1. **Abstract**

Organizations currently use a manual system for management and maintenance of their employee medical information. The current system requires numerous paper forms, with data stored throughout the organizational structure. A significant part of the work of the organizations Medical Personnel involves the acquisition, management and timely retrieval of great volumes of the medical information. This information typically involves medical history, hospital records and payment records. All of this information must be managed in an efficient and cost wise manner so that organizational resources may be effectively utilized. The new EMPLOYEE HEALTH MANAGEMENT SYSTEM will automate the management of the employee medical information, making it more efficient and error free. It aims at standardizing and consolidating data, thereby ensuring integrity and reducing inconsistencies.

1. **Introduction about the project**

The project titled as “Staff Health Management System” is a windows-based application. This software provides the facility for registering organization employees and storing their health-related records. The organization medical personnel will be responsible for the management of this system. A registered user first enters their login details to be authorized access to the system. An Access database must be maintained for storage of the data being entered into the system.

1. **Existing System**

The existing employee health management system is not standardized and efficient. Information is often written on paper forms or entered into a Microsoft Excel file. Often, information on forms is incomplete, or does not follow a uniform standard. Forms are often lost in transit between departments and multiple copies of the same information exist in the organization which leads to inconsistences in the information.

**Drawbacks of Existing System**

* Lack of security for the data
* Time consuming
* Needs a lot of paper work
* Leads to a duplication of data
* Leads to a lot of inconsistent data

To avoid all these limitations and make the working more accurate and efficient, the system needs to be computerized and a have a central database to store all information.

1. **Proposed System**

The aim of proposed system is to develop an improved system. The proposed system can overcome all the limitations of the existing system. The system provides proper security and reduces the manual work. The existing system has several disadvantages and many more difficulties to work well. The proposed system tries to eliminate or reduce these difficulties up to some extent. The proposed system will help the user to reduce the workload and mental conflict. The proposed system helps the user to work in a user-friendly environment and he can easily do his jobs without time lagging.

**Expected Advantages of Proposed System**

The system is very simple in design and to implement. The system requires very low system resources and the system will work in almost all configurations. It has got following features

* Ensured data accuracy.
* Minimized manual data entry.
* Minimum time needed for various processes like reporting.
* Greater efficiency.
* User friendliness.
* Minimum time required.

1. **MODULES**
2. Login

* this module gives authentication to users of the system

1. Search Employees

* this module allows the user to search for the organizations employees registered in the system

1. Employee Registration

* this module allows the system user to add new Employees into the new system

1. Records Entry

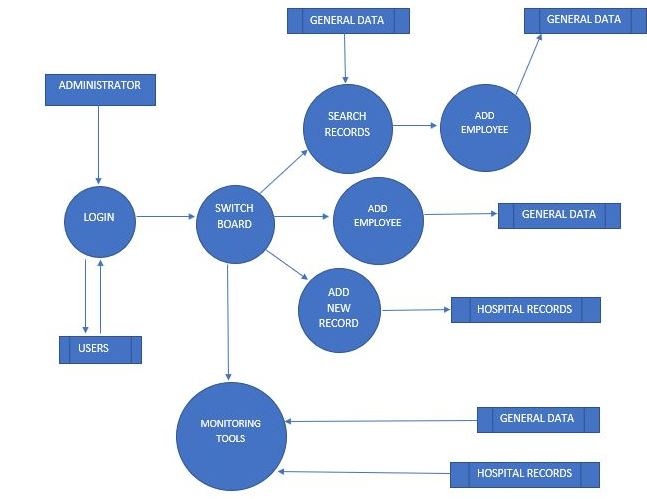
* this module will allow the system user to add Employees health records into the system.

1. Monitoring tools

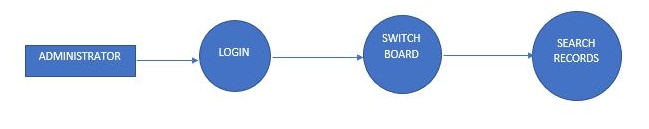
* this module will allow the user to view the employees registered in the system and their hospital records and access other reporting features.

