Contents

[Changes from prototype 1](#_Toc159840561)

[Changes from post-prototype review and feedback 1](#_Toc159840562)

[Additions to enhance the prototype and meet success criteria 1](#_Toc159840563)

[Validation 1](#_Toc159840564)

[Add patient 1](#_Toc159840565)

[Create account 2](#_Toc159840566)

[Edit patient (in view selected patient) 4](#_Toc159840567)

[Variables, data structures and files 5](#_Toc159840568)

[Files 5](#_Toc159840569)

[Program variables and data structures 5](#_Toc159840570)

[User interface 10](#_Toc159840571)

[Patients.vb 10](#_Toc159840572)

[AddPatient.vb 10](#_Toc159840573)

[SelectedPatient.vb 10](#_Toc159840574)

[AddUser.vb 10](#_Toc159840575)

[MainMenu.vb 10](#_Toc159840576)

[DishchargePatient.vb 10](#_Toc159840577)

[DischargedPatients.vb 10](#_Toc159840578)

[Login.vb 10](#_Toc159840579)

# Changes from prototype

### Changes from post-prototype review and feedback

* Confirmation that the data was saved was added

A screenshot of a computer error message

Description automatically generated

* The delete patient and save patient buttons were made different colours to indicate their function and or importance. The delete button was also moved away from the others so it is less likely to be accidentally pressed (This happened when the third party was testing)

A close-up of a pink rectangle

Description automatically generated 

* Units were added to some of the weight and height field which were missing them

### Additions to enhance the prototype and meet success criteria

* The view discharged patients menu was added and to work with this the discharge patient module was changed so that when the user is removed from the main patients data-grid it is then added to the second data-grid on the discharged patients menu. This involved the creation of a new CSV file to store the discharged patients
* The contact patient GP button was given functionality displaying the GP’s email to the user
* The login system was added. User accounts are created by already logged in accounts (The program starts with a sysadmin account) and stored in a new CSV containing all user information. Entered username is searched for in the CSV using a binary search algorithm and when found the password associated with that username is checked against the user’s entered password
* There is now a new form containing a data-grid and set of text boxes so that a signed in user can view all existing accounts (without passwords) and create new ones
* More validation and error trapping has been added across the program

# Validation

### Add patient

* Data is read and assigned to variables (with predetermined data types) within a try statement. This means that if they are not suitable for that datatype (i.e. a letter in a letter in a field linked to a double variable) a message box can be output telling the user the data is not suitable.
* Validation is not required for the sex field and the ethnicity field since it they use radial button and combo boxes to limit user input to those which are expected.

|  |
| --- |
| Adding patient code |
| 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 |

### Create account

* Data is read and subject to presence checks. If they do not pass, a message box can be output telling the user the data is not suitable.
* The program also checks that the entered email address contains an ‘@’ symbol showing a message box with an error if it does not.

|  |
| --- |
| Creating account code |
| If TxtName.Text <> "" Then  If TxtUsername.Text <> "" Then 'presence check validation  If TxtEmail.Text <> "" And TxtEmail.Text.Contains("@") Then  'Checking both passwords match and exist  If TxtPassword.Text <> "" And TxtPasswordCon.Text = TxtPassword.Text Then  Success = True  'Taking data from text boxes  FullName = TxtName.Text  Username = TxtUsername.Text  Email = TxtEmail.Text  Password = TxtPassword.Text  GridUsers.Rows.Add(FullName, Username, Email, Password)  'Data already loaded into datagrid and will be added so old file is to be deleted  If Dir(Location) <> "" Then  Kill(Location)  End If  'Saving The users table  Using s As StreamWriter = New StreamWriter(File.OpenWrite(Location))  'for each individual user  For Each row As DataGridViewRow In GridUsers.Rows  Dim rowData As New List(Of String)  'adding each peice of data about user  For Each column As DataGridViewColumn In GridUsers.Columns  rowData.Add(Convert.ToString(row.Cells(column.Name).Value))  Next  s.WriteLine(String.Join(",", rowData))  Next  s.Close()  End Using  End If  End If  End If  End If  If Not (Success) Then  'User entered incorrect data and should be notified  MsgBox("Please fill all fields or check data entered is appropriate", MsgBoxStyle.OkOnly, "Error")  End If |

### Edit patient (in view selected patient)

* Data is read and casted to their data type within a try statement. This means that if they are not suitable for that datatype (i.e. a letter in a letter in a field linked to a double variable) a message box can be output telling the user the data is not suitable.

|  |
| --- |
| Editing patient record code |
| 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  Try  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 = CDbl(TxtHeight.Text)  Patients.GridPatients.Rows(n).Cells("Weight").Value = CDbl(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  Catch ex As Exception  MsgBox("Please fill all fields or check data entered is appropriate", MsgBoxStyle.OkOnly, "Error")  End Try  End Sub |

# Variables, data structures and files

### Files

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| File name | File type | File location | File purpose | Layout |
| Discharged Patients table.csv | csv | Same directory as exe | Stores all data on the patients that have been discharged - Name, Sex, height, weight, date of birth(used for age), medical history, GP name, GP email, ethnicity and a section for other important information) | Name, Sex, height, weight, date of birth(used for age), medical history, GP name, GP email, ethnicity and a section for other important information |
| Patients table.csv | csv | Same directory as exe | Stores all patient data - Name, Sex, height, weight, date of birth(used for age), medical history, GP name, GP email, ethnicity and a section for other important information) | Name, Sex, height, weight, date of birth(used for age), medical history, GP name, GP email, ethnicity and a section for other important information |
| Users table.csv | csv | Same directory as exe | Stores information and login for each user (Doctors etc) – Name, email, username, password | Name, email, username, password |

