CHAPTER 4

SYSTEM DESIGN AND DEVELOPMENT

4.1 INTRODUCTION

This chapter presented the design and development of the HMS system. The uniform modeling language UML was used to illustrate different modules and process involved in the system as well as the data flow and the conception of the database using Microsoft SQL server through the user interface of Visual web developer 2010 express edition.

4.2 USERS REQUIREMENTS

Basically, there are five types of user requirements for researcher's system such as below:

Administrator: The system allows admin to register doctors, staff nurses and patients, as well as editing information about doctor, staff, patients and nurses.

Patient: The system allow patients to view and make a new appointment, as well as to view all his or her medical records.

Doctor: The system allow doctors to view his or her appointments and view patients' medical records.

Nurse: The system allow nurses to view patient's medical record.

Staff: The system allow staffs to view doctors, nurses and patients information.

4.3 SYSTEM DESIGN

There were two types of diagrams explained in this section that are use case diagrams and flowchart diagrams.

4.3.1 Use Case Diagrams

The use case diagrams are charts used to defining system requirements, as well as tool to collaborate with clients by explaining the behavior of the future system, and how it suits the client requirements. It is also useful for generating test cases to track bugs within the system. Figure 4.1, 4.2, 4.3, 4.4 and 4.5 shows use case diagram for users in HMS prototype.

Figure 4.1, shows the Admin use case diagram. In order to register staff or patient or view the audit, the Admin need to login to the system and after that the Admin should logout to secure the system.

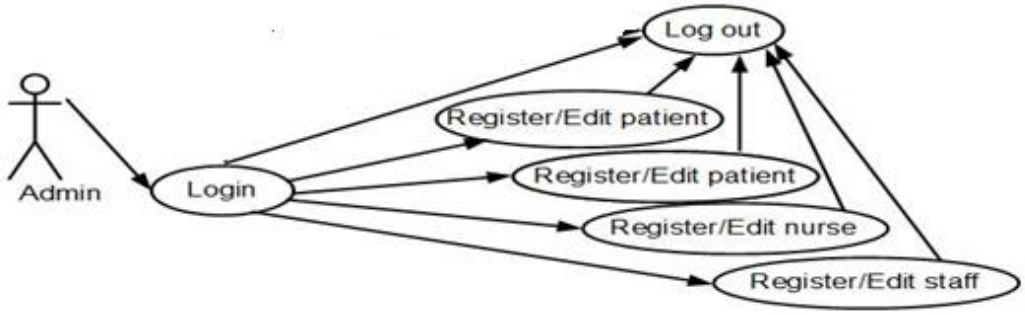

Figure 4.1: Admin's Use Case Diagram

Figure 4.2 shows the Patient's Use Case Diagram. In order to view medical record or make/view appointment, the Patient needs to login to the system and after that the Patient should logout to secure the system.

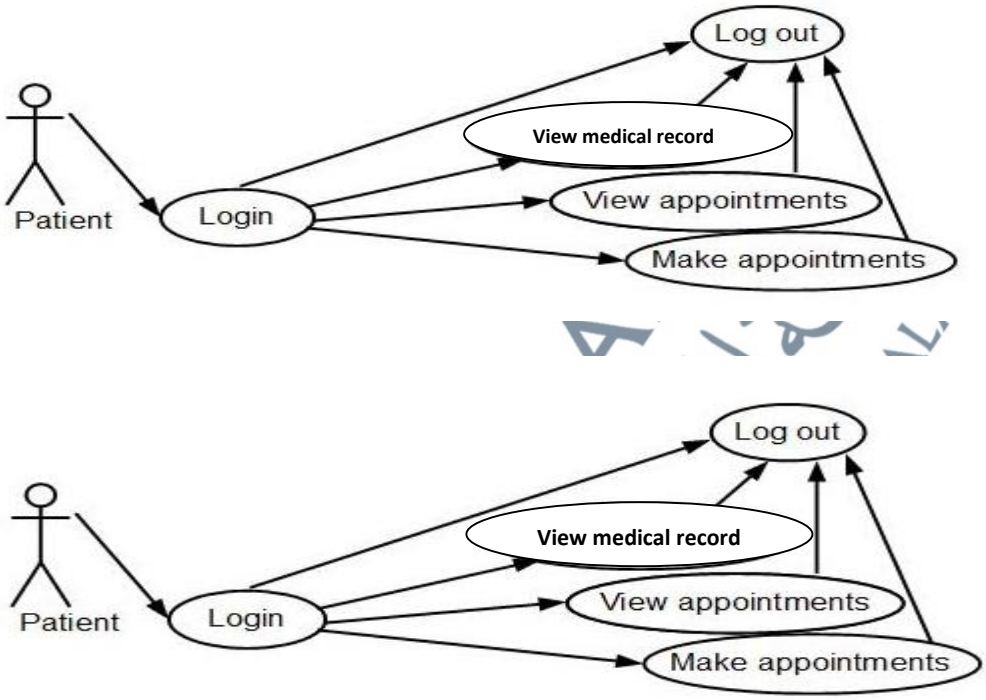

Figure 4.2: Patient's Use Case Diagram

Figure 4.3 shows the Doctor's Use Case Diagram. In order to view the medical record or view appointments or describe treatment, the Doctor needs to login to the system and after that the Doctor should logout to secure the system.

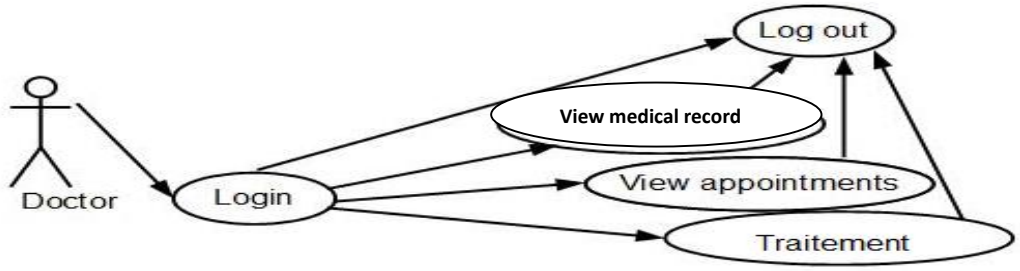

Figure 4.3: Doctor's Use Case Diagram

Figure 4.4 shows Nurse's Use Case Diagram. In order to view the medical record, the Nurse needs to login to the system. After that the Nurse should logout to secure the system.

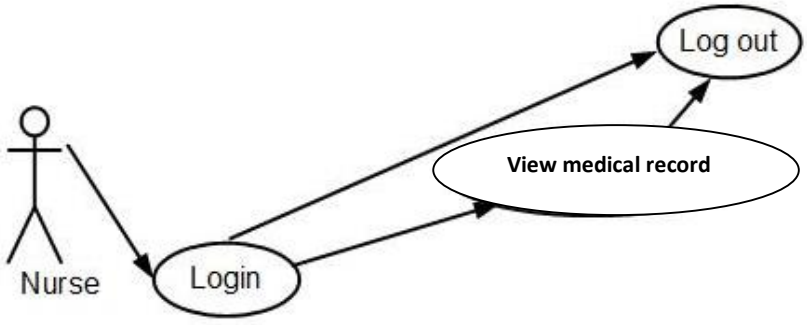

Figure 4.4: Nurse's Use Case Diagram

Figure 4.5 shows Staff's Use Case Diagram. In order to view medical record or view doctor/nurse info, the Staff needs to login to the system and after that the Staff should logout to secure the system.

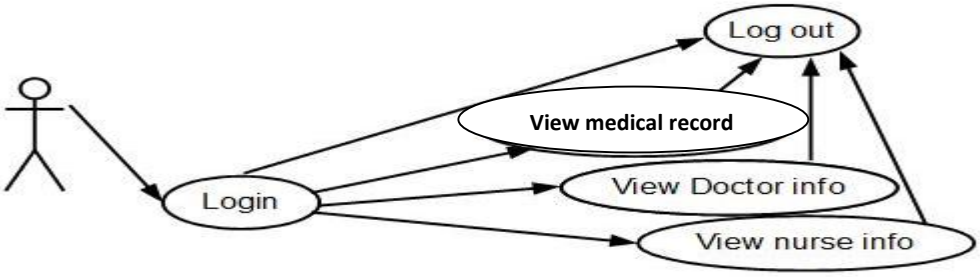

Figure 4.5: Staff's Use Case Diagram

4.3.2 Flowchart Diagrams

Flowchart diagram also known as algorithms are representations of different algorithms and process within the system. Figure 4.6, 4.7, 4.8, 4.9, 4.10 and 4.11 shows flowchart diagrams for users in HMS prototype.

Figure 4.6 shows the Login Flowchart Diagram. Before using the system, any user needs to login to the system. After giving his credentials the user will be logged in.

