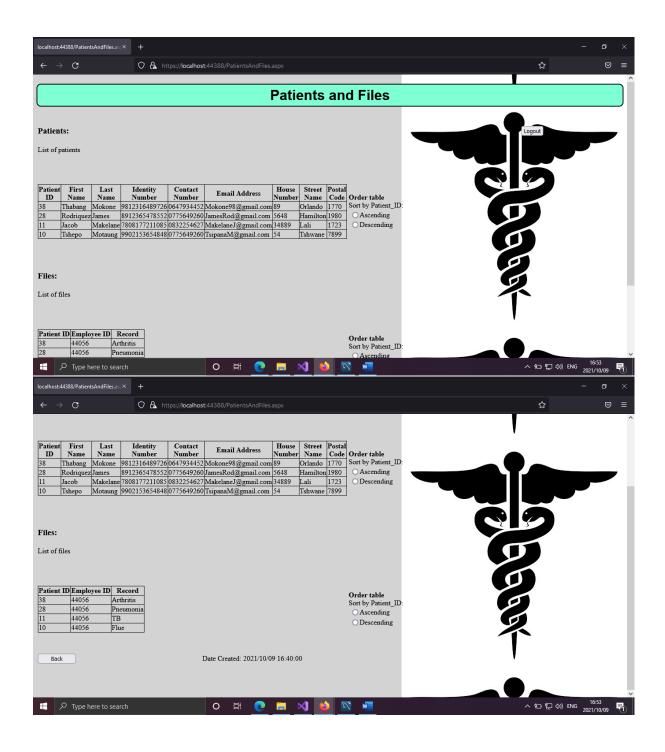# OUTPUT SCREEN OF DIGITAL BOOKING AND FILING SYSTEM

BOPHELONG HOSPITAL

OCTOBER 9, 2021

Screen print of an output screen

Screen 1

Patients and Files

Patients:

List of patients

Patient ID	First Name	Last Name	Identity Number	Contact Number	Email Address	House Number	Street Name	Postal Code
38	Thabang	Mokone	9812316489726	0647934452	Mokone98@gmail.com	89	Orlando	1770
28	Rodriquez	James	8912365478552	0775649260	JamesRod@gmail.com	5648	Hamilton	1980
11	Jacob	Makelane	7808177211085	0832254627	MakelaneJ@gmail.com	34889	Lali	1723
10	Tshepo	Motaung	9902153654848	0775649260	TsipanaM@gmail.com	54	Tshwane	7899

Order table
Sort by Patient_ID:
○ Ascending
○ Descending

Files:

List of files

Patient ID	Employee ID	Record
38	44056	Arthritis
28	44056	Pneumonia

Order table
Sort by Patient_ID:
○ Ascending

Screen 2

Patient ID	First Name	Last Name	Identity Number	Contact Number	Email Address	House Number	Street Name	Postal Code
38	Thabang	Mokone	9812316489726	0647934452	Mokone98@gmail.com	89	Orlando	1770
28	Rodriquez	James	8912365478552	0775649260	JamesRod@gmail.com	5648	Hamilton	1980
11	Jacob	Makelane	7808177211085	0832254627	MakelaneJ@gmail.com	34889	Lali	1723
10	Tshepo	Motaung	9902153654848	0775649260	TsipanaM@gmail.com	54	Tshwane	7899

Order table
Sort by Patient_ID:
○ Ascending
○ Descending

Files:

List of files

Patient ID	Employee ID	Record
38	44056	Arthritis
28	44056	Pneumonia
11	44056	TB
10	44056	Flue

Order table
Sort by Patient_ID:
○ Ascending
○ Descending

Back

Date Created: 2021/10/09 16:40:00

Efficient reuse of code (methods)


```
1 reference
private void createFile()
{
    try
    {
        int patientID = getPatientID();
        con.Open();
        adapter = new SqlDataAdapter();

        command = new SqlCommand($"INSERT INTO [File] (Patient_ID, Employee_ID, Record) VALUES({patientID}, {int.Parse(Ses
        adapter.InsertCommand = command;
        adapter.InsertCommand.ExecuteNonQuery();

        con.Close();

    }
    catch (SqlException ex)
    {
        ScriptManager.RegisterStartupScript(this, this.GetType(), "popup", "alert('Error: An error occurred '" + ex.Messag
    }
}
```

```
1 reference
private int getPatientID()
{
    try
    {
        int patientID = 0;
        con.Open();

        command = new SqlCommand($"SELECT Patient_ID FROM Patient", con);
        dataReader = command.ExecuteReader();

        while (dataReader.Read())
        {
            patientID = int.Parse(dataReader.GetValue(0).ToString());
        }

        con.Close();
        return patientID;



    }
    catch (SqlException ex)
```

