

Appendix F

Full set of code listings:-

Login Screen (215):-

```
public class LoginScreen
{
    static String UsernameA, UserNameN, PasswordA, PasswordN, username;
    static char[] passcharA, passcharN;
    static String passarr;
    JOptionPane frame = new JOptionPane();
    JFrame login_frame = new JFrame ("Bargoed Nurses");
    static JTextField userf = new JTextField();
    static JPasswordField passf = new JPasswordField();
    JPanel login = new JPanel();
    private JPanel heading;
    private ImageIcon image1;
    private JLabel label1;
    public static JPanel mainpage = new JPanel();
    JPanel rest = new JPanel(), sidebar = new JPanel();
    Color buttonBackground = new Color (234,234,234);
    Color blueBackground = new Color (7,32,157);
    Color PanelBackground = new Color (167, 165, 255);
    Font f = new Font("Dialog", Font.PLAIN, 26);
    private static String NName;

    public LoginScreen() throws Exception
    {
        heading = new JPanel();
        heading.setMaximumSize(new Dimension(700,150));
        heading.setBackground(blueBackground);
        image1 = new ImageIcon(getClass().getResource("Bar.jpg"));
        label1 = new JLabel(image1);
        heading.add(label1);

        JButton submit = new JButton("Submit");
        login_frame.setSize(780,570);
        login_frame.setLayout(new FlowLayout());
        login_frame.setBackground(Color.WHITE);

        login.setLayout(new BoxLayout(login, BoxLayout.PAGE_AXIS));
        login.setPreferredSize(new Dimension(600,300));

        mainpage.setPreferredSize(new Dimension(500,400));
        mainpage.removeAll();

        rest.setPreferredSize(new Dimension(800,600));
        //Inputting information into sidebar
        JPanel TestPanel = new JPanel();
        JLabel sidelabel = new JLabel("<html>* Enter personal <br> log in details
```

```

<br>into textboxes<br> provided");
JLabel sidelabelblank = new JLabel(" ");
JLabel sidelabelblank2 = new JLabel(" ");
JLabel sidelabel2 = new JLabel("<html>* If you don't remember your log
in details please contact administrator");

TestPanel.setBackground(PanelBackground);
TestPanel.setPreferredSize(new Dimension(100,400));
TestPanel.setLayout(new BorderLayout(TestPanel, BorderLayout.PAGE_AXIS));
TestPanel.add(sidelabel);
TestPanel.add(sidelabelblank);
TestPanel.add(sidelabel2);
TestPanel.add(sidelabelblank2);

rest.add(TestPanel); rest.add(mainpage);
rest.setBackground(blueBackground);

login_frame.add(heading); login_frame.add(rest);

JLabel userl = new JLabel("Username:");
passl = new JLabel("Password:");
JLabel welcome = new JLabel("<html>Welcome to the Bargoed Nurses
System<br>Please enter login details below:");
login.add(welcome);
userl.setFont(f);
login.add(userl);
userf.setFont(f);
login.add(userf);
userf.setBackground(buttonBackground);
passl.setFont(f);
login.add(passl);
passf.setFont(f);
login.add(passf);
passf.setBackground(buttonBackground);
login.add(submit);
mainpage.add(login);
mainpage.setPreferredSize(new Dimension(650,400));
mainpage.setBackground(Color.WHITE);
login.setBackground(Color.WHITE);

submit.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        try
        {
            //Check Username and Password match against
            Admin details in database
            //Check Username and Password match against
            Nurse details in database

```

```

        //If pair is correct then advance to appropriate
        Screen
        //Else Error message
        String UsernameA = Username();
        String PasswordA = Password();
        Login_Check();
    }
    catch (Exception exc)
    {
        System.out.println(exc);
    }
}
});
login_frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
login_frame.setVisible(true);
}
public void Login_Check() throws Exception
{
    //Check Username and Password match against Admin details in database
    //Check Username and Password match against Nurse details in database
    //If pair is correct then advance to appropriate Screen
    //Else Error message
    Class.forName("org.sqlite.JDBC");
    Connection conn =
    DriverManager.getConnection("jdbc:sqlite:NurseScheduling.sqlite");
    PreparedStatement stat = conn.prepareStatement("select * from
    Administrator where UsernameA=? and PasswordA=? ");
    //SQL Injection Prevention
    stat.setString(1, UsernameA);
    stat.setString(2, PasswordA);

    ResultSet rs = stat.executeQuery();

    Connection connt =
    DriverManager.getConnection("jdbc:sqlite:NurseScheduling.sqlite");
    PreparedStatement statt = connt.prepareStatement("select * from
    DistrictNurse where UserNameN=? and PasswordN=?");

    //SQL Injection Prevention
    statt.setString(1, UsernameA);
    statt.setString(2, PasswordA);
    ResultSet st = statt.executeQuery();

    if (rs.next())
    {
        new MainScreenAdmin();
        login_frame.setVisible(false);
    }
    else if (st.next())
    {

```

```

        new MainScreenNurse();
        login_frame.setVisible(false);
    }
    else
    {
        JOptionPane.showMessageDialog(frame, "Please enter a valid
        Username/Password","Incorrect Login",JOptionPane.ERROR_MESSAGE);
    }
    rs.close();
    st.close();
    conn.close();
    connt.close();
}
//Return the username entered by user
//This username will be used in the login check sql query
public static String Username() throws Exception
{
    UsernameA = userf.getText();
    return UsernameA;
}
//Convert password from charater to a String
//This will be used in login check sql query
//Return the password entered by user
public static String Password() throws Exception
{
    PasswordA = "";
    passcharA = passf.getPassword();
    for (int i=0;i<passcharA.length;i++)

    {
        PasswordA = PasswordA + Character.toString(passcharA[i]);
    }
    return PasswordA;
}
//Return username entered by user
public static String Get_User()
{
    return UsernameA;
}
//Return full name of user entered
public static String Get_UserFullName()
{
    return NName;
}

public static void main(String args[]) throws Exception
{
    new LoginScreen();
}
}

```

Main Screen for District Nurse (205):-

```
public class MainScreenNurse
{
    JOptionPane frame = new JOptionPane();
    JFrame main_frame = new JFrame ("Bargoed Nurses");
    JPanel main = new JPanel();
    private JPanel heading;
    private ImageIcon image1;
    private JLabel label1;
    public static JPanel mainpage = new JPanel();
    JPanel rest = new JPanel(), sidebar = new JPanel();
    Color buttonBackground = new Color (234,234,234);
    Color blueBackground = new Color (7,32,157);
    Color PanelBackground = new Color (167, 165, 255);
    Font f = new Font("Dialog", Font.PLAIN, 36);
    public MainScreenNurse() throws Exception
    {
        heading = new JPanel();
        heading.setMaximumSize(new Dimension(700,150));
        heading.setBackground(blueBackground);
        image1 = new ImageIcon(getClass().getResource("Bar.jpg"));
        label1 = new JLabel(image1);
        heading.add(label1);

        JButton PersonalDetails = new JButton("View Personal Details");
        PersonalDetails.setMaximumSize(new Dimension(700, 100));
        PersonalDetails.setFont(f);
        JButton ViewPatient = new JButton("View Patient Details");
        ViewPatient.setMaximumSize(new Dimension(700, 100));
        ViewPatient.setFont(f);
        JButton ViewCalls = new JButton("View Calls");
        ViewCalls.setMaximumSize(new Dimension(700, 100));
        ViewCalls.setFont(f);
        JButton LogOut = new JButton("Log out");
        LogOut.setMaximumSize(new Dimension(700, 100));
        LogOut.setFont(f);

        main_frame.setSize(780,570);
        main_frame.setLayout(new FlowLayout());
        main_frame.setBackground(Color.WHITE);

        main.setLayout(new BorderLayout(main, BorderLayout.PAGE_AXIS));
        main.setPreferredSize(new Dimension(600,380));
        mainpage.setPreferredSize(new Dimension(500,400));
```

```
mainpage.removeAll();
rest.setPreferredSize(new Dimension(800,600));

//Inputting information into sidebar
JPanel TestPanel = new JPanel();
JLabel sidelabel = new JLabel("<html>* Select the button you require");
JLabel sidelabelblank = new JLabel(" ");
JLabel sidelabelblank2 = new JLabel(" ");
JLabel sidelabelblank3 = new JLabel(" ");
JLabel sidelabelblank4 = new JLabel(" ");
JLabel sidelabelblank5 = new JLabel(" ");

JLabel sidelabel2 = new JLabel("<html>* View Personal details will allow you to
review your details that are stored and check they are up to date");
JLabel sidelabel3 = new JLabel("<html>* View Patient details allows you to search
for a particular patient in order to view their detailed personal details");
JLabel sidelabel4 = new JLabel("<html>* View Calls allows you to view your personal
call list");
JLabel sidelabel5 = new JLabel("<html>* Log out allows you to log out of the
system");

TestPanel.setBackground(PanelBackground);
TestPanel.setPreferredSize(new Dimension(100,400));
TestPanel.setLayout(new BorderLayout(TestPanel, BorderLayout.PAGE_AXIS));
TestPanel.add(sidelabel2);
TestPanel.add(sidelabelblank2);
TestPanel.add(sidelabel3);
TestPanel.add(sidelabelblank3);
TestPanel.add(sidelabel4);
TestPanel.add(sidelabelblank4);
TestPanel.add(sidelabel5);
TestPanel.add(sidelabelblank5);

rest.add(TestPanel); rest.add(mainpage);
rest.setBackground(blueBackground);

main_frame.add(heading); main_frame.add(rest);

main.add(PersonalDetails);
main.add(ViewPatient);
main.add(ViewCalls);
main.add(LogOut);
PersonalDetails.setBackground(buttonBackground);
ViewPatient.setBackground(buttonBackground);
ViewCalls.setBackground(buttonBackground);
```

```
LogOut.setBackground(buttonBackground);
mainpage.add(main);
mainpage.setPreferredSize(new Dimension(650,400));
mainpage.setBackground(Color.WHITE);
main.setBackground(Color.WHITE);
```

```
PersonalDetails.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        try
        {
            //Redirect User to view their personal details
            main_frame.setVisible(false);
            new NursePersonal();
        }
        catch (Exception exc)
        {
            System.out.println(exc);
        }
    }
});
```

```
ViewPatient.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        try
        {
            //Redirect User to search for a patient
            main_frame.setVisible(false);
            new SearchPatientNurse();
        }
        catch (Exception exc)
        {
            System.out.println(exc);
        }
    }
});
```

```
ViewCalls.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent e)
```

```

        {
            try
            {
                //Redirect User to view their personal call list
                main_frame.setVisible(false);
                new CallList();
            }
            catch (Exception exc)
            {
                System.out.println(exc);
            }
        }
    });

    LogOut.addActionListener(new ActionListener()
    {
        public void actionPerformed(ActionEvent e)
        {
            try
            {
                //Dialog box to check whether they want to log out
                Object[] options = {"Yes", "No"};
                int response = JOptionPane.showOptionDialog(frame, "Are
                you sure you want to logout?", "Log Out?",
                JOptionPane.YES_NO_OPTION, JOptionPane.WARNING_MES
                SAGE, null, options, options[1]);
                if (response != JOptionPane.YES_OPTION)
                {
                    return;
                }
                else
                {
                    //Redirect User back to the main log in screen
                    main_frame.setVisible(false);
                    LoginScreen.userf.setText("");
                    LoginScreen.passf.setText("");
                    new LoginScreen();
                }
            }
            catch (Exception exc)
            {
                System.out.println(exc);
            }
        }
    });

```

```

    });

    main_frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    main_frame.setVisible(true);
}

public static void main(String args[]) throws Exception
{
    new MainScreenNurse();
}
}

```

'View Personal details' – District Nurse (251):-

```

public class NursePersonal
{
    static String username;
    JOptionPane frame = new JOptionPane();
    JFrame nurse_frame = new JFrame ("Bargoed Nurses");
    private JPanel heading;
    private ImageIcon image1;
    private JLabel label1;
    public static JPanel mainpage = new JPanel();
    JPanel rest = new JPanel(), sidebar = new JPanel();
    Color buttonBackground = new Color (234,234,234);
    Color blueBackground = new Color (7,32,157);
    Color PanelBackground = new Color (167, 165, 255);
    Font f = new Font("Dialog", Font.PLAIN, 20);

    public NursePersonal() throws Exception
    {
        username = LoginScreen.Get_User();

        heading = new JPanel();
        heading.setMaximumSize(new Dimension(700,150));
        heading.setBackground(blueBackground);
        image1 = new ImageIcon(getClass().getResource("Bar.jpg"));
        label1 = new JLabel(image1);
        heading.add(label1);

        JButton print = new JButton("Print");
        JButton back = new JButton("Back to main screen");
        nurse_frame.setSize(780,570);
        nurse_frame.setLayout(new FlowLayout());
    }
}

```

```

nurse_frame.setBackground(Color.WHITE);

mainpage.setPreferredSize(new Dimension(500,400));
mainpage.removeAll();

rest.setPreferredSize(new Dimension(800,600));
rest.setBackground(blueBackground);

nurse_frame.add(heading); nurse_frame.add(rest);

//Inputting information into sidebar
JPanel TestPanel = new JPanel();
JLabel sidelabel = new JLabel("<html>* Review your details and if they are not up to
date please contact administrator");
JLabel sidelabelblank = new JLabel(" ");
TestPanel.add(sidelabel);
TestPanel.add(sidelabelblank);
TestPanel.setBackground(PanelBackground);

TestPanel.setPreferredSize(new Dimension(100,400));
TestPanel.setLayout(new BorderLayout(TestPanel, BorderLayout.PAGE_AXIS));

rest.add(TestPanel); rest.add(mainpage);

JPanel moduleListPanel = new JPanel();
moduleListPanel.setLayout(new FlowLayout());
moduleListPanel.setPreferredSize(new Dimension(600,600));

JLabel welcome = new JLabel("<html>Your personal details are:");
welcome.setPreferredSize(new Dimension(600, 20));
welcome.setHorizontalTextPosition(JLabel.LEFT);
moduleListPanel.add(welcome);

Class.forName("org.sqlite.JDBC");
Connection conn =
DriverManager.getConnection("jdbc:sqlite:NurseScheduling.sqlite");
Statement stat = conn.createStatement();
ResultSet rs = stat.executeQuery(String.format("select * from DistrictNurse where
UserNameN='%s'", username));

JPanel txtPanel = new JPanel();
txtPanel.setLayout(null);
txtPanel.setLocation(20, 50);
txtPanel.setPreferredSize(new Dimension(200, 330));
txtPanel.setBackground(Color.WHITE);

```

```
moduleListPanel.add(txtPanel);
moduleListPanel.setBackground(Color.WHITE);

JPanel txtFieldPanel = new JPanel();
txtFieldPanel.setLayout(null);
txtFieldPanel.setLocation(150, 60);
txtFieldPanel.setPreferredSize(new Dimension(390, 330));
txtFieldPanel.setBackground(Color.WHITE);
moduleListPanel.add(txtFieldPanel);

JLabel userName = new JLabel("Username:");
userName.setLocation(0, 0);
userName.setSize(300, 30);
userName.setHorizontalTextPosition(JLabel.LEFT);
txtPanel.add(userName);

JLabel NurseName = new JLabel("Nurse Name");
NurseName.setLocation(0,50);
NurseName.setSize(300,30);
NurseName.setHorizontalTextPosition(JLabel.LEFT);
txtPanel.add(NurseName);

JLabel PinNumber = new JLabel("Pin Number");
PinNumber.setLocation(0,100);
PinNumber.setSize(300,30);
PinNumber.setHorizontalTextPosition(JLabel.LEFT);
txtPanel.add(PinNumber);

JLabel Address = new JLabel("Address");
Address.setLocation(0,150);
Address.setSize(300,30);
Address.setHorizontalTextPosition(JLabel.LEFT);
txtPanel.add(Address);

JLabel Town = new JLabel("Town");
Town.setLocation(0,200);
Town.setSize(300,30);
Town.setHorizontalTextPosition(JLabel.LEFT);
txtPanel.add(Town);

JLabel Band = new JLabel("Band");
Band.setLocation(0,250);
Band.setSize(300,30);
Band.setHorizontalTextPosition(JLabel.LEFT);
```

```
txtPanel.add(Band);

JLabel TelNumber = new JLabel("Telephone Number");
TelNumber.setLocation(0,300);
TelNumber.setSize(300,30);
TelNumber.setHorizontalTextPosition(JLabel.LEFT);
txtPanel.add(TelNumber);

JTextField UsernameField = new JTextField(rs.getString("UserNameN"));
UsernameField.setEditable(false);
UsernameField.setLocation(0, 0);
UsernameField.setSize(390, 30);
UsernameField.setBackground(PanelBackground);
txtFieldPanel.add(UsernameField);

JTextField NurseNameField = new JTextField(rs.getString("NurseName"));
NurseNameField.setEditable(false);
NurseNameField.setLocation(0, 50);
NurseNameField.setSize(390, 30);
NurseNameField.setBackground(PanelBackground);
txtFieldPanel.add(NurseNameField);

JTextField PinNumberField = new JTextField(rs.getString("PinNumber"));
PinNumberField.setEditable(false);
PinNumberField.setLocation(0, 100);
PinNumberField.setSize(390, 30);
PinNumberField.setBackground(PanelBackground);
txtFieldPanel.add(PinNumberField);

final JTextArea addressArea = new JTextArea(rs.getString("Address"));
addressArea.setSize(390, 30);
addressArea.setEditable(false);
addressArea.setBackground(PanelBackground);
addressArea.setLinewrap(true);

final JScrollPane scrollPane = new JScrollPane(addressArea);
scrollPane.setLocation(0, 150);
scrollPane.setSize(390, 30);
txtFieldPanel.add(scrollPane);

JTextField TownField = new JTextField(rs.getString("Town"));
TownField.setEditable(false);
TownField.setLocation(0, 200);
TownField.setSize(390, 30);
TownField.setBackground(PanelBackground);
```

```

txtFieldPanel.add(TownField);

JTextField BandField = new JTextField(rs.getString("Band"));
BandField.setEditable(false);
BandField.setLocation(0, 250);
BandField.setSize(390, 30);
BandField.setBackground(PanelBackground);
txtFieldPanel.add(BandField);

JTextField TelNumberField = new JTextField(rs.getString("TelNumberN"));
TelNumberField.setEditable(false);
TelNumberField.setLocation(0, 300);
TelNumberField.setSize(390, 30);
TelNumberField.setBackground(PanelBackground);
txtFieldPanel.add(TelNumberField);

moduleListPanel.add(print);
moduleListPanel.add(back);

mainpage.add(moduleListPanel);
mainpage.setPreferredSize(new Dimension(650,400));
mainpage.setBackground(Color.WHITE);

back.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        try
        {
            //Redirect the district nure main screen
            nurse_frame.setVisible(false);
            new MainScreenNurse();
        }
        catch (Exception exc)
        {
            System.out.println(exc);
        }
    }
});

print.addActionListener(new ActionListener()
{