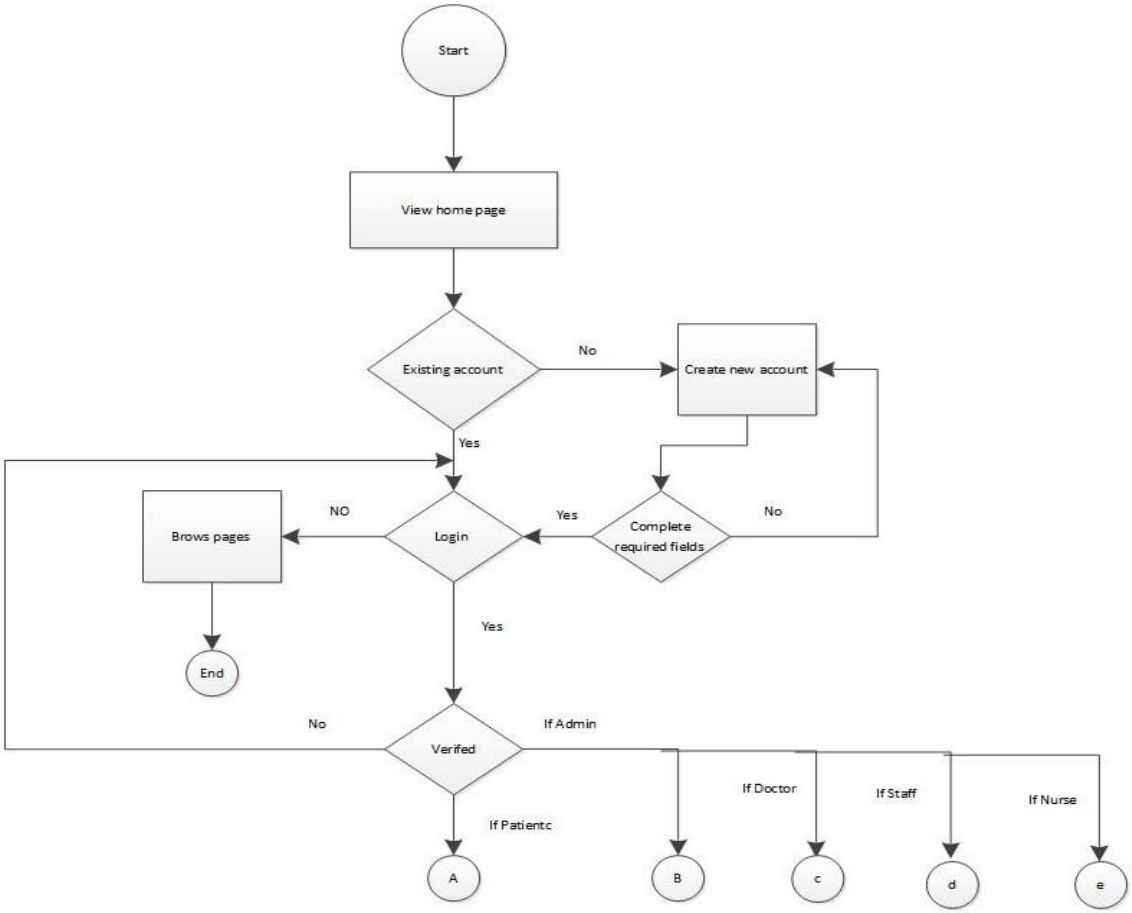

Figure 4.6: Login Flowchart Diagram

Figure 4.7 shows the Admin' Flowchart Diagram. After the Admin has logged in to the system, he or she can register/edit patients, doctors or staff, and then logout.

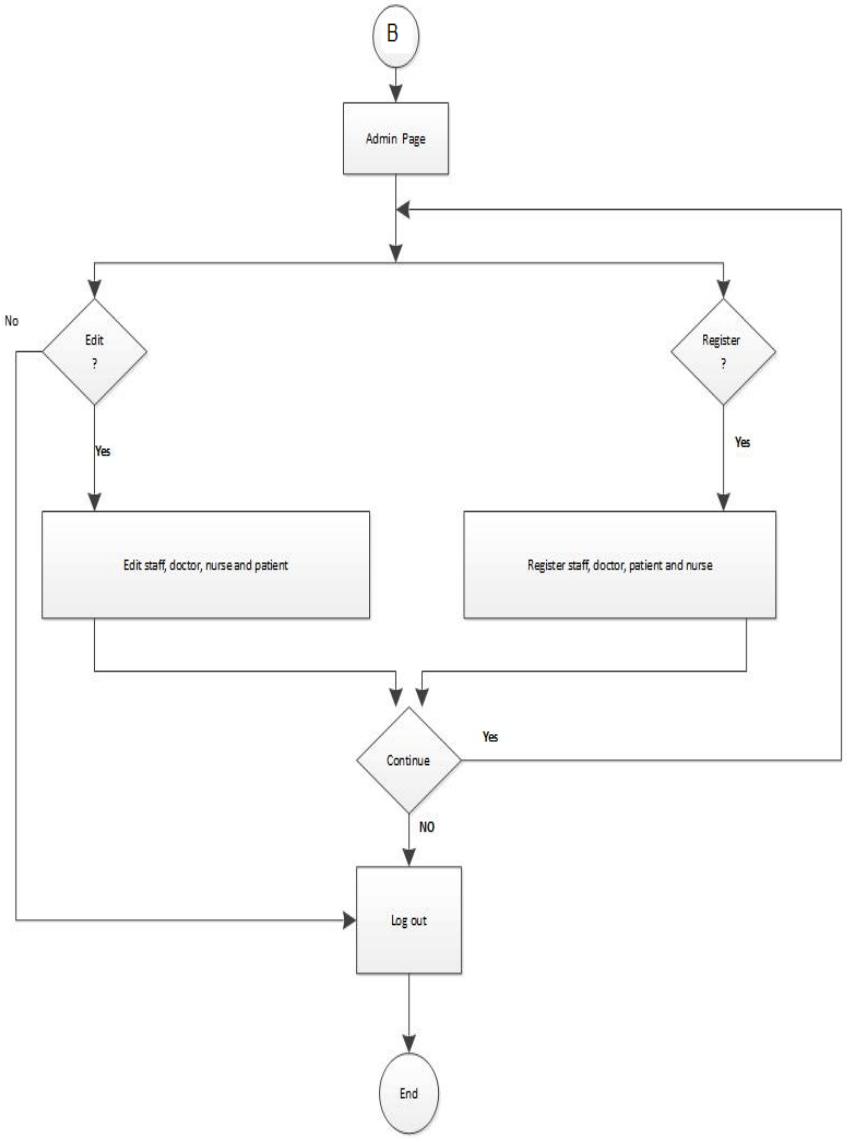

Figure 4.7: Admin's Flowchart Diagram

Figure 4.8 shows the Patient's Flowchart Diagram. After the Patient has logged in to the system he or she can make/view appointment or view medical report, and then logout.

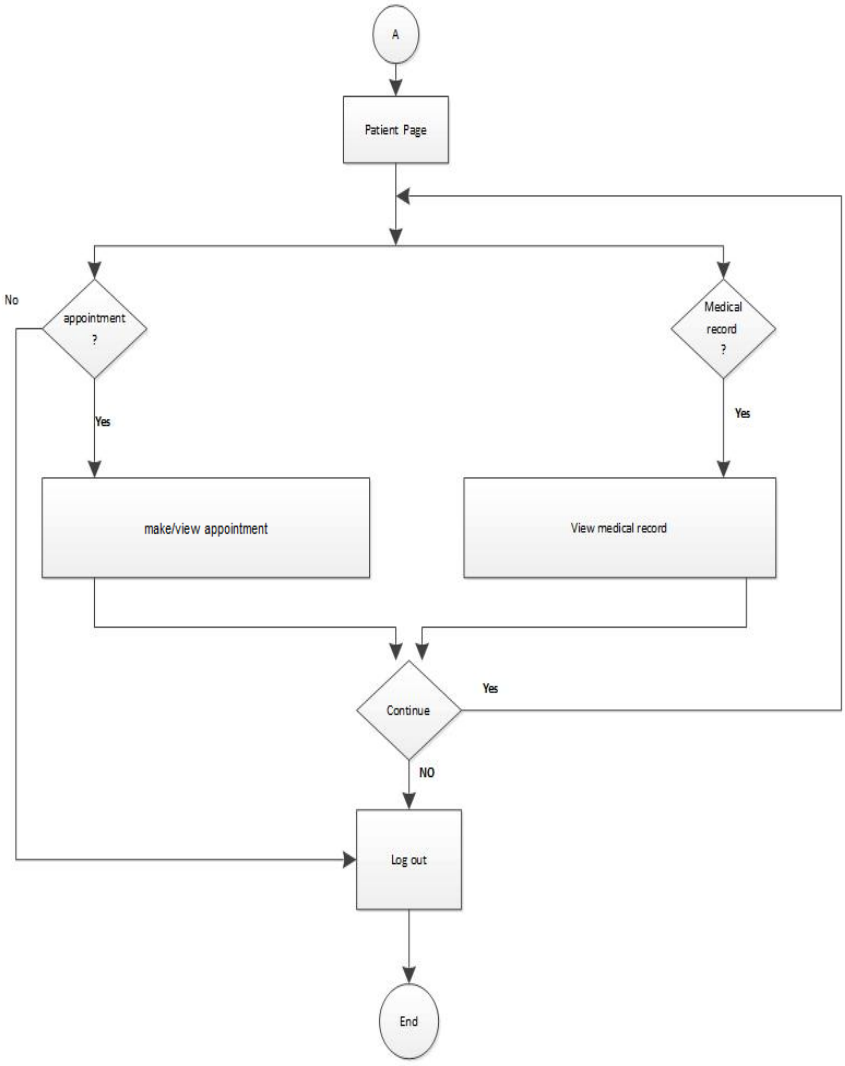

Figure 4.8: Patient's Flowchart Diagram

Figure 4.9 shows the Doctor's Flowchart Diagram. After the Doctor has logged in to the system, he or she can view his appointment or make treatment, and then logout.

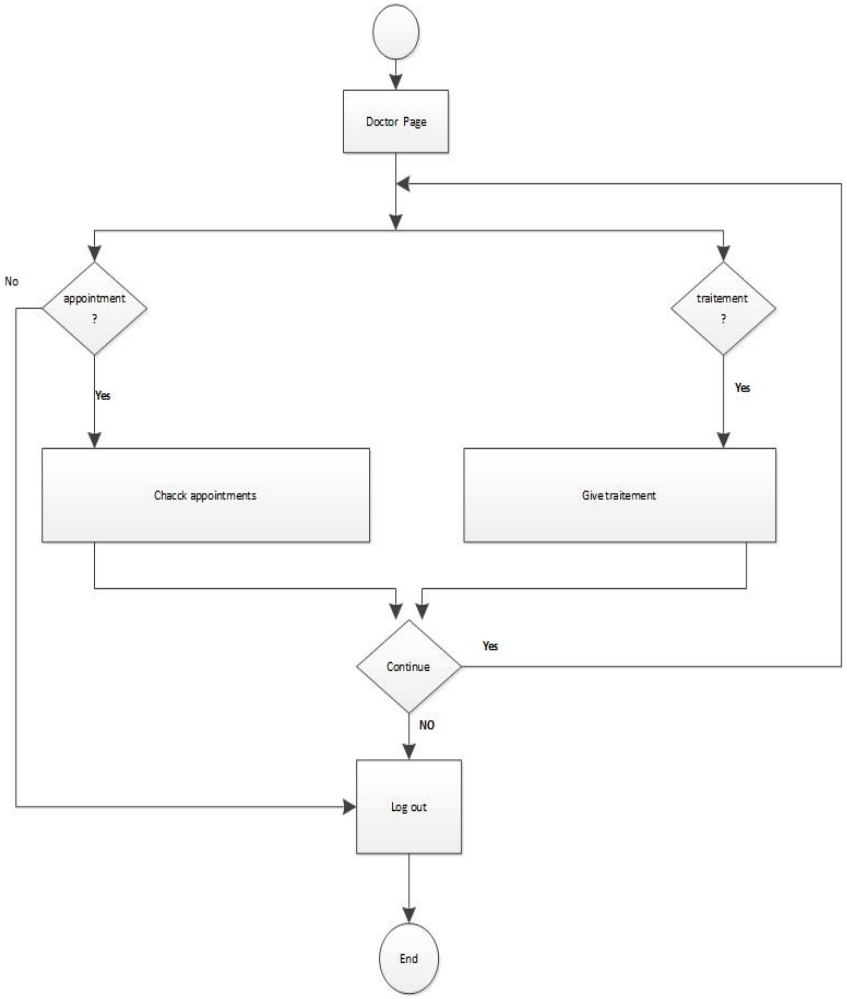

Figure 4.9: Doctor's Flowchart Diagram

Figure 4.10 shows the Nurse's Flowchart Diagram. After the Nurse has logged in to the system, he or she can view medical report, and then logout.