Screenshot 1 — AddNewPatient.aspx.cs

```
85        private void addNewPatient()
86        {
87            try
88            {
89
90                con.Open();
91                adapter = new SqlDataAdapter();
92
93                command = new SqlCommand($"INSERT INTO [Patient] (First_name, Last_name, ID_number, Contact_number, Email_address,
94                adapter.InsertCommand = command;
95                adapter.InsertCommand.ExecuteNonQuery();
96
97                con.Close();
98
99
100               createFile();
101
102               ScriptManager.RegisterStartupScript('                                    uccessfully Added!');", true);
103
104               Response.Redirect("PatientsAndFiles.a
105           }
106       }
```

Screenshot 2 — AdminLogin.aspx.cs

```
40        private void trackActivity(int empID)
41        {
42            try
43            {
44                DateTime date = DateTime.Now;
45
46                con.Open();
47                adapter = new SqlDataAdapter();
48
49                command = new SqlCommand($"INSERT INTO Admin(Employee_ID, Date_accessed, Time_accessed) VALUES({empID}, '{d
50                adapter.InsertCommand = command;
51                adapter.InsertCommand.ExecuteNonQuery();
52
53                con.Close();
54
55            }
56            catch (SqlException ex)
57            {
58                ScriptManager.RegisterStartupScript(this, this.GetType(), "popup", "alert('Error: An error occurred '" + ex
59            }
60        }
61    }
```

```csharp
63        private int getEmployeeID()
64        {
65            try
66            {
67                int employeeID = 0;
68                con.Open();
69                command = new SqlCommand($"SELECT Employee_ID FROM AdminCredentials WHERE Username = '{txtAdminID.Text}' A
70                dataReader = command.ExecuteReader();
71
72                while (dataReader.Read())
73                {
74                    employeeID = int.Parse(dataReader.GetValue(0).ToString());
75                }
76
77                con.Close();
78                return employeeID;
79            }
80            catch (SqlException ex)
81            {
82                ScriptManager.RegisterStartupScript(this, this.GetType(), "popup", "alert('Error: An error occurred '" + e
83            }
84        }
```

```csharp
87            1 reference
88        public bool validate()
89        {
90            try
91            {
92                int count = 0;
93                con.Open();
94                command = new SqlCommand($"SELECT COUNT(*) FROM AdminCredentials WHERE Username = '{txtAdminID.Text}' AND P
95                count = (int)command.ExecuteScalar();
96                con.Close();
97
98                if (count == 0)
99                {
100                   return false;
101               }
102               else
103               {
104                   return true;
105               }
106           }
107           catch (SqlException ex)
108           {
```

```csharp
116    private void backupFiles()
117    {
118        try
119        {
120            StreamWriter output;
121            con.Open();
122
123            command = new SqlCommand($"SELECT * FROM [File] ORDER BY Patient_ID ASC", con);
124            datareader = command.ExecuteReader();
125
126            output = File.AppendText(@"C:\Users\USER\Documents\NWU\Second year\Second semester\CMPG223\Goup Assignments
127            DateTime date = DateTime.Now;
128            output.WriteLine("\nBackup generated on: " + date.ToLocalTime());
129            while (datareader.Read())
130            {
131                output.WriteLine(datareader.GetValue(0) + " - " + datareader.GetValue(1) + " - " + datareader.GetValue
132            }
133
134            output.Close();
135            con.Close();
136
137        }
```

```csharp
87
88     private void backupPatients()
89     {
90         try
91         {
92             StreamWriter output;
93             con.Open();
94
95             command = new SqlCommand($"SELECT * FROM [Patient] ORDER BY Patient_ID ASC", con);
96             datareader = command.ExecuteReader();
97
98             output = File.AppendText(@"C:\Users\USER\Documents\NWU\Second year\Second semester\CMPG223\Goup Assignments
99             DateTime date = DateTime.Now;
100            output.WriteLine("\nBackup generated on: " + date.ToLocalTime());
101            while (datareader.Read())
102            {
103                output.WriteLine(datareader.GetValue(0) + " - " + datareader.GetValue(1) + " - " + datareader.GetValue
104            }
105
106            output.Close();
107            con.Close();
108
```

Two screenshots of Visual Studio showing C# code.

