Prototype

**Login, create account, forgot password**

The account-related modules are not included in the prototype since the prototype is built to be a ‘proof of concept’ and a login system is not vital to the ability for the prototype to effectively carry out the main basic functions of the program (Managing and adding patients).

**View patients**

Viewing and managing the patients in the program is the primary objective of the project and the viewing patients module is therefore vital to include in the prototype to showcase the basic functionality of the program

This module consists of the code run/ subroutine that occurs when the user clicks the *View Patients* button on the main menu. It brings up a window featuring a data grid with all relevant patient data and the options to: *View specific patient, add patient, delete patient, go back, discharge patient or contact patient GP.*

This data is taken from a CSV file in the program files. If the CSV is not there when this module starts, it is created

**Main view patients screen**

A screenshot of a computer

Description automatically generated

**The page for viewing a specific patient**

A screenshot of a computer

Description automatically generated

**If a user tries to start a task that requires a selected user without one selected**

A screenshot of a computer error

Description automatically generated

**If the user attempt to close the form**

**A screenshot of a computer

Description automatically generated**

|  |
| --- |
| **Code to load Data into the DataGrid** |
| Private Sub Patients\_Load(sender As Object, e As EventArgs) Handles MyBase.Load  Dim ReadFile As String = IO.Path.Combine(Application.StartupPath, "Patients table.csv")  Dim LoadData As String = ""  Dim SplitAt() As String  Dim ToContinue As Boolean = True 'ToContinue ensures the loop stops when the reader reaches the end of the file  'Checking file exists - if it doesnt nothing is loaded to avoid error etc  If File.Exists(ReadFile) = True Then  Using objReader As New StreamReader(ReadFile)  If objReader.Peek() = -1 Then  ToContinue = False  End If  Do While ToContinue  'Reading the data  LoadData = objReader.ReadLine()  SplitAt = Split(LoadData, ",")  GridPatients.Rows.Add(SplitAt)  If objReader.Peek() = -1 Then  ToContinue = False  End If  Loop  End Using  End If  End Sub |
| **Saving data to the CSV** |
| Private Sub BtnSave\_Click(sender As Object, e As EventArgs) Handles BtnSave.Click  Dim FileName As String = "Patients table.csv"  'Calling the SavePatients subroutine  SavePatients(IO.Path.Combine(Application.StartupPath, FileName))  End Sub |
| **Saving as CSV through save dialog** |
| Private Sub BtnExport\_Click(sender As Object, e As EventArgs) Handles BtnExport.Click  Dim saving As New SaveFileDialog()  saving.FileName = "Patients table"  saving.Filter = "CSV File | \*.csv"  'MessageBox.Show("Point")  If saving.ShowDialog() = DialogResult.OK Then  SavePatients(IO.Path.Combine(Application.StartupPath, saving.FileName))  End If  End Sub |
| **Saving data subroutine (used it both saving methods)** |
| Private Sub SavePatients(location As String)  ‘If the file already exists, delete it  If Dir(location) <> "" Then  Kill(location)  End If  ‘Writing data into CSV  Using s As StreamWriter = New StreamWriter(File.OpenWrite(location))  For Each row As DataGridViewRow In GridPatients.Rows  Dim rowData As New List(Of String)  For Each column As DataGridViewColumn In GridPatients.Columns  rowData.Add(Convert.ToString(row.Cells(column.Name).Value))  Next  s.WriteLine(String.Join(",", rowData))  Next  s.Close()  End Using  End Sub |

**Sort patients, Search for patient**

The sorting and searching modules are not included in the prototype as they are not essential for the basic function of the program as the core functionality works with or without sorting and searching features. They are therefore unnecessary to include in a proof of concept.

**Add patients**

In order for patients to added and managed by the program, they have to first be added by the user, so for the program to perform its most basic function the add patients module must be included and is therefore in the prototype.

The add patients module consists of a form that appears when the user clicks the ‘add patients’ button on the ‘view patients’ menu. It displays the user with a range of text boxes and other input spaces so that the user can provide all the necessary information to add a patient.

