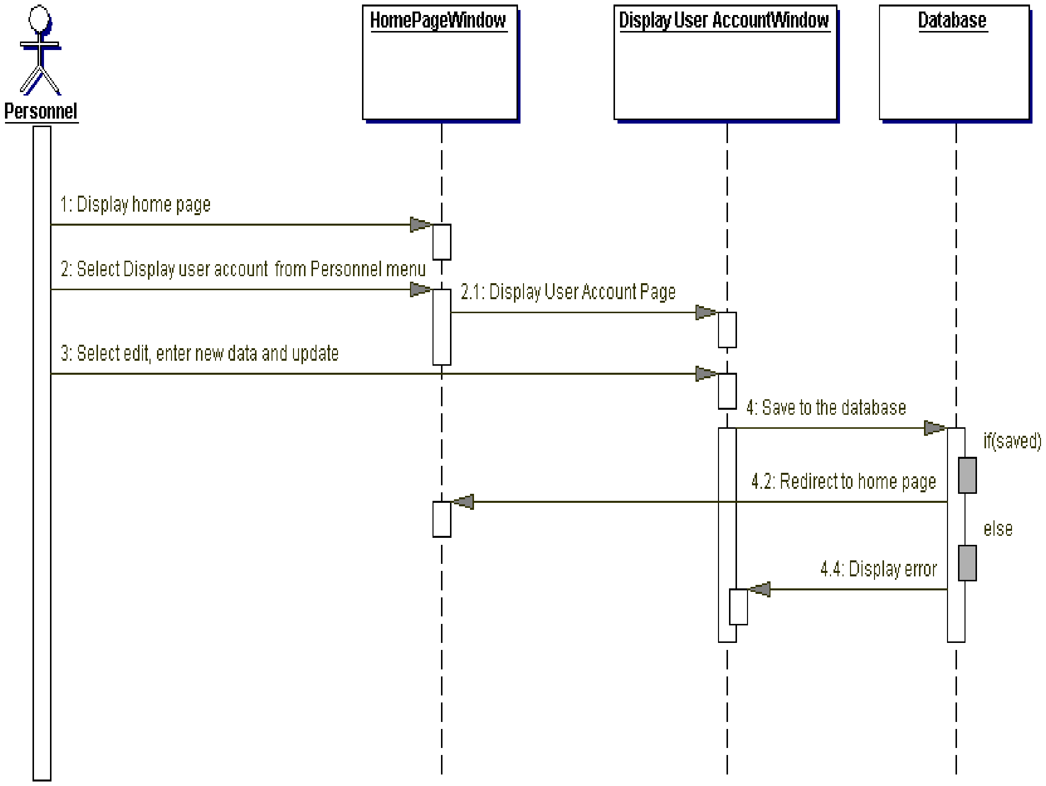

Personnel
HomePageWindow
Display User AccountWindow
Database
1: Display home page
2: Select Display user account from Personnel menu
2.1: Display User Account Page
3: Select edit, enter new data and update
4: Save to the database
if(saved)
4.2: Redirect to home page
else
4.4: Display error

Figure 2-4 Sequence diagram for displaying user account information

The illustration in Figure 2-4 is a sequence diagram for displaying the user account information. When the user (in this case the personnel user) displays the home page, he selects Display User Account from Personnel menu. The page is displayed. If he chooses edit, edit table will be displayed where he will enter new data and click update button. The data is saved to the database and he is redirected to the home page otherwise error message is displayed.

2.1.4 Register customer/product user function

This function allows for registering a new customer to the system. With this the database of the customers can be kept.

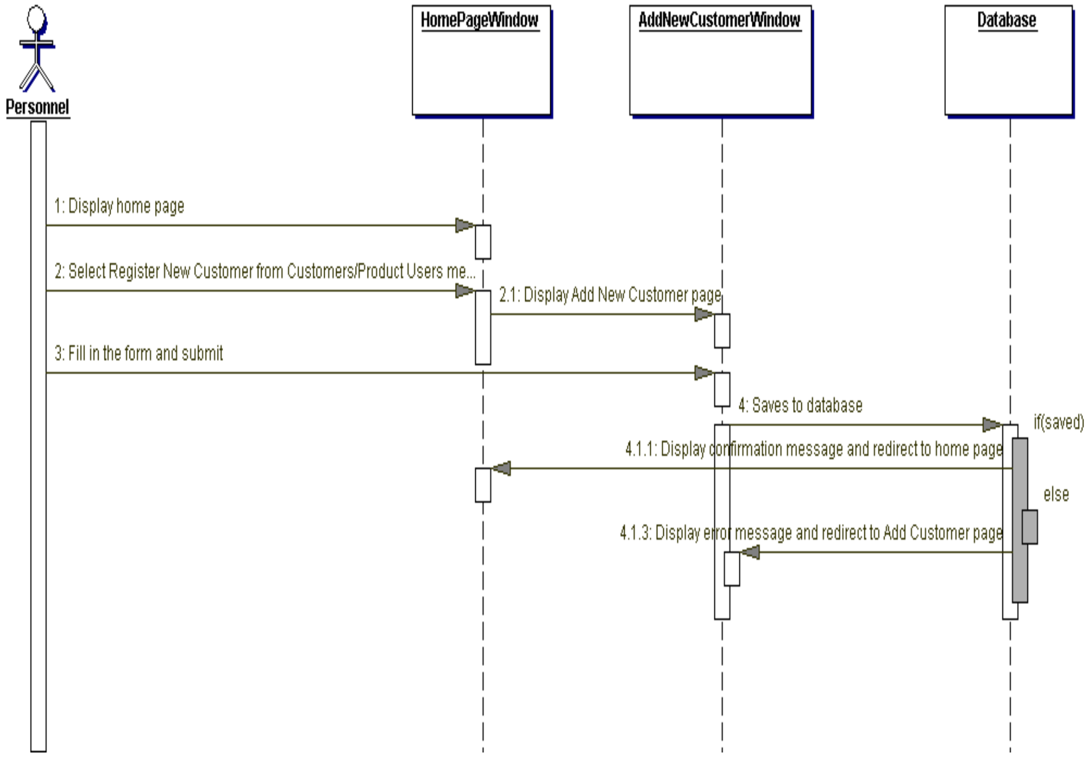

Personnel
HomePageWindow
AddNewCustomerWindow
Database
1: Display home page
2: Select Register New Customer from Customers/Product Users me...
2.1: Display Add New Customer page
3: Fill in the form and submit
4: Saves to database
if(saved)
4.1.1: Display confirmation message and redirect to home page
else
4.1.3: Display error message and redirect to Add Customer page

Figure 2-5 Sequence diagram for registering new customer or product users

A new customer can be registered to the system and the Figure 2-5 shows the sequence diagram for doing that. The application user displays the home page. Then he selects the register new customer/product users from the menu, and it is displayed. He fills the form and hits the submit button. The data will be saved to the database if it is valid and confirmation message is displayed, and he is redirected to the home page. Otherwise error message is displayed.

2.1.5 View Customers/Product Users function

This function allows the application user to display or view the list of the customers/product users registered to the system. The sequence diagram below illustrates the process of viewing the list and updating the data.

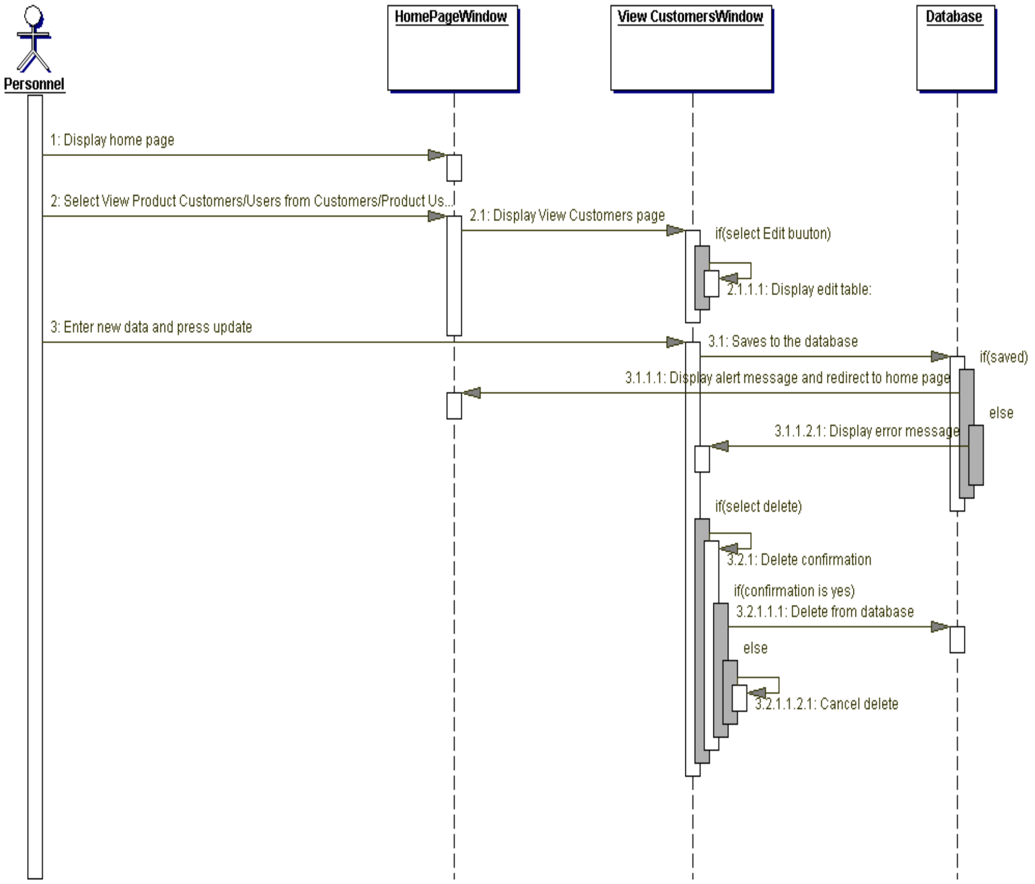

Personnel
HomePageWindow
View CustomersWindow
Database
1: Display home page
2: Select View Product Customers/Users from Customers/Product Us...
2.1: Display View Customers page
if(select Edit buuton)
2.1.1.1: Display edit table:
3: Enter new data and press update
3.1: Saves to the database
if(saved)
3.1.1.1: Display alert message and redirect to home page
else
3.1.1.2.1: Display error message
if(select delete)
3.2.1: Delete confirmation
if(confirmation is yes)
3.2.1.1.1: Delete from database
else
3.2.1.1.2.1: Cancel delete

Figure 2-6 Sequence diagram for viewing customers/product users

As shown in Figure 2-6, the application user displays the home page. He selects the view customers/product users from the Customers/Product Users menu. The page is displayed. He then chooses the action to perform. If he selects 'edit', edit table will be displayed where he will make his changes and click the 'update' button to update the data in the database. If the data is valid, it will be saved, else error message will be displayed. If he chooses 'delete', a confirmation will be displayed. If he selects 'yes', the data will be deleted from the database. Otherwise the delete operation is cancelled.

2.1.6 Add New Product function

This function allows for new products to be registered to the database.

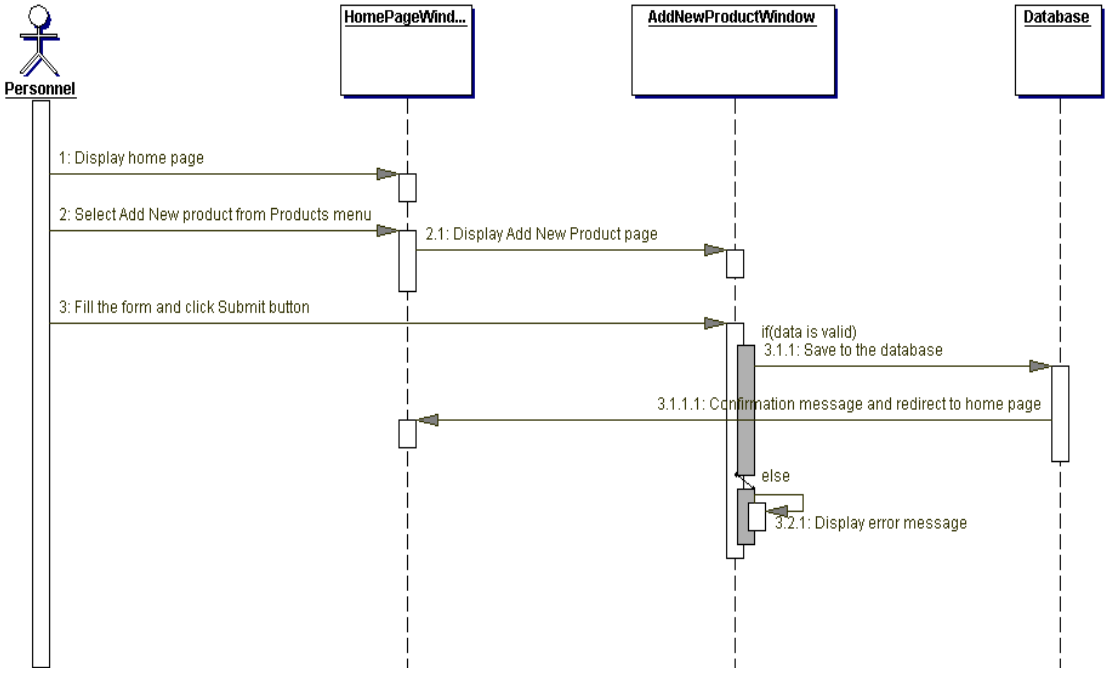

Personnel
HomePageWind...
AddNewProductWindow
Database
1: Display home page
2: Select Add New product from Products menu
2.1: Display Add New Product page
3: Fill the form and click Submit button
if(data is valid)
3.1.1: Save to the database
3.1.1.1: Confirmation message and redirect to home page
else
3.2.1: Display error message

Figure 2-7 Sequence diagram for adding a new product

The process sequence for adding a new product is illustrated in Figure 2-7 above. The home page is displayed. From the home page, the Add new product form is selected from Products menu, which displays it. Then the displayed form is filled out and the submit button is clicked to save to the database. If the data is valid, then it is saved to the database and confirmation displayed. Otherwise an error message is displayed on the screen.