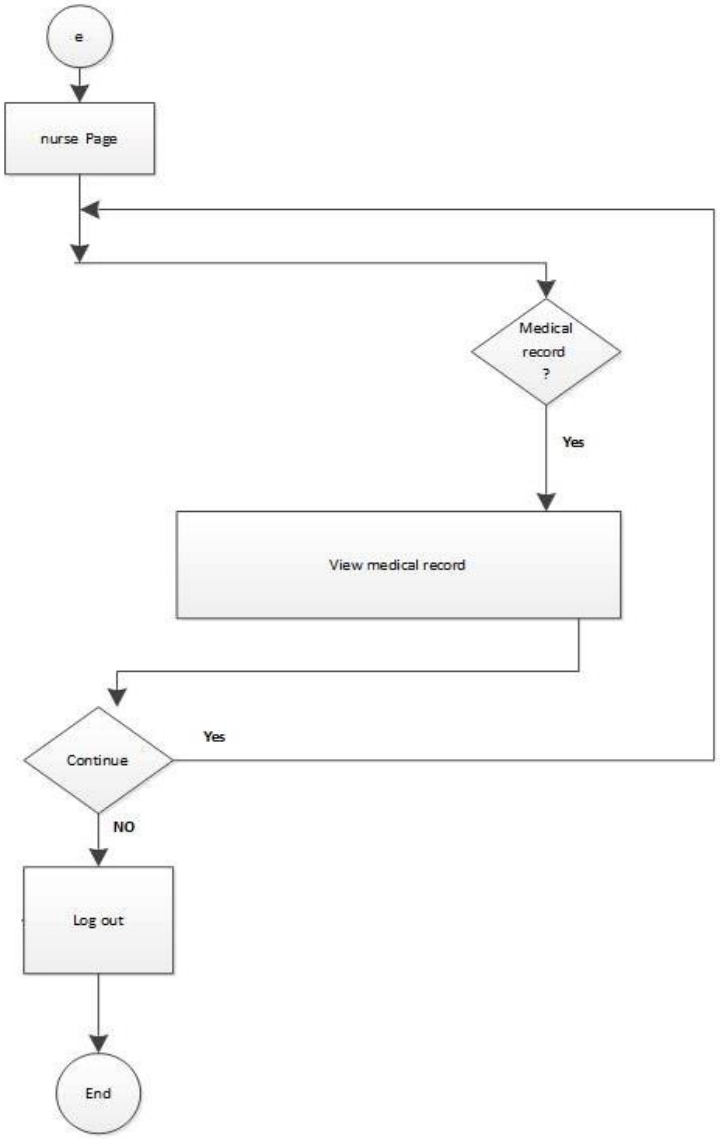

Figure 4.10: Nurse's Flowchart Diagram

Figure 4.11 shows the Staff's Flowchart Diagram. After the Staff has logged in to the system, he or she can find staff info or view medical record, and then logout.

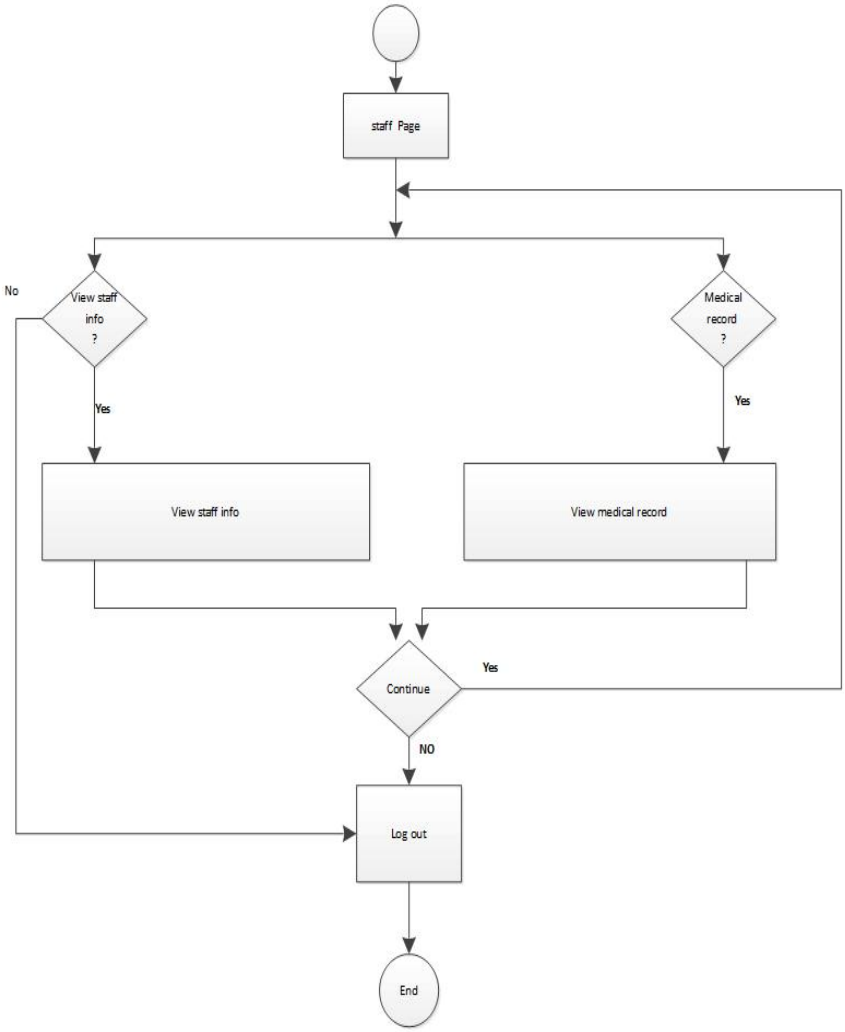

Figure 4.11: Staff's Flowchart Diagram

4.4 DATABASE CREATION

The databases were created using Microsoft SQL server 2008 R2 express edition, through the GUI of Microsoft visual web developer 2010 express edition, where the steps are as follows:

- In Visual Web Developer, open the Solution Explorer, click the right button on the App_Data folder, and then click Add New Item.
- Or if one's application does not have App_Data folder, click the right mouse button on the root folder of the Web application, click Add ASP.NET Folder and click App_Data.
- Click SQL database, type a name for the .mdf database file and then click Add.
- Two files are created: DataBaseName.mdf and DataBaseName_log.ldf. Visual Web Developer automatically moves the focus to the Data Connections section of the Server Explorer window and select the newly created data base.

4.4.1 Entity Relation (ER) Diagrams

The first database to be created is ASPNETDB.MDF and the following is its entity relation diagram as shown in Figure 4.12.

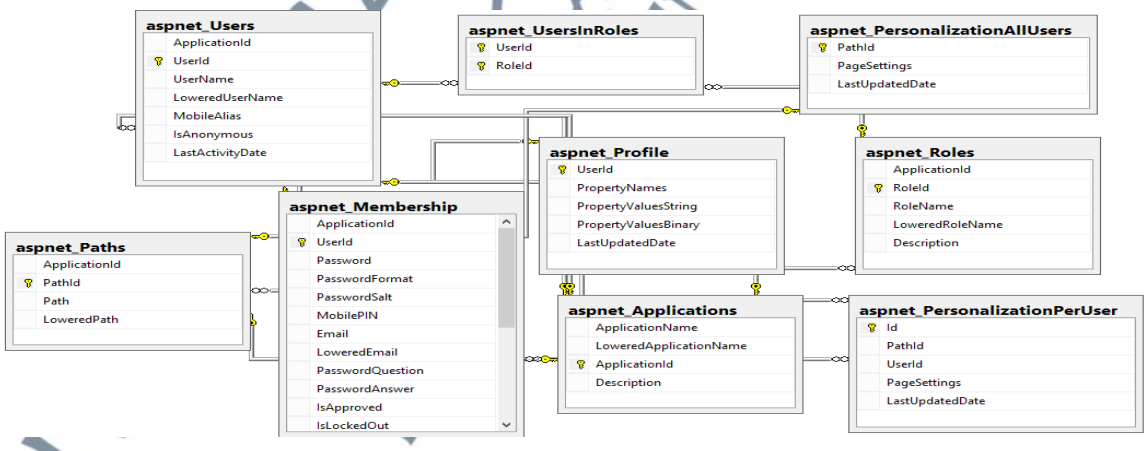

Figure 4.12: First Database Created

The second database to be created is data base.MDF and the following is its entity relation diagram as shown in Figure 4.13.

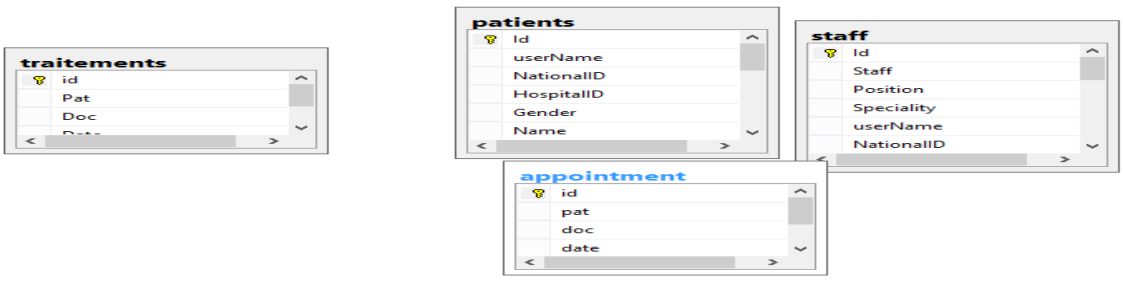

Figure 4.13: Second Database Created

4.4.2 Data Flow Diagram (DFD)

A data flow diagram represents the system database and tables, and the flow of the data during different processes and modules of the system. The charts are related to the tables shown in the entity relation diagrams. Figure 4.14 until 4.26 shows data flow diagrams for HMS prototype.