1. **Data Flow Diagram**

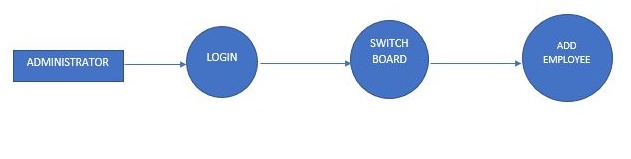
**Context flow diagram**

****

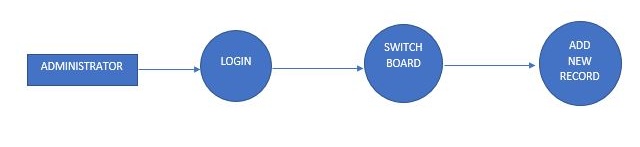
**Search Employees module DFD**

****

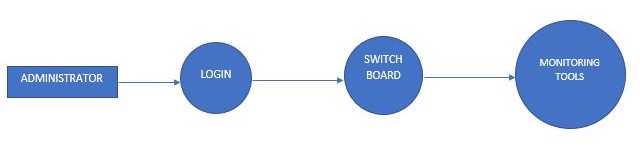
**Employee Registration module DFD**

****

**Records Entry module DFD**

****

**Monitoring tools module DFD**

****

1. **DATABASE DESIGN**

A database is an organized mechanism that has the capability of storing information through which a user can retrieve stored information in an effective and efficient manner. The data is the purpose of any database and must be protected.

The database design is a two-level process. In the first step, user requirements are gathered together and a database is designed which will meet these requirements as clearly as possible. This step is called Information Level Design and it is taken independent of any individual DBMS.

In the second step, this Information level design is transferred into a design for the specific DBMS that will be used to implement the system in question. This step is called Physical Level Design, concerned with the characteristics of the specific DBMS that will be used. A database design runs parallel with the system design.

Normalization is the process of decomposing the attributes in an application, which results in a set of tables with very simple structure. The purpose of normalization is to make tables as simple as possible. Normalization is carried out in this system for the following reasons:

* + To structure the data so that there is no repetition of data.
  + To permit simple retrieval of data in response to query and report request.
  + To simplify the maintenance of the data through updates, insertions,

Deletions.

* + To reduce the need to restructure or reorganize data when new application

Requirements arise.

1. **RELATIONAL DATABASE MANAGEMENT SYSTEM (RDBMS):**

A relational model represents the database as a collection of relations. Each relation resembles a table of values or file of records. In formal relational model terminology, a row is called a tuple, a column header is called an attribute and the table is called a relation. A relational database consists of a collection of tables, each of which is assigned a unique name. A row in a table represents a set of related values.

**RELATIONSHIPS:**

* Table relationships are established using Key. The two main keys of prime importance are Primary Key & Foreign Key. Entity Integrity and Referential Integrity Relationships can be established with these keys.
* Entity Integrity enforces that no Primary Key can have null values.
* Relationships have been set between every table in the database. This ensures both Referential and Entity Relationship Integrity.

**NORMALIZATION:**

As the name implies, it denoted putting things in the normal form. The application developer via normalization tries to achieve a sensible organization of data into proper tables and columns and where names can be easily correlated to the data by the user. Normalization eliminates repeating groups at data and thereby avoids data redundancy which proves to be a great burden on the computer resources. These includes:

* Normalize the data.
* Choose proper names for the tables and columns.
* Choose the proper name for the data.