Note: Once a patient is added in the form they are added to the datagrid and this won’t be saved into the CSV file until the users clicks the save form button from the view patients module.

**Example data being input into the add patient from**

A screenshot of a computer

Description automatically generated

**Added item in datagrid after adding the patient above**



**If fields are missing or data is of the wrong type**

A screenshot of a computer

Description automatically generated

|  |
| --- |
| **Code run when the add patient button is clicked on the add patient form** |
| Public Class AddPatient  Private Sub BtnExecute\_Click(sender As Object, e As EventArgs) Handles BtnExecute.Click  Dim Name As String  Dim Age As Integer  Dim Height As Double  Dim Weight As Double  Dim DOB As Date  Dim History As String  Dim GPName As String  Dim GPEmail As String  Dim Ethnicity As String  Dim OtherInfo As String  Dim Sex As String  Try  'taking data from patient input  Name = TxtName.Text  Height = CDbl(TxtHeight.Text)  Weight = CDbl(TxtWeight.Text)  DOB = DateTimePicker2.Text  History = TxtHistory.Text  GPName = TxtGPname.Text  GPEmail = TxtGPEmail.Text  Ethnicity = CmbEthnicity.Text  OtherInfo = TxtOther.Text  'assigning sex variable  If RdMale.Checked Then  Sex = "Male"  ElseIf RdFemale.Checked Then  Sex = "Female"  Else  Sex = "Other"  End If  'calculating age from DOB  Dim timeDif As TimeSpan = Now - DOB  Age = Int((timeDif.TotalSeconds) / 31536000)  LblDisplay.Text = "Patient added"  'adds patient to data grid  Patients.GridPatients.Rows.Add(Name, Sex, Age, Height, Weight, DOB, History, GPName, GPEmail, Ethnicity, OtherInfo)  '[Add to file]  Me.Close()  Catch ex As Exception  MsgBox("Please fill all fields or check data entered is appropriate", MsgBoxStyle.OkOnly, "Error")  End Try  End Sub |

**Remove patients**

Removing patients is a basic – easy to program section of the view patients module and is therefore included in the prototype to fully demonstrate the view patients functionality

Patients are removed either by being discharged or by being deleted. In the later version of the project discharged patients will be put into a separate file to be accessed however in the prototype delete and discharge effectively do the same thing.

**Delete button being clicked to remove Sarah Johnson**

A screenshot of a computer

Description automatically generated

**Sarah Johnson removed**

**A screenshot of a computer

Description automatically generated**

**Sarah Johnson being discharged**

**A screenshot of a computer

Description automatically generated#**

|  |
| --- |
| **When the delete patient button is clicked** |
| Private Sub BtnDeletePatient\_Click(sender As Object, e As EventArgs) Handles BtnDeletePatient.Click 'delete patient  Try  Dim n As Integer = GridPatients.SelectedRows(0).Index  GridPatients.Rows.RemoveAt(n) 'removes selected item  Catch ex As Exception 'occurs when there is no patient selected  MsgBox("Select patient", MsgBoxStyle.OkOnly, "Error")  End Try  End Sub |
| **Loads selected patient into discharge form** |
| Private Sub DischargePatient\_Load(sender As Object, e As EventArgs) Handles MyBase.Load  Dim n As Integer = Patients.GridPatients.SelectedRows(0).Index  Dim Name As String  'takes patient name and puts it on the label in the discahrge form  Name = Patients.GridPatients.Rows(n).Cells("PatientName").Value  Firstname.Text = Name  End Sub |
| **Discharging patient** |
| Private Sub BtnDischarge\_Click(sender As Object, e As EventArgs) Handles BtnDischarge.Click  Try  Dim n As Integer = Patients.GridPatients.SelectedRows(0).Index  Patients.GridPatients.Rows.RemoveAt(n) 'removes the selected patient  Catch ex As Exception  MsgBox("Error in discharging patient", MsgBoxStyle.OkOnly, "Error") 'if something goes wrong with file handling  Me.Hide()  End Try  End Sub |

**Generate operating list, Generate clinic template**

The List/ template generating modules are not included in the prototype as they are not part of the core function of the program and are therefore not required for a basic proof of concept to showcase and demonstrate the core functionality of the program.