Figure 4.14 shows Data Flow Diagram for Patient. The Admin can insert/update patient, and the patient can view his or her medical report.

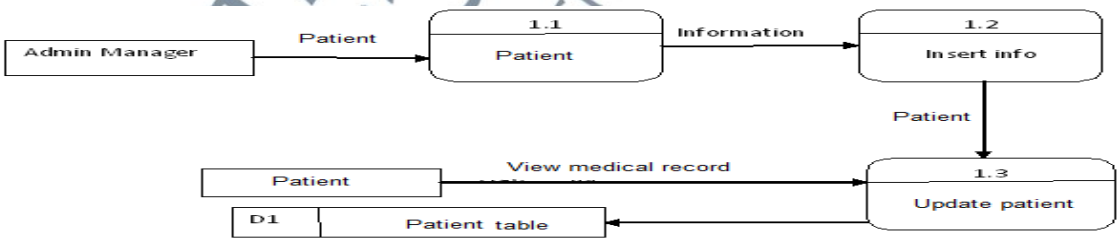

Figure 4.14: Data During Different Processes and Modules of the System

Figure 4.15 shows Data Flow Diagram for Staff. The Admin can insert/update staff, and the staff can view his or her profile.

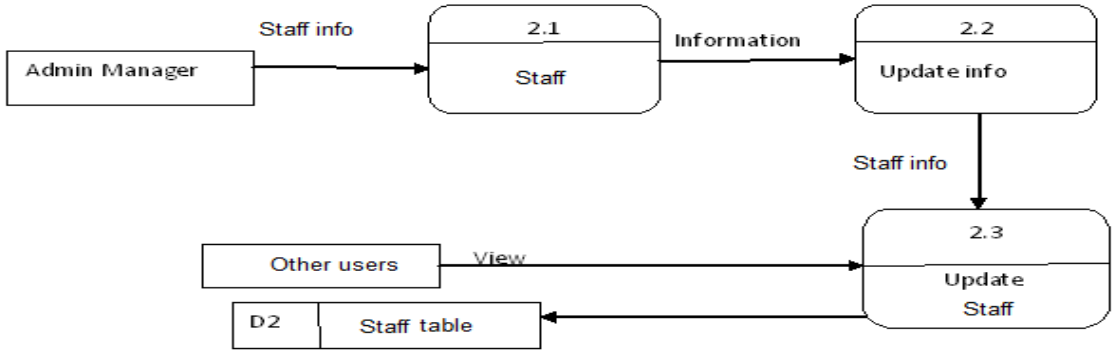

Figure 4.15: Data Flow Diagram for Staff

Figure 4.16 shows Data Flow Diagram for Appointment. The Admin can insert/update appointment, and the patient/doctor can view the appointment.

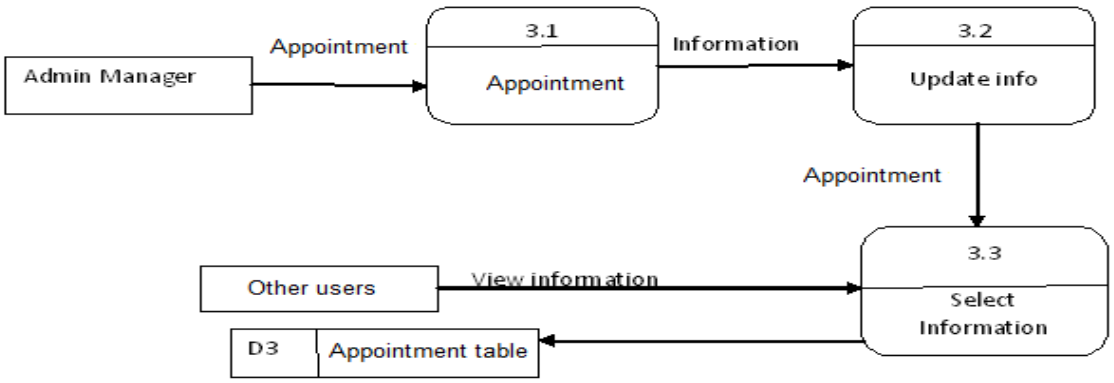

Figure 4.16: Data Flow Diagram for Appointment

Figure 4.17 shows Data Flow Diagram for Treatment. The doctor can insert/update treatment, and the patient can view his or her medical report.

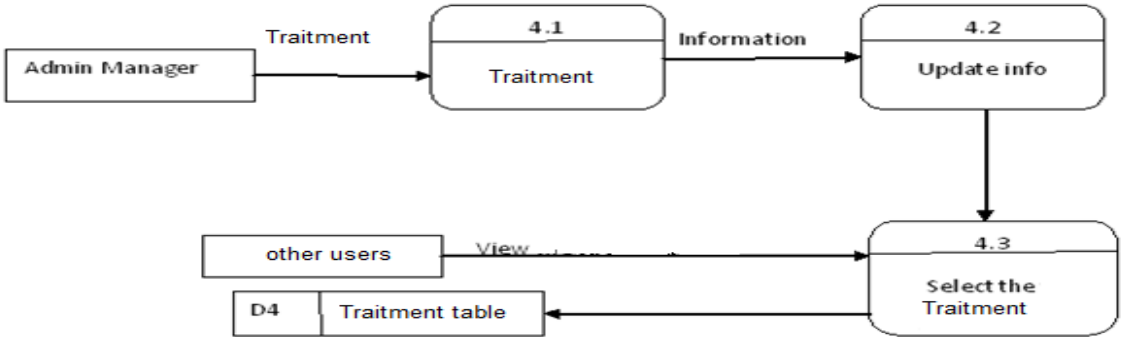

Figure 4.17: Data Flow Diagram for Treatment

Figure 4.18 shows Data Flow Diagram for Profile. The user can insert/update profile.

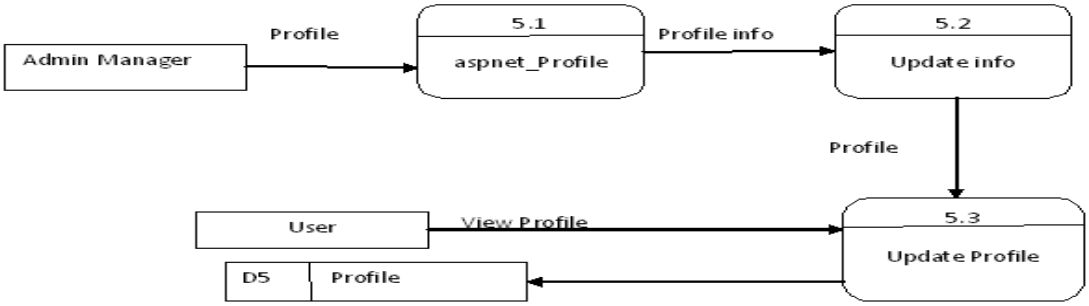

Figure 4.18: Data Flow Diagram for Profile

Figure 4.19 shows Data Flow Diagram for User Personalization. The user can personalise his profile through ASP.Net website configuration.

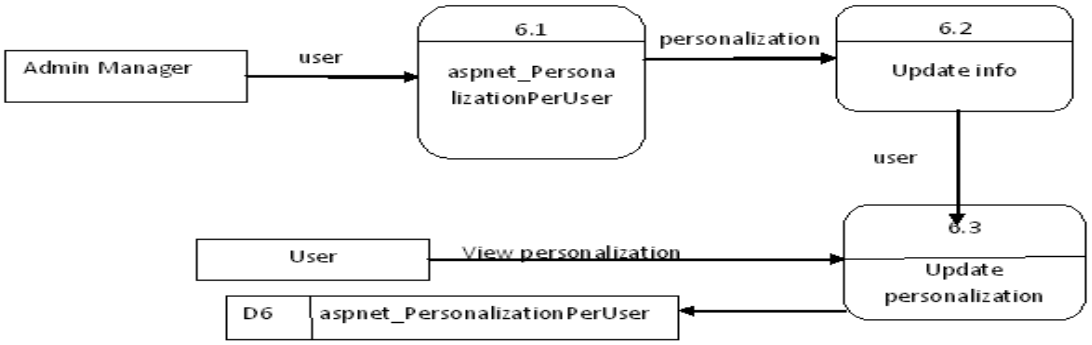

Figure 4.19: Data Flow Diagram for User Personalization

Figure 4.20 shows Data Flow Diagram for User in Role. The Admin can create roles through ASP.Net website configuration. Then users get roles depending on their privilege.

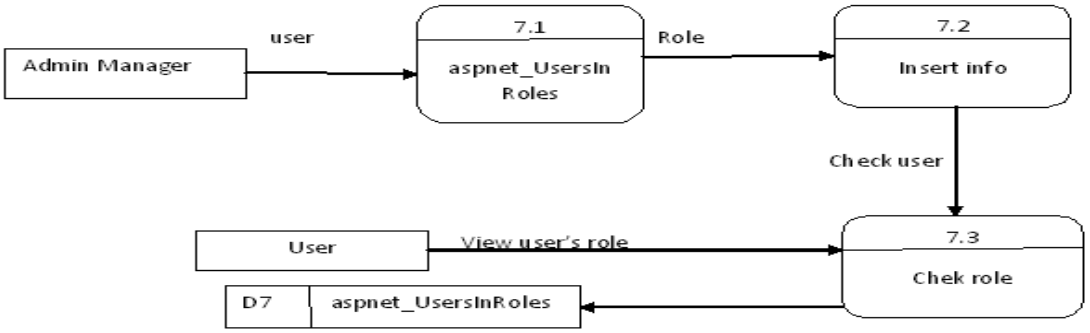

Figure 4.20: Data Flow Diagram for User in Role

Figure 4.21 shows Data Flow Diagram for Personalize All Users. The Admin can personalise the profile through ASP.Net website configuration.