2.1.7 View Product List function

This function lets the user of the application to view the list of products available in the system. That is the list of products stored in the database. This allows the data to be edited or deleted depending on the selection choice.

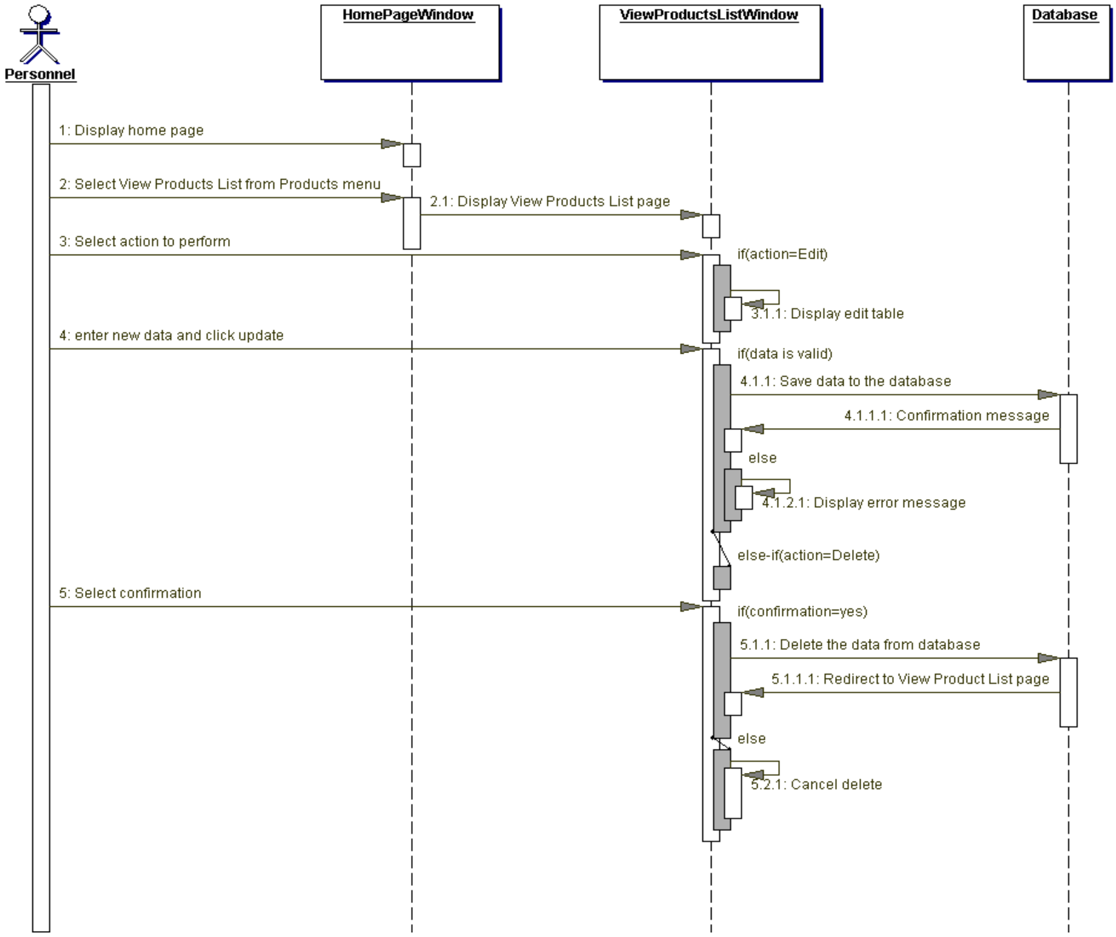

Personnel
HomePageWindow
ViewProductsListWindow
Database
1: Display home page
2: Select View Products List from Products menu
2.1: Display View Products List page
3: Select action to perform
if(action=Edit)
3.1.1: Display edit table
4: enter new data and click update
if(data is valid)
4.1.1: Save data to the database
4.1.1.1: Confirmation message
else
4.1.2.1: Display error message
else-if(action=Delete)
5: Select confirmation
if(confirmation=yes)
5.1.1: Delete the data from database
5.1.1.1: Redirect to View Product List page
else
5.2.1: Cancel delete

Figure 2-8 Sequence diagram for viewing the product list

Figure 2-8 shows the sequence diagram for viewing the list of products. The home page is displayed. Then from the Products menu, View Products List page is selected. If edit is selected from the operations that can be performed, the edit table is displayed. The new data is entered and update button is clicked. The data is saved to the database if it is valid and confirmation is displayed. Otherwise error message is displayed. But on the other hand, if 'delete' is clicked, a confirmation message will be displayed asking to confirm the delete operation. If the confirmation is 'yes', the product will be deleted from the database. Otherwise the delete action is cancelled.

2.1.8 Error Register function

The Error register function allows the application user to register the errors/faults from the customers/product users. This helps to keep track of the errors and later solutions. The following sequence diagram shows the error register operation.

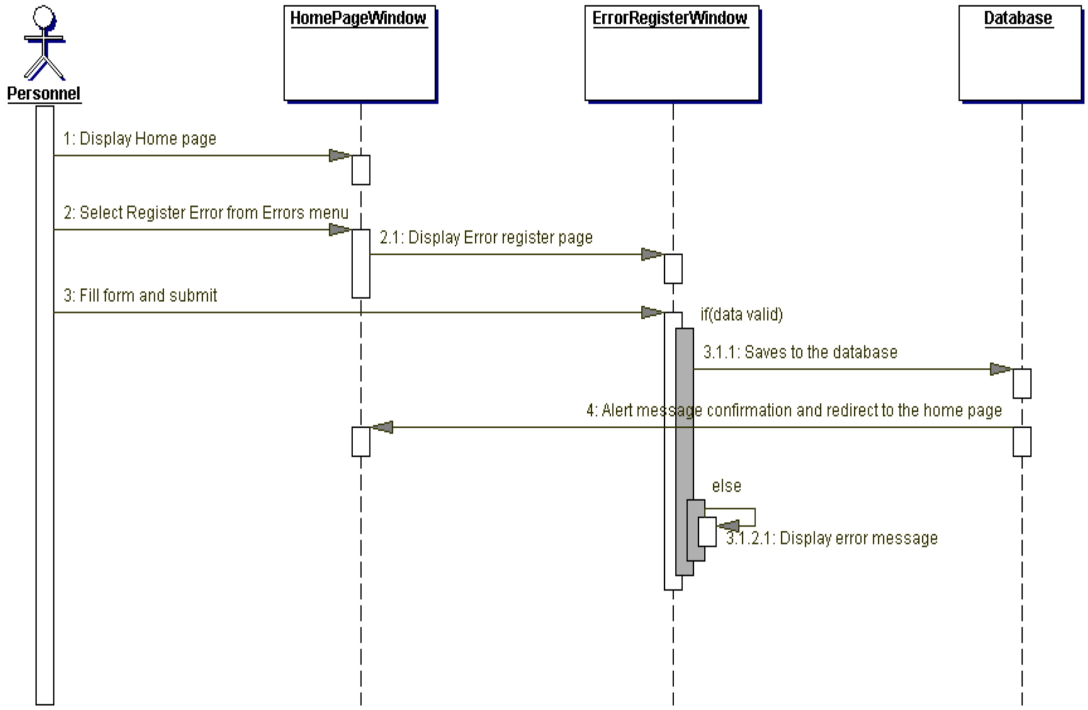

Personnel
HomePageWindow
ErrorRegisterWindow
Database
1: Display Home page
2: Select Register Error from Errors menu
2.1: Display Error register page
3: Fill form and submit
if(data valid)
3.1.1: Saves to the database
4: Alert message confirmation and redirect to the home page
else
3.1.2.1: Display error message

Figure 2-9 Sequence diagram for registering errors

Figure 2-9 is the sequence diagram for registering errors from both customer and product users alike. The home is requested and displayed. Then the Register Error is selected from Errors menu and displayed. Then the form is filled and the submit button is clicked. If the data is valid, it is saved to the database and confirmation is displayed and redirected to the home page. Otherwise, error message is displayed.

2.1.9 View Error List function

This function is used for viewing the list of error registered in the system. Then different operations can be performed.

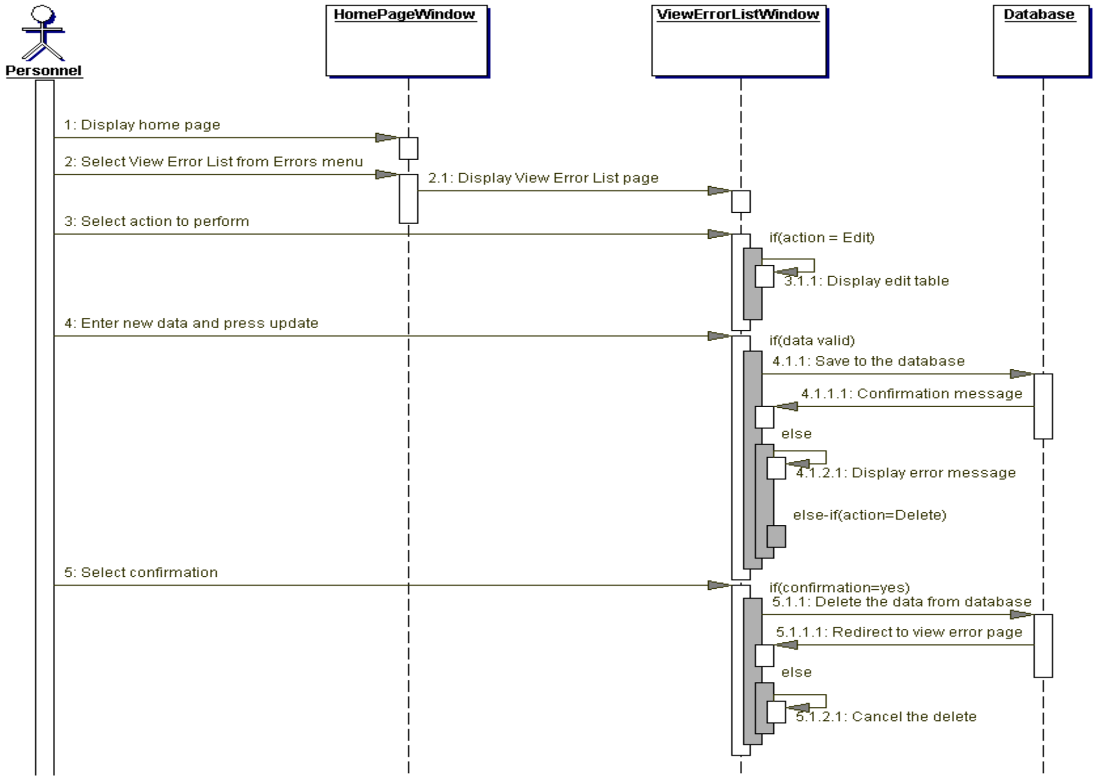

Personnel
HomePageWindow
ViewErrorListWindow
Database
1: Display home page
2: Select View Error List from Errors menu
2.1: Display View Error List page
3: Select action to perform
if(action = Edit)
3.1.1: Display edit table
4: Enter new data and press update
if(data valid)
4.1.1: Save to the database
4.1.1.1: Confirmation message
else
4.1.2.1: Display error message
else-if(action=Delete)
5: Select confirmation
if(confirmation=yes)
5.1.1: Delete the data from database
5.1.1.1: Redirect to view error page
else
5.1.2.1: Cancel the delete

Figure 2-10 Sequence diagram for viewing the error list

The Figure 2-10 above shows the sequence diagram for viewing the list of errors registered to the system. The home page is displayed, from the Errors menu the view error list is selected, and the page is displayed. Then the user selects the action to perform. If the action to perform is ‘edit’, edit table will be displayed. The new data is entered and update button is clicked. This saves the data to the database if the data is valid and confirmation message is displayed, otherwise error message is displayed. On the other hand, if the action performed is chosen to be ‘delete’, a delete confirmation is displayed. If ‘yes’ is selected the data is deleted from the database, otherwise the delete operation is cancelled.

2.1.10 View Unsolved Errors List function

The following diagram is the sequence diagram for the view unsolved errors list function.

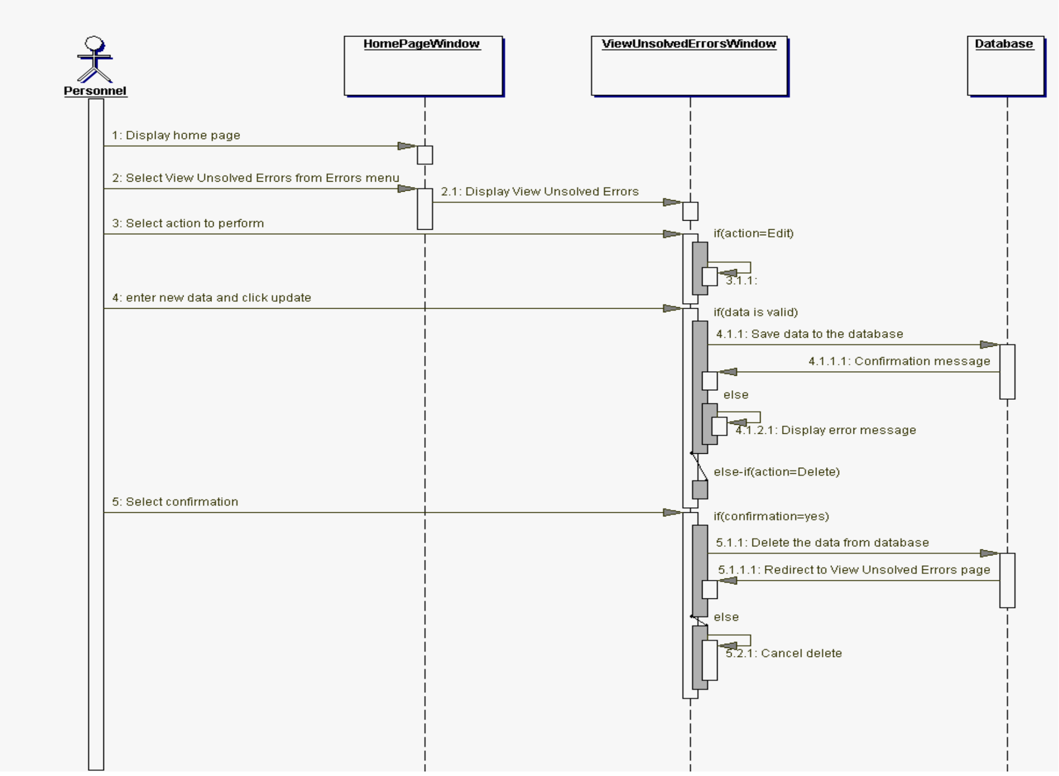

Personnel
HomePageWindow
ViewUnsolvedErrorsWindow
Database
1: Display home page
2: Select View Unsolved Errors from Errors menu
2.1: Display View Unsolved Errors
3: Select action to perform
if(action=Edit)
3.1.1:
4: enter new data and click update
if(data is valid)
4.1.1: Save data to the database
4.1.1.1: Confirmation message
else
4.1.2.1: Display error message
else-if(action=Delete)
5: Select confirmation
if(confirmation=yes)
5.1.1: Delete the data from database
5.1.1.1: Redirect to View Unsolved Errors page
else
5.2.1: Cancel delete

Figure 2-11 Sequence diagram for viewing unsolved error list

The Figure 2-11 above shows the sequence diagram for viewing unsolved error list. The home page is displayed. Then from the Errors menu, the view unsolved error list is selected. The page is displayed. Then he selects the action to perform. If the action to perform is ‘edit’, edit table will be displayed. The new data is entered and update button is clicked. This saves the data to the database if the data is valid and confirmation message is displayed. Otherwise error message is displayed. On the other hand, if the action performed is chosen to be ‘delete’, a delete confirmation is displayed. If ‘yes’ is selected from the options, the data is deleted from the database and redirected to view unsolved errors. Otherwise the delete operation is cancelled.