Prototype overall

|  |  |
| --- | --- |
| Good features | Short comings |
| * The system can effectively take inputs in the form of added patients but the user and put these into the datagrid * The datagrid is easy to use and program with * The program is fairly easy to work with and most elements are easy enough to use for less technical users * It displays a useable version of all core features of the program | * The tab index is not user friendly at times: This can be changed in the properties menu later down the line * The view patient menu has a few   problems with the way it is set out: The layout and design of this section should be changed in order to make this section better suit its purpose   * The elements in the forms are not anchored meaning that the form becomes far less user friendly if it is stretched or compressed: this can be added in the properties menu |

***-----------------------------------------------Full source code: --------------------------------------------------------------***

|  |
| --- |
| Login form |
| Public Class MainPage  Private Sub Button1\_Click(sender As Object, e As EventArgs) Handles Button1.Click  'this is place holder code as no login logic is inlcuded  MainMenu.Show()  TextBox1.Clear()  TextBox2.Clear()  Me.Hide()  End Sub  End Class |
| Main menu form |
| Public Class MainMenu  Private Sub s\_Click(sender As Object, e As EventArgs) Handles s.Click  MainPage.Show()  Me.Hide()  End Sub  Private Sub Viewpatients\_Click(sender As Object, e As EventArgs) Handles Viewpatients.Click  Patients.Show()  Me.Hide()  End Sub  Private Sub GenerateOperatingList\_Click(sender As Object, e As EventArgs) Handles GenerateOperatingList.Click  WIP.Show()  End Sub  Private Sub GenerateClinicTemplate\_Click(sender As Object, e As EventArgs) Handles GenerateClinicTemplate.Click  WIP.Show()  End Sub  Private Sub Stats\_Click(sender As Object, e As EventArgs) Handles Stats.Click  WIP.Show()  End Sub  Private Sub MainMenu\_FormClosing(ByVal sender As Object, ByVal e As System.Windows.Forms.FormClosingEventArgs) Handles Me.FormClosing  If MsgBox("Are you sure you wish to quit form?", MsgBoxStyle.YesNo, "Attention") = MsgBoxResult.No Then  e.Cancel = True  Else  MainPage.Close()  End If  End Sub  End Class |
| Patients form |
| Imports System.IO  Public Class Patients  Private Sub BtnBack\_Click(sender As Object, e As EventArgs) Handles BtnBack.Click 'Back  MainMenu.Show()  Me.Hide()  End Sub  Private Sub BtnAddPatient\_Click(sender As Object, e As EventArgs) Handles BtnAddPatient.Click 'Add patient  AddPatient.Show()  End Sub  Private Sub BtnDischargePatient\_Click(sender As Object, e As EventArgs) Handles BtnDischargePatient.Click 'Discharge patient  Try  Dim n As Integer = GridPatients.SelectedRows(0).Index  Name = GridPatients.Rows(n).Cells("PatientName").Value  If Name <> "" Then  DischargePatient.Show()  Else  MsgBox("You have selected an empty record", MsgBoxStyle.OkOnly, "Error")  End If  Catch ex As Exception  MsgBox("Please select patient", MsgBoxStyle.OkOnly, "Error")  End Try  End Sub  Private Sub BtnDeletePatient\_Click(sender As Object, e As EventArgs) Handles BtnDeletePatient.Click 'delete patient  Try  Dim n As Integer = GridPatients.SelectedRows(0).Index  GridPatients.Rows.RemoveAt(n)  Catch ex As Exception  MsgBox("Select patient", MsgBoxStyle.OkOnly, "Error")  End Try  End Sub  Private Sub