```

```

        public void actionPerformed(ActionEvent e)
        {
            try
            {
                //Display message to user indicating they have selected the
                //print option
                JOptionPane.showMessageDialog(frame, "You have selected
                the print button", "Printing",
                JOptionPane.INFORMATION_MESSAGE);
            }
            catch (Exception exc)
            {
                System.out.println(exc);
            }
        }
    });

    nurse_frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    nurse_frame.setVisible(true);
}

public static void main(String args[]) throws Exception
{
    new NursePersonal();
}
}

```

'Search Patient details' main page – District Nurse (322):-

```

public class SearchPatientNurse
{
    static String PName, FName, TownD, TreatmentD;

    JOptionPane frame = new JOptionPane();
    JFrame patient_frame = new JFrame ("Bargoed Nurses");
    private JPanel heading;
    private ImageIcon image1;
    private JLabel label1;
    public static JPanel mainpage = new JPanel();
    JPanel rest = new JPanel(), sidebar = new JPanel();
    Color buttonBackground = new Color (234,234,234);
    Color blueBackground = new Color (7,32,157);
    Color PanelBackground = new Color (167, 165, 255);
    Font f = new Font("Dialog", Font.PLAIN, 26);
}

```

```

static JTextField patientNameField = new JTextField();

public SearchPatientNurse() throws Exception
{
    heading = new JPanel();
    heading.setMaximumSize(new Dimension(700,150));
    heading.setBackground(blueBackground);
    image1 = new ImageIcon(getClass().getResource("Bar.jpg"));
    label1 = new JLabel(image1);
    heading.add(label1);

    JButton search = new JButton("Search for Patient details");
    JButton back = new JButton("Back to main screen");
    patient_frame.setSize(780,570);
    patient_frame.setLayout(new FlowLayout());
    patient_frame.setBackground(Color.WHITE);

    mainpage.setPreferredSize(new Dimension(500,400));
    mainpage.removeAll();
    rest.setPreferredSize(new Dimension(800,600));
    rest.setBackground(blueBackground);

    patient_frame.add(heading); patient_frame.add(rest);

    //Inputting information into sidebar
    JPanel TestPanel = new JPanel();
    JLabel sidelabel = new JLabel("<html>* Enter known <br> patient details <br>into  
textboxes<br> provided");
    JLabel sidelabelblank = new JLabel(" ");
    JLabel sidelabelblank2 = new JLabel(" ");
    JLabel sidelabelblank3 = new JLabel(" ");
    JLabel sidelabel2 = new JLabel("<html>* A first name for a patient needs to be  
entered");
    JLabel sidelabel3 = new JLabel("<html>* 2 details need <br> to be entered ");
    JLabel sidelabel4 = new JLabel("<html>* Once criteria has been entered then select  
appropriate button at bottom of the page");
    TestPanel.setBackground(PanelBackground);

    TestPanel.setPreferredSize(new Dimension(100,400));
    TestPanel.setLayout(new BorderLayout(TestPanel, BorderLayout.PAGE_AXIS));
    TestPanel.add(sidelabel);
    TestPanel.add(sidelabelblank);
    TestPanel.add(sidelabel2);
    TestPanel.add(sidelabelblank2);
    TestPanel.add(sidelabel3);

```

```
TestPanel.add(sidelabelblank3);
TestPanel.add(sidelabel4);

rest.add(TestPanel); rest.add(mainpage);

JPanel moduleListPanel = new JPanel();
moduleListPanel.setLayout(new FlowLayout());
moduleListPanel.setPreferredSize(new Dimension(600,400));

JLabel welcome = new JLabel("<html>Enter known patient criteria:");
welcome.setFont(f);
welcome.setPreferredSize(new Dimension(600, 30));
welcome.setHorizontalTextPosition(JLabel.LEFT);

moduleListPanel.add(welcome);
JPanel txtPanel = new JPanel();
txtPanel.setLayout(null);
txtPanel.setLocation(20, 50);
txtPanel.setPreferredSize(new Dimension(200, 250));
txtPanel.setBackground(Color.WHITE);

moduleListPanel.add(txtPanel);
moduleListPanel.setBackground(Color.WHITE);

JPanel txtFieldPanel = new JPanel();
txtFieldPanel.setLayout(null);
txtFieldPanel.setLocation(150, 60);
txtFieldPanel.setPreferredSize(new Dimension(390, 250));
txtFieldPanel.setBackground(Color.WHITE);
moduleListPanel.add(txtFieldPanel);

JLabel patientName = new JLabel("Patient first name:");
patientName.setLocation(0, 0);
patientName.setSize(300, 50);
patientName.setHorizontalTextPosition(JLabel.LEFT);
txtPanel.add(patientName);

JLabel FamilyName = new JLabel("Family Name:");
FamilyName.setLocation(0,50);
FamilyName.setSize(300,50);
FamilyName.setHorizontalTextPosition(JLabel.LEFT);
txtPanel.add(FamilyName);

JLabel Town = new JLabel("Town:");
Town.setLocation(0,100);
```

```
Town.setSize(300,50);
Town.setHorizontalTextPosition(JLabel.LEFT);
txtPanel.add(Town);

JLabel treatment = new JLabel("Treatment:");
treatment.setLocation(0,150);
treatment.setSize(300,50);
treatment.setHorizontalTextPosition(JLabel.LEFT);
txtPanel.add(treatment);

final JTextField patientNameField = new JTextField();
patientNameField.setEditable(true);
patientNameField.setLocation(0, 0);
patientNameField.setSize(390, 50);
patientNameField.setBackground(PanelBackground);
txtFieldPanel.add(patientNameField);

final JTextField familyNameField = new JTextField();
familyNameField.setEditable(true);
familyNameField.setLocation(0, 50);
familyNameField.setSize(390, 50);
familyNameField.setBackground(PanelBackground);
txtFieldPanel.add(familyNameField);

final JTextField TownField = new JTextField();
TownField.setEditable(true);
TownField.setLocation(0, 100);
TownField.setSize(390, 50);
TownField.setBackground(PanelBackground);
txtFieldPanel.add(TownField);

final JTextArea Treatment = new JTextArea();
Treatment.setSize(390, 50);
Treatment.setEditable(true);
Treatment.setBackground(PanelBackground);
Treatment.setLineWrap(true);

final JScrollPane scrollPane = new JScrollPane(Treatment);
scrollPane.setLocation(0, 150);
scrollPane.setSize(390, 50);
txtFieldPanel.add(scrollPane);

moduleListPanel.add(search);
moduleListPanel.add(back);
```

```

mainpage.add(moduleListPanel);
mainpage.setPreferredSize(new Dimension(650,400));
mainpage.setBackground(Color.WHITE);

back.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        try
        {
            //Redirect user back to nurse main screen
            patient_frame.setVisible(false);
            new MainScreenNurse();
        }
        catch (Exception exc)
        {
            System.out.println(exc);
        }
    }
});

search.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        try
        {
            //Getting text from text boxes provided to user
            PName = patientNameField.getText();
            FName = familyNameField.getText();
            TownD = TownField.getText();
            TreatmentD = Treatment.getText();

            //converting text to a string
            String str = patientNameField.getText();

            Class.forName("org.sqlite.JDBC");
            Connection conn =
            DriverManager.getConnection("jdbc:sqlite:NurseScheduling.
            sqlite");
            PreparedStatement stat = conn.prepareStatement("select *
            from Patients2 where PName=? and FName=?");
            //SQL Injection Prevention
            stat.setString(1, PName);
            stat.setString(2, FName);

```

```

ResultSet rs = stat.executeQuery();

Class.forName("org.sqlite.JDBC");
Connection connt =
DriverManager.getConnection("jdbc:sqlite:NurseScheduling.
sqlite");
PreparedStatement statt = conn.prepareStatement("select *
from Patients2 where PName=? and PTown=?");
//SQL Injection Prevention
statt.setString(1, PName);
statt.setString(2, TownD);
ResultSet st = statt.executeQuery();

Class.forName("org.sqlite.JDBC");
Connection connr =
DriverManager.getConnection("jdbc:sqlite:NurseScheduling.
sqlite");
PreparedStatement statr = conn.prepareStatement("select *
from Patients2 where PName=? and Treatment=?");
//SQL Injection Prevention
statr.setString(1, PName);
statr.setString(2, TreatmentD);
ResultSet sr = statr.executeQuery();

if (rs.next())
{
    //Confirmation message for a match and then
    redirect to matches page
    JOptionPane.showMessageDialog(frame, "Details
    have been found to match the patient first name
    and family name", "Patient Criteria",
    JOptionPane.ERROR_MESSAGE);
    new SearchPatientDetailsTestNurse();
    patient_frame.setVisible(false);
}
else if (st.next())
{
    String PatientFound = st.getString("PName");
    //Confirmation message for a match and then
    redirect to matches page
    JOptionPane.showMessageDialog(frame, "Details
    have been found to match the patient first name
    and town", "Patient Criteria",
    JOptionPane.ERROR_MESSAGE);
    new SearchPatientDetailsTownTestNurse();
}

```

```

        patient_frame.setVisible(false);
    }
    else if (sr.next())
    {
        //Confirmation message for a match and then
        //redirect to matches page
        JOptionPane.showMessageDialog(frame, "Details
        have been found to match the patient first name
        and treatment", "Patient Criteria",
        JOptionPane.ERROR_MESSAGE);
        new SearchPatientDetailsTreatmentNurse();
        patient_frame.setVisible(false);
    }
    else if (str.equals(""))
    {
        //Message displayed to user if they do not enter a
        //patient first name
        JOptionPane.showMessageDialog(frame, "A first
        name for the patient needs to be entered",
        "Un entered Details",
        JOptionPane.ERROR_MESSAGE);
    }
    else
    {
        //Confirmation message that there are no matches
        //to be found
        JOptionPane.showMessageDialog(frame, "No
        details found for criteria entered", "Unfound
        Patient", JOptionPane.ERROR_MESSAGE);
    }
    rs.close();
    st.close();
    sr.close();
    conn.close();
    connt.close();
    connr.close();
}
catch (Exception exc)
{
    System.out.println(exc);
}
});
patient_frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

```

```

        patient_frame.setVisible(true);
    }

    //Return patient first name
    public static String Get_Patient()
    {
        return PName;
    }

    //Return patient family name
    public static String Get_Family()
    {
        return FName;
    }

    //Return patient town
    public static String Get_Town()
    {
        return TownD;
    }

    //Return patient treatment
    public static String Get_Treatment()
    {
        return TreatmentD;
    }

    public static void main(String args[]) throws Exception
    {
        new SearchPatientNurse();
    }
}

```

'Search Patient details' list of matches for patient first name and family name – District Nurse (294):-

```

public class SearchPatientDetailsTestNurse
{
    String PaName, FaName;

    JOptionPane frame = new JOptionPane();
    private static JFrame main_frame = new JFrame ("Bargoed Nurses");
    JPanel main = new JPanel();
    private JPanel heading, footer;

```

```

private ImageIcon image1;
private JLabel label1, ViewLabel, StaffName;
public static JPanel mainpage = new JPanel();
JScrollPane vertical = new JScrollPane();
JPanel rest = new JPanel(), sidebar = new JPanel();
Color buttonBackground = new Color (234,234,234);
Color blueBackground = new Color (7,32,157);
Font f = new Font("Dialog", Font.PLAIN, 26);
Color PanelBackground = new Color (167, 165, 255);
DefaultListModel data;
final JList TestList = new JList();

static String selectedLeftUsername;
static String lastone;
private static String ending;
private static String PaID;
private static int index;
private static String PaIDTest;
private static String fileName;

public SearchPatientDetailsTestNurse() throws Exception
{
    PaName = SearchPatientNurse.Get_Patient();
    FaName = SearchPatientNurse.Get_Family();

    heading = new JPanel();
    heading.setMaximumSize(new Dimension(700,150));
    heading.setBackground(blueBackground);
    image1 = new ImageIcon(getClass().getResource("Bar.jpg"));
    label1 = new JLabel(image1);
    heading.add(label1);

    footer = new JPanel();
    footer.setMaximumSize(new Dimension(800, 300));
    footer.setBackground(Color.GREEN);
    JButton Print = new JButton("Print Details");
    JButton AnotherPatient = new JButton("Another Patient");
    JButton BackMain = new JButton("Back to Main Screen");
    footer.add(Print); footer.add(AnotherPatient); footer.add(BackMain);
    footer.setBackground(Color.WHITE);

    ViewLabel = new JLabel("Here are the patients' details found:-");
    ViewLabel.setFont(f);
    main_frame.setSize(780,570);
    main_frame.setLayout(new FlowLayout());
}

```

```

main_frame.setBackground(Color.WHITE);
main.setLayout(new BorderLayout(main, BorderLayout.PAGE_AXIS));
main.setPreferredSize(new Dimension(600,350));

final JList TestList = new JList(listPatients());
TestList.setSelectionMode(ListSelectionModel.SINGLE_SELECTION);
TestList.setSize(200, 500);
TestList.setSelectedIndex(0);
TestList.setVisibleRowCount(6);

JScrollPane vertical = new JScrollPane(TestList);

vertical.setVerticalScrollBarPolicy(JScrollPane.VERTICAL_SCROLLBAR_ALWAYS);
vertical.getViewPort().setBackground(Color.white);
mainpage.setPreferredSize(new Dimension(500,400));
mainpage.removeAll();
rest.setPreferredSize(new Dimension(800,600));

//Inputting information into sidebar
JPanel TestPanel = new JPanel();
JLabel sidelabel = new JLabel("<html>* You can double click each name to view
more details on each patient");

TestPanel.setBackground(PanelBackground);
TestPanel.setPreferredSize(new Dimension(100,400));
TestPanel.setLayout(new BorderLayout(TestPanel, BorderLayout.PAGE_AXIS));
TestPanel.add(sidelabel);

rest.add(TestPanel); rest.add(mainpage);
rest.setBackground(blueBackground);

main_frame.add(heading); main_frame.add(rest);

main.add(ViewLabel);
main.add(vertical);
vertical.setMaximumSize(new Dimension(1200, 250));
mainpage.add(main);
mainpage.add(footer);
mainpage.setPreferredSize(new Dimension(650,400));
mainpage.setBackground(Color.WHITE);
main.setBackground(Color.WHITE);

MouseListener mouseListener = new MouseAdapter()
{
    public void mouseClicked(MouseEvent e)

```

```

        {
            try
            {
                //If the user clicks an item in the list twice then
                //redirect to view full patient details
                if (e.getClickCount() == 2)
                {
                    //Get selected value of the item clicked
                    selectedLeftUsername = TestList.getSelectedValue().toString();
                    index = TestList.locationToIndex(e.getPoint());
                    PaIDTest = PaID + index;
                    int test = selectedLeftUsername.indexOf(" ");
                    fileName = selectedLeftUsername.substring(0, test);

                    //Extracting data from within a string
                    String[] dot = selectedLeftUsername.split(" ");
                    lastone = dot[dot.length-1];
                    ending = lastone;

                    new PatientLastNameNurse();
                    main_frame.setVisible(false);

                }

                else
                {

                }

            }

            catch(Exception exc)
            {
                System.out.println(exc);
            }

        }

};

TestList.addMouseListener(mouseListener);

Print.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        try
        {
            //Message to user indicating they have selected the print

```

```

        option
        JOptionPane.showMessageDialog(frame, "You have selected
        the print button", "Printing",
        JOptionPane.INFORMATION_MESSAGE);
    }
    catch (Exception exc)
    {
        System.out.println(exc);
    }
}
});

AnotherPatient.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        try
        {
            //Redirect the user back to main search for patient page
            main_frame.repaint();
            main_frame.setVisible(false);
            new SearchPatientNurse();
        }
        catch (Exception exc)
        {
            System.out.println(exc);
        }
    }
});

BackMain.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        try
        {
            //Redirect user back to main menu for nurse
            main_frame.setVisible(false);
            new MainScreenNurse();
        }
        catch (Exception exc)
        {
            System.out.println(exc);
        }
    }
});