2.1.11 Register Solution function

The register solution function allows solutions for errors to be registered to the system. The following shows the sequence diagram for register solution.

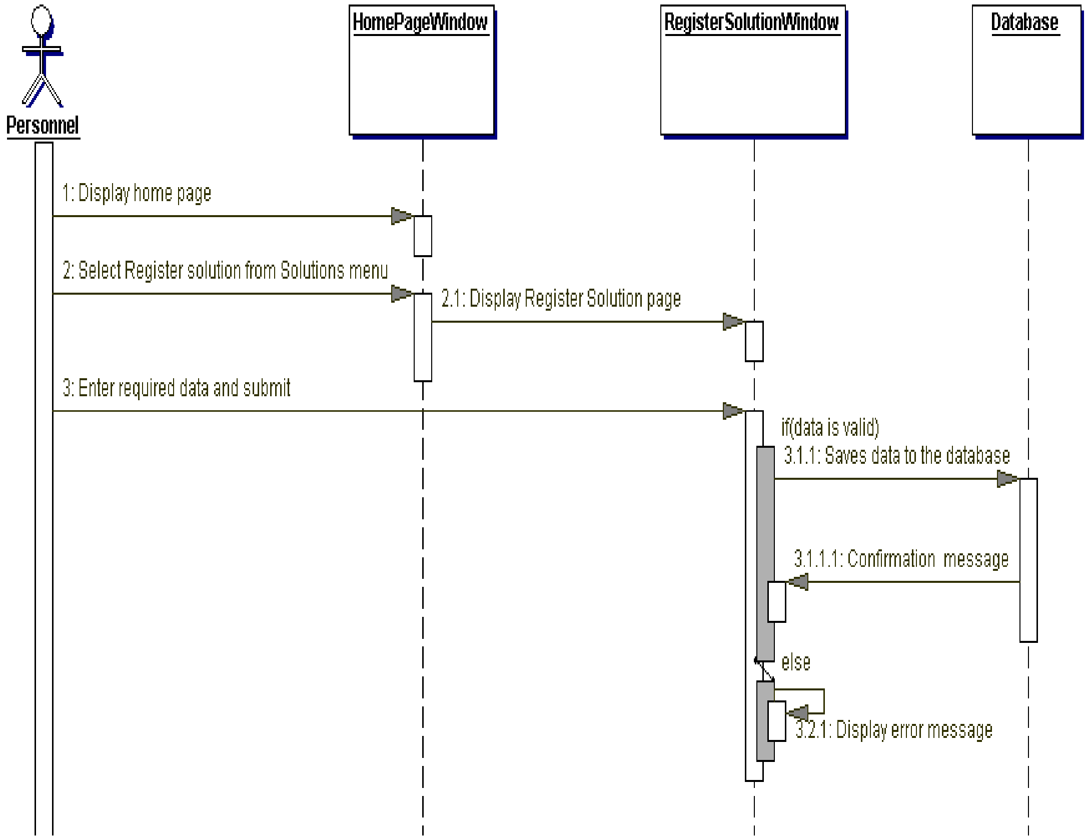

Personnel
HomePageWindow
RegisterSolutionWindow
Database
1: Display home page
2: Select Register solution from Solutions menu
2.1: Display Register Solution page
3: Enter required data and submit
if(data is valid)
3.1.1: Saves data to the database
3.1.1.1: Confirmation message
else
3.2.1: Display error message

Figure 2-12 Sequence diagram for registering solution

Figure 2-12 shows the sequence diagram for registering solution for error. The user displays the home page. From solutions menu, he selects register solution and the page is displayed. He then enters the required data and submits the data. If the data entered is valid, it is saved to the database and confirmation is displayed and redirected to the register solution page. If the data entered is not valid, error message is displayed.

2.1.12 View Solutions List function

The view solutions list allows for the viewing of the registered solution and displaying of the solution file. The next diagram shows the sequence diagram for viewing the list of registered solutions.

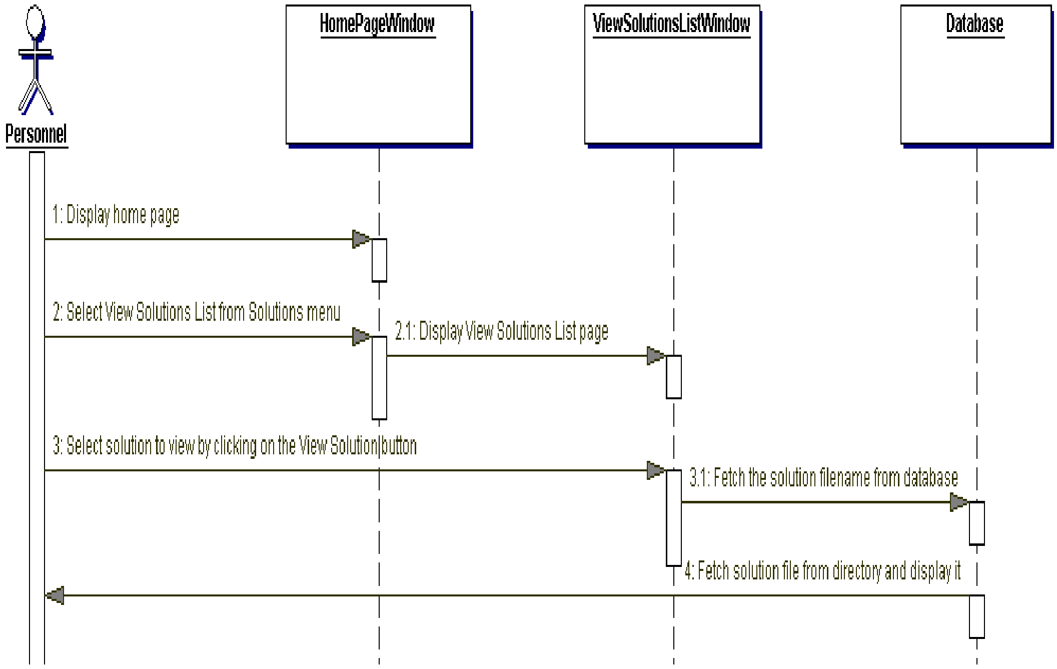

Personnel
HomePageWindow
ViewSolutionsListWindow
Database
1: Display home page
2: Select View Solutions List from Solutions menu
2.1: Display View Solutions List page
3: Select solution to view by clicking on the View Solution button
3.1: Fetch the solution filename from database
4: Fetch solution file from directory and display it

Figure 2-13 Sequence diagram for viewing solutions

As shown in Figure 2-13, the sequence diagram for viewing the solutions, display the home page. From the Solutions menu, select view solutions list page and it is displayed. Select the solution to view by clicking on the View Solution button. The solution filename is fetched and the file is fetched from the server.

2.1.13 Search data function

The search function allows the user to search for errors and solutions that are registered based on search word(s). The following sequence diagram shows the sequence of events.

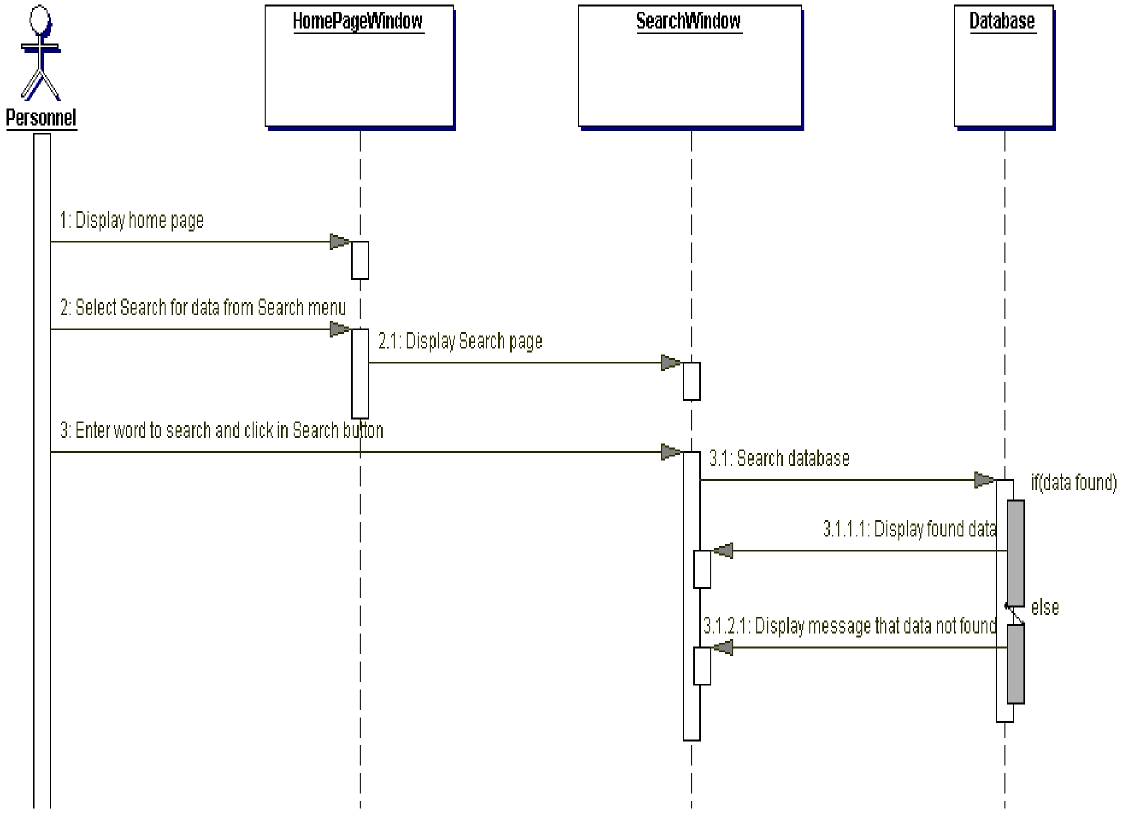

Personnel
HomePageWindow
SearchWindow
Database
1: Display home page
2: Select Search for data from Search menu
2.1: Display Search page
3: Enter word to search and click in Search button
3.1: Search database
if(data found)
3.1.1.1: Display found data
else
3.1.2.1: Display message that data not found

Figure 2-14 Sequence diagram for search function

The illustration in Figure 2-14 for searching for errors and solutions starts with the display of home page. From the search menu, search for data is selected. This displays the search page. The user enters the word(s) for search and clicks the search button. The database is searched and if data is found, it is displayed. Otherwise a message is displayed showing no data found.

2.1.14 Sign Out function

The sign out function allows the user to end the login session and redirect to the login page. The following sequence diagram shows the sign out function.

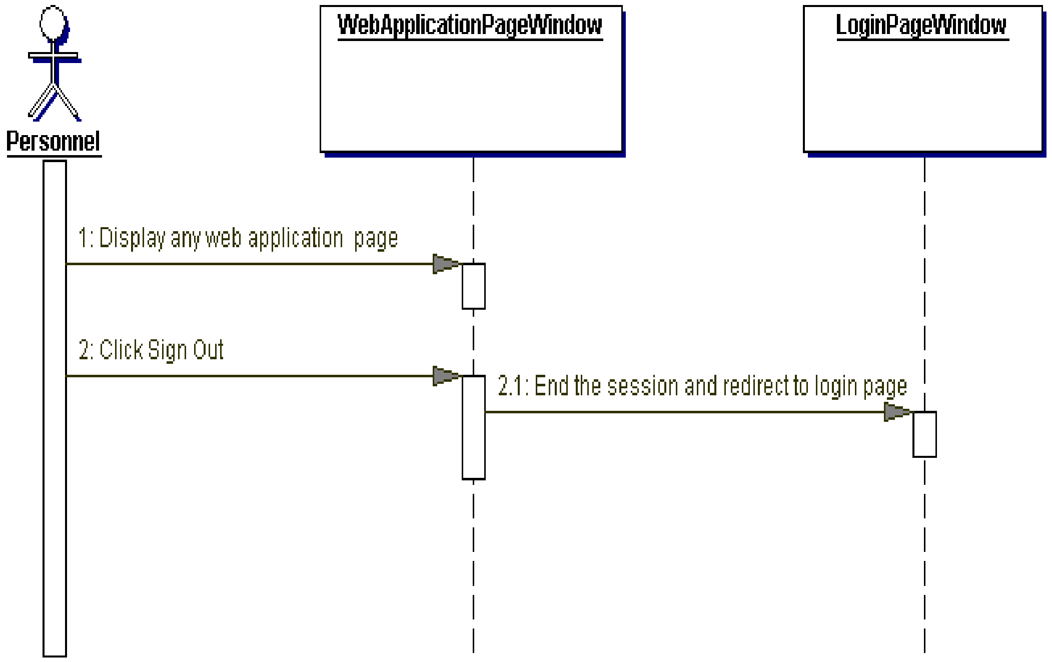

Personnel
WebApplicationPageWindow
LoginPageWindow
1: Display any web application page
2: Click Sign Out
2.1: End the session and redirect to login page

Figure 2-15 Sequence diagram for sign out function

The Figure 2-15 above shows the sequence diagram for sign out function. The sign out link can be accessed from any page apart from the login page. When a page is displayed, clicking on sign out will end the session and redirect to the login page.