BtnViewDis\_Click(sender As Object, e As EventArgs) Handles BtnViewDis.Click 'not in prototype  WIP.Show()  End Sub  Private Sub BtnContact\_Click(sender As Object, e As EventArgs) Handles BtnContact.Click 'not in prototype  WIP.Show()  End Sub  Private Sub Patients\_FormClosing(ByVal sender As Object, ByVal e As System.Windows.Forms.FormClosingEventArgs) Handles Me.FormClosing  If MsgBox("Are you sure you wish to quit form?", MsgBoxStyle.YesNo, "Attention") = MsgBoxResult.No Then  e.Cancel = True  Else  MainPage.Close()  End If  End Sub  Private Sub BtnExport\_Click(sender As Object, e As EventArgs) Handles BtnExport.Click  Dim saving As New SaveFileDialog()  saving.FileName = "Patients table"  saving.Filter = "CSV File | \*.csv"  'MessageBox.Show("Point")  If saving.ShowDialog() = DialogResult.OK Then  SavePatients(IO.Path.Combine(Application.StartupPath, saving.FileName))  End If  End Sub  Private Sub BtnSave\_Click(sender As Object, e As EventArgs) Handles BtnSave.Click  Dim FileName As String = "Patients table.csv"  SavePatients(IO.Path.Combine(Application.StartupPath, FileName))  End Sub  Private Sub SavePatients(location As String)  If Dir(location) <> "" Then  Kill(location)  End If  Using s As StreamWriter = New StreamWriter(File.OpenWrite(location))  For Each row As DataGridViewRow In GridPatients.Rows  Dim rowData As New List(Of String)  For Each column As DataGridViewColumn In GridPatients.Columns  rowData.Add(Convert.ToString(row.Cells(column.Name).Value))  Next  s.WriteLine(String.Join(",", rowData))  Next  s.Close()  End Using  End Sub  Private Sub Patients\_Load(sender As Object, e As EventArgs) Handles MyBase.Load  Dim ReadFile As String = IO.Path.Combine(Application.StartupPath, "Patients table.csv")  Dim LoadData As String = ""  Dim SplitAt() As String  Dim ToContinue As Boolean = True 'To Continue ensures the loop stops when the reader reaches the end of the file  'Checking file exists - if it doesnt nothing is loaded to avoid error etc  If File.Exists(ReadFile) = True Then  Using objReader As New StreamReader(ReadFile)  If objReader.Peek() = -1 Then  ToContinue = False  End If  Do While ToContinue  'Reading the data  LoadData = objReader.ReadLine()  SplitAt = Split(LoadData, ",")  GridPatients.Rows.Add(SplitAt)  If objReader.Peek() = -1 Then  ToContinue = False  End If  Loop  End Using  End If  End Sub  Private Sub BtnViewPatient\_Click(sender As Object, e As EventArgs) Handles BtnViewPatient.Click  Try  Dim n As Integer = GridPatients.SelectedRows(0).Index  SelectedPatient.Show()  Catch ex As Exception  MsgBox("Select patient", MsgBoxStyle.OkOnly, "Error")  End Try  End Sub  End Class |
| View patient |
| Imports System.Windows.Forms.VisualStyles.VisualStyleElement  Public Class SelectedPatient  Private Sub BtnBack\_Click(sender As Object, e As EventArgs) Handles BtnBack.Click  Me.Close()  End Sub  Private Sub SelectedPatient\_Load(sender As Object, e As EventArgs) Handles MyBase.Load  Dim n As Integer = Patients.GridPatients.SelectedRows(0).Index  'Getting Data for patient  LblName.Text = Patients.GridPatients.Rows(n).Cells("PatientName").Value  TxtSex.Text = Patients.GridPatients.Rows(n).Cells("Sex").Value  TxtHeight.Text = Patients.GridPatients.Rows(n).Cells("Height").Value  TxtWeight.Text = Patients.GridPatients.Rows(n).Cells("Weight").Value  TxtNotes.Text = Patients.GridPatients.Rows(n).Cells("Other").Value  