1. **tables structure**

**Table Name: Users**

|  |  |  |  |
| --- | --- | --- | --- |
| **Fieldname** | **Data Type** | **Length** | **Key** |
| UID | Short text | 25 | Primary key |
| UName | Short text | 25 | - |
| Pass | Short text | 10 | - |

**Table Name: GeneralData**

|  |  |  |  |
| --- | --- | --- | --- |
| **Fieldname** | **Data Type** | **Length** | **Key** |
| Staff\_Department | Short Text | 20 | - |
| Client\_ID | Short Text | 20 | Primary key |
| Date\_Registered | Date/Time | - | - |
| First\_Name | Short Text | 20 | - |
| Surname | Short Text | 20 | - |
| Date\_of\_Birth | Date/Time | - | - |
| Address | Long Text | 50 | - |
| Telephone 1 | Short Text | 13 | - |
| Telephone 2 | Short Text | 13 | - |
| Nationality | Short Text | 15 | - |
| Education | Short Text | 15 | - |
| HIV\_Status | Short Text | 15 | - |
| ART\_Start\_Date | Date/Time | - | - |
| TB\_History | Long Text | 50 | - |
| TB\_Diagnosis\_Date | Date/Time | - | - |
| TB\_Treatment\_History | Long Text | 50 | - |

**Table Name: HospitalRecords**

|  |  |  |  |
| --- | --- | --- | --- |
| **Fieldname** | **Data Type** | **Length** | **Key** |
| Employee\_Name | Short Text | 20 | Primary Key |
| Institution\_Name | Long Text | 50 | - |
| Visit\_Date | Date/Time | - | - |
| Conditions | Long Text | 50 | - |
| Treatment | Long Text | 50 | - |
| Bill | Short Text | 20 | - |
| Paid | Short Text | 20 | - |
| Date\_Paid | Date/Time | - | - |
| Remarks | Long Text | 50 | - |

1. **Software and Hardware Specification**

**Hardware Specification**

Processor: Duo Core processor with or above 2.0GHz

Hard disk: 40GB or above

Ram: 4GB or above

**Software Specification**

Operating System: Windows 10

Application Database: Microsoft Access

Application Interface Development: Visual Basic 2012

1. **Code Details**

**Sample codes**

Login

Imports System.Data.OleDb

Public Class Login

Private Sub Cancel\_Click(sender As Object, e As EventArgs) Handles Cancel.Click

Application.Exit()

End Sub

Private Sub Login\_FormClosing(sender As Object, e As FormClosingEventArgs) Handles MyBase.FormClosing

Application.Exit()

End Sub

Private Sub Login\_FormClosed(sender As Object, e As FormClosedEventArgs) Handles MyBase.FormClosed

cn.Close()

End Sub

Private Sub Login\_Load(sender As Object, e As EventArgs) Handles MyBase.Load

ComboBox1.Items.Clear()

PasswordTextBox.Clear()

UsernameTextBox.Clear()

Dim ctr, i As Integer

ds.Clear()

If cn.State = ConnectionState.Open Then

cn.Close()

End If

Module1.conn()

cn.Open()

str = "select \* from Users"

cmd = New OleDbCommand(str, cn)

da.SelectCommand = cmd

da.Fill(ds, "Users")

ctr = ds.Tables("Users").Rows.Count - 1

For i = 0 To ctr

ComboBox1.Items.Add(ds.Tables("Users").Rows(i)(1).ToString)

Next

End Sub

Private Sub ComboBox1\_SelectedIndexChanged(sender As Object, e As EventArgs) Handles ComboBox1.SelectedIndexChanged

UsernameTextBox.Text = ComboBox1.Text

End Sub

Private Sub OK\_Click(sender As Object, e As EventArgs) Handles OK.Click

Dim ctr, i As Integer

Dim che As Integer = 0

Dim emp As String = ""

ds.Clear()

'cn.open()

str = "select \* from Users"

cmd = New OleDbCommand(str, cn)

da.SelectCommand = cmd

da.Fill(ds, "Users")

ctr = ds.Tables("Users").Rows.Count - 1

For i = 0 To ctr

If ds.Tables("Users").Rows(i)(1).ToString = UsernameTextBox.Text And ds.Tables("Users").Rows(i)(2).ToString = PasswordTextBox.Text Then

che = 1

emp = ds.Tables("Users").Rows(i)(3).ToString

End If

Next

If che = 1 Then

cn.Close()