### Program variables and data structures

#### Login.vb

|  |  |  |  |
| --- | --- | --- | --- |
| Variable/ data structure name | Type | Scope | Purpose |
| ReadFile | String | BtnLogin\_Click subroutine | Stores the path of the file to be accessed during the BtnLogin\_Click  subroutine |
| LoadData | String | BtnLogin\_Click subroutine | Used to hold current item in record when reading a line in a CSV |
| SplitAt() | String array | BtnLogin\_Click subroutine | Stores the current record being read |
| ToContinue | Boolean | BtnLogin\_Click subroutine | A Boolean that controls whether to continue reading the file |
| Users | List of string arrays | BtnLogin\_Click subroutine | A data structure used to contain all the data in the Users CSV so it can be searched and sorted without the use of a data-grid |
| UserUsername | String | BtnLogin\_Click subroutine | String to hold the user entered username |
| UserPassword | String | BtnLogin\_Click subroutine | String to hold the user entered password |
| Password | String | BtnLogin\_Click subroutine | String to hold the password associated with the username entered by the user |
| N | integer | BubbleSort function | Hold the length of the Users data structure |
| Swapped | Boolean | BubbleSort function | Stores whether there has been a swap in the current pass of the bubble sort so the program knows when it is sorted |
| tem | String array | BubbleSort function | Temporary data storage used for swapping value |
| MidPoint | Integer | BinarySearch function | Stores the value of the midpoint pointer used in the binary search algorithm |

#### AddUser.vb

|  |  |  |  |
| --- | --- | --- | --- |
| Variable/ data structure name | Type | Scope | Purpose |
| FullName | String | BtnExecute\_Click subroutine | Holds the name entered by the user |
| Username | String | BtnExecute\_Click subroutine | Holds the username entered by the user |
| Email | String | BtnExecute\_Click subroutine | Holds the email entered by the user |
| Password | String | BtnExecute\_Click subroutine | Holds the password entered by the user |
| Success | Boolean | BtnExecute\_Click subroutine | Used to indicate whether entered data passed validation |
| Location | String | BtnExecute\_Click subroutine | Stores the location of the Users table.csv file being written to |
| rowData | List of strings | BtnExecute\_Click subroutine | Data structure containing the data for a single row – this is then written to the CSV file |
| ReadFile | String | AddUser\_Load  subroutine | Stores the path of the file to be accessed during the AddUser\_Load  subroutine |
| LoadData | String | AddUser\_Load  subroutine | Used to hold current item in record when reading a line in a CSV |
| SplitAt() | String array | AddUser\_Load  subroutine | Stores the current record being read |
| ToContinue | Boolean | AddUser\_Load  subroutine | A Boolean that controls whether to continue reading the file |

#### DischargePatient.vb

|  |  |  |  |
| --- | --- | --- | --- |
| Variable/ data structure name | Type | Scope | Purpose |
| n | Integer | DischargePatient\_Load  subroutine | Holds the index of the currently selected record |
| Name | string | DischargePatient\_Load  subroutine | Holds the full name of the currently selected patient |
| Name | String | BtnDischarge\_Click subroutine | Holds the full name of the patient to be discharged |
| Age | Integer | BtnDischarge\_Click subroutine | Holds the age of the patient to be discharged |
| Height | Double | BtnDischarge\_Click subroutine | Holds the heigh of the patient to be discharged |
| Weight | Double | BtnDischarge\_Click subroutine | Holds the weight of the patient to be discharged |
| DOB | DateTime | BtnDischarge\_Click subroutine | Holds the date of birth of the patient to be discharged |
| History | String | BtnDischarge\_Click subroutine | Holds a brief medical history of the patient to be discharged |
| GPName | String | BtnDischarge\_Click subroutine | Holds the name of the GP of the patient to be discharged |
| GPEmail | String | BtnDischarge\_Click subroutine | Holds the Email of the GP of the patient to be discharged |
| Ethnicity | String | BtnDischarge\_Click subroutine | Holds the ethnicity of the patient to be discharged |
| OtherInfo | String | BtnDischarge\_Click subroutine | Holds any extra information of the patient to be discharged |
| Sex | String | BtnDischarge\_Click subroutine | Holds the sex of the patient to be discharged |
| N | Integer | BtnDischarge\_Click subroutine | Holds the index of the currently selected record |
| rowData | List of strings | BtnDischarge\_Click subroutine | Data structure containing the data for a single row – this is then written to the CSV file |

#### DischargedPatients.vb

|  |  |  |  |
| --- | --- | --- | --- |
| Variable/ data structure name | Type | Scope | Purpose |
| ReadFile | String | DischargedPatients\_Load subroutine | Stores the path of the file to be accessed during the DischargedPatients\_Load subroutine |
| LoadData | String | DischargedPatients\_Load subroutine | Used to hold current item in record when reading a line in a CSV |
| SplitAt() | String array | DischargedPatients\_Load subroutine | Stores the current record being read |
| ToContinue | Boolean | DischargedPatients\_Load subroutine | A Boolean that controls whether to continue reading the file |
| Location | String | BtnExport\_Click subroutine | Stores the location of the Users table.csv file being written to |

#### Patients.vb

|  |  |  |  |
| --- | --- | --- | --- |
| Variable/ data structure name | Type | Scope | Purpose |
| N | Integer | Used multiple times across different subroutines (different variables not global) | Holds the index of the currently selected patient record |
| FileName | String | BtnSave\_Click subroutine | Holds the name of the file being written to so it can be passed into the saving subroutine |
| rowData | List of strings | SavePatients subroutine | Stores the location of the Users table.csv file being written to |
| ReadFile | String | Patients\_Load subroutine | Stores the path of the file to be accessed during the Patients\_Load subroutine |
| LoadData | String | Patients\_Load subroutine | Used to hold current item in record when reading a line in a CSV |
| SplitAt() | String array | Patients\_Load subroutine | Stores the current record being read |
| ToContinue | Boolean | Patients\_Load subroutine | A Boolean that controls whether to continue reading the file |