2.2 Administrator functionalities

The administrator is the personnel with additional functions. This means that he performs the functions of the personnel with more added functions exclusive for the administrator. These functions include:

2.2.1 Register New Personnel function

The administrator possesses the sole privilege to create new accounts for new personnel users of the application. When accounts for new users are created, the username and password generated are emailed to the user using the email entered. The following diagram shows the sequence diagram for registering new users.

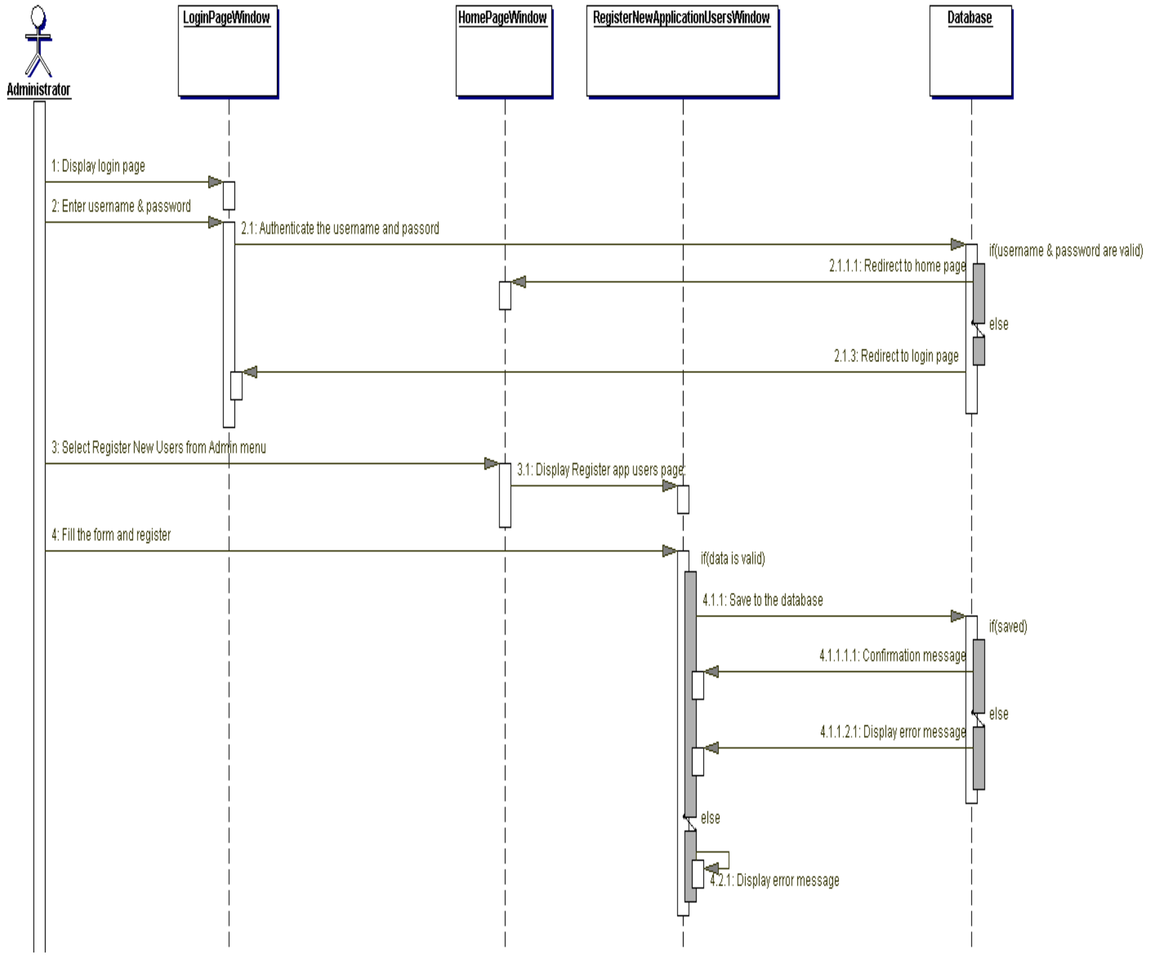

Administrator
LoginPageWindow
HomePageWindow
RegisterNewApplicationUsersWindow
Database
1: Display login page
2: Enter username & password
2.1: Authenticate the username and passord
if(username & password are valid)
2.1.1.1: Redirect to home page
else
2.1.3: Redirect to login page
3: Select Register New Users from Admin menu
3.1: Display Register app users page
4: Fill the form and register
if(data is valid)
4.1.1: Save to the database
if(saved)
4.1.1.1.1: Confirmation message
else
4.1.1.2.1: Display error message
else
4.2.1: Display error message

Figure 2-16 Sequence diagram for the administrator creating user account

The Figure 2-16 shows the steps involved in creating new application user account. The administrator displays the login page. He enters the user name and password. These are authenticated. If they are valid, he gets redirected to the home page, otherwise there is error display. From the home page he selects the register new users from Admin menu, and the page is displayed. He fills the form and clicks the submit button. If data is valid, it is saved into the database and a confirmation message is displayed. Also email will be sent to the user. Otherwise error message is displayed.

2.2.2 View Application Users Accounts function

The administrator, in addition to creating the new application users account, has the exclusive privilege too to view the accounts of all users and to edit them. The following diagram shows a sequence diagram for this.

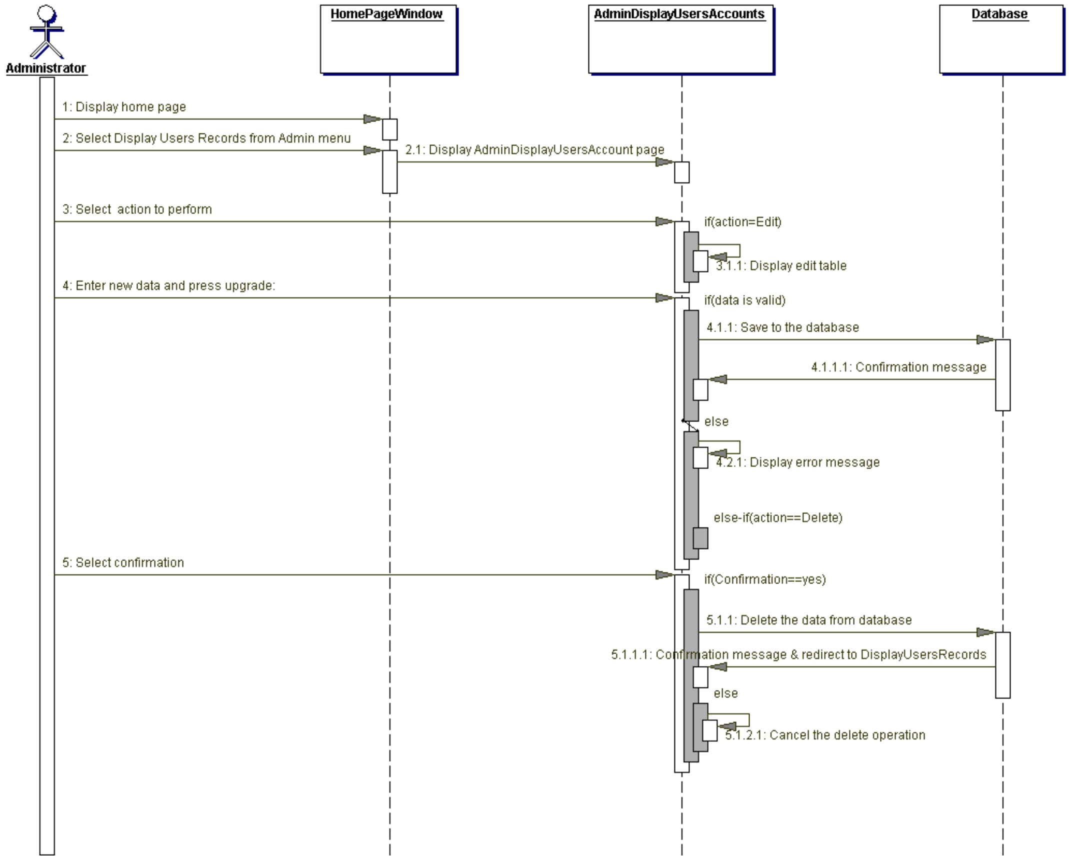

Administrator
HomePageWindow
AdminDisplayUsersAccounts
Database
1: Display home page
2: Select Display Users Records from Admin menu
2.1: Display AdminDisplayUsersAccount page
3: Select action to perform
if(action=Edit)
3.1.1: Display edit table
4: Enter new data and press upgrade:
if(data is valid)
4.1.1: Save to the database
4.1.1.1: Confirmation message
else
4.2.1: Display error message
else-if(action==Delete)
5: Select confirmation
if(Confirmation==yes)
5.1.1: Delete the data from database
5.1.1.1: Confirmation message & redirect to DisplayUsersRecords
else
5.1.2.1: Cancel the delete operation

Figure 2-17 Sequence diagram of administrator viewing user records

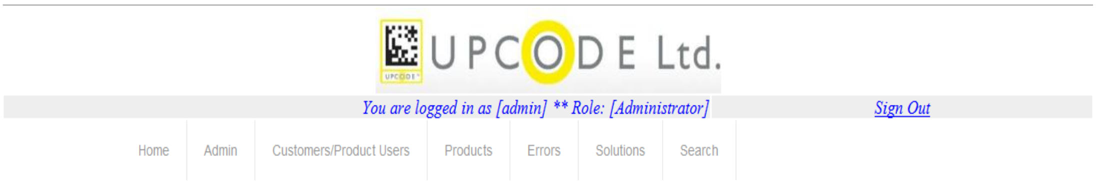
The Figure 2-17 shows the sequence involved in viewing the users records. The administrator displays the home page. From the Administrator menu, he selects the Display Users Records and displays the page. If the action performed is ‘edit’, the edit table is displayed and he enters new data and clicks ‘update’. If the data is valid, it is saved into the database, and a confirmation message is displayed. Otherwise, it displays error message. If the action performed is ‘delete’, a delete confirmation is displayed. If the confirmation is ‘yes’, the data is deleted from the database and confirmation displayed. Otherwise, the delete is cancelled.

3 SYSTEM DESIGN

Design has to do with the translation of analysis model into design model. It is this design model that will be converted to implementation. Design makes implementation easier. I discuss design under three main categories, namely interface, database and security.

3.1 Interface design

The interface provides interactivity to the users of this application- both personnel and administrator. This gives the application user-friendliness- a hallmark for a good software solution.



You are logged in as [admin] ** Role: [Administrator]

Figure 3-1 Uniform interface

The illustration in Figure 3-1 shows the common interface of all the pages in this management solution. The topmost is the logo of the company. This is followed by the login status bar. It shows the user's user name and the role of the user who has logged in to the system. The rightmost in this bar is the sign out link. This is supposed to end the session for a user and takes him back to the login in page. Next row is the menu bar. This shows the menus for the system and brings appropriate menu bar depending on the role of the user.

3.2 Database design

The database is based on MySQL. This stores all the data in this application. There are different tables that are used for storing data. The following tables show all the different tables and data types used in this project.

Table 3-1 Program Users table for storing application users information

Column Name	**Description**	**Data type**
userid	The id of the user (primary key)	Int
lastname	The last name of the user	Varchar
firstname	The first name of the user	Varchar
email	The email of the user	Varchar
phone	The phone of the user	Varchar
username	The user name of the user	Varchar
password	The password of the user	Varchar
role	The user's role (either administrator or personnel)	Varchar

Table 3-1 above shows the program users table with its columns names and data types. This table stores data about the people that use this web application. The userid defines the primary key for the table. The lastname and firstname define the last name and first name respectively. The username stores information about user name of the user and the role identifies whether the user has an administrator or personnel right.

Table 3-2 Product Platform table for storing platform information

Column Name	**Description**	**Data type**
Platform_ID	The ID of the platform (primary key)	Int
Platform_Name	The name of the platform	Varchar
Platform_Category	The category of the platform	Varchar
Platform_Description	The description of the platform	Varchar

Table 3-2 above is the table that stores data about product platform. It also shows the column names and their data types. The Platform_ID stores information about the ID if the platform. Here, it is a primary in this table. The Platform_Name stores the name of the platform. Each platform has a category and Platform_Category stores such information. The information about the platform is stored in Platform_Description. The following table is the show of the table for storing product information.

Table 3-3 Product table for storing product information

Column Name	**Description**	**Data type**
Product_ID	The id of the product (primary key)	Int
Product_Name	The name of the product	Varchar
Product_Version	The version of the product	Varchar
Product_Category	The category of the product	Varchar
Year_Developed	The year the product is developed	Varchar

Product_Platform	The platform of the product	Varchar
Platform_ID	The platform id (foreign key)	Int
Product_Description	The description of the product	Varchar

The above table, Table 3.3, is the database table for storing product information. It shows the names of the fields and their data types in the table. The Product_Name stores the name of product, Product_Version stores the version of product and Year_Dveloped stores the year the product is developed. The Platform_ID is a foreign key in this table and primary in the Platform table. In the Product_Description, the information about the product is stored. The following table is for storing solution about errors.

Table 3-4 The Error Solutions table for storing error solutions information

Column Name	**Description**	**Data type**
Solution_ID	The id of the solution (primary key)	Int
Error_ID	The error id for the solution (foreign key)	Int
Solution_Date	The date of the solution	Date
Solution_Description	The description of the solution	Varchar
Uploaded_FileName	The file name of the solution	Varchar

Table 3-4 shows the structure of error solutions table. It defines its fields and data types. For instance, every solution has Solution ID (Solution_ID) that identifies it. The Error_ID in this table is a foreign key, but a primary in the Error Register table. The Solution_Date stores date information about the error solution. The Solution_Description stores simple information about the solution, while the Uploaded_FileName contains detailed information about the solution.