If emp = "ADMIN" Then

ElseIf emp = "TEACHING STAFF" Then

' Home.StaffToolStripMenuItem.Visible = False

' Home.FeesToolStripMenuItem.Visible = False

'Home.AccountsToolStripMenuItem.Visible = False

' Home.UserToolStripMenuItem.Visible = False

' Home.PaymentsToolStripMenuItem.Visible = False

ElseIf emp = "NON TEACHING STAFF" Then

' Home.StudentToolStripMenuItem.Visible = False

' Home.ClassToolStripMenuItem.Visible = False

'Home.StaffToolStripMenuItem.Visible = False

'Home.ExamToolStripMenuItem.Visible = False

' Home.SubjectToolStripMenuItem.Visible = False

' Home.UserToolStripMenuItem.Visible = False

End If

' Home.Button1.Text = ComboBox1.Text & " - Logout"

Switchboard.Show()

'Me.Hide()

ComboBox1.ResetText()

UsernameTextBox.Clear()

PasswordTextBox.Clear()

Else

MsgBox("Password Incorect for Selected User.")

End If

End Sub

End Class

Search Records

Public Class Search\_Records

Private Sub GeneralDataBindingNavigatorSaveItem\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles GeneralDataBindingNavigatorSaveItem.Click

Me.Validate()

Me.GeneralDataBindingSource.EndEdit()

Me.TableAdapterManager.UpdateAll(Me.Staff\_Health\_ManagementDataSet)

End Sub

Private Sub Search\_Records\_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

'TODO: This line of code loads data into the 'Staff\_Health\_ManagementDataSet.GeneralData' table. You can move, or remove it, as needed.

'Me.GeneralDataTableAdapter.Fill(Me.Staff\_Health\_ManagementDataSet.GeneralData)

End Sub

Private Sub Button1\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click

End Sub

Private Sub txtsearch\_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles txtsearch.TextChanged

Dim search As String = "%" + txtsearch.Text + "%"

Me.GeneralDataTableAdapter.FillBySearchName(Me.Staff\_Health\_ManagementDataSet.GeneralData, search, search)

End Sub

Private Sub Button2\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button2.Click

Enrolment.Show()

End Sub

End Class

Employee Registration

Imports System.Data.OleDb

Public Class Enrolment

Dim provider As String

Dim dataFile As String

Dim connString As String

Dim myConnection As OleDbConnection = New OleDbConnection

Private Sub TextBox1\_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles txtID.TextChanged

End Sub

Private Sub TextBox5\_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles txtaddress.TextChanged

End Sub

Private Sub btnClose\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnClose.Click

Me.Close()

End Sub

Private Sub btnClear\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnClear.Click

Me.txtaddress.Text = ""

Me.txtID.Text = ""

Me.txtname.Text = ""

Me.txtnum1.Text = ""

Me.txtnum2.Text = ""

Me.txtsurname.Text = ""

Me.cbodept.Text = ""

Me.cboedu.Text = ""

Me.cbonat.Text = ""

Me.cbohiv.Text = ""

Me.cbotb.Text = ""

Me.txttbhis.Text = ""

End Sub

Private Sub Enrolment\_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

End Sub

Private Sub btnAdd\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnAdd.Click

provider = "Provider=Microsoft.Jet.OLEDB.4.0;Data Source="

dataFile = "C:\Users\PHILLIP\Documents\Staff Health Management.mdb"

connString = provider & dataFile

myConnection.ConnectionString = connString

myConnection.Open()

Dim str As String

str = "Insert into GeneralData([Client\_ID],[Address],[Telephone 1],[Telephone 2],[Nationality],[Education],[Staff\_Department],[Date\_Registered],[First\_Name],[Surname],[Date\_of\_Birth],[HIV\_Status],[ART\_Start\_Date],[TB\_History],[TB\_Diagnosis\_Date],[TB\_Treatment\_History]) Values (?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?)"