#### SelectedPatient.vb

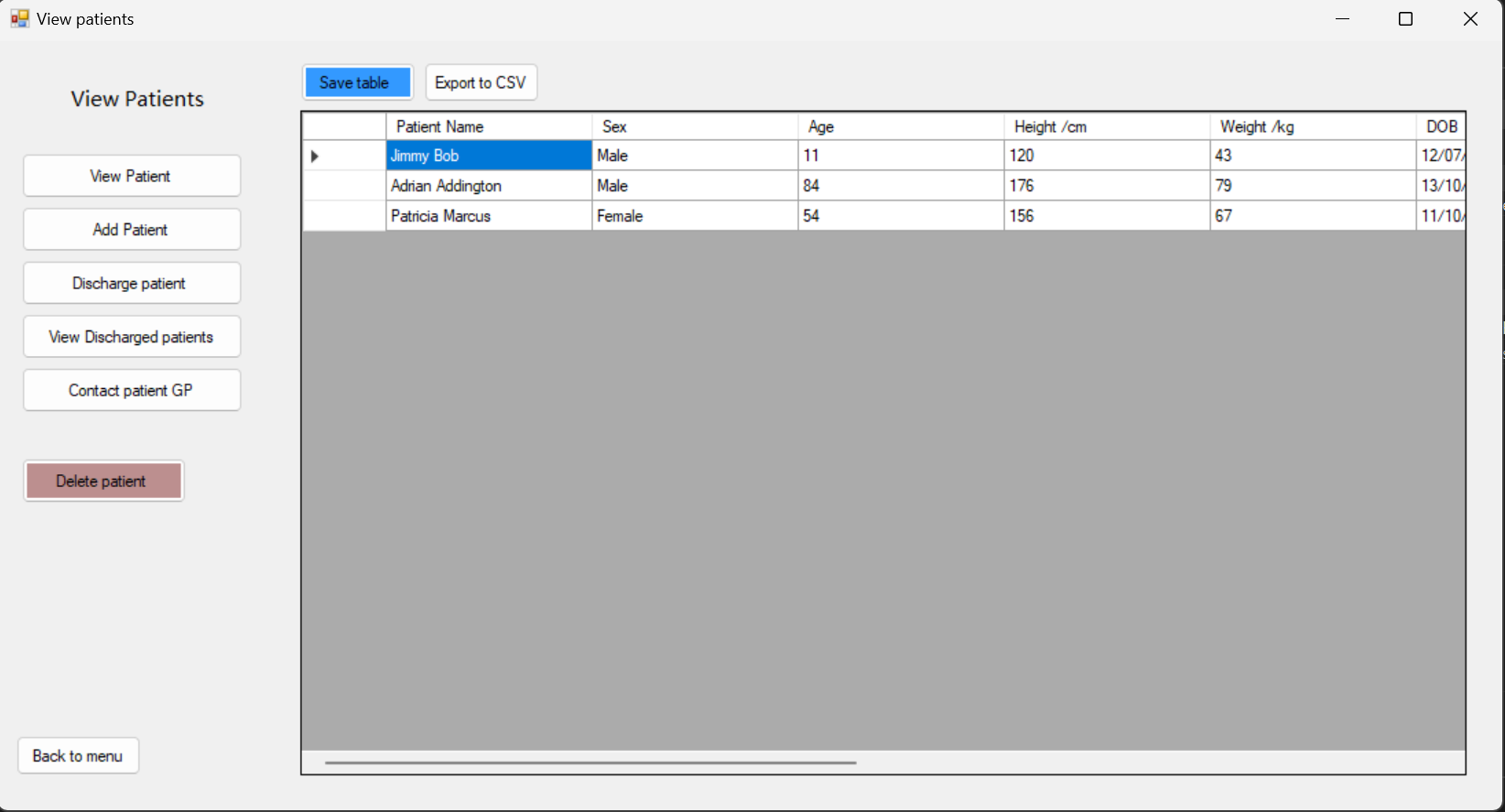
|  |  |  |  |
| --- | --- | --- | --- |
| Variable/ data structure name | Type | Scope | Purpose |
| N | Integer | Used multiple times across different subroutines (different variables not global) | Holds the index of the currently selected patient record |
| timeDif | TimeSpan | BtnSave\_Click subroutine | Used to hold the difference in time between now and the entered date of birth |
| Age | Integer | BtnSave\_Click subroutine | Used to hold the new calculated age of the selected patient |

#### AddPatient.vb

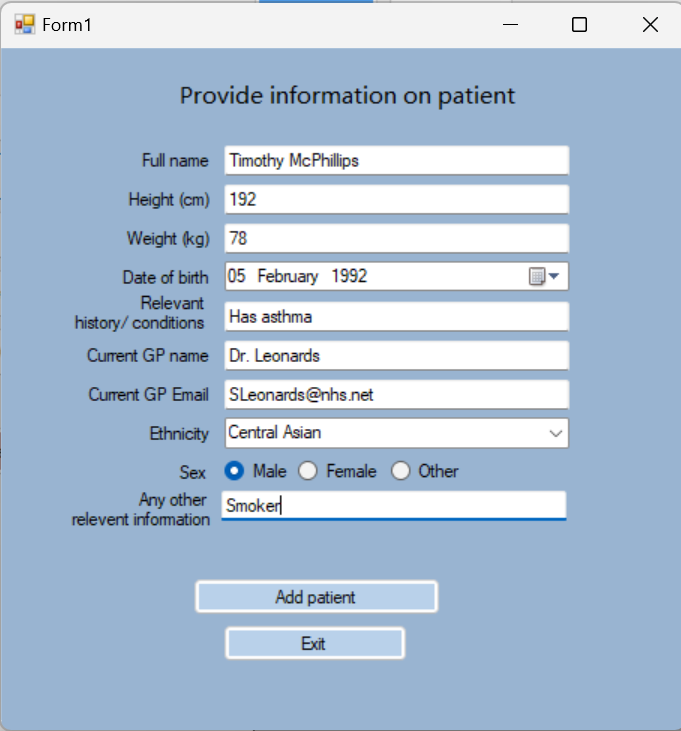
|  |  |  |  |
| --- | --- | --- | --- |
| Variable/ data structure name | Type | Scope | Purpose |
| Name | String | BtnExecute\_Click subroutine | Holds the full name of the patient to be added |
| Age | Integer | BtnExecute\_Click subroutine | Holds the age of the patient to be Added |
| Height | Double | BtnExecute\_Click subroutine | Holds the heigh of the patient to be added |
| Weight | Double | BtnExecute\_Click subroutine | Holds the weight of the patient to be added |
| DOB | DateTime | BtnDischarge\_Click subroutine | Holds the date of birth of the patient to be added |
| History | String | BtnExecute\_Click subroutine | Holds a brief medical history of the patient to be added |
| GPName | String | BtnExecute\_Click subroutine | Holds the name of the GP of the patient to be added |
| GPEmail | String | BtnExecute\_Click subroutine | Holds the Email of the GP of the patient to be added |
| Ethnicity | String | BtnExecute\_Click subroutine | Holds the ethnicity of the patient to be discharged |
| OtherInfo | String | BtnExecute\_Click subroutine | Holds any extra information of the patient to be added |
| Sex | String | BtnExecute\_Click subroutine | Holds the sex of the patient to be added |
| timeDif | TimeSpan | BtnExecute\_Click subroutine | Used to hold the difference in time between now and the entered date of birth |