```

```

    });

    main_frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    main_frame.setVisible(true);
}

public String[] listPatients() throws Exception
{
    //Comparing the values entered for family name and patient first name
    //Comparing these values with values in the database
    //Create a list of these matches
    //Then return created list to frame to be viewed by user
    Class.forName("org.sqlite.JDBC");
    Connection conn =
    DriverManager.getConnection("jdbc:sqlite:NurseScheduling.sqlite");
    Statement stat = conn.createStatement();
    ResultSet rs = stat.executeQuery(String.format("select * from Patients2 where
    FName='%s' and PName='%s'", FaName, PaName));

    String[] rawModList = new String[200];
    int index = 0;
    while (rs.next())
    {
        rawModList[index] = rs.getString("PID") + " " + rs.getString("PName") + " " +
        rs.getString("FName");
        FaName = rs.getString("FName");
        PaID = rs.getString("PID");
        index++;
    }
    rs.close();
    conn.close();

    int numElements = 0;
    for (int i = 0; i < 200; i++)
    {
        if (rawModList[i] != null)
        {
            numElements++;
        }
    }

    String[] modList = new String[numElements];
    System.arraycopy(rawModList, 0, modList, 0, numElements);
}

```

```

        return modList;
    }

    //Return patient name
    public static String Get_Patient()
    {
        return ending;
    }

    //Return patient ID value for patient
    public static String Get_ID()
    {
        return fileName;
    }

    //Return main frame
    public static JFrame test_frame()
    {
        return main_frame;
    }

    public static void main(String args[]) throws Exception
    {
        new SearchPatientDetailsTestNurse();
    }
}

```

'Search Patient details' list details of selected patient – District Nurse (325):-

```

public class PatientLastNameNurse
{
    static String username;
    String PaName, FaName, TownDa, TreatmentDa, PatientTr, lastone, PaID;
    JFrame refreshFrame;
    JOptionPane frame = new JOptionPane();
    JFrame nurse_frame = new JFrame ("Bargoed Nurses");
    private JPanel heading;
    private ImageIcon image1;
    private JLabel label1;
    public static JPanel mainpage = new JPanel();
    JPanel rest = new JPanel(), sidebar = new JPanel();
    Color buttonBackground = new Color (234,234,234);
    Color blueBackground = new Color (7,32,157);
    Color PanelBackground = new Color (167, 165, 255);
}

```

```
Font f = new Font("Dialog", Font.PLAIN, 20);
```

```
public PatientLastNameNurse() throws Exception
```

```
{
```

```
    PaName = SearchPatientNurse.Get_Patient();  
    FaName = SearchPatientNurse.Get_Family();  
    TownDa = SearchPatientNurse.Get_Town();  
    lastone = SearchPatientDetailsTestNurse.Get_Patient();  
    PaID = SearchPatientDetailsTestNurse.Get_ID();  
    refreshFrame = SearchPatientDetailsTestNurse.test_frame();
```

```
    heading = new JPanel();  
    heading.setMaximumSize(new Dimension(700,150));  
    heading.setBackground(blueBackground);  
    image1 = new ImageIcon(getClass().getResource("Bar.jpg"));  
    label1 = new JLabel(image1);  
    heading.add(label1);
```

```
    JButton print = new JButton("Print");  
    JButton back = new JButton("Back to list of Patients");  
    nurse_frame.setSize(780,725);  
    nurse_frame.setLayout(new FlowLayout());  
    nurse_frame.setBackground(Color.WHITE);  
    mainpage.setPreferredSize(new Dimension(500,700));  
    mainpage.removeAll();  
    rest.setPreferredSize(new Dimension(800,580));  
    rest.setBackground(blueBackground);  
    JLabel test = new JLabel("Test");  
    nurse_frame.add(heading); nurse_frame.add(rest);
```

```
    JPanel TestPanel = new JPanel();  
    TestPanel.setBackground(PanelBackground);  
    TestPanel.setPreferredSize(new Dimension(100,560));  
    TestPanel.setLayout(new BorderLayout(TestPanel, BorderLayout.PAGE_AXIS));
```

```
    rest.add(TestPanel); rest.add(mainpage);
```

```
    JPanel moduleListPanel = new JPanel();  
    moduleListPanel.setLayout(new FlowLayout());  
    moduleListPanel.setPreferredSize(new Dimension(600,550));
```

```
    JLabel welcome = new JLabel("<html>Details on selected patient are:");  
    welcome.setPreferredSize(new Dimension(600, 20));  
    welcome.setHorizontalTextPosition(JLabel.LEFT);
```

```
moduleListPanel.add(welcome);

//Get information from the database which matches the selected name from
previous frame
Class.forName("org.sqlite.JDBC");
Connection conn =
DriverManager.getConnection("jdbc:sqlite:NurseScheduling.sqlite");
Statement stat = conn.createStatement();
ResultSet rs = stat.executeQuery(String.format("select * from Patients2 where
PName='%s' and FName='%s' and PID ='%s'", PaName, lastone, PaID));

JPanel txtPanel = new JPanel();
txtPanel.setLayout(null);
txtPanel.setLocation(20, 50);
txtPanel.setPreferredSize(new Dimension(200, 480));
txtPanel.setBackground(Color.WHITE);

moduleListPanel.add(txtPanel);
moduleListPanel.setBackground(Color.WHITE);

JPanel txtFieldPanel = new JPanel();
txtFieldPanel.setLayout(null);
txtFieldPanel.setLocation(150, 60);
txtFieldPanel.setPreferredSize(new Dimension(390, 480));
txtFieldPanel.setBackground(Color.WHITE);
moduleListPanel.add(txtFieldPanel);

JLabel patientName = new JLabel("Patient Name:");
patientName.setLocation(0, 0);
patientName.setSize(300, 30);
patientName.setHorizontalTextPosition(JLabel.LEFT);
txtPanel.add(patientName);

JLabel familyName = new JLabel("Family Name:");
familyName.setLocation(0,40);
familyName.setSize(300,30);
familyName.setHorizontalTextPosition(JLabel.LEFT);
txtPanel.add(familyName);

JLabel Address = new JLabel("Address:");
Address.setLocation(0,80);
Address.setSize(300,30);
Address.setHorizontalTextPosition(JLabel.LEFT);
txtPanel.add(Address);
```

```
JLabel DTreatment = new JLabel("Detailed Treatment:");
DTreatment.setLocation(0,160);
DTreatment.setSize(300,30);
DTreatment.setHorizontalTextPosition(JLabel.LEFT);
txtPanel.add(DTreatment);
```

```
JLabel Town = new JLabel("Town:");
Town.setLocation(0,120);
Town.setSize(300,30);
Town.setHorizontalTextPosition(JLabel.LEFT);
txtPanel.add(Town);
```

```
JLabel Allergy = new JLabel("Allergy:");
Allergy.setLocation(0,200);
Allergy.setSize(300,30);
Allergy.setHorizontalTextPosition(JLabel.LEFT);
txtPanel.add(Allergy);
```

```
JLabel ExplainAllergy = new JLabel("ExplainAllergy:");
ExplainAllergy.setLocation(0,240);
ExplainAllergy.setSize(300,30);
ExplainAllergy.setHorizontalTextPosition(JLabel.LEFT);
txtPanel.add(ExplainAllergy);
```

```
JLabel KeyCode = new JLabel("Key Code:");
KeyCode.setLocation(0,280);
KeyCode.setSize(300,30);
KeyCode.setHorizontalTextPosition(JLabel.LEFT);
txtPanel.add(KeyCode);
```

```
JLabel Equipment = new JLabel("Equipment:");
Equipment.setLocation(0,320);
Equipment.setSize(300,30);
Equipment.setHorizontalTextPosition(JLabel.LEFT);
txtPanel.add(Equipment);
```

```
JLabel Telephone = new JLabel("Telephone:");
Telephone.setLocation(0,360);
Telephone.setSize(300,30);
Telephone.setHorizontalTextPosition(JLabel.LEFT);
txtPanel.add(Telephone);
```

```
JLabel Doctor = new JLabel("Doctor:");
Doctor.setLocation(0,400);
Doctor.setSize(300,30);
```

```
Doctor.setHorizontalTextPosition(JLabel.LEFT);  
txtPanel.add(Doctor);
```

```
JLabel Bloods = new JLabel("Bloods:");  
Bloods.setLocation(0,440);  
Bloods.setSize(300,30);  
Bloods.setHorizontalTextPosition(JLabel.LEFT);  
txtPanel.add(Bloods);
```

```
JTextField UsernameField = new JTextField(rs.getString("PName"));  
UsernameField.setEditable(false);  
UsernameField.setLocation(0, 0);  
UsernameField.setSize(390, 30);  
UsernameField.setBackground(PanelBackground);  
txtFieldPanel.add(UsernameField);
```

```
JTextField NurseNameField = new JTextField(rs.getString("FName"));  
NurseNameField.setEditable(false);  
NurseNameField.setLocation(0, 40);  
NurseNameField.setSize(390, 30);  
NurseNameField.setBackground(PanelBackground);  
txtFieldPanel.add(NurseNameField);
```

```
final JTextArea addressArea = new JTextArea(rs.getString("PAddress"));  
addressArea.setSize(390, 30);  
addressArea.setEditable(false);  
addressArea.setBackground(PanelBackground);  
addressArea.setLinewrap(true);
```

```
final JScrollPane scrollPane = new JScrollPane(addressArea);  
scrollPane.setLocation(0, 80);  
scrollPane.setSize(390, 30);  
txtFieldPanel.add(scrollPane);
```

```
JTextField TownField = new JTextField(rs.getString("PTown"));  
TownField.setEditable(false);  
TownField.setLocation(0, 120);  
TownField.setSize(390, 30);  
TownField.setBackground(PanelBackground);  
txtFieldPanel.add(TownField);
```

```
final JTextArea addressArea2 = new JTextArea(rs.getString("DTreatment"));  
addressArea2.setSize(390, 30);  
addressArea2.setEditable(false);  
addressArea2.setBackground(PanelBackground);
```

```
addressArea2.setLinewrap(true);
```

```
final JScrollPane scrollPane2 = new JScrollPane(addressArea2);  
scrollPane2.setLocation(0, 160);  
scrollPane2.setSize(390, 30);  
txtFieldPanel.add(scrollPane2);
```

```
JTextField AllergyField = new JTextField(rs.getString("Allergy"));  
AllergyField.setEditable(false);  
AllergyField.setLocation(0, 200);  
AllergyField.setSize(390, 30);  
AllergyField.setBackground(PanelBackground);  
txtFieldPanel.add(AllergyField);
```

```
final JTextArea addressAreaAllergy = new JTextArea(rs.getString("ExplainAllergy"));  
addressAreaAllergy.setSize(390, 30);  
addressAreaAllergy.setEditable(false);  
addressAreaAllergy.setBackground(PanelBackground);  
addressAreaAllergy.setLinewrap(true);
```

```
final JScrollPane scrollPane3 = new JScrollPane(addressAreaAllergy);  
scrollPane3.setLocation(0, 240);  
scrollPane3.setSize(390, 30);  
txtFieldPanel.add(scrollPane3);
```

```
JTextField KeyCodeField = new JTextField(rs.getString("Kcode"));  
KeyCodeField.setEditable(false);  
KeyCodeField.setLocation(0, 280);  
KeyCodeField.setSize(390, 30);  
KeyCodeField.setBackground(PanelBackground);  
txtFieldPanel.add(KeyCodeField);
```

```
JTextField EquipmentField = new JTextField(rs.getString("Equipment"));  
EquipmentField.setEditable(false);  
EquipmentField.setLocation(0, 320);  
EquipmentField.setSize(390, 30);  
EquipmentField.setBackground(PanelBackground);  
txtFieldPanel.add(EquipmentField);
```

```
JTextField TelephoneField = new JTextField(rs.getString("PTelNumber"));  
TelephoneField.setEditable(false);  
TelephoneField.setLocation(0, 360);  
TelephoneField.setSize(390, 30);
```

```

TelephoneField.setBackground(PanelBackground);
txtFieldPanel.add(TelephoneField);

JTextField DoctorField = new JTextField(rs.getString("Doctor"));
DoctorField.setEditable(false);
DoctorField.setLocation(0, 400);
DoctorField.setSize(390, 30);
DoctorField.setBackground(PanelBackground);
txtFieldPanel.add(DoctorField);

JTextField BloodsField = new JTextField(rs.getString("Bloods"));
BloodsField.setEditable(false);
BloodsField.setLocation(0, 440);
BloodsField.setSize(390, 30);
BloodsField.setBackground(PanelBackground);
txtFieldPanel.add(BloodsField);

moduleListPanel.add(print);
moduleListPanel.add(back);

mainpage.add(moduleListPanel);
mainpage.setPreferredSize(new Dimension(650,560));
mainpage.setBackground(Color.WHITE);

back.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        try
        {
            nurse_frame.setVisible(false);
            refreshFrame.setVisible(true);
        }
        catch (Exception exc)
        {
            System.out.println(exc);
        }
    }
});

print.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        try

```

```

        {
            //Message displayed to used indicating they have selected
            the print option
            JOptionPane.showMessageDialog(frame, "You have selected
            the print button","Printing",
            JOptionPane.INFORMATION_MESSAGE);
        }
        catch (Exception exc)
        {
            System.out.println(exc);
        }
    }
});
nurse_frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
nurse_frame.setVisible(true);
}

public static void main(String args[]) throws Exception
{
    new PatientLastNameNurse();
}
}

```

'Search Patient details' list of matches for patient first name and town – District Nurse (280):-

```

public class SearchPatientDetailsTownTestNurse
{
    String PaName, FaName, TownDa, TreatmentDa;

    JOptionPane frame = new JOptionPane();
    private static JFrame main_frame = new JFrame ("Bargoed Nurses");
    JPanel main = new JPanel();
    private JPanel heading, footer;
    private ImageIcon image1;
    private JLabel label1, ViewLabel, StaffName;
    public static JPanel mainpage = new JPanel();
    JScrollPane vertical = new JScrollPane();
    JPanel rest = new JPanel(), sidebar = new JPanel();
    Color buttonBackground = new Color (234,234,234);
    Color blueBackground = new Color (7,32,157);
    Font f = new Font("Dialog", Font.PLAIN, 26);
    Color PanelBackground = new Color (167, 165, 255);
    DefaultListModel data;
}

```

```

final JList TestList = new JList();

static String selectedLeftUsername;
static String lastone;
private static String ending;
private static String PaID;
private static String fileName;

public SearchPatientDetailsTownTestNurse() throws Exception
{
    PaName = SearchPatientNurse.Get_Patient();
    FaName = SearchPatientNurse.Get_Family();
    TownDa = SearchPatientNurse.Get_Town();
    TreatmentDa = SearchPatientNurse.Get_Treatment();

    heading = new JPanel();
    heading.setMaximumSize(new Dimension(700,150));
    heading.setBackground(blueBackground);
    image1 = new ImageIcon(getClass().getResource("Bar.jpg"));
    label1 = new JLabel(image1);
    heading.add(label1);

    footer = new JPanel();
    footer.setMaximumSize(new Dimension(800, 300));
    footer.setBackground(Color.GREEN);
    JButton Print = new JButton("Print Details");
    JButton AnotherPatient = new JButton("Another Patient");
    JButton BackMain = new JButton("Back to Main Screen");
    footer.add(Print); footer.add(AnotherPatient); footer.add(BackMain);
    footer.setBackground(Color.WHITE);

    ViewLabel = new JLabel("Here are the patients' details found:-");
    ViewLabel.setFont(f);

    main_frame.setSize(780,570);
    main_frame.setLayout(new FlowLayout());
    main_frame.setBackground(Color.WHITE);

    main.setLayout(new BoxLayout(main, BoxLayout.PAGE_AXIS));
    main.setPreferredSize(new Dimension(600,350));

    final JList TestList = new JList(listPatients());
    TestList.setSelectionMode(ListSelectionModel.SINGLE_SELECTION);
    TestList.setSize(200, 500);

```

```

TestList.setSelectedIndex(0);
TestList.setVisibleRowCount(6);

JScrollPane vertical = new JScrollPane(TestList);
vertical.setVerticalScrollBarPolicy(JScrollPane.VERTICAL_SCROLLBAR_ALWAYS);
vertical.getViewport().setBackground(Color.white);
mainpage.setPreferredSize(new Dimension(500,400));
mainpage.removeAll();
rest.setPreferredSize(new Dimension(800,600));

//Inputting information into sidebar
JPanel TestPanel = new JPanel();
JLabel sidelabel = new JLabel("<html>* You can double click each name to view
more details on each patient");

TestPanel.setBackground(PanelBackground);
TestPanel.setPreferredSize(new Dimension(100,400));
TestPanel.setLayout(new BorderLayout(TestPanel, BorderLayout.PAGE_AXIS));
TestPanel.add(sidelabel);
rest.add(TestPanel); rest.add(mainpage);
rest.setBackground(blueBackground);
main_frame.add(heading); main_frame.add(rest);
main.add(ViewLabel);
main.add(vertical);
vertical.setMaximumSize(new Dimension(1200, 250));
mainpage.add(main);
mainpage.add(footer);
mainpage.setPreferredSize(new Dimension(650,400));
mainpage.setBackground(Color.WHITE);
main.setBackground(Color.WHITE);

MouseListener mouseListener = new MouseAdapter()
{
    public void mouseClicked(MouseEvent e)
    {
        try
        {
            //If the user clicks an item in the list twice then redirect to
            view full patient details
            if (e.getClickCount() == 2)
            {
                //Get selected value of the item clicked
                selectedLeftUsername = TestList.getSelectedValue().toString();
                final int index = TestList.locationToIndex(e.getPoint());
                int test = selectedLeftUsername.indexOf(" ");
            }
        }
    }
}