Top screenshot — Backup.aspx.cs (displayFiles):

```
            }

    1 reference
    private void displayFiles()
    {
        try
        {
            con.Open();
            adapter = new SqlDataAdapter();
            ds = new DataSet();

            command = new SqlCommand($"SELECT * FROM [File] ORDER BY Patient_ID ASC", con);
            adapter.SelectCommand = command;
            adapter.Fill(ds, "Files");

            gvFiles.DataSource = ds;
            gvFiles.DataBind();

            con.Close();

        }
        catch (SqlException ex)
        {
```

Bottom screenshot — Backup.aspx.cs (displayPatients):

```
    private void displayPatients()
    {
        try
        {
            con.Open();
            adapter = new SqlDataAdapter();
            ds = new DataSet();

            command = new SqlCommand($"SELECT * FROM [Patient] ORDER BY Patient_ID ASC", con);
            adapter.SelectCommand = command;
            adapter.Fill(ds, "Patients");

            gvPatients.DataSource = ds;
            gvPatients.DataBind();

            con.Close();

        }
        catch (SqlException ex)
        {
            ScriptManager.RegisterStartupScript(this, this.GetType(), "popup", "alert('Error: An error occurred '" + ex
```

Top window - BookingsHome.aspx.cs - displayBookings():

```
34              private void displayBookings()
35              {
36                  try
37                  {
38                      con.Open();
39                      adapter = new SqlDataAdapter();
40                      ds = new DataSet();
41
42                      command = new SqlCommand($"SELECT Booking_ID, First_name, Last_name, Appointment_date, Appointment_time FR(
43                      adapter.SelectCommand = command;
44                      adapter.Fill(ds, "Files");
45
46                      gvBookings.DataSource = ds;
47                      gvBookings.DataBind();
48
49                      con.Close();
50                  }
51                  catch (SqlException ex)
52                  {
53                      ScriptManager.RegisterStartupScript(this, this.GetType(), "popup", "alert('Error: An error occurred '" + e>
54
```

Bottom window - CreatePatientAccount.aspx.cs - create():

```
35              }
36              private void create()
37              {
38                  try
39                  {
40                      con.Open();
41                      adapter = new SqlDataAdapter();
42
43                      command = new SqlCommand($"INSERT INTO PatientCredentials(Username, Password, Name) VALUES('{txtSurname.Te>
44                      adapter.InsertCommand = command;
45                      adapter.InsertCommand.ExecuteNonQuery();
46
47                      con.Close();
48
49                      //System.Threading.Thread.Sleep(5000);
50                      ScriptManager.RegisterStartupScript(this, this.GetType(), "popup", "alert('Account Created Successfully');"}
51
52                      Response.Redirect("PatientLogin.aspx");
53                  }
54                  catch (SqlException ex)
55                  {
```

```
38          }
        1 reference
39          public bool validate()
40          {
41              try
42              {
43                  int count = 0;
44                  con.Open();
45                  command = new SqlCommand($"SELECT COUNT(*) FROM [Patient] WHERE Patient_ID = {int.Parse(Session["PatientID"
46                  count = (int)command.ExecuteScalar();
47                  con.Close();
48
49                  if (count == 0)
50                  {
51                      return false;
52                  }
53                  else
54                  {
55                      return true;
56                  }
57              }
58              catch (SqlException ex)
```



```
        1 reference
77          private void deleteFile()
78          {
79              try
80              {
81                  con.Open();
82                  adapter = new SqlDataAdapter();
83
84                  command = new SqlCommand($"DELETE FROM [File] WHERE Patient_ID = {int.Parse(Session["PatientID"].ToString(
85                  adapter.DeleteCommand = command;
86                  adapter.DeleteCommand.ExecuteNonQuery();
87
88                  con.Close();
89
90              }
91              catch (SqlException ex)
92              {
93                  ScriptManager.RegisterStartupScript(this, this.GetType(), "popup", "alert('Error: An error occurred '" + ex
94              }
95          }
96      }
97
        1 reference
```

Screenshot 1 (top window):

File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help   Search (Ctrl+Q)   Proj   HM