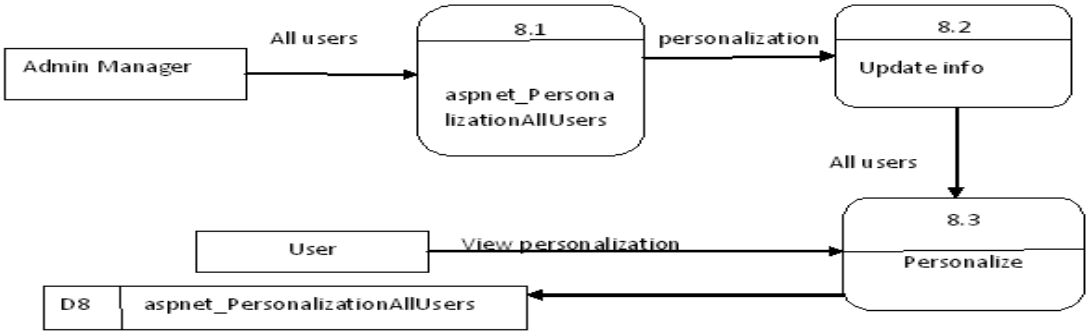

Figure 4.21: Data Flow Diagram for Personalize All Users

Figure 4.22 shows Data Flow Diagram for Roles. The Admin can create/update/delete roles through ASP.Net website configuration. The roles will be attributed to users based on their privilege.

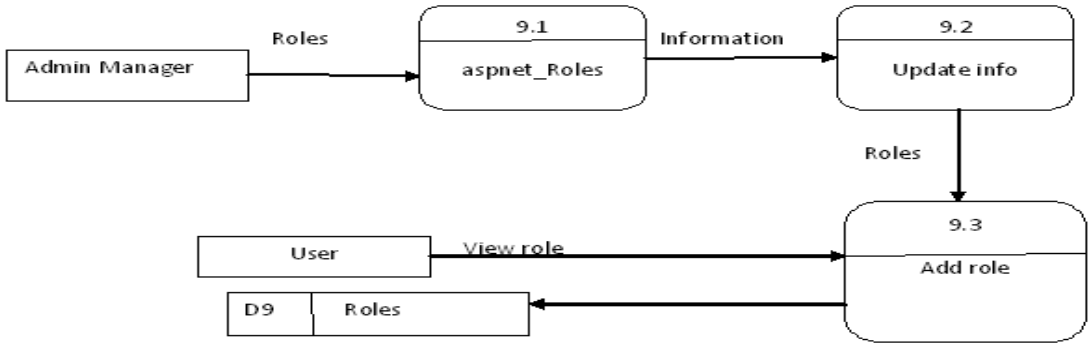

Figure 4.22: Data Flow Diagram for Roles

Figure 4.23 shows Data Flow Diagram for Aspnet_Path. Each time the user visits a page the asp.net save the page name to personalize user experience.

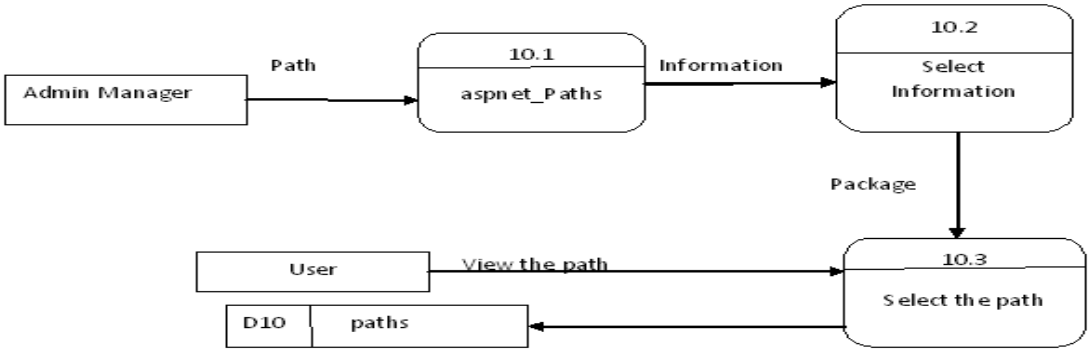

Figure 4.23: Data Flow Diagram for Path

Figure 4.24 shows Data Flow Diagram for Membership. When the admin registers a new user account, the information is saved in Membership table.

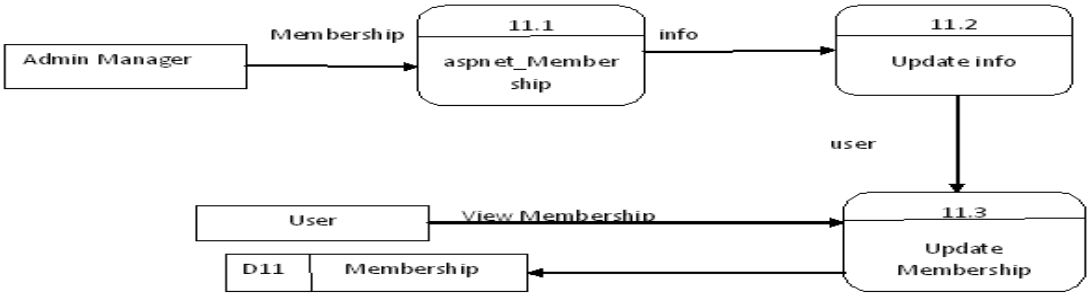

Figure 4.24: Data Flow Diagram for Membership

Figure 4.25 shows Data Flow Diagram for Users. When the admin registers a new user account, the information is saved in Membership table as well as in the aspnet_users table.

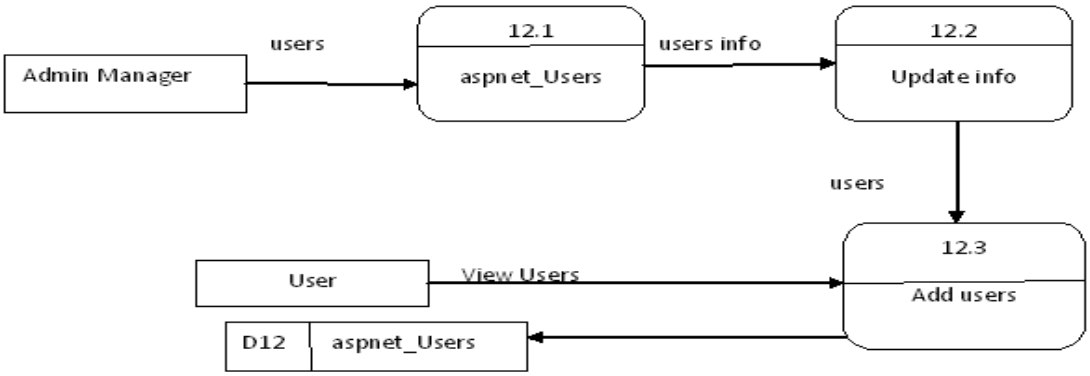

Figure 4.25: Data Flow Diagram for Users

Figure 4.26 shows Data Flow Diagram for Application. When the website is created the name of the website as well as other information are saved in the aspnet_application.

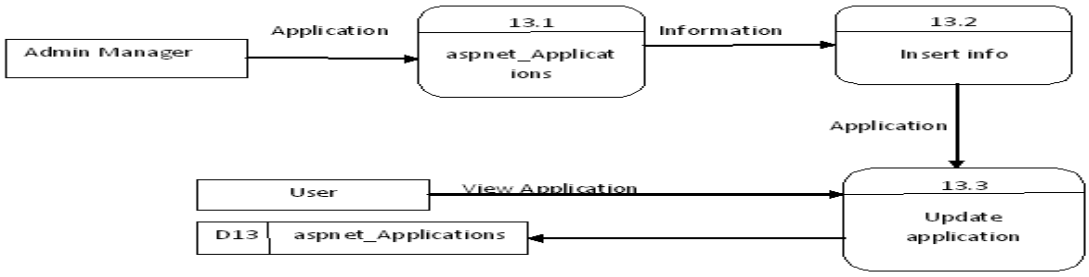

Figure 4.26: Data Flow Diagram for Application

4.5 SECURITY IMPLEMENTATION IN THE HMS PROTOTYPE

4.5.1 Triple DES Implementation

In order to encrypt and decrypt the medical record in the proposed system a class called CryptorEngine is developed. This class contains two procedures: Encrypt and Decrypt which uses the Microsoft ".net" object: "System. Security. Cryptography"

The encryption key was stored in the web configuration file as an application settings under the name " SecurityKey " as shown in Figure 4.27.

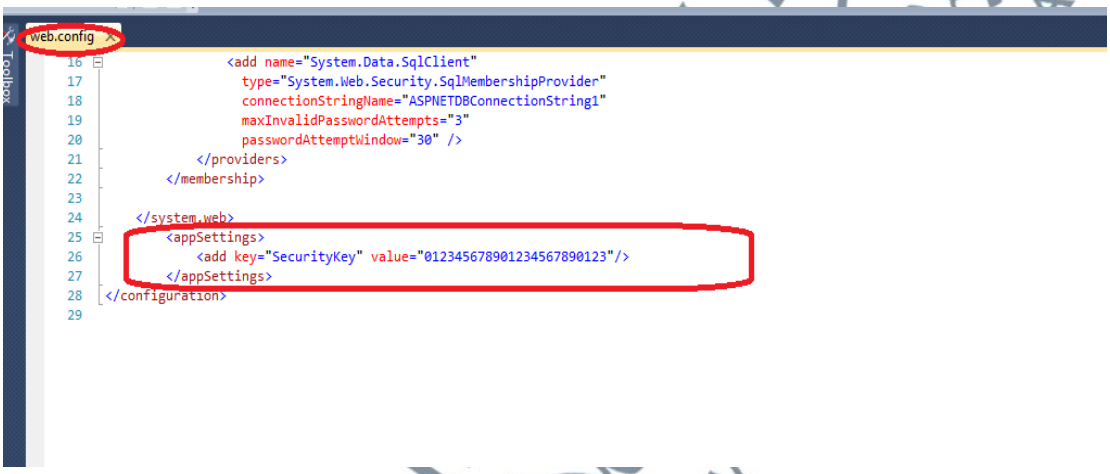

Figure 4.27: Encryption Key

The result of the encryption is shown in the "Patients" table (Figure 4.28).