# User interface

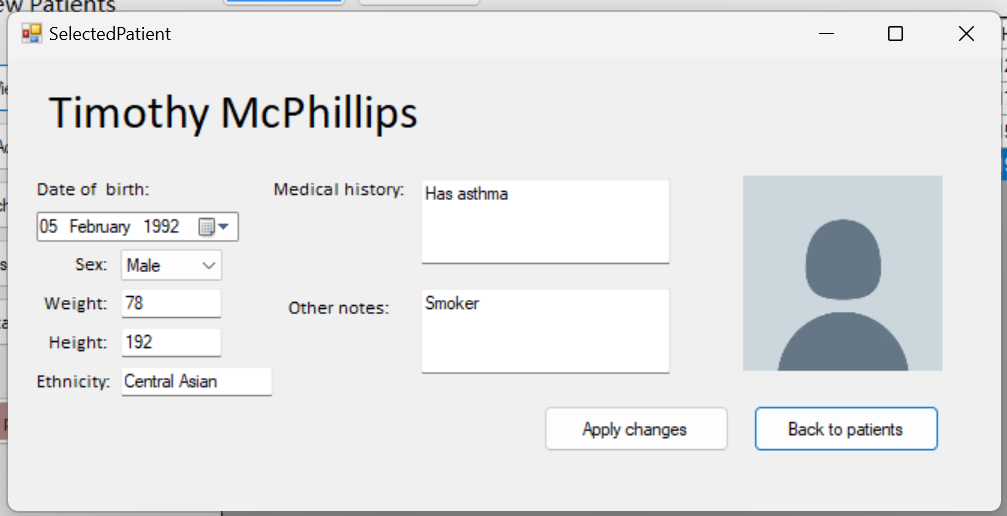
### Patients.vb



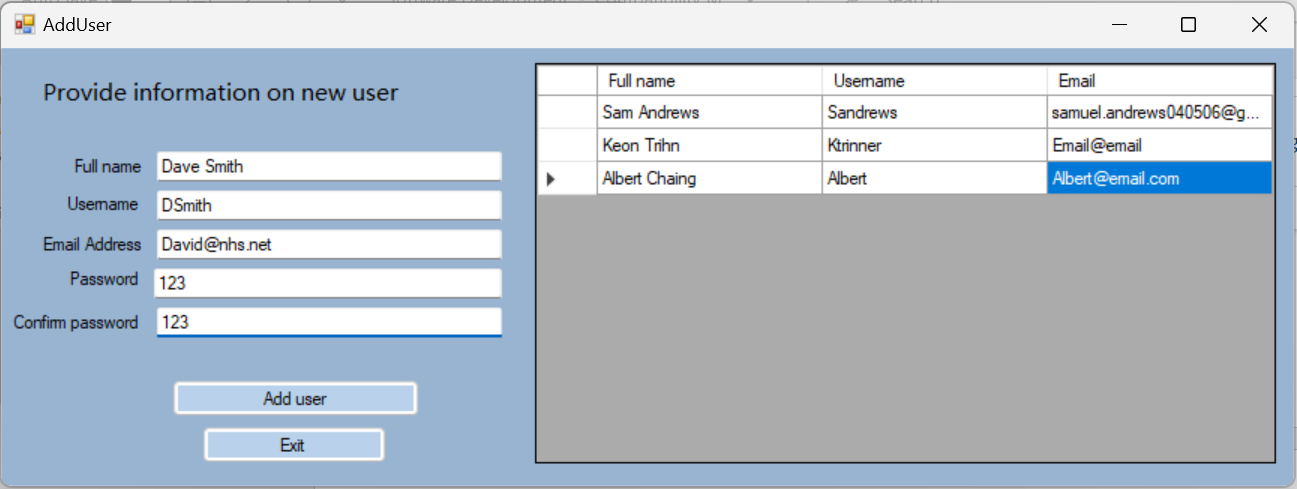
### AddPatient.vb



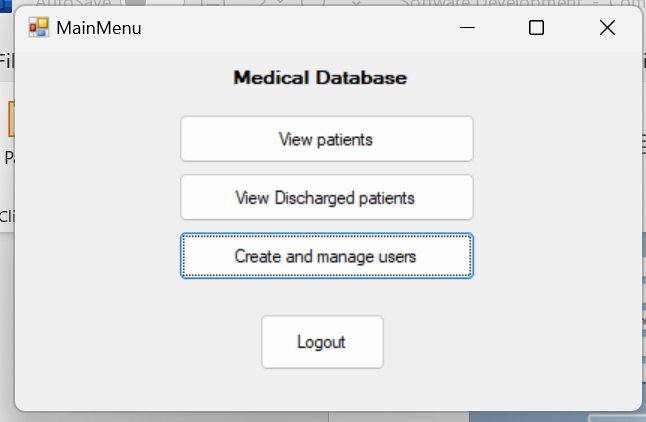
### SelectedPatient.vb



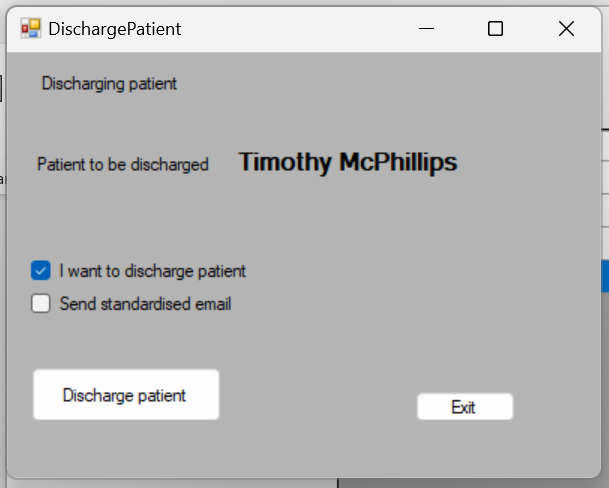
### AddUser.vb



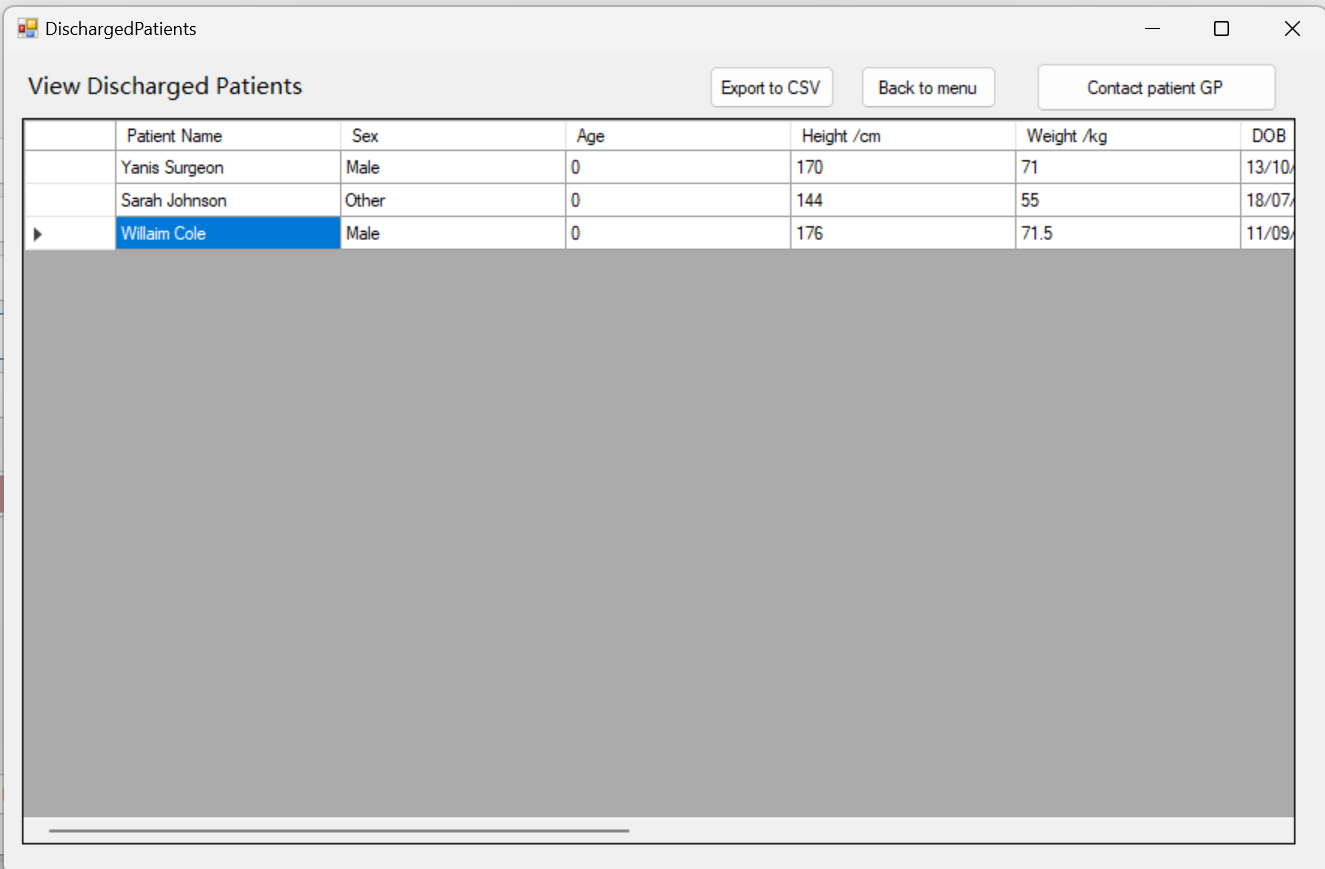
### MainMenu.vb



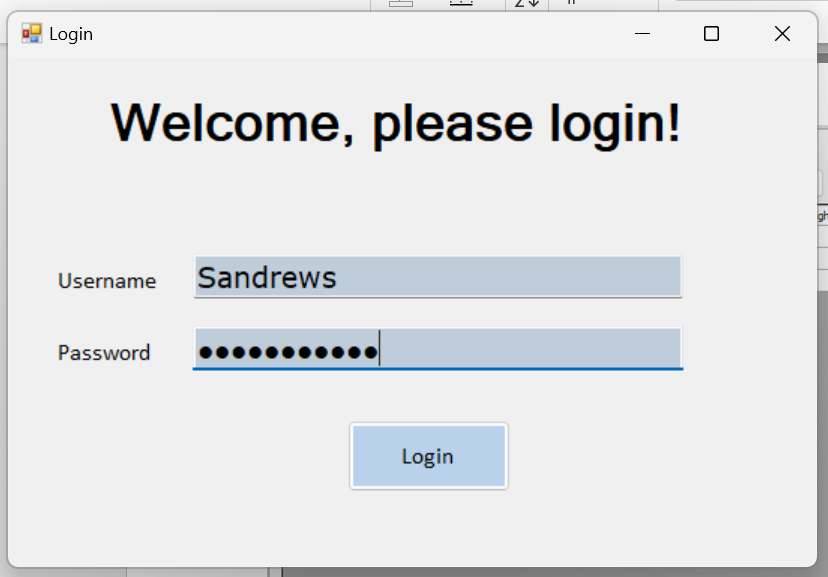
### DishchargePatient.vb



### DischargedPatients.vb



### Login.vb



# Full source code

### Patients.vb

|  |
| --- |
| Imports System.IO  ''' <summary>  ''' This form shows a datagrid with all patients on it  ''' It has a variaty of differnet ways in which the user can manage the patients  ''' </summary>  Public Class Patients  Private Sub BtnBack\_Click(sender As Object, e As EventArgs) Handles BtnBack.Click  MainMenu.Show()  Me.Hide()  End Sub  Private Sub BtnAddPatient\_Click(sender As Object, e As EventArgs) Handles BtnAddPatient.Click  AddPatient.Show()  End Sub  Private Sub BtnDischargePatient\_Click(sender As Object, e As EventArgs) Handles BtnDischargePatient.Click  Try  'Error trapping to ensure the user has selected a viable record to discharge  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  Try  'Error trapping to ensure the user has selected a viable record to delete  Dim n As Integer = GridPatients.SelectedRows(0).Index  GridPatients.Rows.RemoveAt(n) 'Deleting  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  DischargedPatients.Show()  End Sub  Private Sub BtnContact\_Click(sender As Object, e As EventArgs) Handles BtnContact.Click  Try  'Error trapping to ensure the user has selected a viable record  Dim n As Integer = GridPatients.SelectedRows(0).Index  MsgBox(GridPatients.Rows(n).Cells("GPEmail").Value, MsgBoxStyle.OkOnly, "Patients GP email") 'displaying GP email  Catch ex As Exception  MsgBox("No patient selected", MsgBoxStyle.OkOnly, "Error")  End Try  End Sub  Private Sub Patients\_FormClosing(ByVal sender As Object, ByVal e As System.Windows.Forms.FormClosingEventArgs) Handles Me.FormClosing  'Other forms are dependant on this form so if it is closed without using the given buttons the user is asked and then the whole program closes to avoid error  If MsgBox("Are you sure you wish to quit form?", MsgBoxStyle.YesNo, "Attention") = MsgBoxResult.No Then  e.Cancel = True  Else  Login.