Dim cmd As OleDbCommand = New OleDbCommand(str, myConnection)

cmd.Parameters.Add(New OleDbParameter("Client\_ID", CType(txtID.Text, String)))

cmd.Parameters.Add(New OleDbParameter("Address", CType(txtaddress.Text, String)))

cmd.Parameters.Add(New OleDbParameter("Telephone 1", CType(txtnum1.Text, String)))

cmd.Parameters.Add(New OleDbParameter("Telephone 2", CType(txtnum2.Text, String)))

cmd.Parameters.Add(New OleDbParameter("Nationality", CType(cbonat.Text, String)))

cmd.Parameters.Add(New OleDbParameter("Education", CType(cboedu.Text, String)))

cmd.Parameters.Add(New OleDbParameter("Staff\_Department", CType(cbodept.Text, String)))

cmd.Parameters.Add(New OleDbParameter("Date\_Registered", CType(Datereg.Text, Date)))

cmd.Parameters.Add(New OleDbParameter("First\_Name", CType(txtname.Text, String)))

cmd.Parameters.Add(New OleDbParameter("Surname", CType(txtsurname.Text, String)))

cmd.Parameters.Add(New OleDbParameter("Date\_Of\_Birth", CType(Datedob.Text, String)))

cmd.Parameters.Add(New OleDbParameter("HIV\_Status", CType(cbohiv.Text, String)))

cmd.Parameters.Add(New OleDbParameter("ART\_Start\_Date", CType(datehiv.Text, Date)))

cmd.Parameters.Add(New OleDbParameter("TB\_History", CType(cbotb.Text, String)))

cmd.Parameters.Add(New OleDbParameter("TB\_Diagnosis\_Date", CType(datetb.Text, Date)))

cmd.Parameters.Add(New OleDbParameter("TB\_Treatment\_History", CType(txttbhis.Text, String)))

Try

cmd.ExecuteNonQuery()

cmd.Dispose()

myConnection.Close()

txtaddress.Clear()

txtID.Clear()

txtname.Clear()

txtnum1.Clear()

txtnum2.Clear()

txtsurname.Clear()

txttbhis.Clear()

Catch ex As Exception

MsgBox(ex.Message)

End Try

End Sub

Private Sub GroupBox1\_Enter(sender As Object, e As EventArgs) Handles GroupBox1.Enter

End Sub

End Class

Records Entry

Imports System.Data.OleDb

Public Class Employee\_Hospital\_Visit

Dim provider As String

Dim dataFile As String

Dim connString As String

Dim myConnection As OleDbConnection = New OleDbConnection

Private Sub Employee\_Hospital\_Visit\_Load(sender As Object, e As EventArgs) Handles MyBase.Load

End Sub

Private Sub ComboBox2\_SelectedIndexChanged(sender As Object, e As EventArgs) Handles cbopaid.SelectedIndexChanged

End Sub

Private Sub Button2\_Click(sender As Object, e As EventArgs) Handles Button2.Click

Monitoring\_Tools.Show()

End Sub

Private Sub Button3\_Click(sender As Object, e As EventArgs) Handles Button3.Click

Me.Close()

End Sub

Private Sub ADD\_Click(sender As Object, e As EventArgs) Handles ADD.Click

provider = "Provider=Microsoft.Jet.OLEDB.4.0;Data Source="

dataFile = "C:\Users\PHILLIP\Documents\Staff Health Management.mdb"

connString = provider & dataFile

myConnection.ConnectionString = connString

myConnection.Open()

Dim str As String

str = "Insert into HospitalRecords([Employee\_Name],[Institution\_Name],[Visit\_Date],[Conditions],[Treatment],[Bill],[Paid],[Date\_Paid],[Remarks]) Values (?,?,?,?,?,?,?,?,?)"