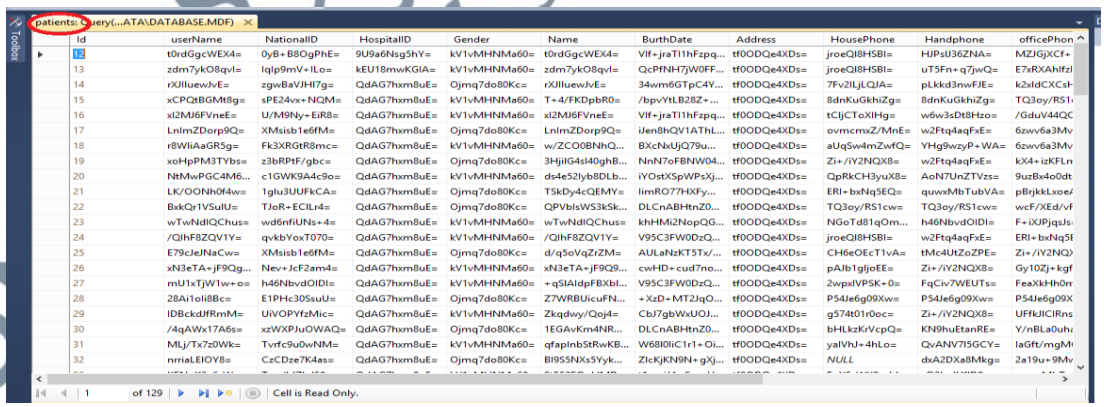

Figure 4.28: Encryption Results

First of all, the “Register new patients” form is displayed. Then, after filling the form “Create User” button need to be clicked as shown in Figure 4.29.

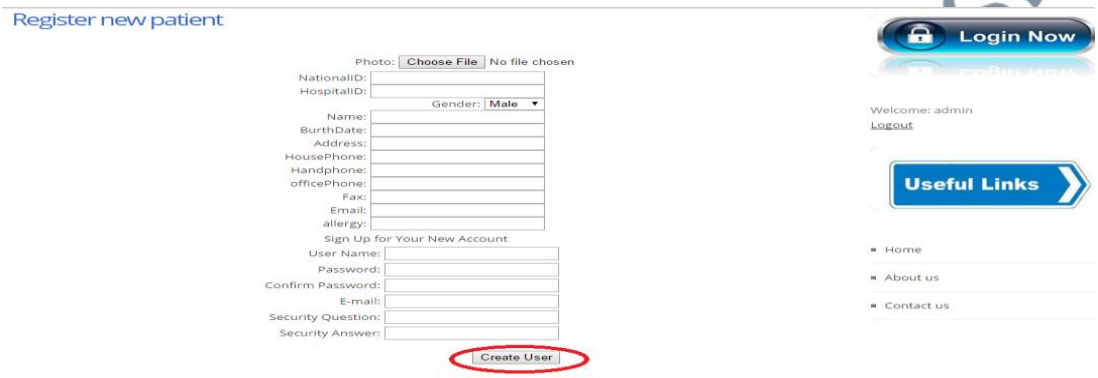

Figure 4.29: “Register New Patients” Form

Then the code in the “Register.aspx.vb” is run, as shown in Figure 4.30, before inserting the record in the database. The procedure is called EncryptParameter. This procedure calls the Encrypt method from the class: EncryptorEngine, which conducted the encryption of the medical record before inserting it in the database.

```
ptorEngine.vb   Admin/Register.aspx.vb   patients: Query(...ATA\DATABASE.MDF)
                                         (Declarations)
    End Sub

    Protected Sub SqlDataSource1_Inserting(ByVal sender As Object, ByVal e As System.Web.UI.WebControls.SqlDataSourceCommandEventArgs) Handles S
        e.Command.Parameters("@userName").Value = CryptorEngine.Encrypt(CreateUserWizard1.UserName, False)
        e.Command.Parameters("@Photo").Value = getImage()
        EncryptParameter("NationalID", e)
        EncryptParameter("HospitalID", e)
        EncryptParameter("Name", e)
        EncryptParameter("BurthDate", e)
        EncryptParameter("Address", e)
        EncryptParameter("HousePhone", e)
        EncryptParameter("Handphone", e)
        EncryptParameter("officePhone", e)
        EncryptParameter("Fax", e)
        EncryptParameter("Email", e)
        EncryptParameter("allergy", e)
        Dim DropDownList1 As DropDownList = TryCast(FormView1.FindControl("DropDownList1"), DropDownList)
        If DropDownList1 IsNot Nothing Then
            e.Command.Parameters("@Gender").Value = CryptorEngine.Encrypt(DropDownList1.Text, False)
        End If
    End Sub

    Protected Sub EncryptParameter(ByVal p As String, ByVal e As System.Web.UI.WebControls.SqlDataSourceCommandEventArgs)
        Dim txtBox As TextBox = TryCast(FormView1.FindControl(p + "TextBox0"), TextBox)
        If txtBox IsNot Nothing Then
            e.Command.Parameters("@" + p).Value = CryptorEngine.Encrypt(txtBox.Text, False)
        End If
    End Sub
    Function getImage() As Byte()
        Dim Image As Byte() = Nothing
        If ImageUploadToDB.PostedFile IsNot Nothing AndAlso ImageUploadToDB.PostedFile.FileName <> "" Then
```

Figure 4.30: Encrypt Parameter

Figure 4.31 shows the Encrypt procedure that takes the application defined key stored in the application settings to perform the triple DES encryption using the Microsoft object: “System.Security. Cryptography”.

App_Code/CryptorEngine.vb | Admin/Register.aspx.vb | patients: Query(...ATA\DATABASE.MDF)

(General) | (Declarations)

```
''' <param name="useHashing">use hashing? send to for extra secirity</param>
''' <returns></returns>
Public Shared Function Encrypt(ByVal toEncrypt As String, ByVal useHashing As Boolean) As String
    If toEncrypt = vbNullString Then
        Return vbNullString
    End If
    Dim keyArray As Byte()
    Dim toEncryptArray As Byte() = UTF8Encoding.UTF8.GetBytes(toEncrypt)

    Dim settingsReader As System.Configuration.AppSettingsReader = New AppSettingsReader()
    ' Get the key from config file
    Dim key As String = DirectCast(settingsReader.GetValue("SecurityKey", GetType([String])), String)
    'System.Windows.Forms.MessageBox.Show(key);
    If useHashing Then
        Dim hashmd5 As New MD5CryptoServiceProvider()
        keyArray = hashmd5.ComputeHash(UTF8Encoding.UTF8.GetBytes(key))
        hashmd5.Clear()
    Else
        keyArray = UTF8Encoding.UTF8.GetBytes(key)
    End If

    Dim tdes As New TripleDESCryptoServiceProvider()
    tdes.Key = keyArray
    tdes.Mode = CipherMode.ECB
    tdes.Padding = PaddingMode.PKCS7

    Dim cTransform As ICryptoTransform = tdes.CreateEncryptor()
    Dim resultArray As Byte() = cTransform.TransformFinalBlock(toEncryptArray, 0, toEncryptArray.Length)
    tdes.Clear()
    Return Convert.ToBase64String(resultArray, 0, resultArray.Length)
End Function
```

Figure 4.31: Encrypt Procedure

Since the Patients table is encrypted, then before displaying the patient medical record in a table in the GUI, it is important to decrypt first the information to avoid displaying unreadable encrypted text therefore before displaying each row in the table in the event RowDataBound the procedure Decrypt from the class CryptorEngine is called. After the decryption the readable data is displayed as shown in Figure 4.32.

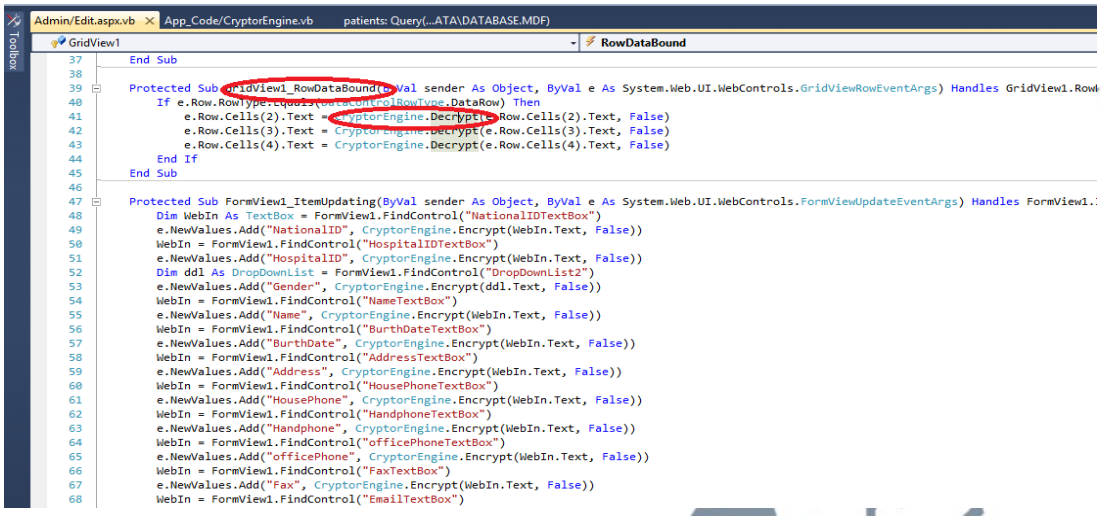

Figure 4.32: Cryptor Engine

Figure 4.33 shows the Decrypt procedure that takes the application defined key stored in the application settings to perform the triple DES decryption using the Microsoft object: “System.Security. Cryptography”.