```

```

        fileName = selectedLeftUsername.substring(0, test);

        //Extracting data from within a String
        String[] dot = selectedLeftUsername.split(".");
        lastone = dot[dot.length-1];
        ending = lastone;
        new PatientTownNurse();
        main_frame.setVisible(false);
    }

    else
    {

    }
}
catch(Exception exc)
{
    System.out.println(exc);
}
}
};

TestList.addMouseListener(mouseListener);

Print.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        try
        {
            //Message is displayed to user indicating they have selected the
            print button
            JOptionPane.showMessageDialog(frame, "You have selected the
            print button", "Printing", JOptionPane.INFORMATION_MESSAGE);

        }
        catch (Exception exc)
        {
            System.out.println(exc);
        }
    }
});

AnotherPatient.addActionListener(new ActionListener()
{

```

```

        public void actionPerformed(ActionEvent e)
        {
            try
            {
                //Redirect user back to main search patients frame
                main_frame.setVisible(false);
                new SearchPatientNurse();
            }
            catch (Exception exc)
            {
                System.out.println(exc);
            }
        }
    });

    BackMain.addActionListener(new ActionListener()
    {
        public void actionPerformed(ActionEvent e)
        {
            try
            {
                //Redirect user back to main nurse screen
                main_frame.setVisible(false);
                new MainScreenNurse();
            }
            catch (Exception exc)
            {
                System.out.println(exc);
            }
        }
    });

    main_frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    main_frame.setVisible(true);
}

public String[] listPatients() throws Exception
{
    //Comparing the values entered for town and patient first name
    //Comparing these values with values in the database
    //Create a list of these matches
    //Then return created list to frame to be viewed by user
    Class.forName("org.sqlite.JDBC");
    Connection conn =
    DriverManager.getConnection("jdbc:sqlite:NurseScheduling.sqlite");
    Statement stat = conn.createStatement();
}

```

```

ResultSet rs = stat.executeQuery(String.format("select * from Patients2 where
PTown='%s' and PName='%s'", TownDa, PaName));

String[] rawModList = new String[200];
int index = 0;
while (rs.next())
{
    rawModList[index] = rs.getString("PID") + " " + rs.getString("PName") + " " +
rs.getString("FName");
    PaID = rs.getString("PID");
    index++;
}
rs.close();
conn.close();

int numElements = 0;
for (int i = 0; i < 200; i++)
{
    if (rawModList[i] != null)
    {
        numElements++;
    }
}
String[] modList = new String[numElements];
System.arraycopy(rawModList, 0, modList, 0, numElements);

return modList;
}

//Return patient
public static String Get_Patient()
{
    return ending;
}

//Return patient ID
public static String Get_ID()
{
    return fileName;
}

//Return the main frame
public static JFrame test_frame()
{
    return main_frame;
}

```

```

    }

    public static void main(String args[]) throws Exception
    {
        new SearchPatientDetailsTownTestNurse();
    }
}

```

'Search Patient details' list details of selected patient – District Nurse (324):-

```

public class PatientTownNurse
{
    static String username;

    String PaName, FaName, TownDa, lastone, PaID;
    JFrame refreshFrame;

    JOptionPane frame = new JOptionPane();
    JFrame nurse_frame = new JFrame ("Bargoed Nurses");
    private JPanel heading;
    private ImageIcon image1;
    private JLabel label1;
    public static JPanel mainpage = new JPanel();
    JPanel rest = new JPanel(), sidebar = new JPanel();
    Color buttonBackground = new Color (234,234,234);
    Color blueBackground = new Color (7,32,157);
    Color PanelBackground = new Color (167, 165, 255);
    Font f = new Font("Dialog", Font.PLAIN, 20);

    public PatientTownNurse() throws Exception
    {
        PaName = SearchPatientNurse.Get_Patient();
        FaName = SearchPatientNurse.Get_Family();
        TownDa = SearchPatientNurse.Get_Town();
        lastone = SearchPatientDetailsTownTestNurse.Get_Patient();
        PaID = SearchPatientDetailsTownTestNurse.Get_ID();
        refreshFrame = SearchPatientDetailsTownTestNurse.test_frame();

        heading = new JPanel();
        heading.setMaximumSize(new Dimension(700,150));
        heading.setBackground(blueBackground);
        image1 = new ImageIcon(getClass().getResource("Bar.jpg"));
        label1 = new JLabel(image1);
    }
}

```

```

heading.add(label1);

JButton print = new JButton("Print");
JButton back = new JButton("Back to list of Patients");
nurse_frame.setSize(780,725);
nurse_frame.setLayout(new FlowLayout());
nurse_frame.setBackground(Color.WHITE);

mainpage.setPreferredSize(new Dimension(500,700));
mainpage.removeAll();

rest.setPreferredSize(new Dimension(800,580));
rest.setBackground(blueBackground);
nurse_frame.add(heading); nurse_frame.add(rest);
JPanel TestPanel = new JPanel();
TestPanel.setBackground(PanelBackground);
TestPanel.setPreferredSize(new Dimension(100,560));
TestPanel.setLayout(new BorderLayout(TestPanel, BorderLayout.PAGE_AXIS));
rest.add(TestPanel); rest.add(mainpage);

JPanel moduleListPanel = new JPanel();
moduleListPanel.setLayout(new FlowLayout());
moduleListPanel.setPreferredSize(new Dimension(600,550));
JLabel welcome = new JLabel("<html>Details on selected patient are:");
welcome.setPreferredSize(new Dimension(600, 20));
welcome.setHorizontalTextPosition(JLabel.LEFT);
moduleListPanel.add(welcome);

//Get information from the database which matches the selected name from
previous frame
Class.forName("org.sqlite.JDBC");
Connection conn =
DriverManager.getConnection("jdbc:sqlite:NurseScheduling.sqlite");
Statement stat = conn.createStatement();
ResultSet rs = stat.executeQuery(String.format("select * from Patients2 where
PName='%s' and FName='%s' and PID='%s'", PaName, lastone, PaID));

JPanel txtPanel = new JPanel();
txtPanel.setLayout(null);
txtPanel.setLocation(20, 50);
txtPanel.setPreferredSize(new Dimension(200, 480));
txtPanel.setBackground(Color.WHITE);

moduleListPanel.add(txtPanel);
moduleListPanel.setBackground(Color.WHITE);

```

```
JPanel txtFieldPanel = new JPanel();
txtFieldPanel.setLayout(null);
txtFieldPanel.setLocation(150, 60);
txtFieldPanel.setPreferredSize(new Dimension(390, 480));
txtFieldPanel.setBackground(Color.WHITE);
moduleListPanel.add(txtFieldPanel);
```

```
JLabel patientName = new JLabel("Patient Name:");
patientName.setLocation(0, 0);
patientName.setSize(300, 30);
patientName.setHorizontalTextPosition(JLabel.LEFT);
txtPanel.add(patientName);
```

```
JLabel familyName = new JLabel("Family Name: ");
familyName.setLocation(0,40);
familyName.setSize(300,30);
familyName.setHorizontalTextPosition(JLabel.LEFT);
txtPanel.add(familyName);
```

```
JLabel Address = new JLabel("Address:");
Address.setLocation(0,80);
Address.setSize(300,30);
Address.setHorizontalTextPosition(JLabel.LEFT);
txtPanel.add(Address);
```

```
JLabel DTreatment = new JLabel("Detailed Treatment:");
DTreatment.setLocation(0,160);
DTreatment.setSize(300,30);
DTreatment.setHorizontalTextPosition(JLabel.LEFT);
txtPanel.add(DTreatment);
```

```
JLabel Town = new JLabel("Town:");
Town.setLocation(0,120);
Town.setSize(300,30);
Town.setHorizontalTextPosition(JLabel.LEFT);
txtPanel.add(Town);
```

```
JLabel Allergy = new JLabel("Allergy:");
Allergy.setLocation(0,200);
Allergy.setSize(300,30);
Allergy.setHorizontalTextPosition(JLabel.LEFT);
txtPanel.add(Allergy);
```

```
JLabel ExplainAllergy = new JLabel("ExplainAllergy:");
```

```
ExplainAllergy.setLocation(0,240);
ExplainAllergy.setSize(300,30);
ExplainAllergy.setHorizontalTextPosition(JLabel.LEFT);
txtPanel.add(ExplainAllergy);
```

```
JLabel KeyCode = new JLabel("Key Code:");
KeyCode.setLocation(0,280);
KeyCode.setSize(300,30);
KeyCode.setHorizontalTextPosition(JLabel.LEFT);
txtPanel.add(KeyCode);
```

```
JLabel Equipment = new JLabel("Equipment:");
Equipment.setLocation(0,320);
Equipment.setSize(300,30);
Equipment.setHorizontalTextPosition(JLabel.LEFT);
txtPanel.add(Equipment);
```

```
JLabel Telephone = new JLabel("Telephone:");
Telephone.setLocation(0,360);
Telephone.setSize(300,30);
Telephone.setHorizontalTextPosition(JLabel.LEFT);
txtPanel.add(Telephone);
```

```
JLabel Doctor = new JLabel("Doctor:");
Doctor.setLocation(0,400);
Doctor.setSize(300,30);
Doctor.setHorizontalTextPosition(JLabel.LEFT);
txtPanel.add(Doctor);
```

```
JLabel Bloods = new JLabel("Bloods:");
Bloods.setLocation(0,440);
Bloods.setSize(300,30);
Bloods.setHorizontalTextPosition(JLabel.LEFT);
txtPanel.add(Bloods);
```

```
JTextField UsernameField = new JTextField(rs.getString("PName"));
UsernameField.setEditable(false);
UsernameField.setLocation(0, 0);
UsernameField.setSize(390, 30);
UsernameField.setBackground(PanelBackground);
txtFieldPanel.add(UsernameField);
```

```
JTextField NurseNameField = new JTextField(rs.getString("FName"));
NurseNameField.setEditable(false);
NurseNameField.setLocation(0, 40);
```

```
NurseNameField.setSize(390, 30);
NurseNameField.setBackground(PanelBackground);
txtFieldPanel.add(NurseNameField);

final JTextArea addressArea = new JTextArea(rs.getString("PAddress"));
addressArea.setSize(390, 30);
addressArea.setEditable(false);
addressArea.setBackground(PanelBackground);
addressArea.setLineWrap(true);

final JScrollPane scrollPane = new JScrollPane(addressArea);
scrollPane.setLocation(0, 80);
scrollPane.setSize(390, 30);
txtFieldPanel.add(scrollPane);

JTextField TownField = new JTextField(rs.getString("PTown"));
TownField.setEditable(false);
TownField.setLocation(0, 120);
TownField.setSize(390, 30);
TownField.setBackground(PanelBackground);
txtFieldPanel.add(TownField);

final JTextArea addressArea2 = new JTextArea(rs.getString("DTreatment"));
addressArea2.setSize(390, 30);
addressArea2.setEditable(false);
addressArea2.setBackground(PanelBackground);
addressArea2.setLineWrap(true);

final JScrollPane scrollPane2 = new JScrollPane(addressArea2);
scrollPane2.setLocation(0, 160);
scrollPane2.setSize(390, 30);
txtFieldPanel.add(scrollPane2);

JTextField AllergyField = new JTextField(rs.getString("Allergy"));
AllergyField.setEditable(false);
AllergyField.setLocation(0, 200);
AllergyField.setSize(390, 30);
AllergyField.setBackground(PanelBackground);
txtFieldPanel.add(AllergyField);

final JTextArea addressAreaAllergy = new JTextArea(rs.getString("ExplainAllergy"));
addressAreaAllergy.setSize(390, 30);
addressAreaAllergy.setEditable(false);
addressAreaAllergy.setBackground(PanelBackground);
```

```
addressAreaAllergy.setLinewrap(true);

final JScrollPane scrollPane3 = new JScrollPane(addressAreaAllergy);
scrollPane3.setLocation(0, 240);
//scrollPane.setEditable(false);
scrollPane3.setSize(390, 30);
txtFieldPanel.add(scrollPane3);

JTextField KeyCodeField = new JTextField(rs.getString("Kcode"));
KeyCodeField.setEditable(false);
KeyCodeField.setLocation(0, 280);
KeyCodeField.setSize(390, 30);
KeyCodeField.setBackground(PanelBackground);
txtFieldPanel.add(KeyCodeField);

JTextField EquipmentField = new JTextField(rs.getString("Equipment"));
EquipmentField.setEditable(false);
EquipmentField.setLocation(0, 320);
EquipmentField.setSize(390, 30);
EquipmentField.setBackground(PanelBackground);
txtFieldPanel.add(EquipmentField);

JTextField TelephoneField = new JTextField(rs.getString("PTelNumber"));
TelephoneField.setEditable(false);
TelephoneField.setLocation(0, 360);
TelephoneField.setSize(390, 30);
TelephoneField.setBackground(PanelBackground);
txtFieldPanel.add(TelephoneField);

JTextField DoctorField = new JTextField(rs.getString("Doctor"));
DoctorField.setEditable(false);
DoctorField.setLocation(0, 400);
DoctorField.setSize(390, 30);
DoctorField.setBackground(PanelBackground);
txtFieldPanel.add(DoctorField);

JTextField BloodsField = new JTextField(rs.getString("Bloods"));
BloodsField.setEditable(false);
BloodsField.setLocation(0, 440);
BloodsField.setSize(390, 30);
BloodsField.setBackground(PanelBackground);
txtFieldPanel.add(BloodsField);

moduleListPanel.add(print);
moduleListPanel.add(back);
```

```

mainpage.add(moduleListPanel);
mainpage.setPreferredSize(new Dimension(650,560));
mainpage.setBackground(Color.WHITE);

back.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        try
        {
            //Redirect back to list of found matches
            nurse_frame.setVisible(false);
            refreshFrame.setVisible(true);
        }
        catch (Exception exc)
        {
            System.out.println(exc);
        }
    }
});

print.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        try
        {
            //Message to user indicating they have selected the print option
            JOptionPane.showMessageDialog(frame, "You have selected the
            print button", "Printing", JOptionPane.INFORMATION_MESSAGE);
        }
        catch (Exception exc)
        {
            System.out.println(exc);
        }
    }
});

nurse_frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
nurse_frame.setVisible(true);
}

```

```

public static void main(String args[]) throws Exception
{
    new PatientTownNurse();

}
}

```

'Search Patient details' list of matches for patient first name and treatment – District Nurse (281):-

```

public class SearchPatientDetailsTreatmentNurse
{
    String PaName, FaName, TownDa, TreatmentDa;

    JOptionPane frame = new JOptionPane();
    private static JFrame main_frame = new JFrame ("Bargoed Nurses");
    JPanel main = new JPanel();
    private JPanel heading, footer;
    private ImageIcon image1;
    private JLabel label1, ViewLabel, StaffName;
    public static JPanel mainpage = new JPanel();
    JScrollPane vertical = new JScrollPane();
    JPanel rest = new JPanel(), sidebar = new JPanel();
    Color buttonBackground = new Color (234,234,234);
    Color blueBackground = new Color (7,32,157);
    Font f = new Font("Dialog", Font.PLAIN, 26);
    Color PanelBackground = new Color (167, 165, 255);
    DefaultListModel data;
    final JList TestList = new JList();

    static String selectedLeftUsername;
    static String lastone;
    private static String ending;
    private static String PaID;
    private static String fileName;

    public SearchPatientDetailsTreatmentNurse() throws Exception
    {
        PaName = SearchPatientNurse.Get_Patient();
        FaName = SearchPatientNurse.Get_Family();
        TownDa = SearchPatientNurse.Get_Town();
        TreatmentDa = SearchPatientNurse.Get_Treatment();

        heading = new JPanel();
    }
}

```

```

heading.setMaximumSize(new Dimension(700,150));
heading.setBackground(blueBackground);
image1 = new ImageIcon(getClass().getResource("Bar.jpg"));
label1 = new JLabel(image1);
heading.add(label1);

footer = new JPanel();
footer.setMaximumSize(new Dimension(800, 300));
footer.setBackground(Color.GREEN);
JButton Print = new JButton("Print Details");
JButton AnotherPatient = new JButton("Another Patient");
JButton BackMain = new JButton("Back to Main Screen");
footer.add(Print); footer.add(AnotherPatient); footer.add(BackMain);
footer.setBackground(Color.WHITE);

ViewLabel = new JLabel("Here are the patients' details found:-");
ViewLabel.setFont(f);
main_frame.setSize(780,570);
main_frame.setLayout(new FlowLayout());
main_frame.setBackground(Color.WHITE);
main.setLayout(new BorderLayout(main, BorderLayout.PAGE_AXIS));
main.setPreferredSize(new Dimension(600,350));

final JList TestList = new JList(listPatients());
TestList.setSelectionMode(ListSelectionModel.SINGLE_SELECTION);
TestList.setSize(200, 500);
TestList.setSelectedIndex(0);
TestList.setVisibleRowCount(6);

JScrollPane vertical = new JScrollPane(TestList);
vertical.setVerticalScrollBarPolicy(JScrollPane.VERTICAL_SCROLLBAR_ALWAYS);
vertical.getViewPort().setBackground(Color.white);
mainpage.setPreferredSize(new Dimension(500,400));
mainpage.removeAll();
rest.setPreferredSize(new Dimension(800,600));

//Inputting information into sidebar
JPanel TestPanel = new JPanel();
JLabel sidelabel = new JLabel("<html>* You can double click each name to view
more details on each patient");

TestPanel.setBackground(PanelBackground);
TestPanel.setPreferredSize(new Dimension(100,400));
TestPanel.setLayout(new BorderLayout(TestPanel, BorderLayout.PAGE_AXIS));
TestPanel.add(sidelabel);