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"  'Saving patinet using a save dialog (exporting)  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  'Saving patinets to program files (without dialog)  Dim FileName As String = "Patients table.csv"  SavePatients(IO.Path.Combine(Application.StartupPath, FileName))  End Sub  Private Sub SavePatients(location As String)  'Subroutine to save patients in either way  'Data already loaded into datagrid And will be added so old file Is to be deleted  If Dir(location) <> "" Then  Kill(location)  End If  Using s As StreamWriter = New StreamWriter(File.OpenWrite(location))  'for each record  For Each row As DataGridViewRow In GridPatients.Rows  Dim rowData As New List(Of String)  'adding data for each record  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  MsgBox("The table has been saved and will load next time the program is opened.", MsgBoxStyle.OkOnly, "Data saved")  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)  'Checking if there is another record after  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)  'Checking if there is another record after  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  'error trapping to ensure a patient is selected  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 |

### AddPatient.vb

|  |
| --- |
| ''' <summary>  ''' Basic form used to take data input from user and add patient  ''' </summary>  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 |

### SelectedPatient.vb

|  |
| --- |
| Imports System.Windows.Forms.VisualStyles.VisualStyleElement  ''' <summary>  ''' A menu used to edit and show a specific Patient  ''' </summary>  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  Try  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 = CDbl(TxtHeight.Text)  Patients.GridPatients.Rows(n).Cells("Weight").Value = CDbl(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  Catch ex As Exception  MsgBox("Please fill all fields or check data entered is appropriate", MsgBoxStyle.OkOnly, "Error")  End Try  End Sub  End Class |

### AddUser.vb

|  |
| --- |