Debug ▾   Any CPU ▾   ▶ IIS Express (Firefox) ▾   Live Share

Tabs: AdminLogin.aspx.cs   Backup.aspx.cs   BookingsHome.aspx.cs   CreatePatientAccount.aspx.cs   **DeletePatient.aspx.cs ×**

Proj   ▾ Proj.DeletePatient   ▾ btnSubmit_Click(object sender, EventArgs e)

```
        1 reference
 98     private void deletePatient()
 99     {
100         try
101         {
102             con.Open();
103             adapter = new SqlDataAdapter();
104
105             command = new SqlCommand($"DELETE FROM [Patient] WHERE Patient_ID = {int.Parse(Session["PatientID"].ToStri
106             adapter.DeleteCommand = command;
107             adapter.DeleteCommand.ExecuteNonQuery();
108
109             con.Close();
110
111         }
112         catch (SqlException ex)
113         {
114             ScriptManager.RegisterStartupScript(this, this.GetType(), "popup", "alert('Error: An error occurred '" + e
115
116         }
117     }
118
```

100 %   ⊘ No issues found   Ln: 35   Ch: 39   SPC   CRLF

Error List
Entire Solution ▾   ⊗ 0 Errors   ⚠ 0 Warnings   ⓘ 0 of 70 Messages   Build + IntelliSense ▾   Search Error List

Code   Description   Project   File   Line   Suppression State

Solution Explorer

Search Solution Explorer (Ctrl+;)

▲ 📄 BookingsHome.aspx
  ▷ 📄 BookingsHome.aspx.cs
  ▷ 📄 BookingsHome.aspx.designer.cs
▲ 📄 CreatePatientAccount.aspx
  ▷ 📄 CreatePatientAccount.aspx.cs
  ▷ 📄 CreatePatientAccount.aspx.designer.cs
▲ 📄 DeletePatient.aspx
  ▷ 📄 DeletePatient.aspx.cs
  ▷ 📄 DeletePatient.aspx.designer.cs
  📄 face.css
▷ 📄 Home.aspx
  📄 packages.config

Solution Explorer   Git Changes

Properties

Ready   ↑ Add to Source Control ▲

Screenshot 2 (bottom window):

File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help   Search (Ctrl+Q)   Proj   HM

Debug ▾   Any CPU ▾   ▶ IIS Express (Firefox) ▾   Live Share

Tabs: Backup.aspx.cs   BookingsHome.aspx.cs   CreatePatientAccount.aspx.cs   DeletePatient.aspx.cs   **PatientLogin.aspx.cs ×**

Proj   ▾ Proj.PatientLogin   ▾ btnLogin_Click(object sender, EventArgs e)

```
        1 reference
 38     public bool validate()
 39     {
 40         try
 41         {
 42             int count = 0;
 43             con.Open();
 44             command = new SqlCommand($"SELECT COUNT(*) FROM PatientCredentials WHERE Username = '{txtPatientUsername.T
 45             count = (int)command.ExecuteScalar();
 46             con.Close();
 47
 48             if(count == 0)
 49             {
 50                 return false;
 51             }
 52             else
 53             {
 54                 con.Open();
 55
 56                 command = new SqlCommand($"SELECT Name FROM [PatientCredentials] WHERE Username = '{txtPatientUsername
 57                 dataReader = command.ExecuteReader();
 58
```

100 %   ⊘ No issues found   Ln: 32   Ch: 113   SPC   CRLF

Error List
Entire Solution ▾   ⊗ 0 Errors   ⚠ 0 Warnings   ⓘ 0 of 70 Messages   Build + IntelliSense ▾   Search Error List

Code   Description   Project   File   Line   Suppression State

Solution Explorer

Search Solution Explorer (Ctrl+;)

▲ 📄 DeletePatient.aspx
  ▷ 📄 DeletePatient.aspx.cs
  ▷ 📄 DeletePatient.aspx.designer.cs
  📄 face.css
▷ 📄 Home.aspx
  📄 packages.config
▲ 📄 PatientLogin.aspx
  ▷ 📄 PatientLogin.aspx.cs
  ▷ 📄 PatientLogin.aspx.designer.cs
▷ 📄 Patients.aspx
▷ 📄 PatientsAndFiles.aspx
▷ 📄 ReserveBooking.aspx

Solution Explorer   Git Changes

Properties
PatientLogin.aspx.cs File Properties

Browse to URL
Build Action   Compile
Copy to Output Directory   Do not copy
Custom Tool
Custom Tool Namespace
File Name   PatientLogin.aspx.cs
Full Path   C:\Users\USER\Documents\NV