```

```

rest.add(TestPanel); rest.add(mainpage);
rest.setBackground(blueBackground);
main_frame.add(heading); main_frame.add(rest);

main.add(ViewLabel);
main.add(vertical);
vertical.setMaximumSize(new Dimension(1200, 250));
mainpage.add(main);
mainpage.add/footer);
mainpage.setPreferredSize(new Dimension(650,400));
mainpage.setBackground(Color.WHITE);
main.setBackground(Color.WHITE);

MouseListener mouseListener = new MouseAdapter()
{
    public void mouseClicked(MouseEvent e)
    {
        try
        {
            //If the user clicks an item in the list twice then redirect to
            view full patient details
            if (e.getClickCount() == 2)
            {
                //Get selected value of the item clicked
                selectedLeftUsername = TestList.getSelectedValue().toString();
                final int index = TestList.locationToIndex(e.getPoint());
                int test = selectedLeftUsername.indexOf(" ");
                fileName = selectedLeftUsername.subString(0, test);

                //Extracting data from within a String
                String[] dot = selectedLeftUsername.split(" ");
                lastone = dot[dot.length-1];
                ending = lastone;

                new PatientTreatmentNurse();
                main_frame.setVisible(false);
            }
            else
            {
            }
        }
        catch (Exception exc)
        {
            System.out.println(exc);
        }
    }
}

```

```

    }
};

TestList.addMouseListener(mouseListener);

Print.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        try
        {
            //Message displayed to user indicating the print option has
            //been selected
            JOptionPane.showMessageDialog(frame, "You have selected
            the print button", "Printing",
            JOptionPane.INFORMATION_MESSAGE);
        }
        catch (Exception exc)
        {
            System.out.println(exc);
        }
    }
});

AnotherPatient.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        try
        {
            //Redirect main search patients frame
            main_frame.setVisible(false);
            new SearchPatientNurse();
        }
        catch (Exception exc)
        {
            System.out.println(exc);
        }
    }
});

BackMain.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {

```

```

        try
        {
            //Redirect main nurse frame
            main_frame.setVisible(false);
            new MainScreenNurse();
        }
        catch (Exception exc)
        {
            System.out.println(exc);
        }
    }
});
main_frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
main_frame.setVisible(true);
}

```

```

public String[] listPatients() throws Exception
{
    //Comparing the values entered for treatment and patient first name
    //Comparing these values with values in the database
    //Create a list of these matches
    //Then return created list to frame to be viewed by user
    Class.forName("org.sqlite.JDBC");
    Connection conn =
    DriverManager.getConnection("jdbc:sqlite:NurseScheduling.sqlite");
    Statement stat = conn.createStatement();
    ResultSet rs = stat.executeQuery(String.format("select * from Patients2 where
    Treatment='%s' and PName='%s'", TreatmentDa, PaName));

    String[] rawModList = new String[200];
    int index = 0;
    while (rs.next())
    {
        rawModList[index] = rs.getString("PID") + " " + rs.getString("PName") + " " +
        rs.getString("FName");
        FaName = rs.getString("FName");
        index++;
    }
    rs.close();
    conn.close();

    int numElements = 0;
    for (int i = 0; i < 200; i++)
    {
        if (rawModList[i] != null)

```

```

        {
            numElements++;
        }
    }
    String[] modList = new String[numElements];

    System.arraycopy(rawModList, 0, modList, 0, numElements);

    return modList;
}

//Return patient
public static String Get_Patient()
{
    return ending;
}

//Return patient ID
public static String Get_ID()
{
    return fileName;
}

//Return main frame
public static JFrame test_frame()
{
    return main_frame;
}

public static void main(String args[]) throws Exception
{
    new SearchPatientDetailsTreatmentNurse();
}
}

```

'Search Patient details' list details of selected patient – District Nurse (323):-

```

public class PatientTreatmentNurse
{
    static String username;
    String PaName, FaName, TownDa, TreatmentDa, lastone, PaID;
    JFrame testFrame;
    JOptionPane frame = new JOptionPane();
    JFrame nurse_frame = new JFrame ("Bargoed Nurses");
    private JPanel heading;
}

```

```

private ImageIcon image1;
private JLabel label1;
public static JPanel mainpage = new JPanel();
JPanel rest = new JPanel(), sidebar = new JPanel();
Color buttonBackground = new Color (234,234,234);
Color blueBackground = new Color (7,32,157);
Color PanelBackground = new Color (167, 165, 255);
Font f = new Font("Dialog", Font.PLAIN, 20);

public PatientTreatmentNurse() throws Exception
{
    PaName = SearchPatientNurse.Get_Patient();
    FaName = SearchPatientNurse.Get_Family();
    TownDa = SearchPatientNurse.Get_Town();
    TreatmentDa = SearchPatientNurse.Get_Treatment();
    lastone = SearchPatientDetailsTreatmentNurse.Get_Patient();
    PaID = SearchPatientDetailsTreatmentNurse.Get_ID();
    testFrame = SearchPatientDetailsTreatmentNurse.test_frame();

    heading = new JPanel();
    heading.setMaximumSize(new Dimension(700,150));
    heading.setBackground(blueBackground);
    image1 = new ImageIcon(getClass().getResource("Bar.jpg"));
    label1 = new JLabel(image1);
    heading.add(label1);

    JButton print = new JButton("Print");
    JButton back = new JButton("Back to list of Patients");
    nurse_frame.setSize(780,725);
    nurse_frame.setLayout(new FlowLayout());
    nurse_frame.setBackground(Color.WHITE);
    mainpage.setPreferredSize(new Dimension(500,700));
    mainpage.removeAll();
    rest.setPreferredSize(new Dimension(800,580));
    rest.setBackground(blueBackground);
    JLabel test = new JLabel("Test");

    nurse_frame.add(heading); nurse_frame.add(rest);

    JPanel TestPanel = new JPanel();
    TestPanel.setBackground(PanelBackground);
    TestPanel.setPreferredSize(new Dimension(100,560));
    TestPanel.setLayout(new BorderLayout(TestPanel, BorderLayout.PAGE_AXIS));
    rest.add(TestPanel); rest.add(mainpage);
    JPanel moduleListPanel = new JPanel();

```

```

moduleListPanel.setLayout(new FlowLayout());
moduleListPanel.setPreferredSize(new Dimension(600,550));
JLabel welcome = new JLabel("<html>Details on selected patient are:");
welcome.setPreferredSize(new Dimension(600, 20));
welcome.setHorizontalTextPosition(JLabel.LEFT);
moduleListPanel.add(welcome);

//Get information from the database to match the selected item
Class.forName("org.sqlite.JDBC");
Connection conn =
DriverManager.getConnection("jdbc:sqlite:NurseScheduling.sqlite");
Statement stat = conn.createStatement();
ResultSet rs = stat.executeQuery(String.format("select * from Patients2 where
PName='%s' and FName='%s' and PID='%s'", PaName, lastone, PaID));

JPanel txtPanel = new JPanel();
txtPanel.setLayout(null);
txtPanel.setLocation(20, 50);
txtPanel.setPreferredSize(new Dimension(200, 480));
txtPanel.setBackground(Color.WHITE);

moduleListPanel.add(txtPanel);
moduleListPanel.setBackground(Color.WHITE);

JPanel txtFieldPanel = new JPanel();
txtFieldPanel.setLayout(null);
txtFieldPanel.setLocation(150, 60);
txtFieldPanel.setPreferredSize(new Dimension(390, 480));
txtFieldPanel.setBackground(Color.WHITE);
moduleListPanel.add(txtFieldPanel);

JLabel patientName = new JLabel("Patient Name:");
patientName.setLocation(0, 0);
patientName.setSize(300, 30);
patientName.setHorizontalTextPosition(JLabel.LEFT);
txtPanel.add(patientName);

JLabel familyName = new JLabel("Family Name:");
familyName.setLocation(0,40);
familyName.setSize(300,30);
familyName.setHorizontalTextPosition(JLabel.LEFT);
txtPanel.add(familyName);

JLabel Address = new JLabel("Address:");
Address.setLocation(0,80);

```

```
Address.setSize(300,30);
Address.setHorizontalTextPosition(JLabel.LEFT);
txtPanel.add(Address);
```

```
JLabel DTreatment = new JLabel("Detailed Treatment:");
DTreatment.setLocation(0,160);
DTreatment.setSize(300,30);
DTreatment.setHorizontalTextPosition(JLabel.LEFT);
txtPanel.add(DTreatment);
```

```
JLabel Town = new JLabel("Town");
Town.setLocation(0,120);
Town.setSize(300,30);
Town.setHorizontalTextPosition(JLabel.LEFT);
txtPanel.add(Town);
```

```
JLabel Allergy = new JLabel("Allergy:");
Allergy.setLocation(0,200);
Allergy.setSize(300,30);
Allergy.setHorizontalTextPosition(JLabel.LEFT);
txtPanel.add(Allergy);
```

```
JLabel ExplainAllergy = new JLabel("ExplainAllergy:");
ExplainAllergy.setLocation(0,240);
ExplainAllergy.setSize(300,30);
ExplainAllergy.setHorizontalTextPosition(JLabel.LEFT);
txtPanel.add(ExplainAllergy);
```

```
JLabel KeyCode = new JLabel("Key Code:");
KeyCode.setLocation(0,280);
KeyCode.setSize(300,30);
KeyCode.setHorizontalTextPosition(JLabel.LEFT);
txtPanel.add(KeyCode);
```

```
JLabel Equipment = new JLabel("Equipment:");
Equipment.setLocation(0,320);
Equipment.setSize(300,30);
Equipment.setHorizontalTextPosition(JLabel.LEFT);
txtPanel.add(Equipment);
```

```
JLabel Telephone = new JLabel("Telephone:");
Telephone.setLocation(0,360);
Telephone.setSize(300,30);
Telephone.setHorizontalTextPosition(JLabel.LEFT);
txtPanel.add(Telephone);
```

```
JLabel Doctor = new JLabel("Doctor:");
Doctor.setLocation(0,400);
Doctor.setSize(300,30);
Doctor.setHorizontalTextPosition(JLabel.LEFT);
txtPanel.add(Doctor);

JLabel Bloods = new JLabel("Bloods:");
Bloods.setLocation(0,440);
Bloods.setSize(300,30);
Bloods.setHorizontalTextPosition(JLabel.LEFT);
txtPanel.add(Bloods);

JTextField UsernameField = new JTextField(rs.getString("PName"));
UsernameField.setEditable(false);
UsernameField.setLocation(0, 0);
UsernameField.setSize(390, 30);
UsernameField.setBackground(PanelBackground);
txtFieldPanel.add(UsernameField);

JTextField NurseNameField = new JTextField(rs.getString("FName"));
NurseNameField.setEditable(false);
NurseNameField.setLocation(0, 40);
NurseNameField.setSize(390, 30);
NurseNameField.setBackground(PanelBackground);
txtFieldPanel.add(NurseNameField);

final JTextArea addressArea = new JTextArea(rs.getString("PAddress"));
addressArea.setSize(390, 30);
addressArea.setEditable(false);
addressArea.setBackground(PanelBackground);
addressArea.setLineWrap(true);

final JScrollPane scrollPane = new JScrollPane(addressArea);
scrollPane.setLocation(0, 80);
scrollPane.setSize(390, 30);
txtFieldPanel.add(scrollPane);

JTextField TownField = new JTextField(rs.getString("PTown"));
TownField.setEditable(false);
TownField.setLocation(0, 120);
TownField.setSize(390, 30);
TownField.setBackground(PanelBackground);
txtFieldPanel.add(TownField);
```

```
final JTextArea addressArea2 = new JTextArea(rs.getString("DTreatment"));
addressArea2.setSize(390, 30);
addressArea2.setEditable(false);
addressArea2.setBackground(PanelBackground);
addressArea2.setLineWrap(true);
```

```
final JScrollPane scrollPane2 = new JScrollPane(addressArea2);
scrollPane2.setLocation(0, 160);
scrollPane2.setSize(390, 30);
txtFieldPanel.add(scrollPane2);
```

```
JTextField AllergyField = new JTextField(rs.getString("Allergy"));
AllergyField.setEditable(false);
AllergyField.setLocation(0, 200);
AllergyField.setSize(390, 30);
AllergyField.setBackground(PanelBackground);
txtFieldPanel.add(AllergyField);
```

```
final JTextArea addressAreaAllergy = new JTextArea(rs.getString("ExplainAllergy"));
addressAreaAllergy.setSize(390, 30);
addressAreaAllergy.setEditable(false);
addressAreaAllergy.setBackground(PanelBackground);
addressAreaAllergy.setLineWrap(true);
```

```
final JScrollPane scrollPane3 = new JScrollPane(addressAreaAllergy);
scrollPane3.setLocation(0, 240);
scrollPane3.setSize(390, 30);
txtFieldPanel.add(scrollPane3);
```

```
JTextField KeyCodeField = new JTextField(rs.getString("Kcode"));
KeyCodeField.setEditable(false);
KeyCodeField.setLocation(0, 280);
KeyCodeField.setSize(390, 30);
KeyCodeField.setBackground(PanelBackground);
txtFieldPanel.add(KeyCodeField);
```

```
JTextField EquipmentField = new JTextField(rs.getString("Equipment"));
EquipmentField.setEditable(false);
EquipmentField.setLocation(0, 320);
EquipmentField.setSize(390, 30);
EquipmentField.setBackground(PanelBackground);
txtFieldPanel.add(EquipmentField);
```

```
JTextField TelephoneField = new JTextField(rs.getString("PTelNumber"));
```

```

TelephoneField.setEditable(false);
TelephoneField.setLocation(0, 360);
TelephoneField.setSize(390, 30);
TelephoneField.setBackground(PanelBackground);
txtFieldPanel.add(TelephoneField);

JTextField DoctorField = new JTextField(rs.getString("Doctor"));
DoctorField.setEditable(false);
DoctorField.setLocation(0, 400);
DoctorField.setSize(390, 30);
DoctorField.setBackground(PanelBackground);
txtFieldPanel.add(DoctorField);

JTextField BloodsField = new JTextField(rs.getString("Bloods"));
BloodsField.setEditable(false);
BloodsField.setLocation(0, 440);
BloodsField.setSize(390, 30);
BloodsField.setBackground(PanelBackground);
txtFieldPanel.add(BloodsField);

moduleListPanel.add(print);
moduleListPanel.add(back);
mainpage.add(moduleListPanel);
mainpage.setPreferredSize(new Dimension(650,560));
mainpage.setBackground(Color.WHITE);

back.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        try
        {
            //Redirect back to frame with list of matches
            nurse_frame.setVisible(false);
            testFrame.setVisible(true);
        }
        catch (Exception exc)
        {
            System.out.println(exc);
        }
    }
});

print.addActionListener(new ActionListener()

```

```

        {
            public void actionPerformed(ActionEvent e)
            {
                try
                {
                    //Message displayed to user indicating the selection of the
                    //print option
                    JOptionPane.showMessageDialog(frame, "You have selected
                    the print button", "Printing",
                    JOptionPane.INFORMATION_MESSAGE);
                }
                catch (Exception exc)
                {
                    System.out.println(exc);
                }
            }
        });
        nurse_frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        nurse_frame.setVisible(true);
    }

    public static void main(String args[]) throws Exception
    {
        new PatientTreatmentNurse();
    }
}
}
}

```

'View call list' list individual calls for a specific member of staff – District Nurse (340):-

```

public class CallList
{
    String PaName, FaName, date;
    JList trial;
    JOptionPane frame = new JOptionPane();
    private static JFrame main_frame = new JFrame ("Bargoed Nurses");
    JPanel main = new JPanel();
    private JPanel heading, footer;
    private ImageIcon image1;
    private JLabel label1, ViewLabel, StaffName;
    public static JPanel mainpage = new JPanel();
    JScrollPane vertical = new JScrollPane();
    JPanel rest = new JPanel(), sidebar = new JPanel();
}