| Imports System.IO  Imports System.Windows.Forms.VisualStyles.VisualStyleElement  ''' <summary>  ''' Form to add user  ''' takes users details and adds them to a CSV so they can login  ''' </summary>  '''  Public Class AddUser  Private Sub BtnExecute\_Click(sender As Object, e As EventArgs) Handles BtnExecute.Click  'Adding a user  Dim FullName As String  Dim Username As String  Dim Email As String  Dim Password As String  Dim Success As Boolean = False  Dim Location As String = IO.Path.Combine(Application.StartupPath, "Users table.csv") 'file that this form will use  If TxtName.Text <> "" Then  If TxtUsername.Text <> "" Then 'presence check validation  If TxtEmail.Text <> "" And TxtEmail.Text.Contains("@") Then  'Checking both passwords match and exist  If TxtPassword.Text <> "" And TxtPasswordCon.Text = TxtPassword.Text Then  Success = True  'Taking data from text boxes  FullName = TxtName.Text  Username = TxtUsername.Text  Email = TxtEmail.Text  Password = TxtPassword.Text  GridUsers.Rows.Add(FullName, Username, Email, Password)  'Data already loaded into datagrid and will be added so old file is to be deleted  If Dir(Location) <> "" Then  Kill(Location)  End If  'Saving The users table  Using s As StreamWriter = New StreamWriter(File.OpenWrite(Location))  'for each individual user  For Each row As DataGridViewRow In GridUsers.Rows  Dim rowData As New List(Of String)  'adding each peice of data about user  For Each column As DataGridViewColumn In GridUsers.Columns  rowData.Add(Convert.ToString(row.Cells(column.Name).Value))  Next  s.WriteLine(String.Join(",", rowData))  Next  s.Close()  End Using  End If  End If  End If  End If  If Not (Success) Then  'User entered incorrect data and should be notified  MsgBox("Please fill all fields or check data entered is appropriate", MsgBoxStyle.OkOnly, "Error")  End If  End Sub  Private Sub Cancel\_Click(sender As Object, e As EventArgs) Handles Cancel.Click  Me.Close()  End Sub  Private Sub AddUser\_Load(sender As Object, e As EventArgs) Handles MyBase.Load  Dim ReadFile As String = IO.Path.Combine(Application.StartupPath, "Users 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 to datagrid  LoadData = objReader.ReadLine()  SplitAt = Split(LoadData, ",")  GridUsers.Rows.Add(SplitAt)  'Checking if there is another record  If objReader.Peek() = -1 Then  ToContinue = False  End If  Loop  End Using  End If  End Sub  End Class |