Browse to URL

Ready   ↑ Add to Source Control ▲

```
27
28              private void displayPatients()
29              {
30                  try
31                  {
32                      con.Open();
33                      adapter = new SqlDataAdapter();
34                      ds = new DataSet();
35
36                      string sortPatients = "ASC";
37                      if(Session["SortPatients"] != null)
38                      {
39                          sortPatients = Session["SortPatients"].ToString();
40                      }
41
42                      if(sortPatients == "DESC")
43                      {
44                          command = new SqlCommand($"SELECT * FROM [Patient] ORDER BY Patient_ID ASC", con);
45                      }
46                      else
47                      {
48                          command = new SqlCommand($"SELECT * FROM [Patient] ORDER BY Patient_ID DESC", con);
```

Second screenshot code:

```
44                      command = new SqlCommand($"SELECT * FROM [Patient] ORDER BY Patient_ID ASC", con);
45                  }
46                  else
47                  {
48                      command = new SqlCommand($"SELECT * FROM [Patient] ORDER BY Patient_ID DESC", con);
49                  }
50
51                  adapter.SelectCommand = command;
52                  adapter.Fill(ds, "Patients");
53
54                  gvPatients.DataSource = ds;
55                  gvPatients.DataBind();
56
57                  con.Close();
58              }
59          }
60          catch (SqlException ex)
61          {
62              ScriptManager.RegisterStartupScript(this, this.GetType(), "popup", "alert('Error: An error occurred '" + ex
63
64          }
65      }
66  }
```

Page 10 of 13

Screenshot 1 (top):

```
        1 reference
67      private void displayFiles()
68      {
69          try
70          {
71              con.Open();
72              adapter = new SqlDataAdapter();
73              ds = new DataSet();
74
75              string sortFiles = "ASC";
76              if (Session["SortFiles"] != null)
77              {
78                  sortFiles = Session["SortFiles"].ToString();
79              }
80
81              if(sortFiles == "DESC")
82              {
83                  command = new SqlCommand($"SELECT * FROM [File] ORDER BY Patient_ID ASC", con);
84              }
85              else
86              {
87                  command = new SqlCommand($"SELECT * FROM [File] ORDER BY Patient_ID DESC", con);
```

Screenshot 2 (bottom):

```
86          {
87              command = new SqlCommand($"SELECT * FROM [File] ORDER BY Patient_ID DESC", con);
88          }
89
90
91          adapter.SelectCommand = command;
92          adapter.Fill(ds, "Files");
93
94          gvFiles.DataSource = ds;
95          gvFiles.DataBind();
96
97          con.Close();
98
99      }
100     catch (SqlException ex)
101     {
102         ScriptManager.RegisterStartupScript(this, this.GetType(), "popup", "alert('Error: An error occurred '" + e)
103
104     }
105 }
106
107 protected void rbPatients_SelectedIndexChanged(object sender, EventArgs e)
```

```
private void book()
{
    try
    {
        con.Open();
        adapter = new SqlDataAdapter();

        command = new SqlCommand($"INSERT INTO Appointments(First_name, Last_name, Appointment_date, Appointment_time) V
        adapter.InsertCommand = command;
        adapter.InsertCommand.ExecuteNonQuery();

        con.Close();

        ScriptManager.RegisterStartupScript(this, this.GetType(), "popup", "alert('Booking Created Successfully');", tr

        if(Session["AdminID"] != null)
        {
            Response.Redirect("AdminHome.aspx");
        }
        else
```

```
private void updatePatient()
{
    try
    {
        string sortPatients = "Ascending";
        if (Session["SortPatients"] != null)
        {
            sortPatients = Session["SortPatients"].ToString();
        }

        con.Open();
        adapter = new SqlDataAdapter();

        command = new SqlCommand($"UPDATE [File] SET Record = '{txtRecord.Text}' WHERE Patient_ID = {int.Parse(Sess
        adapter.UpdateCommand = command;
        adapter.UpdateCommand.ExecuteNonQuery();

        con.Close();

        Response.Redirect("PatientsA

    }
    catch (SqlException ex)
```

int SqlCommand.ExecuteNonQuery()
Executes a Transact-SQL statement against the connection and returns the number of rows affected.

Returns:
The number of rows affected.

Exceptions:
InvalidCastException
SqlException
System.IO.IOException
InvalidOperationException
ObjectDisposedException