```

```

Color buttonBackground = new Color (234,234,234);
Color blueBackground = new Color (7,32,157);
Color PanelBackground = new Color (167, 165, 255);
Font f = new Font("Dialog", Font.PLAIN, 26);
DefaultListModel data;
final JList TestList = new JList();

static String selectedPatient;
static String lastone;
private static String ending;
private static String PaID;
private static int index;
private static String PaIDTest;
private static String fileName;
private static String firstSplit;
private static String familyName;
private static String Address;
private static String Town;
private static String Treatment;
private static String KCode;
private static int index2;
static String username;

public CallList() throws Exception
{
    username = LoginScreen.Get_User();
    //Get the date of last saved call list
    Getdate();
    trial = SearchPatient.Get_allocation();
    PaName = SearchPatient.Get_Patient();
    FaName = SearchPatient.Get_Family();

    heading = new JPanel();
    heading.setMaximumSize(new Dimension(700,150));
    heading.setBackground(blueBackground);
    image1 = new ImageIcon(getClass().getResource("Bar.jpg"));
    label1 = new JLabel(image1);
    heading.add(label1);

    footer = new JPanel();
    footer.setMaximumSize(new Dimension(800, 300));
    footer.setBackground(Color.GREEN);
    JButton Print = new JButton("Print Details");
    JButton BackMain = new JButton("Back to Main Screen");
    footer.add(Print); footer.add(BackMain);
}

```

```

footer.setBackground(Color.WHITE);

ViewLabel = new JLabel("Here are the patients' details found:-" + date);
ViewLabel.setFont(f);
main_frame.setSize(780,570);
main_frame.setLayout(new FlowLayout());
main_frame.setBackground(Color.WHITE);
main.setLayout(new BorderLayout(main, BorderLayout.PAGE_AXIS));
main.setPreferredSize(new Dimension(600,350));

final JList TestList = new JList(listPatients());
TestList.setSelectionMode(ListSelectionModel.SINGLE_SELECTION);
TestList.setSize(200, 500);
TestList.setSelectedIndex(0);
TestList.setVisibleRowCount(6);

JScrollPane vertical = new JScrollPane(TestList);
vertical.setVerticalScrollBarPolicy(JScrollPane.VERTICAL_SCROLLBAR_ALWAYS);
vertical.getViewport().setBackground(Color.white);
mainpage.setPreferredSize(new Dimension(500,400));
mainpage.removeAll();
rest.setPreferredSize(new Dimension(800,600));

//Inputting information into sidebar
JPanel TestPanel = new JPanel();
JLabel sidelabel = new JLabel("<html>* Here are your list of calls");
JLabel sidelabelblank = new JLabel(" ");
JLabel sidelabelblank2 = new JLabel(" ");
JLabel sidelabel2 = new JLabel("<html>* You can double click each patient in order
to find out more details");
TestPanel.setBackground(PanelBackground);

TestPanel.setPreferredSize(new Dimension(100,400));
TestPanel.setLayout(new BorderLayout(TestPanel, BorderLayout.PAGE_AXIS));
TestPanel.add(sidelabel);
TestPanel.add(sidelabelblank);
TestPanel.add(sidelabel2);
TestPanel.add(sidelabelblank2);

rest.add(TestPanel); rest.add(mainpage);
rest.setBackground(blueBackground);
main_frame.add(heading); main_frame.add(rest);
main.add(ViewLabel);
main.add(vertical);
vertical.setMaximumSize(new Dimension(1200, 250));

```

```

mainpage.add(main);
mainpage.add(footer);
mainpage.setPreferredSize(new Dimension(650,400));
mainpage.setBackground(Color.WHITE);
main.setBackground(Color.WHITE);

MouseListener mouseListener = new MouseAdapter()
{
    public void mouseClicked(MouseEvent e)
    {
        try
        {
            //Once an item is double clicked then redirect to view
            details on the item selected
            if (e.getClickCount() == 2)
            {
                //Get selected value of item double clicked
                selectedPatient = TestList.getSelectedValue().toString();
                index = TestList.locationToIndex(e.getPoint());
                PaIDTest = PaID + index;

                //Break down the selected item and extract it's Strings
                String words[] = selectedPatient.split(" ");
                firstSplit = words[0];
                familyName = words[1];
                Address = words[2] + " " + words[3] + " " + words[4];
                Town = words[5];
                Treatment = words[6];
                KCode = words[7];
                main_frame.setVisible(false);
                new PatientDetailed();
            }

            else
            {
            }
        }
        catch(Exception exc)
        {
            System.out.println(exc);
        }
    }
};

TestList.addMouseListener(mouseListener);

```

```

Print.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        try
        {
            //Message indicating that the user has selected the print option
            JOptionPane.showMessageDialog(frame, "You have selected the
            print button", "Printing", JOptionPane.INFORMATION_MESSAGE);
        }
        catch (Exception exc)
        {
            System.out.println(exc);
        }
    }
});

BackMain.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        try
        {
            //Redirect back to the nurse main frame
            main_frame.setVisible(false);
            new MainScreenNurse();
        }
        catch (Exception exc)
        {
            System.out.println(exc);
        }
    }
});
main_frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
}

public String[] listPatients() throws Exception
{
    //Get the individual call list based on the log in details entered
    Class.forName("org.sqlite.JDBC");
    Connection conn =
    DriverManager.getConnection("jdbc:sqlite:NurseScheduling.sqlite");
    Statement stat = conn.createStatement();
    ResultSet rs = stat.executeQuery(String.format("SELECT * FROM Calls, DistrictNurse

```

```
where Calls.NurseName = DistrictNurse.NurseName and DistrictNurse.UserNameN
='%s'", username));
```

```
String[] rawModList = new String[200];
index = 0;
```

```
while (rs.next())
{
    rawModList[index] = rs.getString("PName") + " " + rs.getString("FName") + "
    " + rs.getString("PAddress") + " " + rs.getString("PTown") + " " +
    rs.getString("Treatment") + " " + rs.getString("Kcode");
    index++;
}
rs.close();
conn.close();
```

```
int numElements = 0;
for (int i = 0; i < 200; i++)
```

```
{
    if (rawModList[i] != null)
    {
        numElements++;
    }
}
```

```
String[] modList = new String[numElements];
System.arraycopy(rawModList, 0, modList, 0, numElements);
```

```
if (index == 0)
{
    //Message to user indicating no details could be found and then redirected
    back to main frame
    JOptionPane.showMessageDialog(frame, "<html>No details found could
    mean calls have not been allocated <br>for tomorrow yet or you are not on the rota
    to work tomorrow", "No details", JOptionPane.ERROR_MESSAGE);
```

```
    main_frame.setVisible(false);
    new MainScreenNurse();
```

```
}
else
{
    main_frame.setVisible(true);
}
```

```
return modList;
```

```
}
```

```

//This method gets the date on the last call list saved
public void Getdate() throws Exception
{
    Class.forName("org.sqlite.JDBC");
    Connection connt = (Connection)
    DriverManager.getConnection("jdbc:sqlite:NurseScheduling.sqlite");
    Statement stt = connt.createStatement();
    ResultSet rst = stt.executeQuery("SELECT Date FROM CallLists2");

    date = rst.getString("Date");
}

public static String Get_Patient()
{
    return ending;
}

public static String Get_ID()
{
    return fileName;
}

public static JFrame test_frame()
{
    return main_frame;
}

public static String Get_PName()
{
    return firstSplit;
}

public static String Get_FName()
{
    return familyName;
}

public static String Get_Address()
{
    return Address;
}

public static String Get_Town()
{
    return Town;
}

```

```

}

public static String Get_Treatment()
{
    return Treatment;
}

public static void main(String args[]) throws Exception
{

    new CallList();
}
}

```

'View call list' detailed version of each call (patient) – District Nurse (340):-

```

public class PatientDetailed
{
    static String username;
    String patientName, familyName, Address, Town, Treatment;
    JFrame refreshFrame;
    JOptionPane frame = new JOptionPane();
    JFrame nurse_frame = new JFrame ("Bargoed Nurses");
    private JPanel heading;
    private ImageIcon image1;
    private JLabel label1;
    public static JPanel mainpage = new JPanel();
    JPanel rest = new JPanel(), sidebar = new JPanel();
    Color buttonBackground = new Color (234,234,234);
    Color blueBackground = new Color (7,32,157);
    Color PanelBackground = new Color (167, 165, 255);
    Font f = new Font("Dialog", Font.PLAIN, 20);

    public PatientDetailed() throws Exception
    {
        patientName = CallList.Get_PName();
        familyName = CallList.Get_FName();
        Address = CallList.Get_Address();
        Town = CallList.Get_Town();
        Treatment = CallList.Get_Treatment();
        refreshFrame = CallList.test_frame();

        heading = new JPanel();
        heading.setMaximumSize(new Dimension(700,150));
        heading.setBackground(blueBackground);
        image1 = new ImageIcon(getClass().getResource("Bar.jpg"));
    }
}

```

```

label1 = new JLabel(image1);
heading.add(label1);

JButton print = new JButton("Print");
JButton back = new JButton("Back to list of Patients");
nurse_frame.setSize(780,725);
nurse_frame.setLayout(new FlowLayout());
nurse_frame.setBackground(Color.WHITE);

mainpage.setPreferredSize(new Dimension(500,700));
mainpage.removeAll();
rest.setPreferredSize(new Dimension(800,580));
rest.setBackground(blueBackground);

nurse_frame.add(heading); nurse_frame.add(rest);

JPanel TestPanel = new JPanel();
TestPanel.setBackground(PanelBackground);
TestPanel.setPreferredSize(new Dimension(100,560));
TestPanel.setLayout(new BorderLayout(TestPanel, BorderLayout.PAGE_AXIS));
rest.add(TestPanel); rest.add(mainpage);

JPanel moduleListPanel = new JPanel();
moduleListPanel.setLayout(new FlowLayout());
moduleListPanel.setPreferredSize(new Dimension(600,550));
JLabel welcome = new JLabel("<html>Patient's details are:");
welcome.setPreferredSize(new Dimension(600, 20));
welcome.setHorizontalTextPosition(JLabel.LEFT);
moduleListPanel.add(welcome);

//Get details based on the information selected from the individual call list
Class.forName("org.sqlite.JDBC");
Connection conn =
DriverManager.getConnection("jdbc:sqlite:NurseScheduling.sqlite");
Statement stat = conn.createStatement();
ResultSet rs = stat.executeQuery(String.format("select * from Patients2 where
PName='%s' and FName='%s' and PAddress='%s' and PTown='%s' and
Treatment='%s'", patientName, familyName, Address, Town, Treatment));

JPanel txtPanel = new JPanel();
txtPanel.setLayout(null);
txtPanel.setLocation(20, 50);
txtPanel.setPreferredSize(new Dimension(200, 480));
txtPanel.setBackground(Color.WHITE);

```

```
moduleListPanel.add(txtPanel);
moduleListPanel.setBackground(Color.WHITE);

JPanel txtFieldPanel = new JPanel();
txtFieldPanel.setLayout(null);
txtFieldPanel.setLocation(150, 60);
txtFieldPanel.setPreferredSize(new Dimension(390, 480));
txtFieldPanel.setBackground(Color.WHITE);
moduleListPanel.add(txtFieldPanel);

JLabel patientName = new JLabel("Patient Name:");
patientName.setLocation(0, 0);
patientName.setSize(300, 30);
patientName.setHorizontalTextPosition(JLabel.LEFT);
txtPanel.add(patientName);

JLabel familyName = new JLabel("Family Name: ");
familyName.setLocation(0,40);
familyName.setSize(300,30);
familyName.setHorizontalTextPosition(JLabel.LEFT);
txtPanel.add(familyName);

JLabel Address = new JLabel("Address: ");
Address.setLocation(0,80);
Address.setSize(300,30);
Address.setHorizontalTextPosition(JLabel.LEFT);
txtPanel.add(Address);

JLabel DTreatment = new JLabel("Detailed Treatment: ");
DTreatment.setLocation(0,160);
DTreatment.setSize(300,30);
DTreatment.setHorizontalTextPosition(JLabel.LEFT);
txtPanel.add(DTreatment);

JLabel Town = new JLabel("Town: ");
Town.setLocation(0,120);
Town.setSize(300,30);
Town.setHorizontalTextPosition(JLabel.LEFT);
txtPanel.add(Town);

JLabel Allergy = new JLabel("Allergy: ");
Allergy.setLocation(0,200);
Allergy.setSize(300,30);
Allergy.setHorizontalTextPosition(JLabel.LEFT);
txtPanel.add(Allergy);
```

```
JLabel ExplainAllergy = new JLabel("ExplainAllergy:");
ExplainAllergy.setLocation(0,240);
ExplainAllergy.setSize(300,30);
ExplainAllergy.setHorizontalTextPosition(JLabel.LEFT);
txtPanel.add(ExplainAllergy);
```

```
JLabel KeyCode = new JLabel("Key Code:");
KeyCode.setLocation(0,280);
KeyCode.setSize(300,30);
KeyCode.setHorizontalTextPosition(JLabel.LEFT);
txtPanel.add(KeyCode);
```

```
JLabel Equipment = new JLabel("Equipment:");
Equipment.setLocation(0,320);
Equipment.setSize(300,30);
Equipment.setHorizontalTextPosition(JLabel.LEFT);
txtPanel.add(Equipment);
```

```
JLabel Telephone = new JLabel("Telephone:");
Telephone.setLocation(0,360);
Telephone.setSize(300,30);
Telephone.setHorizontalTextPosition(JLabel.LEFT);
txtPanel.add(Telephone);
```

```
JLabel Doctor = new JLabel("Doctor:");
Doctor.setLocation(0,400);
Doctor.setSize(300,30);
Doctor.setHorizontalTextPosition(JLabel.LEFT);
txtPanel.add(Doctor);
```

```
JLabel Bloods = new JLabel("Bloods:");
Bloods.setLocation(0,440);
Bloods.setSize(300,30);
Bloods.setHorizontalTextPosition(JLabel.LEFT);
txtPanel.add(Bloods);
```

```
JTextField UsernameField = new JTextField(rs.getString("PName"));
UsernameField.setEditable(false);
UsernameField.setLocation(0, 0);
UsernameField.setSize(390, 30);
UsernameField.setBackground(PanelBackground);
txtFieldPanel.add(UsernameField);
```

```
JTextField NurseNameField = new JTextField(rs.getString("FName"));
```

```
NurseNameField.setEditable(false);
NurseNameField.setLocation(0, 40);
NurseNameField.setSize(390, 30);
NurseNameField.setBackground(PanelBackground);
txtFieldPanel.add(NurseNameField);
```

```
final JTextArea addressArea = new JTextArea(rs.getString("PAddress"));
addressArea.setSize(390, 30);
addressArea.setEditable(false);
addressArea.setBackground(PanelBackground);
addressArea.setLinewrap(true);
```

```
final JScrollPane scrollPane = new JScrollPane(addressArea);
scrollPane.setLocation(0, 80);
scrollPane.setSize(390, 30);
txtFieldPanel.add(scrollPane);
```

```
JTextField TownField = new JTextField(rs.getString("PTown"));
TownField.setEditable(false);
TownField.setLocation(0, 120);
TownField.setSize(390, 30);
TownField.setBackground(PanelBackground);
txtFieldPanel.add(TownField);
```

```
final JTextArea addressArea2 = new JTextArea(rs.getString("DTreatment"));
addressArea2.setSize(390, 30);
addressArea2.setEditable(false);
addressArea2.setBackground(PanelBackground);
addressArea2.setLinewrap(true);
```

```
final JScrollPane scrollPane2 = new JScrollPane(addressArea2);
scrollPane2.setLocation(0, 160);
scrollPane2.setSize(390, 30);
txtFieldPanel.add(scrollPane2);
```

```
JTextField AllergyField = new JTextField(rs.getString("Allergy"));
AllergyField.setEditable(false);
AllergyField.setLocation(0, 200);
AllergyField.setSize(390, 30);
AllergyField.setBackground(PanelBackground);
txtFieldPanel.add(AllergyField);
```

```
final JTextArea addressAreaAllergy = new JTextArea(rs.getString("ExplainAllergy"));
addressAreaAllergy.setSize(390, 30);
```

```
addressAreaAllergy.setEditable(false);
addressAreaAllergy.setBackground(PanelBackground);
addressAreaAllergy.setLinewrap(true);

final JScrollPane scrollPane3 = new JScrollPane(addressAreaAllergy);
scrollPane3.setLocation(0, 240);
scrollPane3.setSize(390, 30);
txtFieldPanel.add(scrollPane3);

JTextField KeyCodeField = new JTextField(rs.getString("Kcode"));
KeyCodeField.setEditable(false);
KeyCodeField.setLocation(0, 280);
KeyCodeField.setSize(390, 30);
KeyCodeField.setBackground(PanelBackground);
txtFieldPanel.add(KeyCodeField);

JTextField EquipmentField = new JTextField(rs.getString("Equipment"));
EquipmentField.setEditable(false);
EquipmentField.setLocation(0, 320);
EquipmentField.setSize(390, 30);
EquipmentField.setBackground(PanelBackground);
txtFieldPanel.add(EquipmentField);

JTextField TelephoneField = new JTextField(rs.getString("PTelNumber"));
TelephoneField.setEditable(false);
TelephoneField.setLocation(0, 360);
TelephoneField.setSize(390, 30);
TelephoneField.setBackground(PanelBackground);
txtFieldPanel.add(TelephoneField);

JTextField DoctorField = new JTextField(rs.getString("Doctor"));
DoctorField.setEditable(false);
DoctorField.setLocation(0, 400);
DoctorField.setSize(390, 30);
DoctorField.setBackground(PanelBackground);
txtFieldPanel.add(DoctorField);

JTextField BloodsField = new JTextField(rs.getString("Bloods"));
BloodsField.setEditable(false);
BloodsField.setLocation(0, 440);
BloodsField.setSize(390, 30);
BloodsField.setBackground(PanelBackground);
txtFieldPanel.add(BloodsField);

moduleListPanel.add(print);
```

```

moduleListPanel.add(back);
mainpage.add(moduleListPanel);
mainpage.setPreferredSize(new Dimension(650,560));
mainpage.setBackground(Color.WHITE);

back.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        try
        {
            //Redirect back to individual call list
            nurse_frame.setVisible(false);
            refreshFrame.setVisible(true);
        }
        catch (Exception exc)
        {
            System.out.println(exc);
        }
    }
});

print.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        try
        {
            //Message indicating that the user has selected the print
            option
            JOptionPane.showMessageDialog(frame, "You have selected
            the print button", "Printing",
            JOptionPane.INFORMATION_MESSAGE);
        }
        catch (Exception exc)
        {
            System.out.println(exc);
        }
    }
});
nurse_frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
nurse_frame.setVisible(true);
}

public static void main(String args[]) throws Exception