### MainMenu.vb

|  |
| --- |
| Imports System.IO  Imports System.Windows.Forms.VisualStyles.VisualStyleElement  ''' <summary>  ''' Form to add user  ''' takes users details and adds them to a CSV so they can login  ''' </summary>  '''  Public Class AddUser  Private Sub BtnExecute\_Click(sender As Object, e As EventArgs) Handles BtnExecute.Click  'Adding a user  Dim FullName As String  Dim Username As String  Dim Email As String  Dim Password As String  Dim Success As Boolean = False  Dim Location As String = IO.Path.Combine(Application.StartupPath, "Users table.csv") 'file that this form will use  If TxtName.Text <> "" Then  If TxtUsername.Text <> "" Then 'presence check validation  If TxtEmail.Text <> "" And TxtEmail.Text.Contains("@") Then  'Checking both passwords match and exist  If TxtPassword.Text <> "" And TxtPasswordCon.Text = TxtPassword.Text Then  Success = True  'Taking data from text boxes  FullName = TxtName.Text  Username = TxtUsername.Text  Email = TxtEmail.Text  Password = TxtPassword.Text  GridUsers.Rows.Add(FullName, Username, Email, Password)  'Data already loaded into datagrid and will be added so old file is to be deleted  If Dir(Location) <> "" Then  Kill(Location)  End If  'Saving The users table  Using s As StreamWriter = New StreamWriter(File.OpenWrite(Location))  'for each individual user  For Each row As DataGridViewRow In GridUsers.Rows  Dim rowData As New List(Of String)  'adding each peice of data about user  For Each column As DataGridViewColumn In GridUsers.Columns  rowData.Add(Convert.ToString(row.Cells(column.Name).Value))  Next  s.WriteLine(String.Join(",", rowData))  Next  s.Close()  End Using  End If  End If  End If  End If  If Not (Success) Then  'User entered incorrect data and should be notified  MsgBox("Please fill all fields or check data entered is appropriate", MsgBoxStyle.OkOnly, "Error")  End If  End Sub  Private Sub Cancel\_Click(sender As Object, e As EventArgs) Handles Cancel.Click  Me.Close()  End Sub  Private Sub AddUser\_Load(sender As Object, e As EventArgs) Handles MyBase.Load  Dim ReadFile As String = IO.Path.Combine(Application.StartupPath, "Users 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 to datagrid  LoadData = objReader.ReadLine()  SplitAt = Split(LoadData, ",")  GridUsers.Rows.Add(SplitAt)  'Checking if there is another record  If objReader.Peek() = -1 Then  ToContinue = False  End If  Loop  End Using  End If  End Sub  End Class |

### DishchargePatient.vb

|  |
| --- |
| Imports System.IO  Public Class DischargePatient  ''' <summary>  ''' A form to discharge patients  ''' Moves selected patinet record from main patients csv to discharged patients CSV  ''' </summary>  Private Sub DischargePatient\_Load(sender As Object, e As EventArgs) Handles MyBase.Load  Dim n As Integer = Patients.GridPatients.SelectedRows(0).Index 'validation already done before this is called so not required here  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 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  'File this form will be using  Dim Location As String = IO.Path.Combine(Application.StartupPath, "Discharged Patients table.csv")  DischargedPatients.Show()  'Defines selected patinet  Dim n As Integer = Patients.GridPatients.SelectedRows(0).Index  'Takes data from record on original table  Name = Patients.GridPatients.Rows(n).Cells("PatientName").Value  Sex = Patients.GridPatients.Rows(n).Cells("Sex").Value  Height = Patients.GridPatients.Rows(n).Cells("Height").Value  Weight = Patients.GridPatients.Rows(n).Cells("Weight").Value  OtherInfo = Patients.GridPatients.Rows(n).Cells("Other").Value  Ethnicity = Patients.GridPatients.Rows(n).Cells("Ethnicity").Value  History = Patients.GridPatients.Rows(n).Cells("History").Value  DOB = Patients.GridPatients.Rows(n).Cells("PatientDOB").Value  GPName = Patients.GridPatients.Rows(n).Cells("GPName").Value  GPEmail = Patients.GridPatients.Rows(n).Cells("GPEmail").Value  'adds record to discharge table  DischargedPatients.GridDischargedPatients.Rows.Add(Name, Sex, Age, Height, Weight, DOB, History, GPName, GPEmail, Ethnicity, OtherInfo)  'Data already loaded into datagrid and will be added so old file is to be deleted  If Dir(Location) <> "" Then  Kill(Location)  End If  'Saving The discharged paitnets table  Using s As StreamWriter = New StreamWriter(File.OpenWrite(Location))  'for each record  For Each row As DataGridViewRow In DischargedPatients.GridDischargedPatients.Rows  Dim rowData As New List(Of String)  'Data for each patient added to datagrid  For Each column As DataGridViewColumn In DischargedPatients.GridDischargedPatients.Columns  rowData.Add(Convert.ToString(row.Cells(column.Name).Value))  Next  s.WriteLine(String.Join(",", rowData))  Next  s.Close()  End Using  'Changing focus to patinets CSV file  Location = IO.Path.Combine(Application.StartupPath, "Patients table.csv")  'Removing old record  Patients.GridPatients.Rows.RemoveAt(n)  'Data already loaded into datagrid And will be added so old file Is to be deleted  If Dir(Location) <> "" Then  Kill(Location)  End If  'saving normal patients table  Using s As StreamWriter = New StreamWriter(File.OpenWrite(Location))  'for each record  For Each row As DataGridViewRow In Patients.GridPatients.Rows  Dim rowData As New List(Of String)  'Data for each patient added to datagrid  For Each column As DataGridViewColumn In Patients.GridPatients.Columns  rowData.Add(Convert.ToString(row.Cells(column.Name).Value))  Next  s.WriteLine(String.Join(",", rowData))  Next  s.Close()  End Using  MsgBox("The patient has been successfully discharged", MsgBoxStyle.OkOnly, "Patient Discharged")  Me.Close()  Catch ex As Exception  MsgBox("Error in discharging patient", MsgBoxStyle.OkOnly, "Error")  Me.Close()  End Try  End Sub  Private Sub Cancel\_Click(sender As Object, e As EventArgs) Handles Cancel.Click  Me.Close() 'exit button  End Sub  End Class |