Dim cmd As OleDbCommand = New OleDbCommand(str, myConnection)

cmd.Parameters.Add(New OleDbParameter("Employee\_Name", CType(cboname.Text, String)))

cmd.Parameters.Add(New OleDbParameter("Institution\_Name", CType(txtinstitution.Text, String)))

cmd.Parameters.Add(New OleDbParameter("Visit\_Date", CType(datevisit.Text, Date)))

cmd.Parameters.Add(New OleDbParameter("Conditions", CType(txtconditions.Text, String)))

cmd.Parameters.Add(New OleDbParameter("Treatment", CType(txttreatment.Text, String)))

cmd.Parameters.Add(New OleDbParameter("Bill", CType(txtbill.Text, String)))

cmd.Parameters.Add(New OleDbParameter("Paid", CType(cbopaid.Text, String)))

cmd.Parameters.Add(New OleDbParameter("Date\_Paid", CType(datepaid.Text, Date)))

cmd.Parameters.Add(New OleDbParameter("Remarks", CType(txtremarks.Text, String)))

Try

cmd.ExecuteNonQuery()

cmd.Dispose()

myConnection.Close()

txtbill.Clear()

txtconditions.Clear()

txtinstitution.Clear()

txtremarks.Clear()

txttreatment.Clear()

Catch ex As Exception

MsgBox(ex.Message)

End Try

End Sub

Private Sub cboname\_SelectedIndexChanged(sender As Object, e As EventArgs) Handles cboname.SelectedIndexChanged

End Sub

End Class

Monitoring Tools

Public Class Monitoring\_Tools

Private Sub LinkLabel1\_LinkClicked(sender As Object, e As LinkLabelLinkClickedEventArgs) Handles LinkLabel1.LinkClicked

Hospital\_Records.Show()

End Sub

Private Sub LinkLabel2\_LinkClicked(sender As Object, e As LinkLabelLinkClickedEventArgs) Handles LinkLabel2.LinkClicked

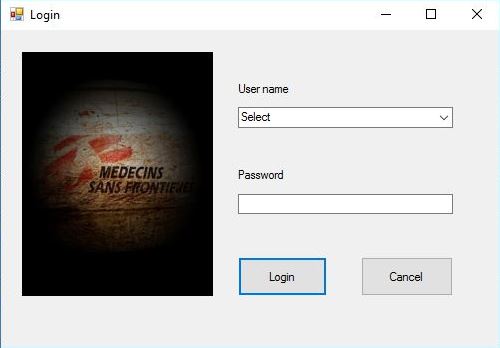
All\_Registered\_Employees.Show()