```

```

    {
        new PatientDetailed();
    }
}

```

Main Screen for Administrator (210):-

```

public class MainScreenAdmin
{
    JOptionPane frame = new JOptionPane();
    JFrame main_frame = new JFrame ("Bargoed Nurses");
    JPanel main = new JPanel();
    private JPanel heading;
    private Imugelcon image1;
    private JLabel label1;
    public static JPanel mainpage = new JPanel();
    JPanel rest = new JPanel(), sidebar = new JPanel();
    Color buttonBackground = new Color (234,234,234);
    Color blueBackground = new Color (7,32,157);
    Color PanelBackground = new Color (167, 165, 255);
    Font f = new Font("Dialog", Font.PLAIN, 36);

    public MainScreenAdmin() throws Exception
    {
        heading = new JPanel();
        heading.setMaximumSize(new Dimension(700,150));
        heading.setBackground(blueBackground);
        image1 = new Imugelcon(getClass().getResource("Bar.jpg"));
        label1 = new JLabel(image1);
        heading.add(label1);

        JButton AllocateCalls = new JButton("Allocate Calls");
        AllocateCalls.setMaximumSize(new Dimension(700, 100));
        AllocateCalls.setFont(f);
        JButton ViewStaff = new JButton("View Staff Details");
        ViewStaff.setMaximumSize(new Dimension(700, 100));
        ViewStaff.setFont(f);
        JButton ViewPatient = new JButton("View Patient Details");
        ViewPatient.setMaximumSize(new Dimension(700, 100));
        ViewPatient.setFont(f);
        JButton ViewCalls = new JButton("View Calls");
        ViewCalls.setMaximumSize(new Dimension(700, 100));
        ViewCalls.setFont(f);
        JButton LogOut = new JButton("Log out");
        LogOut.setMaximumSize(new Dimension(700, 100));
    }
}

```

```
LogOut.setFont(f);
```

```
main_frame.setSize(780,570);  
main_frame.setLayout(new FlowLayout());  
main_frame.setBackground(Color.WHITE);  
main.setLayout(new BorderLayout(main, BorderLayout.PAGE_AXIS));  
main.setPreferredSize(new Dimension(600,380));  
mainpage.setPreferredSize(new Dimension(500,400));  
mainpage.removeAll();  
rest.setPreferredSize(new Dimension(800,600));
```

```
//Inputting information into sidebar
```

```
JPanel TestPanel = new JPanel();  
JLabel sidelabel2 = new JLabel("<html>* Allocate calls allows you to select staff and  
allocate calls");  
JLabel sidelabel21 = new JLabel("<html> * View staff details allows you to search for  
a particular staff member to view their details");  
JLabel sidelabel3 = new JLabel("<html>* View Patient details allows you to search  
for a particular patient to view their details");  
JLabel sidelabel4 = new JLabel("<html>* View Calls allows you to view lists for calls  
that have been allocated");  
JLabel sidelabel5 = new JLabel("<html>* Log out allows you to log out of the  
System");
```

```
TestPanel.setBackground(PanelBackground);  
TestPanel.setPreferredSize(new Dimension(100,400));  
TestPanel.setLayout(new BorderLayout(TestPanel, BorderLayout.PAGE_AXIS));  
TestPanel.add(sidelabel2);  
TestPanel.add(sidelabel21);  
TestPanel.add(sidelabel3);  
TestPanel.add(sidelabel4);  
TestPanel.add(sidelabel5);  
rest.add(TestPanel);  
rest.add(mainpage);  
rest.setBackground(blueBackground);
```

```
main_frame.add(heading); main_frame.add(rest);  
main.add(AllocateCalls);  
main.add(ViewStaff);  
main.add(ViewPatient);  
main.add(ViewCalls);  
main.add(LogOut);  
AllocateCalls.setBackground(buttonBackground);  
ViewStaff.setBackground(buttonBackground);  
ViewPatient.setBackground(buttonBackground);
```

```
ViewCalls.setBackground(buttonBackground);
LogOut.setBackground(buttonBackground);
mainpage.add(main);
mainpage.setPreferredSize(new Dimension(650,400));
mainpage.setBackground(Color.WHITE);
main.setBackground(Color.WHITE);
```

```
AllocateCalls.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        try
        {
            //Redirect to main allocation page
            new AdminAllocateMainTest();
            main_frame.setVisible(false);
        }
        catch (Exception exc)
        {
            System.out.println(exc);
        }
    }
});
```

```
ViewStaff.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        try
        {
            //Redirect to main search staff frame
            main_frame.setVisible(false);
            new SearchStaff();
        }
        catch (Exception exc)
        {
            System.out.println(exc);
        }
    }
});
```

```
ViewPatient.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
```

```

        try
        {
            //Redirect to main search patient frame
            main_frame.setVisible(false);
            new SearchPatient();
        }
        catch (Exception exc)
        {
            System.out.println(exc);
        }
    }
});

ViewCalls.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        try
        {
            //Redirect to view all staff that are working
            main_frame.setVisible(false);
            new SearchStaffWorking();
        }
        catch (Exception exc)
        {
            System.out.println(exc);
        }
    }
});

LogOut.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        try
        {
            //Dialog box to check they want to log out
            Object[] options = {"Yes", "No"};
            int response = JOptionPane.showOptionDialog(frame, "Are
you sure you want to logout?", "Log Out?",
JOptionPane.YES_NO_OPTION, JOptionPane.WARNING_MES
SAGE, null, options, options[1]);
            if (response != JOptionPane.YES_OPTION)
            {
                return;
            }
        }
    }
});

```

```

        }
        else
        {
            main_frame.setVisible(false);
            LoginScreen.userf.setText("");
            LoginScreen.passf.setText("");
            new LoginScreen();
        }
    }
    catch (Exception exc)
    {
        System.out.println(exc);
    }
}
});
main_frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
main_frame.setVisible(true);
main_frame.revalidate();
}

public static void main(String args[] throws Exception
{
    new MainScreenAdmin();
}
}

```

'Allocate Calls' main page - Administrator (479):-

```

public class AdminAllocateMainTest
{
    static String username;
    JOptionPane frame = new JOptionPane();
    JFrame main_frame = new JFrame ("Bargoed Nurses");
    JPanel main = new JPanel();
    JPanel mainButton = new JPanel();
    private JPanel heading, footer;
    private ImageIcon image1;
    final JList TestList = new JList();
    private JList list;
    private String [] testlist;
    private String [] modList;
    private JLabel label1, AllocateLabel, EnterLabel;
    JButton move = new JButton();
    JButton moveBack = new JButton();
}

```

```

public static JPanel mainpage = new JPanel();
JPanel rest = new JPanel(), sidebar = new JPanel();
Color buttonBackground = new Color (234,234,234);
Color blueBackground = new Color (7,32,157);
Font f = new Font("Dialog", Font.PLAIN, 36);
Color PanelBackground = new Color (167, 165, 255);
private static DefaultListModel data;
private DefaultListModel data1;
final JScrollPane side = new JScrollPane();
final JPanel moduleListPanel = new JPanel();
final JScrollPane vertical = new JScrollPane();
private String selectedLeftUsername;
private String selectedRightUsername;

public AdminAllocateMainTest() throws Exception
{
    heading = new JPanel();
    heading.setMaximumSize(new Dimension(700,150));
    heading.setBackground(blueBackground);
    image1 = new ImageIcon(getClass().getResource("Bar.jpg"));
    label1 = new JLabel(image1);
    heading.add(label1);

    footer = new JPanel();
    footer.setMaximumSize(new Dimension(800, 300));
    footer.setBackground(Color.GREEN);
    JButton CreateCall = new JButton("Create Call List");
    JButton BackMain = new JButton("Back to Main Screen");
    footer.add(CreateCall); footer.add(BackMain);
    footer.setBackground(Color.WHITE);

    AllocateLabel = new JLabel("Allocate Calls");
    AllocateLabel.setFont(f);
    EnterLabel = new JLabel("Please select 6 members of staff:-");
    data = new DefaultListModel();
    data.ensureCapacity(6);
    final JList TestList = new JList(data);
    TestList.setSelectionMode(ListSelectionModel.SINGLE_SELECTION);
    TestList.setSize(200, 500);
    TestList.setModel(data);
    TestList.setSelectedIndex(0);
    TestList.setVisibleRowCount(6);

    data1 = new DefaultListModel();
    data1.addElement(modList);
}

```

```

final JList list = new JList(listUsers());
list.setSelectionMode(ListSelectionModel.SINGLE_SELECTION);
list.setSize(500, 500);
list.setVisibleRowCount(10);

move = new JButton("Move Nurse");
final JList Testlist;
JButton moveBack = new JButton("Remove from list");
move.setEnabled(false);
MouseListener mouseListener = new MouseAdapter()
{
    public void mouseClicked(MouseEvent e)
    {
        try
        {
            //If item in list is clicked once then get value of selected item
            //Left hand list (Original List)
            if (e.getClickCount() == 1)
            {
                move.setEnabled(true);
                selectedLeftUsername = list.getSelectedValue().toString();
                moduleListPanel.revalidate();
            }
            else
            {
            }
        }
        catch (Exception exc)
        {
            System.out.println(exc);
        }
    }
};

list.addMouseListener(mouseListener);

MouseListener mouseListener1 = new MouseAdapter()
{
    public void mouseClicked(MouseEvent e)
    {
        try
        {
            //If item in list is clicked once then get value of selected item
            //Right hand list (Created by user)
            if (e.getClickCount() == 1)

```

```
        {
            selectedRightUsername = TestList.getSelectedValue().toString();
        }
        else
        {
        }
        }
        catch(Exception exc)
        {
            System.out.println(exc);
        }
    }
};
```

```
TestList.addMouseListener(mouseListener1);
```

```
ActionListener moveListener = new ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        try
        {
            //If move button is selected then call the MoveName method
            MoveName();
        }
        catch (Exception exc)
        {
        }
    }
};
```

```
move.addActionListener(moveListener);
```

```
ActionListener moveBackListener = new ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        try
        {
            //If move back button is selected then call the moveBack method
            moveBack();
        }
        catch (Exception exc)
```

```

        {
        }
    }
};

moveBack.addActionListener(moveBackListener);
main_frame.setSize(780,570);
main_frame.setLayout(new FlowLayout());
main_frame.setBackground(Color.WHITE);
main.setLayout(new BorderLayout(main, BorderLayout.PAGE_AXIS));
main.setPreferredSize(new Dimension(600,70));
mainButton.setLayout(new BorderLayout(mainButton, BorderLayout.PAGE_AXIS));
mainButton.add(move);
mainButton.add(moveBack);
mainButton.setBackground(Color.WHITE);

final JScrollPane vertical = new JScrollPane(list);
JScrollPane side = new JScrollPane(TestList);
side.getViewport().setBackground(Color.white);

side.setVerticalScrollBarPolicy(JScrollPane.VERTICAL_SCROLLBAR_ALWAYS);
vertical.setVerticalScrollBarPolicy(JScrollPane.VERTICAL_SCROLLBAR_ALWAYS);
vertical.setPreferredSize(new Dimension(240, 180));
side.setPreferredSize(new Dimension(240, 180));

JPanel moduleListPanel = new JPanel();
moduleListPanel.setLayout(new FlowLayout());
moduleListPanel.setPreferredSize(new Dimension(650,200));
moduleListPanel.setBackground(Color.WHITE);
moduleListPanel.add(vertical); moduleListPanel.add(mainButton);
moduleListPanel.add(side);
mainpage.setPreferredSize(new Dimension(500,400));
mainpage.removeAll();
rest.setPreferredSize(new Dimension(800,600));

//Inputting information into sidebar
JPanel TestPanel = new JPanel();
JLabel sidelabel = new JLabel("<html>* Select a name from the left list and select
move name");
JLabel sidelabel2 = new JLabel("<html>* Continue to add names to the right list until
there are 6 staff members");
JLabel sidelabel21 = new JLabel("<html> * You can remove a name from the right list
by selecting it and selecting remove name");
JLabel sidelabel3 = new JLabel("<html>* Use the staff rota to know which names to
select");

```

```
JLabel sidelabel4 = new JLabel("<html>* Once happy with the right hand list select allocate calls");
```

```
TestPanel.setBackground(PanelBackground);  
TestPanel.setPreferredSize(new Dimension(100,400));  
TestPanel.setLayout(new BorderLayout(TestPanel, BorderLayout.PAGE_AXIS));  
TestPanel.add(sidelabel);  
TestPanel.add(sidelabel2);  
TestPanel.add(sidelabel21);  
TestPanel.add(sidelabel3);  
TestPanel.add(sidelabel4);  
rest.add(TestPanel);  
rest.add(mainpage);  
rest.setBackground(blueBackground);
```

```
main_frame.add(heading); main_frame.add(rest);  
main.add(AllocateLabel);  
main.add(EnterLabel);  
mainpage.add(main);  
mainpage.add(moduleListPanel);  
mainpage.add(footer);  
mainpage.setPreferredSize(new Dimension(650,400));  
mainpage.setBackground(Color.WHITE);  
main.setBackground(Color.WHITE);
```

```
CreateCall.addActionListener(new ActionListener()  
{  
    public void actionPerformed(ActionEvent e)  
    {  
        try  
        {  
            int destSize = data.getSize();  
  
            if (destSize < 6)  
            {  
                //Checking the right number of staff is selected and if not 6  
                then display message to user  
                JOptionPane.showMessageDialog(frame, "There is not  
                enough members in right list", "Right List is not full",  
                JOptionPane.ERROR_MESSAGE);  
            }  
            else  
            {  
                new AdminAllocate();  
                main_frame.setVisible(false);  
            }  
        }  
    }  
});
```

```

        }
    }
    catch (Exception exc)
    {
        System.out.println(exc);
    }
}
});

BackMain.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        try
        {
            //Redirect back to the main administrator frame
            new MainScreenAdmin();
            main_frame.setVisible(false);
        }
        catch (Exception exc)
        {
            System.out.println(exc);
        }
    }
});
main_frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
main_frame.setVisible(true);
}

//Creating left hand list
//Getting Staff Names from the District Nurse table
public String[] listUsers() throws Exception
{
    Class.forName("org.sqlite.JDBC");
    Connection conn =
    DriverManager.getConnection("jdbc:sqlite:NurseScheduling.sqlite");
    Statement stat = conn.createStatement();
    ResultSet rs = stat.executeQuery("select NurseName, Town from DistrictNurse;");

    String[] rawModList = new String[200];
    int index = 0;

    while (rs.next())
    {

```

```

        rawModList[index] = rs.getString("NurseName");
        index++;
    }
    rs.close();
    conn.close();

    int numElements = 0;
    for (int i = 0; i < 200; i++)
    {
        if (rawModList[i] != null)
        {
            numElements++;
        }
    }
    modList = new String[numElements];
    System.arraycopy(rawModList, 0, modList, 0, numElements);

    return modList;
}

```

//This method is called once the move button is selected after an item from the left hand list is selected

//This method will check the size of right hand list before moving a name

//If right hand list size is already 6 then the new selected cannot be added

//This method will also check whether the new item selected matches an item already in right hand list

//If there is a match then the name will not be moved and message displayed to user

//If right hand list is not 6 and does not contain new selection then item is moved to right hand list

```
public void MoveName() throws Exception
```

```

{
    move.setEnabled(false);
    int destSize = data.getSize();

    if (destSize >= 6)
    {
        JOptionPane.showMessageDialog(frame, "You have reached max 6
        nurses", "Right List is full", JOptionPane.ERROR_MESSAGE);
    }
    else if (data.contains(selectedLeftUsername))
    {
        JOptionPane.showMessageDialog(frame, "No duplicates please!");
    }
    else
    {

```

```

        data.addElement(selectedLeftUsername);
        int selectedIndex = list.getSelectedIndex();
    }
    if (destSize == 6)
    {
        move.setEnabled(false);
    }
}

//This method is called once a name from the right hand list is selected
//Then the move back button is selected
//This method will remove the selected item from the right hand list
public void moveBack() throws Exception
{
    move.setEnabled(false);
    data.removeElement(selectedRightUsername);
}

//Return username
public static String Get_User()
{
    return username;
}

//Return defaultlist model data
public static DefaultListModel Get_list()
{
    return data;
}

public static void main(String args[]) throws Exception
{
    new AdminAllocateMainTest();
}
}

```

'Allocate Calls' confirmation date page - Administrator (565):-

```

public class AdminAllocate
{
    DefaultListModel tempList;
    JOptionPane frame = new JOptionPane();
    JFrame main_frame = new JFrame ("Bargoed Nurses");
}