### DischargedPatients.vb

|  |
| --- |
| Imports System.IO  Public Class DischargedPatients  ''' <summary>  ''' Shows a datagrid where all discharged patients can be viewed  ''' </summary>  '''  Private Sub DischargedPatients\_Load(sender As Object, e As EventArgs) Handles MyBase.Load  Dim ReadFile As String = IO.Path.Combine(Application.StartupPath, "Discharged 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)  'checks if there is a next record  If objReader.Peek() = -1 Then  ToContinue = False  End If  Do While ToContinue  'Reading the data and adding to datagrid  LoadData = objReader.ReadLine()  SplitAt = Split(LoadData, ",")  GridDischargedPatients.Rows.Add(SplitAt)  'checks if there is a next record  If objReader.Peek() = -1 Then  ToContinue = False  End If  Loop  End Using  End If  End Sub  Private Sub BtnExport\_Click(sender As Object, e As EventArgs) Handles BtnExport.Click  'Allows user to save file using a save dialog  Dim Location  Dim saving As New SaveFileDialog()  saving.FileName = "Discharged Patients table"  saving.Filter = "CSV File | \*.csv"  If saving.ShowDialog() = DialogResult.OK Then  Location = IO.Path.Combine(Application.StartupPath, saving.FileName)  Using s As StreamWriter = New StreamWriter(File.OpenWrite(Location))  'for each record  For Each row As DataGridViewRow In GridDischargedPatients.Rows  Dim rowData As New List(Of String)  'adding each item of data for each record  For Each column As DataGridViewColumn In GridDischargedPatients.Columns  rowData.Add(Convert.ToString(row.Cells(column.Name).Value))  Next  s.WriteLine(String.Join(",", rowData))  Next  s.Close()  End Using  End If  End Sub  Private Sub BtnBack\_Click(sender As Object, e As EventArgs) Handles BtnBack.Click  Me.Close()  End Sub  Private Sub BtnContact\_Click(sender As Object, e As EventArgs) Handles BtnContact.Click  Try  'displaying email of selected patinets GP  Dim n As Integer = GridDischargedPatients.SelectedRows(0).Index  MsgBox(GridDischargedPatients.Rows(n).Cells("GPEmail").Value, MsgBoxStyle.OkOnly, "Patients GP email")  Catch ex As Exception  MsgBox("No patient selected", MsgBoxStyle.OkOnly, "Error")  End Try  End Sub  End Class |

### Login.vb

|  |
| --- |
| Imports System.IO  ''' <summary>  ''' Starting form  ''' Login form, checks username and password against User csv file  ''' </summary>  Public Class Login  Private Sub BtnLogin\_Click(sender As Object, e As EventArgs) Handles BtnLogin.Click  Dim ReadFile As String = IO.Path.Combine(Application.StartupPath, "Users table.csv") 'File this form will be working with  Dim LoadData As String = ""  Dim SplitAt() As String  Dim ToContinue As Boolean = True  Dim Users As List(Of String()) = New List(Of String()) 'List of string arrays - each array will contain a User record  'Taking data from textboxes  Dim UserUsername As String = TxtUsername.Text  Dim UserPassword As String = TxtPassword.Text  Dim Password As String  'Taking users from the Users csv file and adding them into the users List of arrays  If File.Exists(ReadFile) = True Then  Dim n As Integer = 0  Using objReader As New StreamReader(ReadFile)  'Checking if there is another record after  If objReader.Peek() = -1 Then  ToContinue = False  End If  Do While ToContinue  'adding user arrays to list  LoadData = objReader.ReadLine()  SplitAt = Split(LoadData, ",")  Users.Add(SplitAt)  'Checking if there is another record after  If objReader.Peek() = -1 Then  ToContinue = False  End If  n += 1  Loop  End Using  End If  Users = BubbleSort(Users)  Password = BinarySearch(Users, UserUsername, ((Users.Count) - 1), 0)  If Password = "" Then  MsgBox("Your username is not recognised. Please contact your administrator to create an account.", MsgBoxStyle.OkOnly, "Incorrect username")  ElseIf UserPassword = Password Then  'Login correct - entering program  MainMenu.Show()  TxtUsername.Clear()  TxtPassword.Clear()  Me.Hide()  Else  MsgBox("Your password is incorrect. Please try again or contact your administrator.", MsgBoxStyle.OkOnly, "Incorrect passwrd")  End If  End Sub  Private Function BubbleSort(Users As List(Of String()))  Dim n As Integer = (Users.Count)  Dim swapped As Boolean = False  Dim Tem As String()  'loop that goes through the list in Users  For i = 0 To (n - 1)  swapped = False  For j = 0 To (n - i - 2)  'swapping  If Users(j)(1).CompareTo(Users(j + 1)(1)) = -1 Then  swapped = True  Tem = Users(j)  Users(j) = Users(j + 1)  Users(j + 1) = Tem  End If  Next j  'When sorted  If Not (swapped) Then  Exit For  End If  Next i  Return Users  End Function  Private Function BinarySearch(Users As List(Of String()), Target As String, Top As Integer, Bottom As Integer)  Dim MidPoint As Integer  MidPoint = (Top + Bottom) \ 2  Try  'When item found  If Target = Users(MidPoint)(1) Then  Return Users(MidPoint)(3)  'item not found  ElseIf Bottom > Top Or Top < Bottom Then  Return ""  'Target item below current midpoint  ElseIf Users(MidPoint)(1).CompareTo(Target) = -1 Then  Return BinarySearch(Users, Target, (MidPoint - 1), Bottom) ''''''''''''''''''''''''''''''''''''''''''Recursion''''''''  'Target item below current midpoint  ElseIf Users(MidPoint)(1).CompareTo(Target) = 1 Then  Return BinarySearch(Users, Target, Top, (MidPoint + 1)) '''''''''''''''''''''''''''''''''''''''''Recursion''''''''  Else  Return ""  End If  Catch ex As Exception  Return ""  End Try  End Function  End Class |