Contents

[Changes from prototype 1](#_Toc159790024)

[Changes from post-prototype review and feedback 1](#_Toc159790025)

[Additions to enhance the prototype and meet success criteria 1](#_Toc159790026)

[Validation 1](#_Toc159790027)

[Add patient 1](#_Toc159790028)

[Create account 2](#_Toc159790029)

[Edit patient (in view selected patient) 4](#_Toc159790030)

[Variables, data structures and files 5](#_Toc159790031)

[Files 5](#_Toc159790032)

[Program variables and data structures 5](#_Toc159790033)

[User interface 5](#_Toc159790034)

# 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 |
| 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) |
| 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) |
| Users table.csv | csv | Same directory as exe | Stores information and login for each user (Doctors etc) – 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 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

#### Patients.vb

|  |  |  |  |
| --- | --- | --- | --- |
| Variable/ data structure name | Type | Scope | Purpose |
|  |  |  |  |
|  |  |  |  |

#### MainMenu.vb

|  |  |  |  |
| --- | --- | --- | --- |
| Variable/ data structure name | Type | Scope | Purpose |
|  |  |  |  |
|  |  |  |  |

#### SelectedPatient.vb

|  |  |  |  |
| --- | --- | --- | --- |
| Variable/ data structure name | Type | Scope | Purpose |
|  |  |  |  |
|  |  |  |  |

#### AddPatient.vb

|  |  |  |  |
| --- | --- | --- | --- |
| Variable/ data structure name | Type | Scope | Purpose |
|  |  |  |  |
|  |  |  |  |

#### Patients.vb

|  |  |  |  |
| --- | --- | --- | --- |
| Variable/ data structure name | Type | Scope | Purpose |
|  |  |  |  |
|  |  |  |  |

# User interface