Table 3-5 The Error Register table for storing error information

Column Name	**Description**	**Data type**
Error_ID	The id of the error registered (primary key)	Int
CustomerOrUser_Name	The Customer or the user reporting the error	Varchar
Contact_Person	The contact person	Varchar
Email	The email	Varchar
Phone	The phone	Varchar
Product_Name	The name of the product having the error	Varchar
Product_Manufacturer	The manufacturer of the product	Varchar
Product_Model	The model of the product	Varchar
Entry_Personnel	The personnel registering the error	Varchar
Entry_Date	The date for registering the error	Date
Entry_Time	The time the error is registered	Time
Problem_Description	The description of the error	Varchar
Error_Status	The status of error (either solved or unsolved)	Varchar
NumberOfDaysAssigned	The number of days assigned to the develop solution for the error	Varchar
Solution_Due_Date	The due date	Date

Table 3-5 is the database table for error register. The Error_ID identifies the ID of the error, the CustomerOrUser_Name stores information about the reporter of the error. Contact_Person, Email and Phone identify the contact information of the reporter. Product_Name, Product_Manufacturer and Product_Model store information about the reported error. Problem_Description stores information about the error and Solution_Date about the date the solution is registered.

Table 3-6 The Customers table for storing customer information

Column Name	**Description**	**Data type**
Customer_ID	The id of the customer (primary key)	Int
Customer_Name	The name of the customer	Varchar
Address	The address of the customer	Varchar
Country	The country of the customer	Varchar
Email	The email	Varchar
Contact_Person	The person to contact	Varchar
Contact_Person_Designation	The designation of the contact person	Varchar
Contact_Person_Phone	The person of the contact person	Varchar

Table 3-6 is the table for customer table. Each customer has ID (Customer_ID), and Customer_Name identifies the customer or the reporter. Customer_Name, Address, Country, Email and Contact_Person_Phone store information about the customer.

Table 3-7 The Countries table for storing information about countries

Column Name	**Description**	**Data type**
Country_Code	The code of the country	Int
Country_Name	The name of the country	Varchar

The last table, Table 3-7, stores information about the countries of the customers. Country_Code is the code of the country and the Country_Name is the name of the country.

3.2.1 Entity Relationship diagram

Entity relationship diagram is a good tool to represent the entities in a system and how they all interact with each other. It is useful for database design, as they represent what can be stored in database tables. The figure below represents the entities identified and their relationship with one another. Each entity has attributes with the first attribute being the primary key.

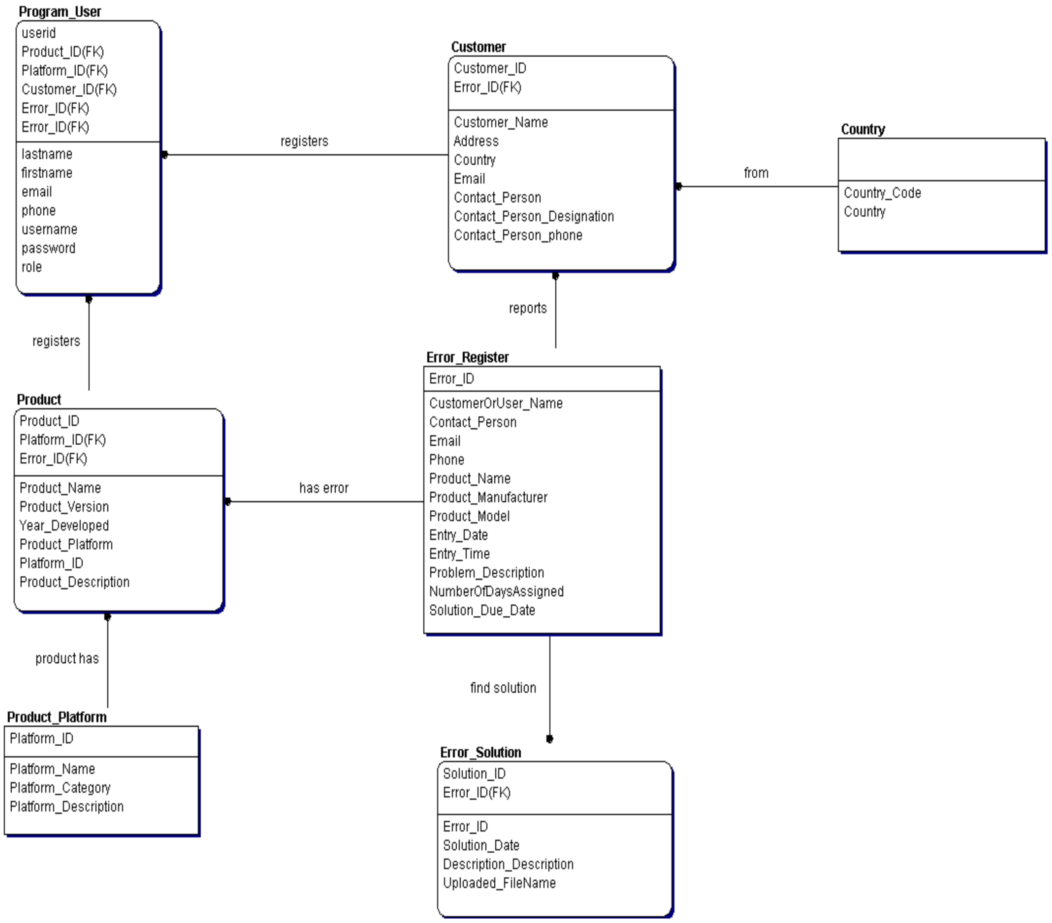


Figure 3-2 Entity relationship diagram for the Web Based Error Reporting Management Application

Figure 3-2 above is the entity relationship diagram for the application. Program_User entity has ‘registers’ relationship with Customer and Product entities. This means that program user registers bother customers and products into the system. Product entity has ‘product has’ relationship with Product_Platform entity, meaning that each product has product platform. Customer entity has ‘from’ relationship with Country entity while it has ‘reports’ relationship with Error_Register entity. Each customer is from a country and may report an error. Error_Register entity on the other hand has the ‘find solution’ relationship with Error_Solution entity. There is a relationship between registered error and solution when a solution is developed for an error.

3.3 Security design

The security of the application is another important concern, as this limits access to users. It authenticates users, logs unanimous users out of the application but allows authenticated users into the application. Authentication is based on form. Every form is authenticated before releasing the form to the user. This means that when a page is requested, it makes sure that it is delivered only to the right user. For example, there are pages that are only viewed by an administrator. Even if the personnel should type the page name on the address bar, it will not be delivered.

3.4 Application deployment

Deployment diagram is a diagram showing the run-time configuration of the processing nodes and the components that run on those nodes. The deployment diagram shows the hardware for the system, the software that is installed on that hardware, and the middleware used to connect separate machines to one another. The following diagram shows the deployment diagram for the system.

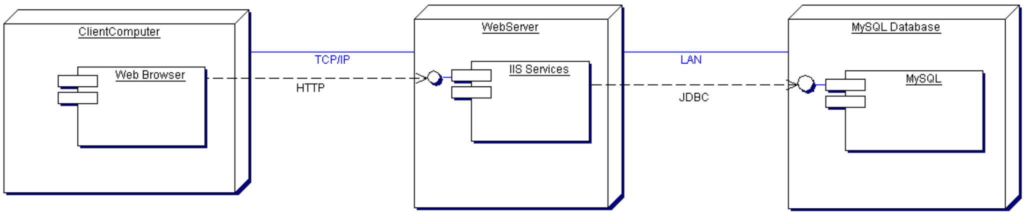


Figure 3-3 Deployment diagram showing the nodes and components

Figure 3-3 shown above, with any web browser running on a client computer, the user can access the IIS Services running on the UpCode Ltd.'s web server via HTTP. The web server receives requests from the client and delivers the page requested if the user is authenticated and has right to the requested page. From time to time, there is need to call the MySQL database. This is done through JDBC connector driver for MySQL and the communication between MySQL database and IIS web server is done via LAN. When all the components running on the nodes and the nodes too work properly and together, the application is delivered to the user.

4 IMPLEMENTATION

The design model is at this level is translated into implementation. Implementation involves coding the various parts of the application. This project work, Web Based Error Reporting Management Application, is implemented using ASP.NET 2.0 with C# language as the programming language. The coding of the pages was done in Visual Studio 2005, which is a powerful development tool (IDE), and all the data is stored and retrieved from MySQL database, which uses SQL syntax.

4.1 Menu Implementation

The menu was built in a class Menu.cs and returned as string to the webform. The following illustrates the menu implementation.

```
public string BuildAdminMenu()
{
 // Building menu
 StringBuilder buildMenu = new StringBuilder();
 buildMenu.Append("<div id=\"nav\">");
 // The wrapping div
 buildMenu.Append("<ul>");
 // Home menu item
 buildMenu.Append("<li class=\"first\"><a
 href=\"Default.aspx\">Home</a></li>");
 // Admin menu item
 buildMenu.Append("<li class=\"top\"><a
 href=\"DisplayUserAccount.aspx\">Admin</a>");
 buildMenu.Append("<ul>");
 buildMenu.Append("<li><a
 href=\"RegisterNewUsers.aspx\">Register
 New Users</a></li>");
 buildMenu.Append("<li><a
 href=\"AdminDisplayUsersAccounts.aspx\">Display Users
 Records</a></li>");      // View and edit
 buildMenu.Append("<li><a
href=\"ChangePassword.aspx\">Change
Password</a></li>");
buildMenu.Append("<li><a
href=\"DisplayUserAccount.aspx\">Display
User Account</a></li>");    // View and edit
buildMenu.Append("</ul>");
buildMenu.Append("</li>");

// Customers/Product Users menu item
buildMenu.Append("<li class=\"top\"><a
href=\"ViewCustomersUsersList.aspx\">Customers/Product
Users</a>");
buildMenu.Append("<ul>");
buildMenu.Append("<li><a
href=\"AddNewCustomer.aspx\">Register
New Customer</a></li>");
buildMenu.Append("<li><a
href=\"ViewCustomersUsersList.aspx\">View Product
Customers/Users</a></li>");   // View and edit
buildMenu.Append("</ul>");
buildMenu.Append("</li>");
...
buildMenu.Append("</ul>");
// End of unordered list
buildMenu.Append("</div>");
// End of wrapping div
// Return the menu string
return buildMenu.ToString();
}
```

Figure 4-1 Part of the code for building menu

Figure 4-1 shows part of the code for building the solutions menu. There are two methods that are used to build the menus- personnel menu and administrator menu. The above figure builds administrator menu. After the menu is built, the method returns it as string, which is styled by the MenuStylesheet.css file shown in Figure 4-2 to give it the appearance we see in the solutions. The figure below is the solution explorer showing the MenuStylesheet.css.

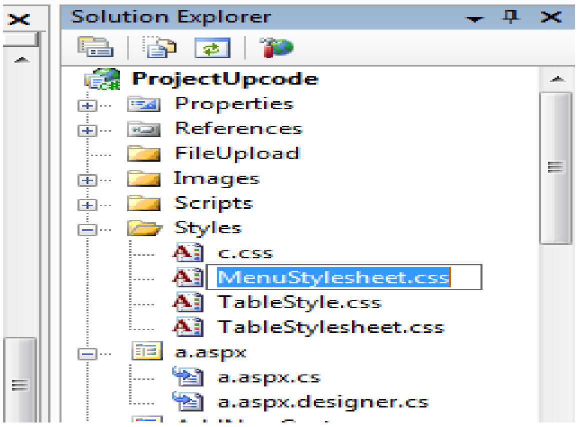


Figure 4-2 Solution explorer showing the MenuStylesheet.css

Figure 4-3 below shows the part of the code for the MenuStylesheet.css. The codes are used to style or format the menu. This gives it the appearance that it has.

```
*{
   margin:0px; padding: 0px;
 }

#nav {
font-family: arial, sans-serif;
position: relative;
width: 1000px;
height:56px;
font-size:14px;
color:#999;
margin: 0 auto;
}

#nav ul {
list-style-type: none;
}

#nav ul li {
float: left;
position: relative;
}

#nav ul li a {
text-align: center;
border-right:1px solid #e9e9e9;
padding:20px;
display:block;
text-decoration:none;
color:#999;
}
```

```
#nav ul li ul {
display: none
}

#nav ul li:hover ul {
display: block;
position: absolute;
}
…

#loginStatusRight:hover
{
      background: blue;
      color:Red;
}
```

Figure 4-3 MenuStylesheet.css for styling the menu

4.2 Authentication and Authorization

ASP.NET offers some security mechanisms to the web applications. Authentication makes sure that only the right users log in to the system and the anonymous users are prevented from having access. On the other hand, authorization makes it possible for authenticated users to have access to the right resources when they request such resources. These two mechanisms are employed in this project.

4.2.1 Form-Based Authentication and Authorization

Forms authentication is an ASP.NET's security mechanism that authenticates each user by verifying the credentials entered (usually the user name and password) into the web form, in this case login form, against the data stored in the database. On the other hand, authorization determines whether an identity should be granted access to a specific resource. This application uses the following configurations in the Web.config located in the root of the application directory. The Web.config file is the right place to make the necessary configurations for the entire application. The following is the content of the application's Web.config file.

```
<?xml version="1.0"?>
<configuration>
   <system.web>
         <authentication mode="Forms">
          <forms name="FormsAuthDB.AspxAuth" loginUrl="Login.aspx"
              protection="All" timeout="120"/>
         </authentication>
         <authorization>
           <deny users="?"/>
           <!--allow users="*"/-->
         </authorization>
    </system.web>
          <location path="Images/upcode_logo.jpg">
              <system.web>
                <authorization>
                        <allow users="*"  />
                </authorization>
            </system.web>
           </location>
</configuration>
```

Figure 4-4 The Web.config for the application

Figure 4-4 above shows the configuration of the Web.config to implement authentication and authorization. The following lines show that the type of authentication is forms based.

```
    <authentication mode="Forms">
        <forms name="FormsAuthDB.AspxAuth" loginUrl="Login.aspx"
            protection="All" timeout="120"/>
```

The loginUrl points to the login page where the user name and password are entered. The protection is set for all the pages. The timeout is set to 120 minutes, under which if no activity has taken place, the session is ended. The first authorization tag denies permission to anonymous users and gives permission to only authenticated users. The following lines illustrate this.

```
<authorization>
     <deny users="?"/>
     <!--allow users="*"/-->
 </authorization
```

The following lines shows how access to some resources cab be allowed.

```
      <location path="Images/upcode_logo.jpg">
              <system.web>
                <authorization>
                        <allow users="*"  />
                </authorization>
            </system.web>
        </location>
```

The location path is used to allow the resources, in this case the logo- upcode.jpg, to all users. This makes it possible for the logo to be displayed even in the login page. Without this line, the logo will be displayed only for authenticated users, which will not show the logo in the login page because it is at this stage that users are authenticated.