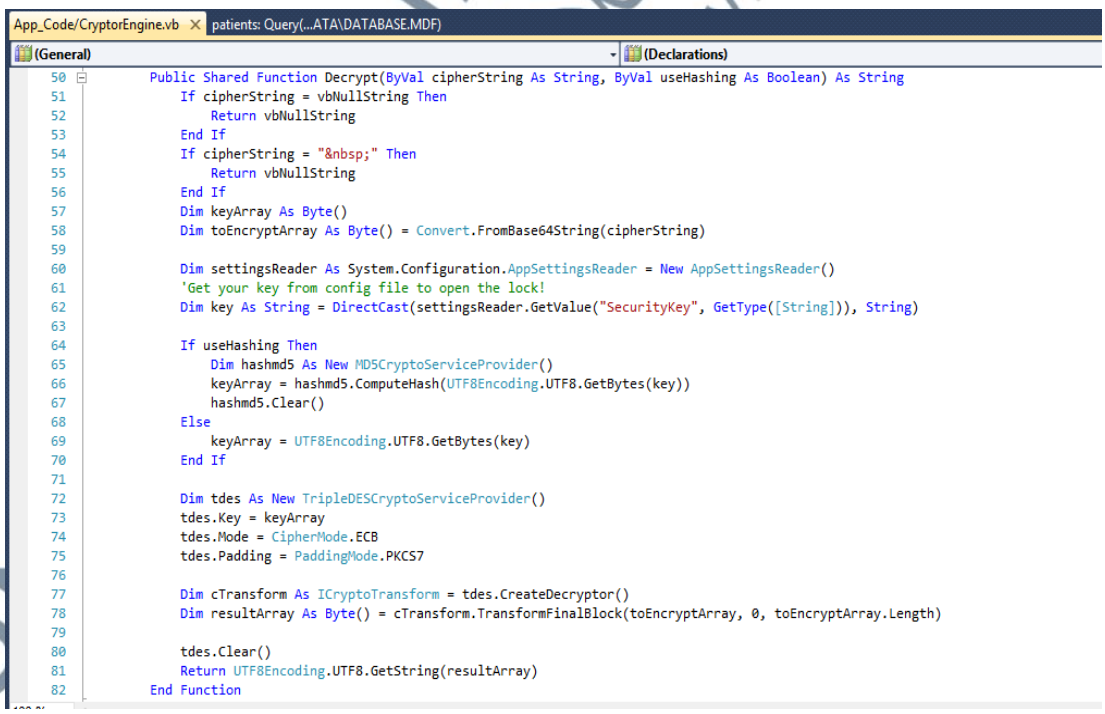

Figure 4.33: Decrypt Procedure

The result of the decryption is shown in the Figure 4.34 which shows data from the Patients table. Even though the data was encrypted in the database, however it arrives decrypted and readable to the user.

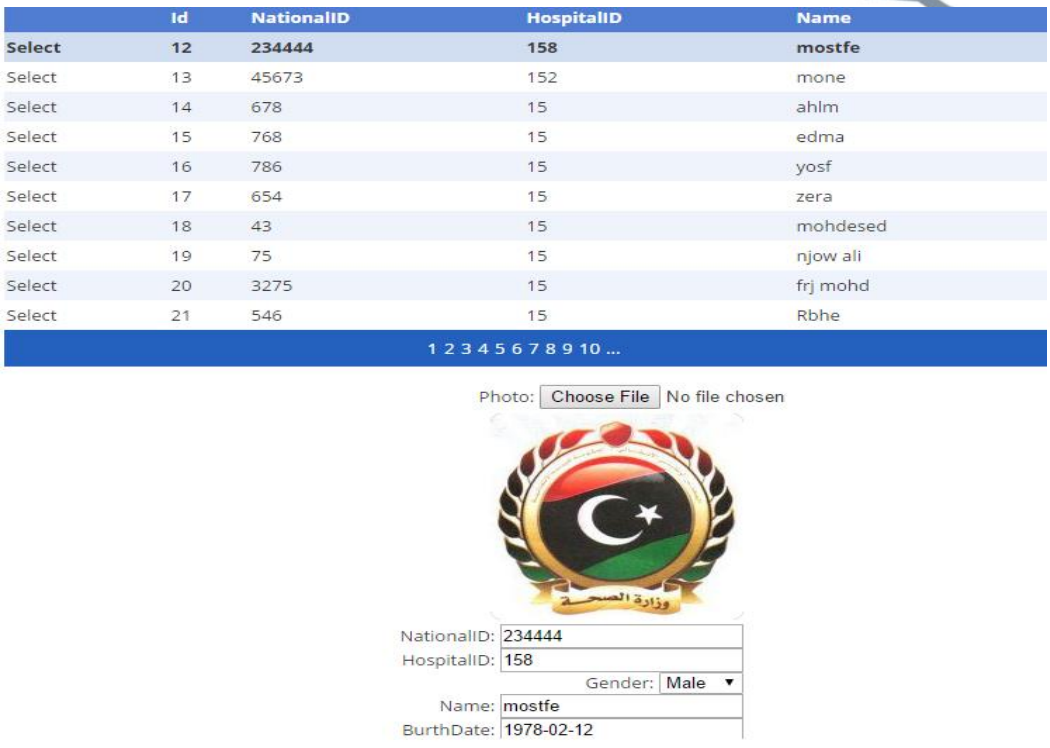

Figure 4.34: Decryption Result

4.5.2 Other Security Measures

4.5.2.1 System Menu

Figure 4.35 shows the anonymous user can only see the front login page.

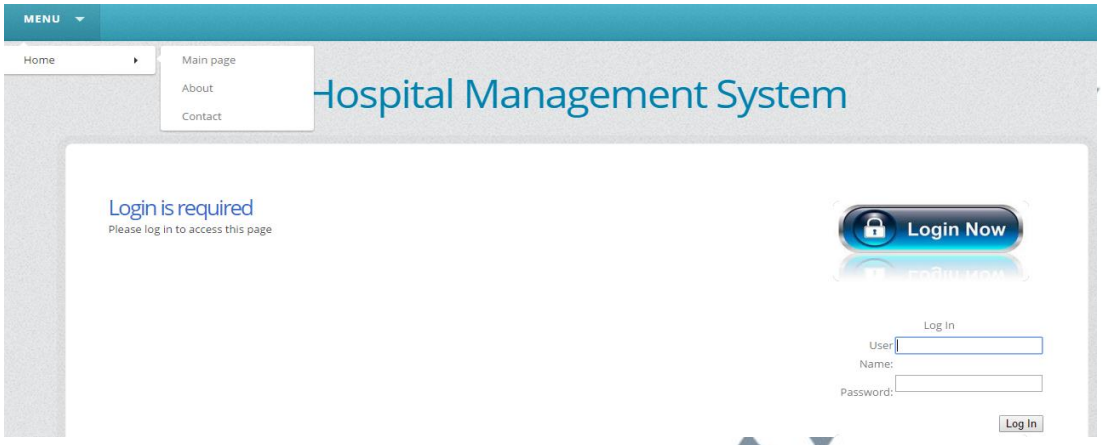

Figure 4.35: Front Login Page

However the logged in user can see only its own menu, not the other users' menu as shown for the admin in Figure 4.36.

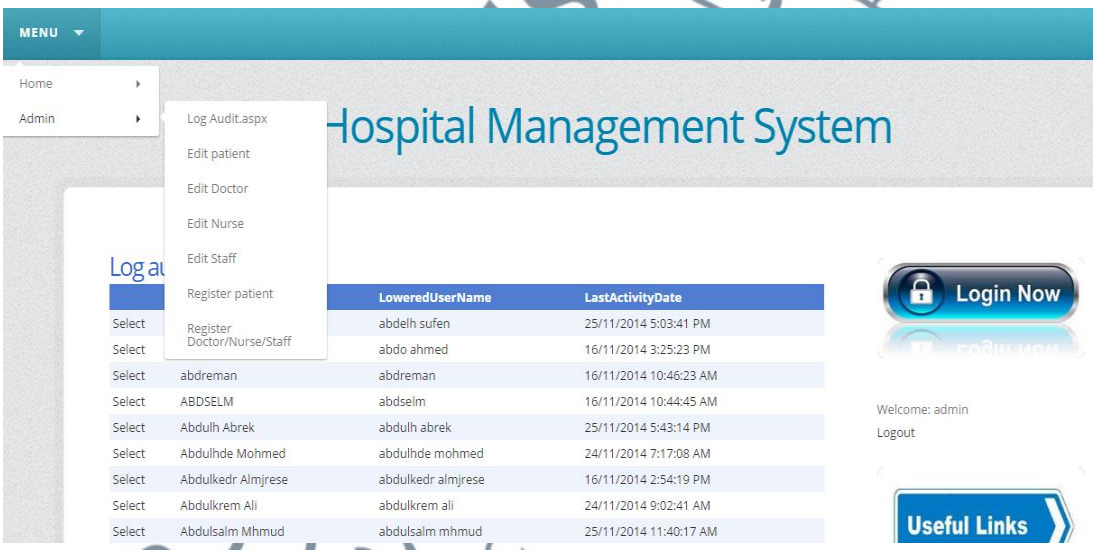

Figure 4.36: Menu for Admin

4.5.2.2 Password Attempts Limitation

The configuration file is adapted under the membership key. As shown in Figure 4.37, the number of attempts is limited to three.

Figure 4.37: Password Attempts Limitation

If the user fail to enter the correct password for more than three attempts his account will be locked. Even if he gives the correct password after that, he cannot log in to the system as shown in Figure 4.38.

Figure 4.38: Failed Login

Figure 4.39 shows that Doctor's account is locked because of more than three login failure attempts.

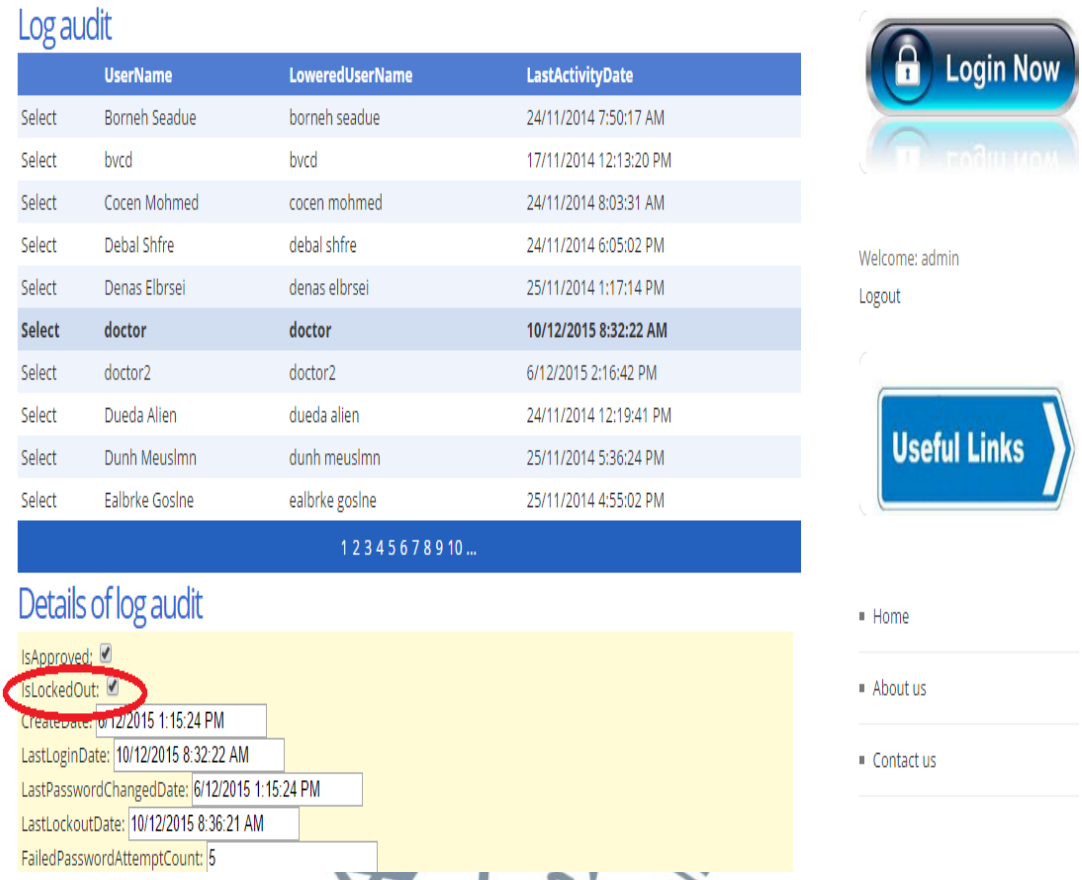

Figure 4.39: Locked Account

Therefore the admin need to unlock the account and then click the update button, as shown in Figure 4.40.