End Sub

End Class

1. **SCREEN SHOTS**

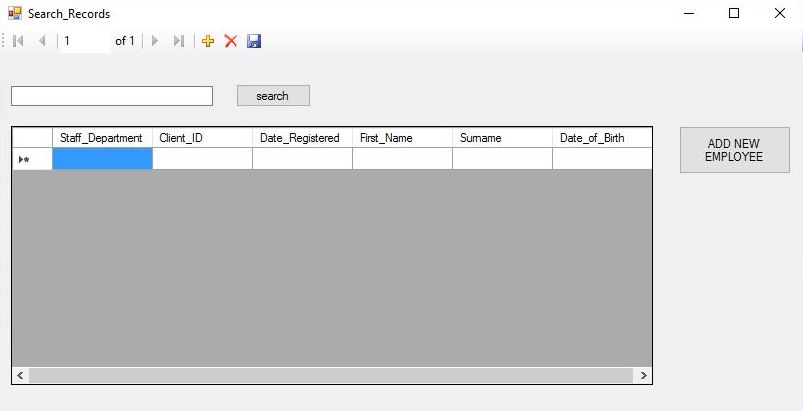
**Login**

****

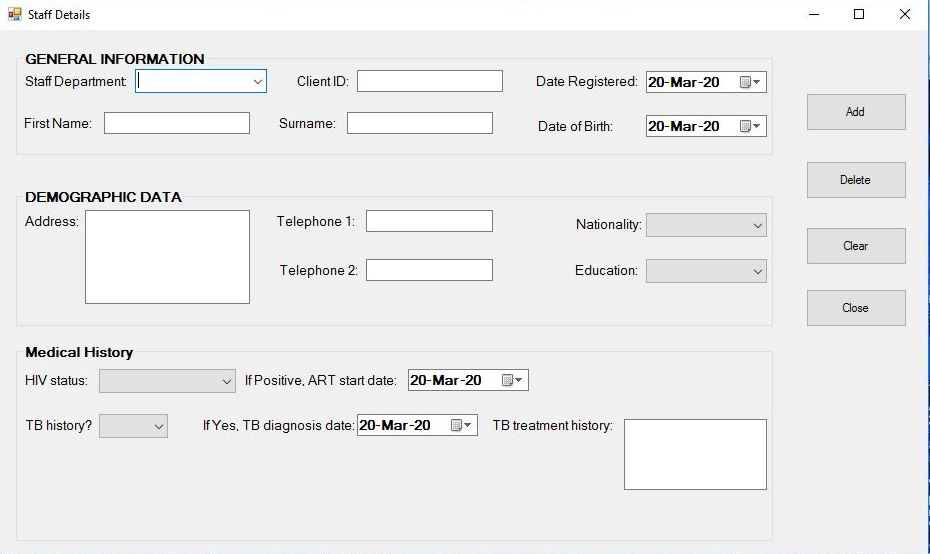
**Switch board**

****

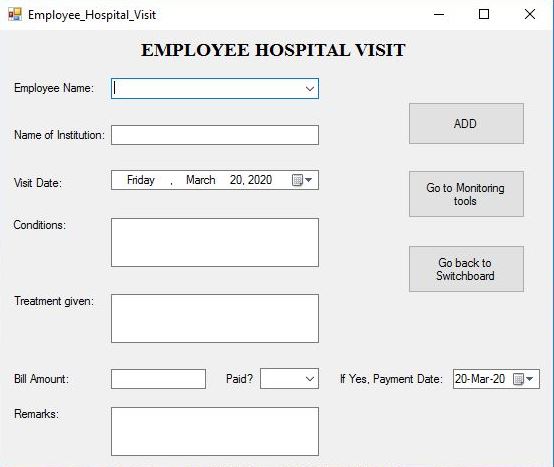
**Search Employee**

****

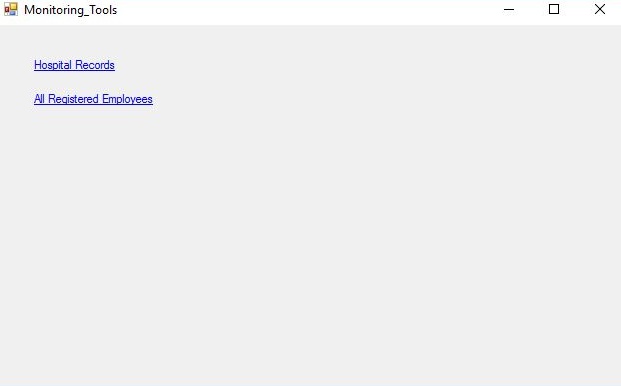
**Staff Registration**

****

**Hospital Visit**

****

**Monitoring Tools**

****

1. **Conclusion**

The project titled as “Employee Health Management System” is a database application. This software provides facility for registering all organization employees, capturing their health details and their hospital visits for easy administrative management. This software is developed with scalability in mind. Additional modules can be easily added when necessary. The software is developed with modular approach. All modules in the system have been tested with valid data and invalid data and everything work successfully. Thus, the system has fulfilled all the objectives identified and is able to replace the existing system.

The project has been completed successfully with the maximum satisfaction of the organization. The constraints are met and overcome successfully. The system is designed as it was decided in the design phase. The project gives good idea on developing a full-fledged application satisfying the user requirements.

The system is very flexible and versatile. This software has a user-friendly screen that enables the user to use without any inconvenience. Validation checks induced have greatly reduced errors. Provisions have been made to upgrade the software. The application has been tested with live data and has provided a successful result. Hence the software has proved to work efficiently.

1. **Future Enhancement**

* The software can be developed further to include a lot of modules because the proposed system is developed on the view of future.