4.2.2 Form-Based Authentication and Role-Based Security

This application uses role-based form authentication. For authenticated users, their roles stored in the database are mapped to the users and the right menu is displayed depending on whether they are personnel or administrator. To do this, the best place to do the mapping is the AuthenticateRequest event in Global.asax. This event is fired each time a request is made. The application works in such a way that when a request is made, the role of the user is determined. From this, it is determined if he has the right to the page requested. If he has, the page is delivered. Otherwise, the page is not delivered. The following lines of code illustrate this.

```
protected void Application_AuthenticateRequest(Object sender,
EventArgs e)
{
    HttpApplication app = (HttpApplication) sender;
    if (app.Request.IsAuthenticated && app.User.Identity is
     FormsIdentity)
    {
       FormsIdentity identity = (FormsIdentity) app.User.Identity;

       // Get the role from the database for the user
       // First get the instance of class DatabaseConnect and pass
       //  identity name to the method
       DatabaseConnect databaseConnect = new DatabaseConnect();
       string role = databaseConnect.GetUserRole(identity.Name);

       // Create a GenericPrincipal  for the username and assign
       // to the request
        if(role != null)
           app.Context.User = new GenericPrincipal(identity, new
             string[] {role});
    }
}
```

Figure 4-5 AuthenticateRequest event in the Global.asax

Figure 4-5 illustrates the codes for AuthenticateRequest event in the Global.asax file. This file links roles onto forms authentication. After verifying that the user has been authenticated and that the authentication was done using forms authentication, a valid authentication cookie is attached to the request. The method, Application_AuthenticateRequest, will extract the user name from the cookie. It does this indirectly by casting app.User.Identity to FormsIdentity. A call is made to the database with the username from the FormsIdentity object's Name property as the parameter. This returns the role of the user. Then a new object, GenericPrincipal, is created that contains the role of the user which is passed to it. When this event is fired, ASP.NET compares the role name in the GenericPrincipal to the role of the user given access in Web.config.

The following figure shows the method that gets the user role and returns it for the Application_AuthenticateRequest event's FormsIdentity object.

```
public string GetUserRole(string name)
{
    // Read from the database
    MySqlCommand cmd = conn.CreateCommand();
    string role = "";
    try
    {
        cmd.CommandText = "SELECT role FROM program_users WHERE
          username='" +  name  +  "'";
        conn.Open();
        MySqlDataReader reader = cmd.ExecuteReader();

        if (reader.Read())
        {
            role = reader.GetString(0);
        }
        else
           return null;
    }
    catch (MySqlException)
    {
        return null;
    }
    finally
    {
        conn.Close();
    }

        return role;
   }
```

Figure 4-6 GetUserRole method in DatabaseConnect class

Figure 4-6 is a method in the Database class for getting the role of the user. It calls MySQL database with the username entered as parameter and returns the role to the caller. This role is assigned to the current request.

4.3 Application Functions Implementation

The application is composed of functions. These functions work together to achieve this web based error reporting system. All the functions in the design stage were successfully implemented. They include:

- Login
- Search
- Register solution
- View solutions list
- Register error
- View/edit error list
- Change Password
- View User Records
- Register New Customer
- View Customers list
- Add new product
- View product list
- Add new platform
- View platform list
- Register new application user
- Send email to new application user
- View application users records
- Edit/delete application users records

4.4 Database Connection

For the application to connect to the MySQL database, the Connector/Net MySQL database driver was installed. In the DatabaseConnect class, which is the class dedicated to handle all database operations, the following code illustrates the parameters that are initialized.

```
public class DatabaseConnect
{
   // Initialize mysql connection
   static  string connString = "SERVER=localhost;" +
    "DATABASE=upcode;" + "UID=root;" + "PASSWORD=";
   static MySqlConnection conn = new
      MySqlConnection(connString);
   …
}
```

Figure 4-7 Database connection parameters in DatabaseConnect class

The second line in Figure 4-7 above initializes the connection string which is a string. The server name is the ‘localhost’, database name is ‘upcode’, username is ‘root’ and the password is empty string. The third line shows the connection string passed to the object of the MySqlConnect. This object is used to connect to the database in all the methods in the DatabaseConnect class.

4.5 Organizing Application Files

The application was organized in folders for ease of files location. Figure 2-4 shows such organization. The application namespace ProjectUpcode is the application folder. This contains the *.cs class files, the *.aspx files, the Global.asax, the Web.config file and four folders - FileUpload, Images and Styles. The table below, Table 4-1, shows the organization of the application files.

Table 4-1 Organization of the application files

*.aspx files	These files are presentation files in pure html codes.
*.cs files	These files are code-behind files. They contain the business logic.
Global.asax	This file stores application event handlers, application declarations and other global application elements. It is located in the virtual root directory of the application.
Web.config	This file stores the configuration of the application.
FileUpload folder	This folder contains uploaded solution files.
Images folder	This folder contains the company logo picture file.
Styles folder	This folder stores the *.css files for styling the content of the aspx files.

5 TESTING

Software testing is a very important stage of software development. This is done with the intent of finding error and correcting them, and thereby making sure that the system meets specifications. This application, Web Based Error Reporting Management Application, was tested and bugs were found and corrected. In addition to my personal testing, I tested the application with the administrator. From this testing, he pointed out areas needing adjustments and corrections. Some of these were not necessarily bugs, but the way he wanted the output to look. These corrections were made. The testing revealed the following:

1. Different browsers have their different ways of rendering pages. The testing was done using Internet Explorer, Mozilla Firefox, Safari and Chrome as browsers. They give similar results, except that Internet Explorer, from Internet Explorer 7 and below, has some issues with the menu that that uses *.css file for styling the menu. The submenus tend to disappear on hovering. But this works fine in other browsers, and has been corrected in Internet Explorer 8.

2. The forms authorization was tested and found to be working. After each user is authenticated, authorization confirms that the user has the right to have the requested page. This makes it possible that the personnel users do not have the pages that only the administrator has the exclusive right.

3. The table sometimes displayed out of order especially when the fields contained lengthy text for display. To correct this, the lengthy of the field was tested before displaying the table. This only displays the limited text in the fields with such lengthy text while displaying the whole text in the edit form.

4. Mail server is needed for the application to be able to send email from it to users who have been created.

5. The application can save and retrieve data from the database. This feature works properly.

6. The upload file feature works perfectly well. The application makes it possible that only required file formats are allowed for file upload. This removes the risk of uploading files that contain dangerous scripts.

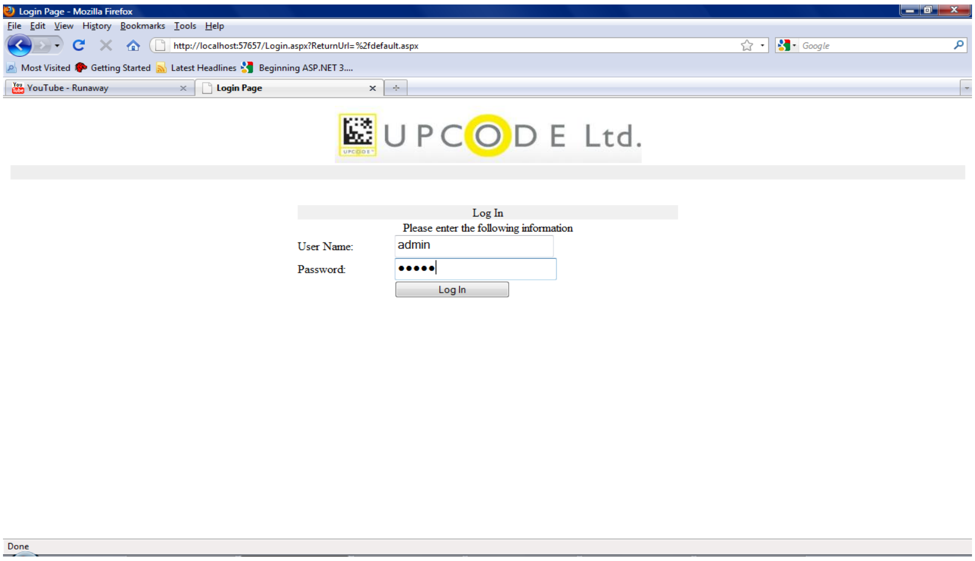
The web based error reporting system was deployed as shown in figure 3-3. The application runs in IIS (Internet Information Services) server. This is a web server application developed by Microsoft for use with Microsoft Windows. It is the second most used server behind Apache HTTP Server. Using web browser in any client computer, a connection is made to the IIS web server running on the company's server through HTTP. The IIS web server connects to the database server. The web server receives requests from the client and delivers the pages to the client browser in the form of HMTL. The IIS web server is configured to accept upload of files from users. Without adding the permission from users, file upload by users is not possible. That will generate errors in the browser.

5.1 Results

The following are the results from the application showing how it works. In this section, I show and explain how each function of this web application works.

5.1.1 Login Window

The following figures illustrate how the login page looks like. Whatever page that is requested from the address bar, the application redirects to the login page.


Login Page - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://localhost:57657/Login.aspx?ReturnUrl=%2fdefault.aspx
Google
Most Visited
Getting Started
Latest Headlines
Beginning ASP.NET 3....
YouTube - Runaway
Login Page
UPCODE Ltd.
Done

Log In

Please enter the following information

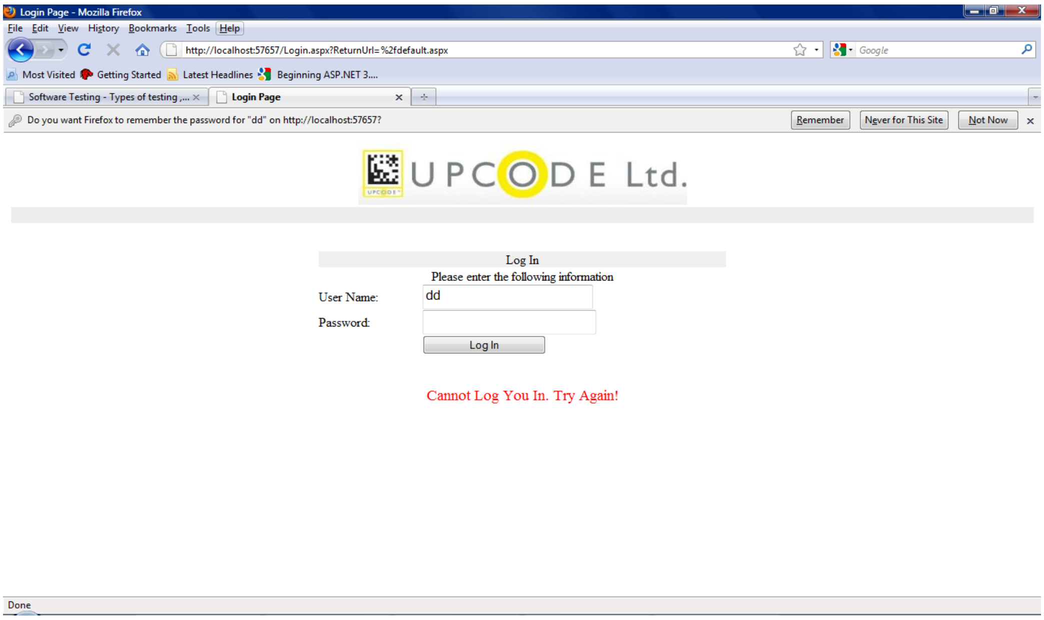
User Name: admin

Password: •••••

Log In

Figure 5-1 Login page

Figure 5-1 shows the login page. The user enters the username and password and hits the 'Log In' button. After authentication, the user is logged in to use the resources, this time, to access the pages, if the credentials are correct. The following illustration, Figure 5-2, shows the login page with incorrect credentials.


Login Page - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://localhost:57657/Login.aspx?ReturnUrl=%2fdefault.aspx
Google
Most Visited
Getting Started
Latest Headlines
Beginning ASP.NET 3....
Software Testing - Types of testing ,...
Login Page
Remember
Never for This Site
Not Now
UPCODE Ltd.
Done

Do you want Firefox to remember the password for "dd" on http://localhost:57657?

Log In

Please enter the following information

User Name: dd

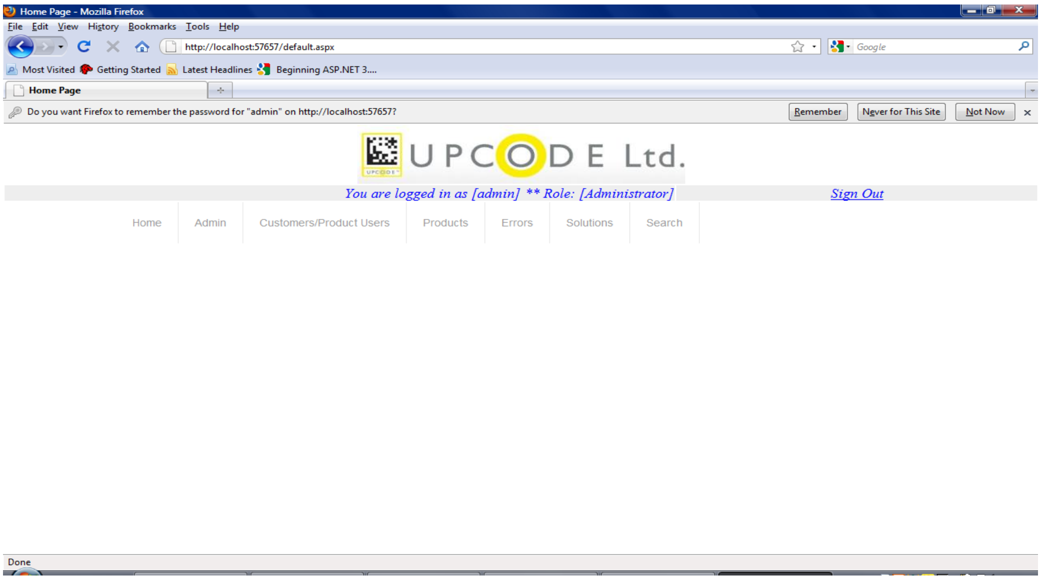
Password:

Log In

Cannot Log You In. Try Again!