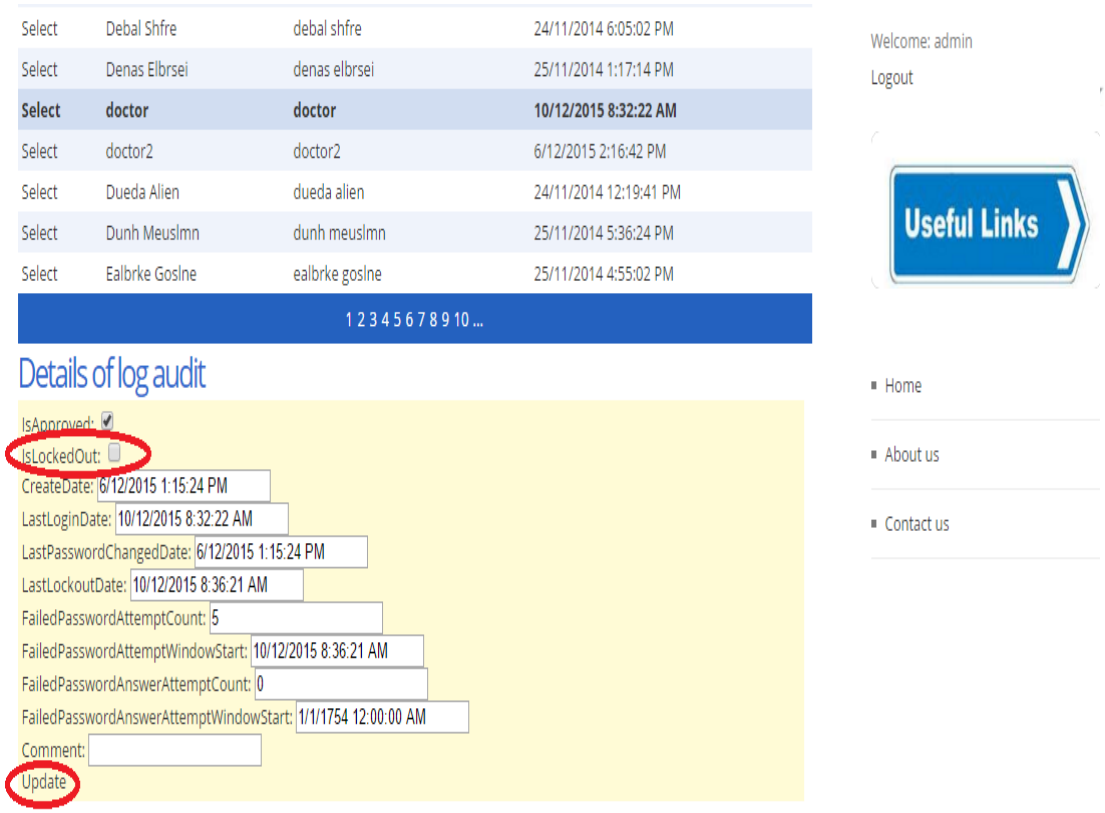

Figure 4.40: Unlock Account

After the account was unlocked by admin, the user can login again as shown in Figure 4.41.

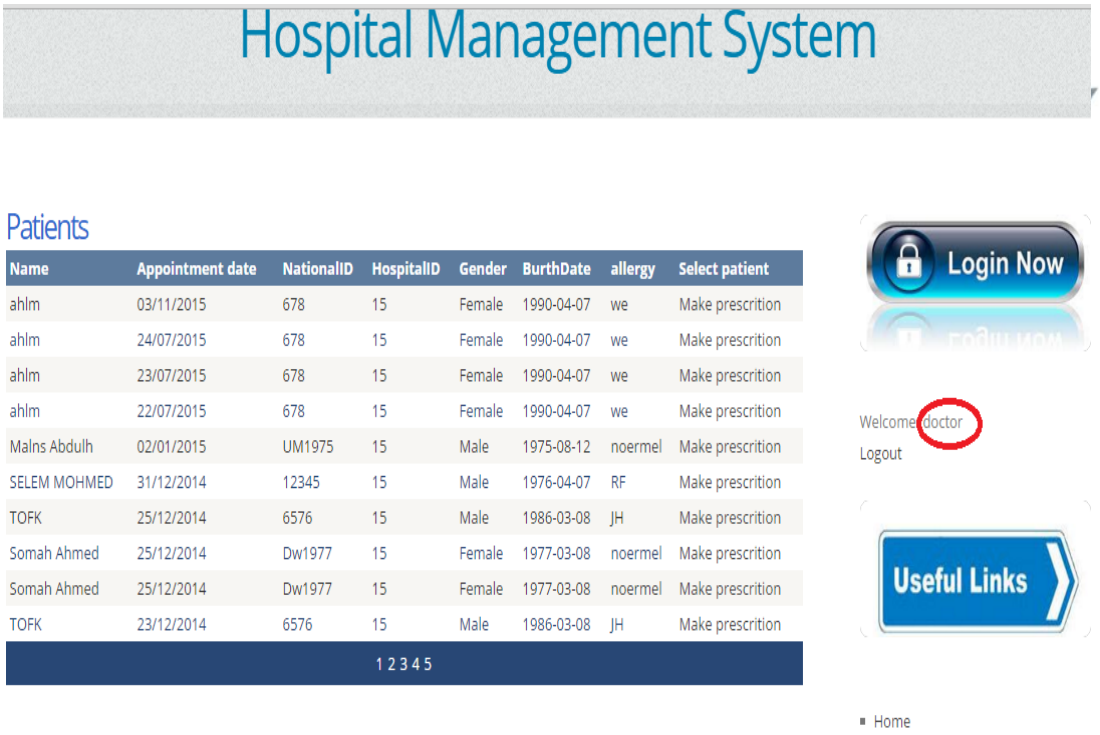

Figure 4.41: Re-login attempt

4.5.2.3 Login Audit

To create the login audit, a master detail GUI is used to combine the table membership with the user table. Each time we select a user in the grid, the user's log audit detail is shown in the form, as shown in Figure 4.42.

For each user registered in the system, the login audit shows useful information about the user such as:

"isapproved" "isLocked" the last login date; last lockout date.

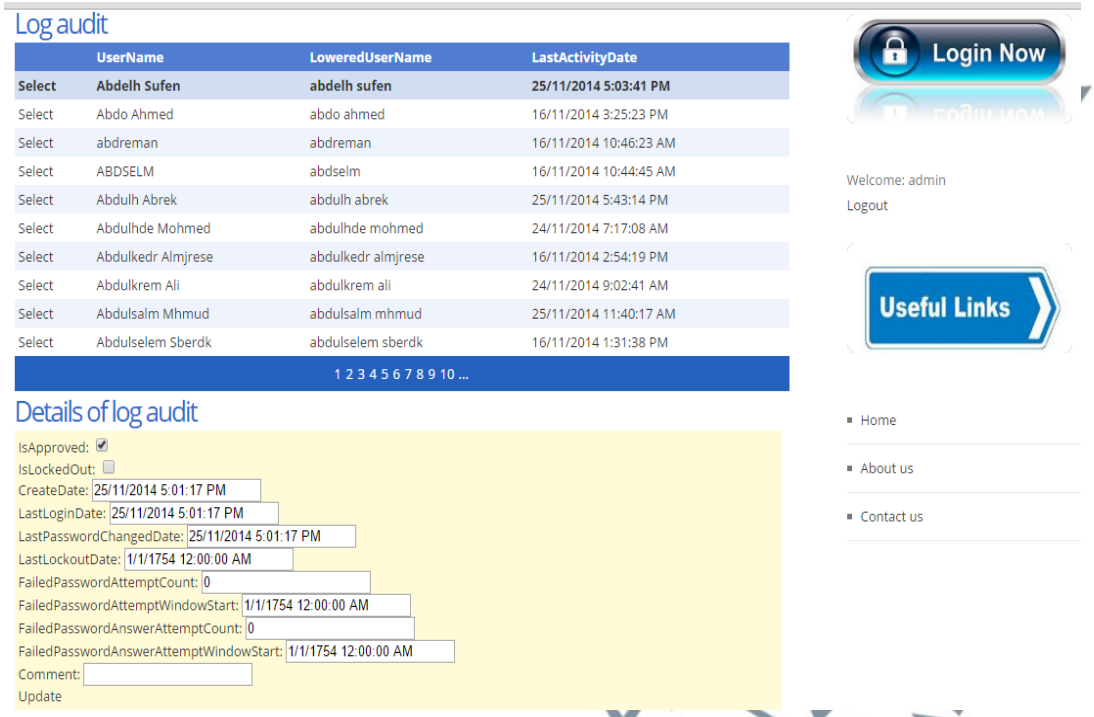

Figure 4.42: User's Log Audit Detail

A complete audit of the database can be done using OmniAudit software or "LBE Desktop Help" which respectively costs: 399USD and 675USD, but unfortunately researcher couldn't afford them.

4.6 CONCLUSION

The design of the system as well as the database was presented in this chapter. The uniform modelling language (UML) was used to design the system.