TxtEthnicity.Text = Patients.GridPatients.Rows(n).Cells("Ethnicity").Value  TxtMedicalHist.Text = Patients.GridPatients.Rows(n).Cells("History").Value  DtpDOB.Value = Patients.GridPatients.Rows(n).Cells("PatientDOB").Value  End Sub  Private Sub BtnSave\_Click(sender As Object, e As EventArgs) Handles BtnSave.Click  Dim n As Integer = Patients.GridPatients.SelectedRows(0).Index  'Applying the changes to the datagrid  Patients.GridPatients.Rows(n).Cells("PatientName").Value = LblName.Text  Patients.GridPatients.Rows(n).Cells("Sex").Value = TxtSex.Text  Patients.GridPatients.Rows(n).Cells("Height").Value = TxtHeight.Text  Patients.GridPatients.Rows(n).Cells("Weight").Value = TxtWeight.Text  Patients.GridPatients.Rows(n).Cells("Other").Value = TxtNotes.Text  Dim timeDif As TimeSpan = Now - DtpDOB.Value  Dim Age As Integer = Int((timeDif.TotalSeconds) / 31536000)  Patients.GridPatients.Rows(n).Cells("Age").Value = Age  Patients.GridPatients.Rows(n).Cells("Ethnicity").Value = TxtEthnicity.Text  Patients.GridPatients.Rows(n).Cells("History").Value = TxtMedicalHist.Text  Patients.GridPatients.Rows(n).Cells("PatientDOB").Value = DtpDOB.Value  End Sub  End Class |
| Add paiteint menu |
| Public Class AddPatient  Private Sub BtnExecute\_Click(sender As Object, e As EventArgs) Handles BtnExecute.Click  Dim Name As String  Dim Age As Integer  Dim Height As Double  Dim Weight As Double  Dim DOB As Date  Dim History As String  Dim GPName As String  Dim GPEmail As String  Dim Ethnicity As String  Dim OtherInfo As String  Dim Sex As String  Try  'taking data from patient input  Name = TxtName.Text  Height = CDbl(TxtHeight.Text)  Weight = CDbl(TxtWeight.Text)  DOB = DateTimePicker2.Text  History = TxtHistory.Text  GPName = TxtGPname.Text  GPEmail = TxtGPEmail.Text  Ethnicity = CmbEthnicity.Text  OtherInfo = TxtOther.Text  'assigning sex variable  If RdMale.Checked Then  Sex = "Male"  ElseIf RdFemale.Checked Then  Sex = "Female"  Else  Sex = "Other"  End If  'calculating age from DOB  Dim timeDif As TimeSpan = Now - DOB  Age = Int((timeDif.TotalSeconds) / 31536000)  LblDisplay.Text = "Patient added"  'adds patient to data grid  Patients.GridPatients.Rows.Add(Name, Sex, Age, Height, Weight, DOB, History, GPName, GPEmail, Ethnicity, OtherInfo)  '[Add to file]  Me.Close()  Catch ex As Exception  MsgBox("Please fill all fields or check data entered is appropriate", MsgBoxStyle.OkOnly, "Error")  End Try  End Sub  Private Sub Cancel\_Click(sender As Object, e As EventArgs) Handles Cancel.Click  Me.Close() 'exits program  End Sub  End Class |
| Discharge patient |
| Public Class DischargePatient  Private Sub DischargePatient\_Load(sender As Object, e As EventArgs) Handles MyBase.Load  Dim n As Integer = Patients.GridPatients.SelectedRows(0).Index  Dim Name As String  'takes patient name and puts it on the label in the discahrge form  Name = Patients.GridPatients.Rows(n).Cells("PatientName").Value  Firstname.Text = Name  End Sub  Private Sub BtnDischarge\_Click(sender As Object, e As EventArgs) Handles BtnDischarge.Click  Try  Dim n As Integer = Patients.GridPatients.SelectedRows(0).Index  Patients.GridPatients.Rows.RemoveAt(n)  Catch ex As Exception  MsgBox("Error in discharging patient", MsgBoxStyle.OkOnly, "Error")  Me.Hide()  End Try  End Sub  Private Sub Cancel\_Click(sender As Object, e As EventArgs) Handles Cancel.Click  Me.Hide() 'exit button  End Sub  End Class |