Figure 5-2 Login page with incorrect credentials

The following figure shows the outlook when a user is authenticated and is logged in to the home page.



http://localhost:57657/default.aspx

Do you want Firefox to remember the password for "admin" on http://localhost:57657?

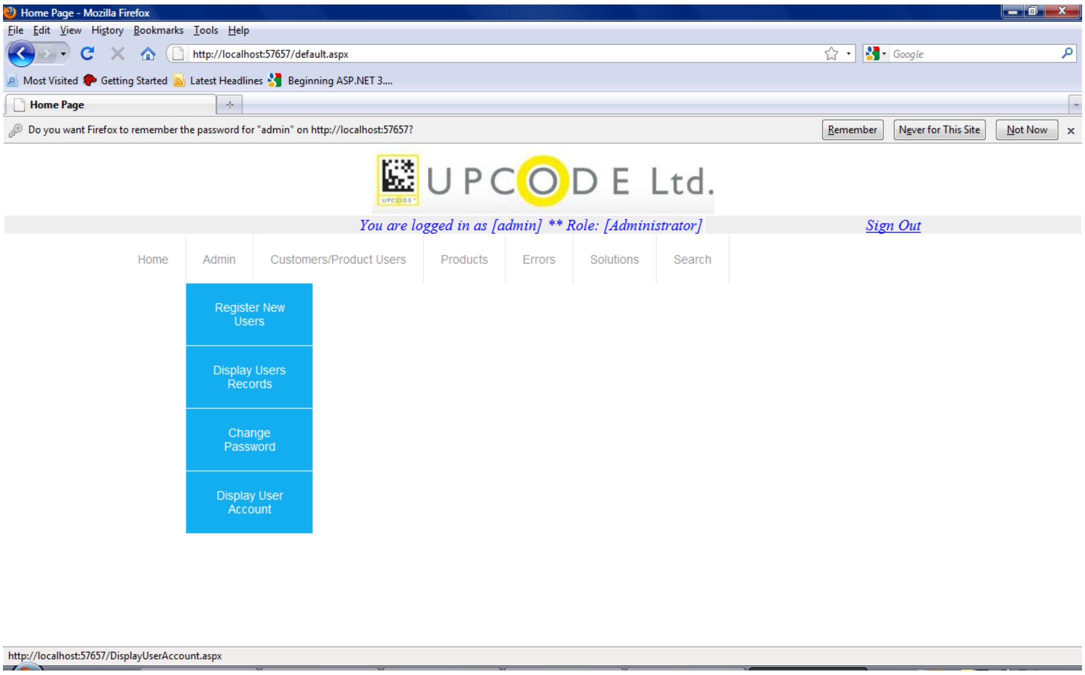
*You are logged in as [admin] ** Role: [Administrator]*

Figure 5-3 Default page after authentication

The login status in the home page indentifies the user with a login name and the role as shown in Figure 5-3.

5.1.2 Administrator and Personnel Roles Menus

The user is logged in as either an administrator or personnel. When this is done the right menu is displayed. The following two figures illustrate what the menus look like.

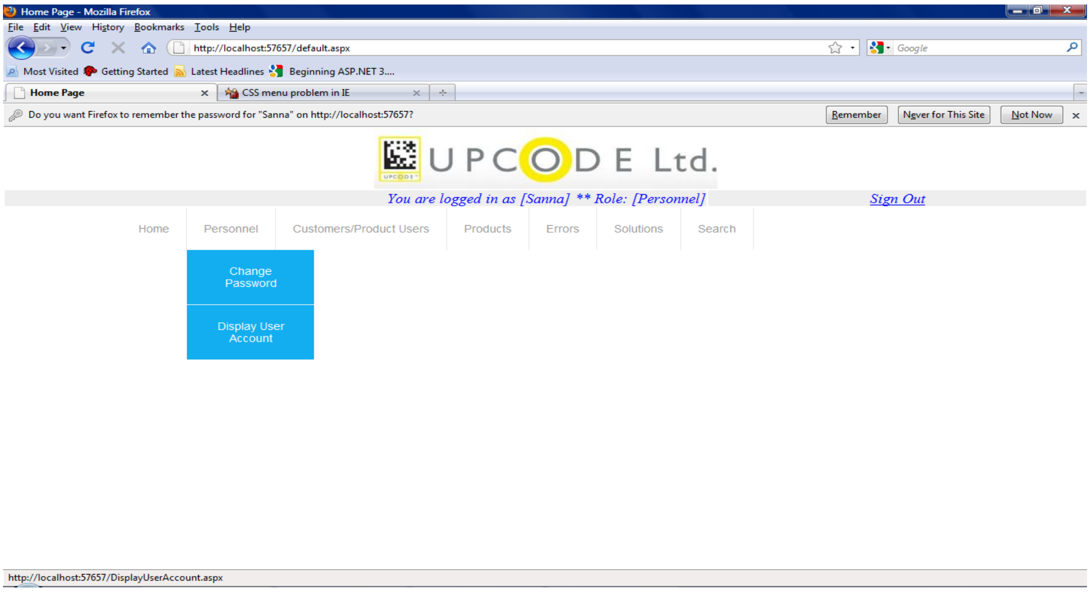


Do you want Firefox to remember the password for "admin" on http://localhost:57657?

You are logged in as [admin] ** Role: [Administrator]

Figure 5-4 Administrator functions menus page

Figure 5-4 shows the administrator functions menus. A look at the menu shows that the administrator has two additional menus that are not found in the personnel functions menus. With these functions the administrator will perform administrator-related operation. For example, creating new accounts for application users and displaying all the account users' information are operations performed with the administrator right.

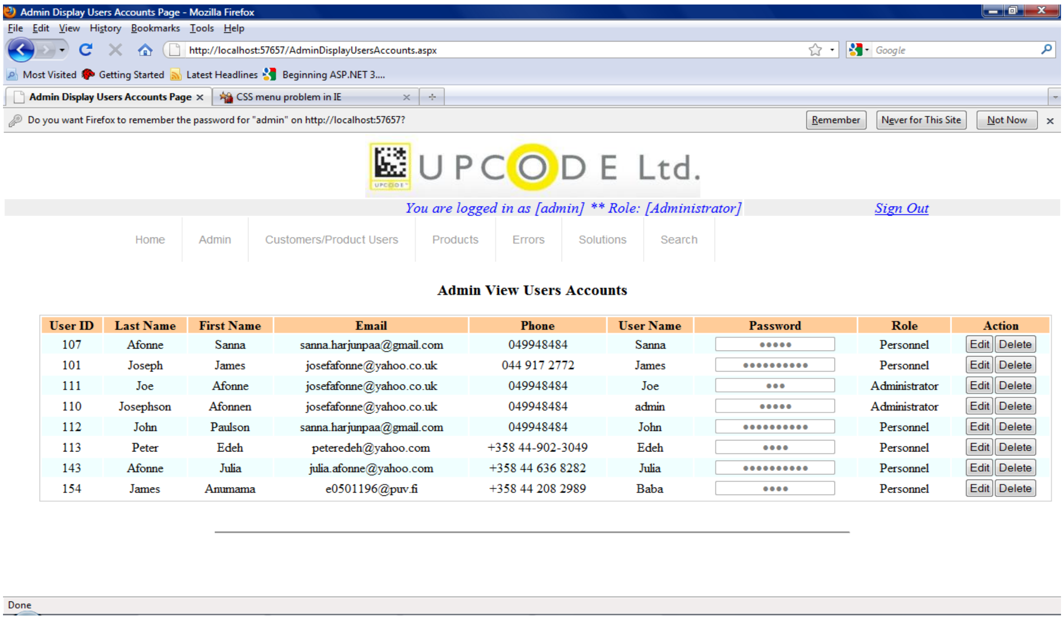

Home Page - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://localhost:57657/default.aspx
Google
Most Visited Getting Started Latest Headlines Beginning ASP.NET 3....
Home Page
CSS menu problem in IE
Remember
Never for This Site
Not Now
UPCODE Ltd.
Sign Out
Home
Personnel
Customers/Product Users
Products
Errors
Solutions
Search
Change Password
Display User Account
http://localhost:57657/DisplayUserAccount.aspx

Do you want Firefox to remember the password for "Sanna" on http://localhost:57657?

*You are logged in as [Sanna] ** Role: [Personnel]*

Figure 5-5 Personnel functions menus page

Figure 5-5 is the personnel functions menus. This shows that the personnel user has only two menus that will allow him or her to perform some operations on his or her account. For example, change password and display user account information.


Admin Display Users Accounts Page - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://localhost:57657/AdminDisplayUsersAccounts.aspx
Google
Most Visited Getting Started Latest Headlines Beginning ASP.NET 3....
Admin Display Users Accounts Page
CSS menu problem in IE
Remember
Never for This Site
Not Now
UPCODE Ltd.
Sign Out
Home
Admin
Customers/Product Users
Products
Errors
Solutions
Search
Done

Do you want Firefox to remember the password for "admin" on http://localhost:57657?

*You are logged in as [admin] ** Role: [Administrator]*

Admin View Users Accounts

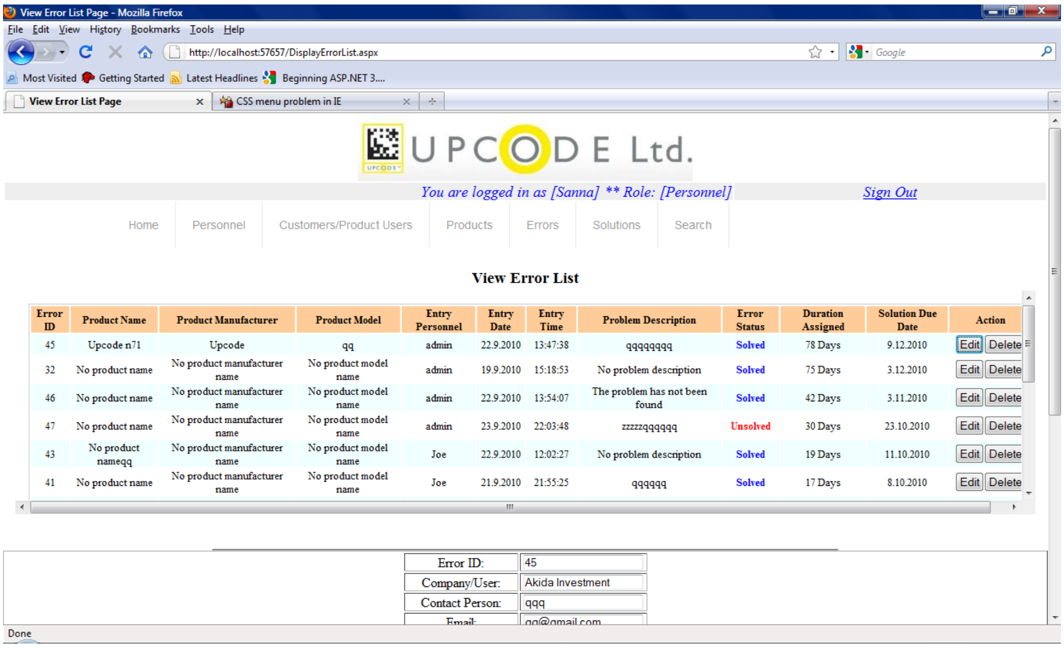
User ID	Last Name	First Name	Email	Phone	User Name	Password	Role	Action
107	Afonne	Sanna	sanna.harjunpaa@gmail.com	049948484	Sanna	●●●●●	Personnel	Edit Delete
101	Joseph	James	josefafonne@yahoo.co.uk	044 917 2772	James	●●●●●●●●●●	Personnel	Edit Delete
111	Joe	Afonne	josefafonne@yahoo.co.uk	049948484	Joe	●●●	Administrator	Edit Delete
110	Josephson	Afonnen	josefafonne@yahoo.co.uk	049948484	admin	●●●●●	Administrator	Edit Delete
112	John	Paulson	sanna.harjunpaa@gmail.com	049948484	John	●●●●●●●●●●	Personnel	Edit Delete
113	Peter	Edeh	peteredeh@yahoo.com	+358 44-902-3049	Edeh	●●●●	Personnel	Edit Delete
143	Afonne	Julia	julia.afonne@yahoo.com	+358 44 636 8282	Julia	●●●●●●●●●●	Personnel	Edit Delete
154	James	Anumama	e0501196@puv.fi	+358 44 208 2989	Baba	●●●●	Personnel	Edit Delete

Figure 5-6 Administrator view of user accounts page

Figure 5-6 shows the administrator's view users' accounts. This displays all the users with their roles. Clicking on the 'edit' button will display each user's data and allow the administrator to update them. Clicking on 'delete' button will delete the selected user record.

5.1.3 View Error List

The view error list allows for viewing of error list. It also allows for editing/deleting of registered errors.


View Error List Page - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://localhost:57657/DisplayErrorList.aspx
Google
Most Visited Getting Started Latest Headlines Beginning ASP.NET 3....
View Error List Page
CSS menu problem in IE
UPCODE Ltd.
You are logged in as [Sanna] ** Role: [Personnel]
Sign Out
Home
Personnel
Customers/Product Users
Products
Errors
Solutions
Search
Done

View Error List

Error ID	Product Name	Product Manufacturer	Product Model	Entry Personnel	Entry Date	Entry Time	Problem Description	Error Status	Duration Assigned	Solution Due Date	Action
45	Upcode n71	Upcode	qq	admin	22.9.2010	13:47:38	qqqqqqqq	Solved	78 Days	9.12.2010	Edit Delete
32	No product name	No product manufacturer name	No product model name	admin	19.9.2010	15:18:53	No problem description	Solved	75 Days	3.12.2010	Edit Delete
46	No product name	No product manufacturer name	No product model name	admin	22.9.2010	13:54:07	The problem has not been found	Solved	42 Days	3.11.2010	Edit Delete
47	No product name	No product manufacturer name	No product model name	admin	23.9.2010	22:03:48	zzzzzqqqqqq	Unsolved	30 Days	23.10.2010	Edit Delete
43	No product nameqq	No product manufacturer name	No product model name	Joe	22.9.2010	12:02:27	No problem description	Solved	19 Days	11.10.2010	Edit Delete
41	No product name	No product manufacturer name	No product model name	Joe	21.9.2010	21:55:25	qqqqqq	Solved	17 Days	8.10.2010	Edit Delete

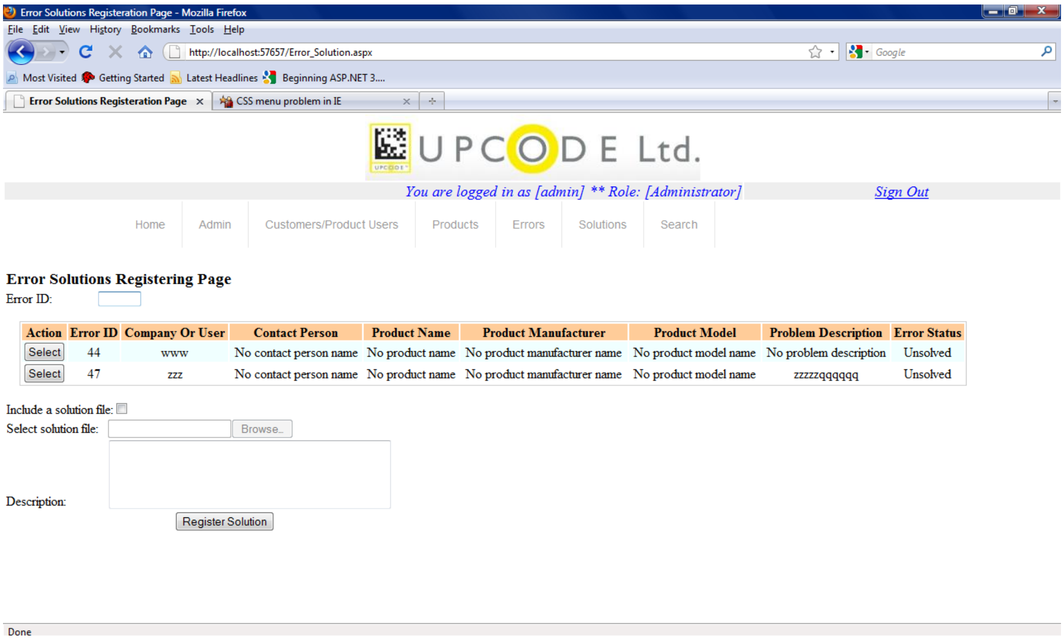
Error ID:	45
Company/User:	Akida Investment
Contact Person:	qqq
Email:	qq@gmail.com

Figure 5-7 View error list page

Figure 5-7 shows the display of all the registered errors in the system. Clicking on 'edit' button allows errors to be edited and updated while clicking on 'delete' button deletes the selected error record from the system.

5.1.4 Error Solution Register

This page allows for entering error solution into the system. From ‘Solutions’ menu, click on ‘register solution’. This will display the register error solution page.



*You are logged in as [admin] ** Role: [Administrator]*

Error Solutions Registering Page

Error ID:

Action	Error ID	Company Or User	Contact Person	Product Name	Product Manufacturer	Product Model	Problem Description	Error Status
Select	44	www	No contact person name	No product name	No product manufacturer name	No product model name	No problem description	Unsolved
Select	47	zzz	No contact person name	No product name	No product manufacturer name	No product model name	zzzzzqqqqqq	Unsolved

Include a solution file: □

Select solution file:

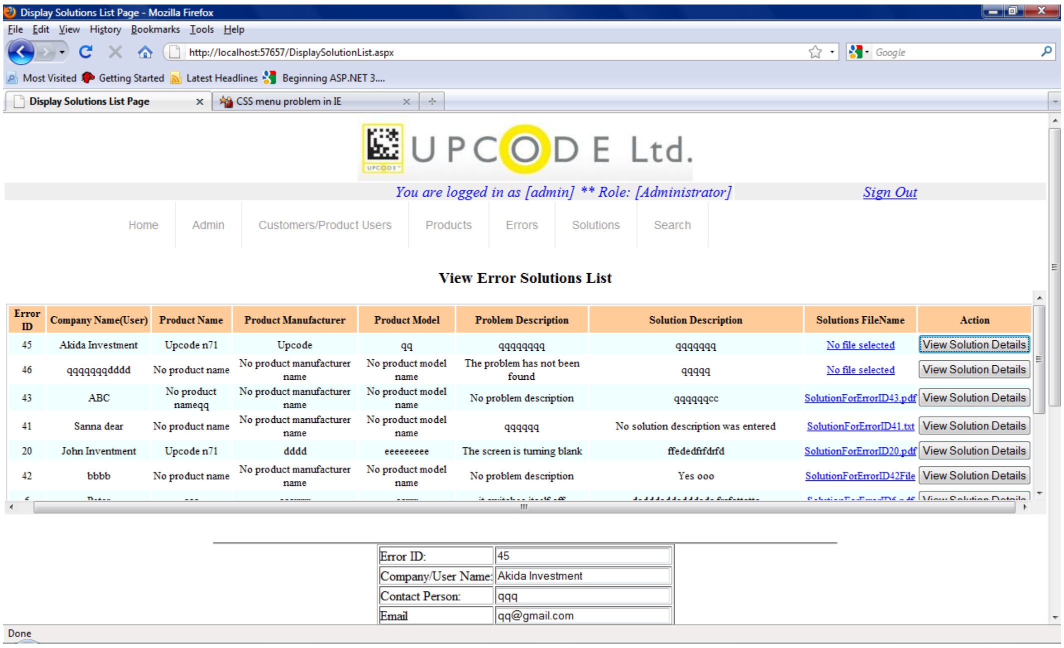
Description:

Figure 5-8 Error solution register page

Figure 5-8 is the show of the page to register error solutions. The error is selected from the action. The ‘include solution file’ is checked if a solution file is to be uploaded, this enables the select file control, but left unchecked if no file solution is included in the solution.

5.1.5 View Error Solutions

From the ‘solutions’ menu, select ‘view error solutions’. This will display the following figure.


Display Solutions List Page - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://localhost:57657/DisplaySolutionList.aspx
Most Visited Getting Started Latest Headlines Beginning ASP.NET 3....
Display Solutions List Page
CSS menu problem in IE
UPCODE Ltd.
You are logged in as [admin] ** Role: [Administrator]
Sign Out
Home
Admin
Customers/Product Users
Products
Errors
Solutions
Search
Done

View Error Solutions List

Error ID	Company Name(User)	Product Name	Product Manufacturer	Product Model	Problem Description	Solution Description	Solutions FileName	Action
45	Akida Investment	Upcode n71	Upcode	qq	qqqqqqqq	qqqqqqq	No file selected	View Solution Details
46	qqqqqqqdddd	No product name	No product manufacturer name	No product model name	The problem has not been found	qqqqq	No file selected	View Solution Details
43	ABC	No product nameqq	No product manufacturer name	No product model name	No problem description	qqqqqqcc	SolutionForErrorID43.pdf	View Solution Details
41	Sanna dear	No product name	No product manufacturer name	No product model name	qqqqqq	No solution description was entered	SolutionForErrorID41.txt	View Solution Details
20	John Inventment	Upcode n71	dddd	eeeeeeeee	The screen is turning blank	ffededfrfdrfd	SolutionForErrorID20.pdf	View Solution Details
42	bbbb	No product name	No product manufacturer name	No product model name	No problem description	Yes ooo	SolutionForErrorID42File	View Solution Details

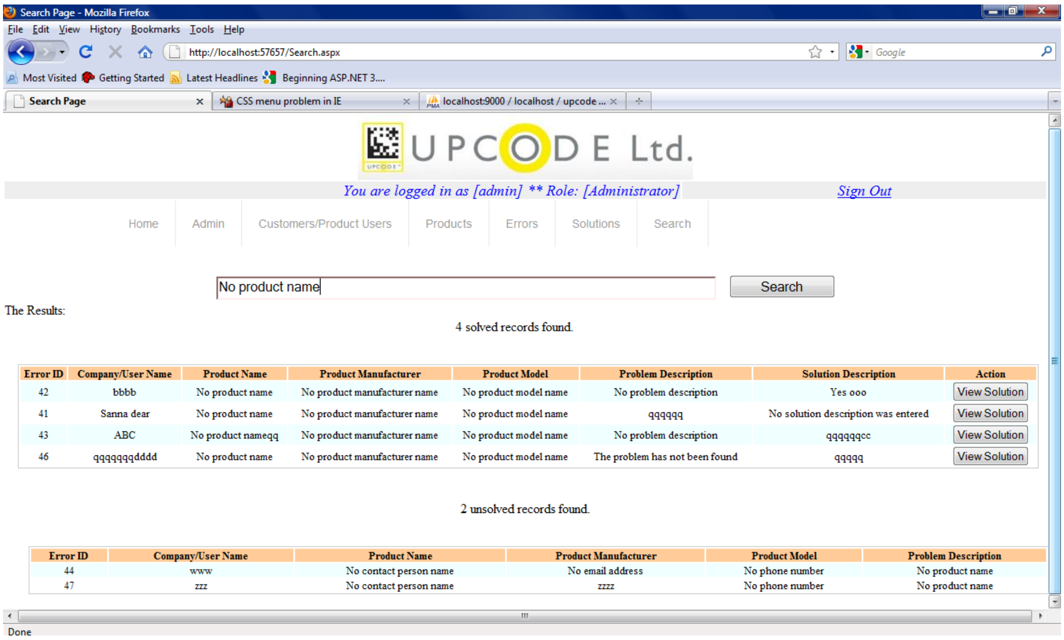
Error ID:	45
Company/User Name:	Akida Investment
Contact Person:	qqq
Email	qq@gmail.com

Figure 5-9 View error solutions list page

The Figure 5-9 shows the display of the list of error solutions. Clicking on the 'Solutions FileNname' will display the uploaded solutions file and clicking on the 'View Solution Details' will display the entire error solution record. The record shows the entire record including the data not showing on the 'View Error Solutions List' display.

5.1.6 Searching data

From the home page, clicking on 'search' menu displays search page. The following figure shows the search page.


Search Page - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://localhost:57657/Search.aspx
Google
Most Visited Getting Started Latest Headlines Beginning ASP.NET 3....
Search Page
CSS menu problem in IE
localhost:9000 / localhost / upcode ...
UPCODE Ltd.
Sign Out
Home
Admin
Customers/Product Users
Products
Errors
Solutions
Search
Search
Done

You are logged in as [admin] ** Role: [Administrator]

No product name

The Results:

4 solved records found.

Error ID	Company/User Name	Product Name	Product Manufacturer	Product Model	Problem Description	Solution Description	Action
42	bbbb	No product name	No product manufacturer name	No product model name	No problem description	Yes ooo	View Solution
41	Sanna dear	No product name	No product manufacturer name	No product model name	qqqqqq	No solution description was entered	View Solution
43	ABC	No product nameqq	No product manufacturer name	No product model name	No problem description	qqqqqqcc	View Solution
46	qqqqqqqdddd	No product name	No product manufacturer name	No product model name	The problem has not been found	qqqqq	View Solution

2 unsolved records found.

Error ID	Company/User Name	Product Name	Product Manufacturer	Product Model	Problem Description
44	www	No contact person name	No email address	No phone number	No product name
47	zzz	No contact person name	zzzz	No phone number	No product name

Figure 5-10 Search data page

Figure 5-10 shows the search page. Typing the error word(s) to search for and clicking the 'Search' button will display the result of the search for the solved solutions and unsolved solutions. Clicking the 'View Solution' will display the solution file containing the solution of steps to solve a particular error.

6 CONCLUSION

The purpose of this project work was to develop a Web Based Error Reporting Application using ASP.NET technology. This application will especially aid the helpdesk staff of UpCode Ltd. to manage reports of errors from users of its many products. Requirements for the application were gathered from the company by way of discussion and interviews with those involved in the use of the application. This project implemented all the functions mentioned in the specification and design of the application and offers user-friendliness. By using the application error reporting management's efficiency will be improved.

I have especially benefited from this project work. For one thing, it has helped me get familiar with how ASP.NET technology collaborates with other technologies such as HTML, CSS, MYSQL, XML and JavaScript. Because of the fact that the company asked me to use simple technology that they could easily maintain, put a lot of pressure on me to figure out other ways that ideas can be implemented with simple HTML and C# code-behind programming.

ASP.NET platform has developed APIs and technologies that make web application development easier, but these technologies have their associated problems and shortcomings. For example, some ASP.NET technologies generate a lot codes that may make debugging pretty difficult. Additionally, the ASP.NET 2.0 limited ability to make asynchronous calls to the server leaves the developer to think about ways to handle calls to the server.

I also observed that different browsers have their different ways of rendering the same pages. The application was tested using Firefox, Internet Explorer and Safari. To a large extent, they did their best in rendering pages similar except Internet Explorer that has issues with submenus disappearing on hovering on the submenus after the menus have been selected. This usually is the case when the menu is styled using cascading style sheet.

REFERENCES

1. Matthew MacDonald and Mario Szpuszta. 2005. Pro ASP.NET 2.0 in C# 2005. Apress.

2. Simeon Bennett [etal]. 2001. Schaum’s Outlines UML. McGRAW-HILL.

3. Jeff Prosise. 2002. Programming Microsoft .NET. Microsoft Press.

4. Don Gosselin. 2000. JavaScript: Comprehensive. Course Technology.

5. http://www.upcode.fi/.