```

```

JPanel main = new JPanel();
private JPanel heading, footer;
private ImageIcon image1;
private JLabel label1, AllocateLabel, EnterLabel, DateLabel, StaffLabel;
static JTextField DateCall = new JTextField("dd/mm/yyyy");
private static String DateEntered;
public static JPanel mainpage = new JPanel();
final JList TestList = new JList();
private static ArrayList now;
JPanel rest = new JPanel(), sidebar = new JPanel();
Color buttonBackground = new Color (234,234,234);
Color blueBackground = new Color (7,32,157);
Font f = new Font("Dialog", Font.PLAIN, 36);
Color PanelBackground = new Color (167, 165, 255);
Font t = new Font("Dialog", Font.PLAIN, 20);

private String s;
private static String fileName;
private static char test;
private static String firstWord;
private static String secondWord;
private static String thirdWord;
private static String fourthWord;
private static String fifthWord;
private static String sixthWord;
private static String firstName;
private static String secondName;
private static String thirdName;
private static String fourthName;
private static String fifthName;
private static String sixthName;
String sixthW;
private static String date;

private static String words;
private static int checkDate;

public AdminAllocate() throws Exception
{
    tempList = AdminAllocateMainTest.Get_list();
    Object[] contents = tempList.toArray();
    String s = tempList.toString();
    int spacePos2 = s.indexOf(",");

    String words[] = s.split(",");

```

```
String firstWord = words[0];
secondWord = words[1];
thirdWord = words[2];
fourthWord = words[3];
fifthWord = words[4];
String sixthWord = words[5];
int sixthTest = sixthWord.length() - 1;
int secondTest = 1 + secondWord.length();
String lastTwo = words[words.length - 2] + " " + words[words.length - 1];

firstName = firstWord.substring(1,spacePos2);
sixthW = sixthWord.substring(0, sixthTest);
sixthName = sixthW.replaceAll("^\\s+", "");
secondName = secondWord.replaceAll("^\\s+", "");
thirdName = thirdWord.replaceAll("^\\s+", "");
fourthName = fourthWord.replaceAll("^\\s+", "");
fifthName = fifthWord.replaceAll("^\\s+", "");

ArrayList now = new ArrayList();
now.add(s);

heading = new JPanel();
heading.setMaximumSize(new Dimension(700,150));
heading.setBackground(blueBackground);
image1 = new ImageIcon(getClass().getResource("Bar.jpg"));
label1 = new JLabel(image1);
heading.add(label1);

footer = new JPanel();
footer.setMaximumSize(new Dimension(800, 300));
footer.setBackground(Color.GREEN);
JButton CreateCall = new JButton("Create Call List");
JButton BackMain = new JButton("Back to Main Screen");
footer.add(CreateCall); footer.add(BackMain);
footer.setBackground(Color.WHITE);

AllocateLabel = new JLabel("Allocate Calls");
AllocateLabel.setFont(f);
EnterLabel = new JLabel("Enter date and then check staff is correct");
DateLabel = new JLabel("Enter date");
StaffLabel = new JLabel("Staff:-");

main_frame.setSize(780,570);
main_frame.setLayout(new FlowLayout());
main_frame.setBackground(Color.WHITE);
```

```

main.setLayout(new BorderLayout(main, BorderLayout.PAGE_AXIS));
main.setPreferredSize(new Dimension(600,250));

final JList TestList = new JList(contents);
final JScrollPane vertical = new JScrollPane(TestList);
vertical.setVerticalScrollBarPolicy(JScrollPane.VERTICAL_SCROLLBAR_ALWAYS);
mainpage.setPreferredSize(new Dimension(500,400));
mainpage.removeAll();
rest.setPreferredSize(new Dimension(800,600));

//Inputting information into sidebar
JPanel TestPanel = new JPanel();
JLabel sidelabel = new JLabel("<html>* Clear enter date text field and enter new
date in the format dd/mm/yyyy");
JLabel sidelabelblank = new JLabel(" ");
JLabel sidelabelblank2 = new JLabel(" ");
JLabel sidelabel2 = new JLabel("<html>* Check the staff members against your rota
list");
JLabel sidelabel21 = new JLabel("<html> * If all correct proceed by selecting create
call list");
TestPanel.setBackground(PanelBackground);
TestPanel.setPreferredSize(new Dimension(100,400));
TestPanel.setLayout(new BorderLayout(TestPanel, BorderLayout.PAGE_AXIS));
TestPanel.add(sidelabel);
TestPanel.add(sidelabelblank);
TestPanel.add(sidelabel2);
TestPanel.add(sidelabelblank2);
TestPanel.add(sidelabel21);
rest.add(TestPanel);
rest.add(mainpage);
rest.setBackground(blueBackground);

main_frame.add(heading); main_frame.add(rest);
main.add(AllocateLabel);
main.add(EnterLabel);
main.add(DateLabel);
DateCall.setMaximumSize(new Dimension(1200,50));
DateCall.setFont(t);
main.add(DateCall);
main.add(StaffLabel);
main.add(vertical);
vertical.setMaximumSize(new Dimension(1200, 100));
mainpage.add(main);
mainpage.add(footer);
mainpage.setPreferredSize(new Dimension(650,400));

```

```
mainpage.setBackground(Color.WHITE);
```

```
main.setBackground(Color.WHITE);
```

```
CreateCall.addActionListener(new ActionListener()
```

```
{
```

```
    public void actionPerformed(ActionEvent e)
```

```
    {
```

```
        try
```

```
        {
```

```
            //Get all values from current Call Lists
```

```
            //Check if table is empty
```

```
            //If table is empty then continue here
```

```
            Class.forName("org.sqlite.JDBC");
```

```
            Connection conn = (Connection)
```

```
            DriverManager.getConnection("jdbc:sqlite:NurseScheduling.sqlite");
```

```
            Statement st = conn.createStatement();
```

```
            ResultSet rs = st.executeQuery("select * from CallLists2");
```

```
            int rowSize = 0;
```

```
            while (rs.next())
```

```
            {
```

```
                rowSize++;
```

```
            }
```

```
            if (rowSize == 0)
```

```
            {
```

```
                rs.close();
```

```
                conn.close();
```

```
                //Call Create Call List method
```

```
                //Call Create Call Patient List method
```

```
                CreateCallList();
```

```
                CreatePatientList();
```

```
            }
```

```
            else
```

```
            {
```

```
                //If table is full then continue here
```

```
                //Create a back up of the Call Lists table
```

```
                //Make a copy of the Call Lists table
```

```
                Class.forName("org.sqlite.JDBC");
```

```
                Connection conntt = (Connection)
```

```
                DriverManager.getConnection("jdbc:sqlite:NurseScheduling.sqlite");
```

```
                Statement sttt = conntt.createStatement();
```

```
                sttt.executeUpdate("INSERT INTO CallListBackUp SELECT *  
FROM CallLists2");
```

```

//Once copied into new table delete the contents of Call Lists
    Class.forName("org.sqlite.JDBC");
    Connection conn = (Connection)
DriverManager.getConnection("jdbc:sqlite:NurseScheduling.sqlite");
    Statement stt = conn.createStatement();
    stt.executeUpdate("delete from CallLists2");

    //Call Create Call List method
    //Call Create Call Patient List method
    CreateCallList();
    CreatePatientList();
}
//Call call List Creation method
callListCreation();
}
catch (Exception exc)
{
    System.out.println(exc);
}
}
});

BackMain.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        try
        {
            //Redirect back to main administrator frame
            main_frame.setVisible(false);
            new MainScreenAdmin();
        }
        catch (Exception exc)
        {
            System.out.println(exc);
        }
    }
});
main_frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
main_frame.setVisible(true);
}

//This method calculates the staff and puts them in the CallLists2 table
//Then calls the Nurse Order method
public void CreateCallList() throws Exception

```

```

{
    Class.forName("org.sqlite.JDBC");
    Connection conn = (Connection)
    DriverManager.getConnection("jdbc:sqlite:NurseScheduling.sqlite");

    String DateE = DateCall.getText();
    checkDate = DateE.length();
    String firstName1 = sixthName;
    String NurseName = "firstName, secondWord, thirdWord, fourthWord, fifthWord,
    sixthName";

    Statement st = conn.createStatement();
    st.executeUpdate("insert into CallLists2 values (" + DateE + ", " + firstName + " ");
    st.executeUpdate("insert into CallLists2 values (" + DateE + ", " + secondName + " ");
    st.executeUpdate("insert into CallLists2 values (" + DateE + ", " + thirdName + " ");
    st.executeUpdate("insert into CallLists2 values (" + DateE + ", " + fourthName + " ");
    st.executeUpdate("insert into CallLists2 values (" + DateE + ", " + fifthName + " ");
    st.executeUpdate("insert into CallLists2 values (" + DateE + ", " + sixthName + " ");

    CreateNurseOrder();
}

//This method checks the amount of blood patients in table Patients2 is correct ... need 30
//Once 30 is found these are stored in temporary table tempBloods
//These are stored in ascending order
public void CreatePatientList() throws Exception
{
    Class.forName("org.sqlite.JDBC");
    Connection connt = (Connection)
    DriverManager.getConnection("jdbc:sqlite:NurseScheduling.sqlite");
    Statement stt = connt.createStatement();
    ResultSet rst = stt.executeQuery("SELECT * FROM Patients2 where
    Bloods='TRUE'");

    int rowSize1 = 0;
    while (rst.next())
    {
        rowSize1++;
    }

    if (rowSize1 > 30)
    {
        JOptionPane.showMessageDialog(frame, "There are too many bloods
        patients", "Failure to create list", JOptionPane.ERROR_MESSAGE);
        rst.close();
    }
}

```

```

        connt.close();
    }
    else if (30 > rowSize1)
    {
        JOptionPane.showMessageDialog(frame, "There isn't enough bloods
        patients", "Failure to create list", JOptionPane.ERROR_MESSAGE);
        rst.close();
        connt.close();
    }
    else if (rowSize1 == 30)
    {
        //Clear the temporary bloods table
        Class.forName("org.sqlite.JDBC");
        Connection connect = (Connection)
        DriverManager.getConnection("jdbc:sqlite:NurseScheduling.sqlite");
        Statement state = connect.createStatement();
        state.executeUpdate("delete from tempBloods");

        //Populate temporary bloods with new list of ordered bloods patients
        Class.forName("org.sqlite.JDBC");
        Connection conn = (Connection)
        DriverManager.getConnection("jdbc:sqlite:NurseScheduling.sqlite");
        Statement sttt = conn.createStatement();
        sttt.executeUpdate("Insert into tempBloods SELECT * FROM Patients2 where
        Bloods='TRUE' order by PTown asc");
        //Call the Create Patient List Other method
        //Call the put Bloods in List method
        CreatePatientListOther();
        putBloodsInList();
    }
}

//This method checks the amount of other patients in table Patients2 is correct ... need 36
//Once 36 is found these are stored in temporary table tempOther
//These are stored in ascending order
public void CreatePatientListOther() throws Exception
{
    Class.forName("org.sqlite.JDBC");
    Connection connt = (Connection)
    DriverManager.getConnection("jdbc:sqlite:NurseScheduling.sqlite");

    Statement stt = connt.createStatement();
    ResultSet rst = stt.executeQuery("SELECT * FROM Patients2 where Bloods='FALSE'");

    int rowSize1 = 0;

```

```

while (rst.next())
{
    rowSize1++;
}

if (rowSize1 > 36)
{
    JOptionPane.showMessageDialog(frame, "There are too many other
patients", "Failure to create list", JOptionPane.ERROR_MESSAGE);
    rst.close();
    connt.close();
}
else if (36 > rowSize1)
{
    JOptionPane.showMessageDialog(frame, "There isn't enough other
patients", "Failure to create list", JOptionPane.ERROR_MESSAGE);
    rst.close();
    connt.close();
}
else if (rowSize1 == 36)
{
    //clear temporary other table ready for new insert
    Class.forName("org.sqlite.JDBC");
    Connection connect = (Connection)
    DriverManager.getConnection("jdbc:sqlite:NurseScheduling.sqlite");
    Statement state = connect.createStatement();
    state.executeUpdate("delete from tempOther");

    //populate tempOther with list of other calls
    Class.forName("org.sqlite.JDBC");
    Connection conn = (Connection)
    DriverManager.getConnection("jdbc:sqlite:NurseScheduling.sqlite");
    Statement sttt = conn.createStatement();
    sttt.executeUpdate("Insert into tempOther SELECT * FROM Patients2 where
Bloods='FALSE' order by PTown asc");

}

}

//This method checks the amount of calls in Calls.. if full then deletes all
//Then inserts a copy of sorted bloods calls from tempBloods into Calls
//NurseName is given the value 'Name' for the time being
public void putBloodsInList() throws Exception

```

```

{
    Class.forName("org.sqlite.JDBC");
    Connection connt = (Connection)
    DriverManager.getConnection("jdbc:sqlite:NurseScheduling.sqlite");
    Statement stt = connt.createStatement();
    ResultSet rst = stt.executeQuery("SELECT * FROM Calls");

    int rowSize1 = 0;
    while (rst.next())
    {
        rowSize1++;
    }

    if (rowSize1 > 1)
    {
        rst.close();
        connt.close();

        Class.forName("org.sqlite.JDBC");
        Connection connect = (Connection)
        DriverManager.getConnection("jdbc:sqlite:NurseScheduling.sqlite");
        Statement state = connect.createStatement();
        state.executeUpdate("delete from Calls");

        Class.forName("org.sqlite.JDBC");
        Connection conn = (Connection)
        DriverManager.getConnection("jdbc:sqlite:NurseScheduling.sqlite");
        Statement sttt = conn.createStatement();
        sttt.executeUpdate("Insert into Calls select PID, PName, FName, PAddress, PTown,
        Treatment, Kcode, Equipment, PTelNumber, 'Name' from tempBloods");
    }
    else
    {
        Class.forName("org.sqlite.JDBC");
        Connection conn = (Connection)
        DriverManager.getConnection("jdbc:sqlite:NurseScheduling.sqlite");
        Statement sttt = conn.createStatement();
        sttt.executeUpdate("Insert into Calls select PID, PName, FName, PAddress, PTown,
        Treatment, Kcode, Equipment, PTelNumber, 'Name' from tempBloods");
    }

    //Call put others in list method
    putOthersInList();
}

```

```

//Then inserts a copy of sorted other calls from temporary other into Calls
//NurseName is given the value 'Name' for the time being
public void putOthersInList() throws Exception
{
    Class.forName("org.sqlite.JDBC");
    Connection conn = (Connection)
    DriverManager.getConnection("jdbc:sqlite:NurseScheduling.sqlite");
    Statement sttt = conn.createStatement();
    sttt.executeUpdate("Insert into Calls select PID, PName, FName, PAddress, PTown,
        Treatment, Kcode, Equipment, PTElNumber, 'Name' from tempOther");
//Call the add Nurses method
addNurses();

}

//This method puts the selected staff in order based on where they live (Town)
public void CreateNurseOrder() throws Exception
{
    Class.forName("org.sqlite.JDBC");
    Connection connt = (Connection)
    DriverManager.getConnection("jdbc:sqlite:NurseScheduling.sqlite");
    Statement stt = connt.createStatement();
    ResultSet rst = stt.executeQuery("SELECT * FROM CallLists2");

int rowSize1 = 0;
while (rst.next())
{
    rowSize1++;
}

if (rowSize1 == 6)
{
    //If table is already populated then clear label before new insertion
    Class.forName("org.sqlite.JDBC");
    Connection connect = (Connection)
    DriverManager.getConnection("jdbc:sqlite:NurseScheduling.sqlite");
    Statement state = connect.createStatement();
    state.executeUpdate("delete from CallLists");

    //Insert into call lists table the list of stored staff based on Town
    Class.forName("org.sqlite.JDBC");
    Connection conn = (Connection)
    DriverManager.getConnection("jdbc:sqlite:NurseScheduling.sqlite");
    Statement sttt = conn.createStatement();

```

```

        sttt.executeUpdate("Insert into CallLists select CallLists2.Date,
        DistrictNurse.NurseName from CallLists2, DistrictNurse where CallLists2.NurseName
        = DistrictNurse.NurseName order by Town asc");
    }
    else
    {
        JOptionPane.showMessageDialog(frame, "Call List has not been created
        yet","Call List",JOptionPane.ERROR_MESSAGE);
    }

}

//This method updates calls table with the list of working staff
public void addNurses() throws Exception
{
    Class.forName("org.sqlite.JDBC");
    Connection conn =
    DriverManager.getConnection("jdbc:sqlite:NurseScheduling.sqlite");

String group1 = "Update Calls Set NurseName=? Where rowid in(1,2,3,4,5,31,32,33,34,35,36)";
PreparedStatement a = conn.prepareStatement(group1);
a.setString(1,firstName);

String group2 = "Update Calls Set NurseName=? Where rowid in(6,7,8,9,10,37,38,39,40,41,42)";
PreparedStatement b = conn.prepareStatement(group2);
b.setString(1,secondName);

String group3 = "Update Calls Set NurseName=? Where rowid in(11,12,13,14,15,43,44,45,46,47,48)";
PreparedStatement c = conn.prepareStatement(group3);
c.setString(1,thirdName);

String group4 = "Update Calls Set NurseName=? Where rowid in(16,17,18,19,20,49,50,51,52,53,54)";
PreparedStatement d = conn.prepareStatement(group4);
d.setString(1,fourthName);

String group5 = "Update Calls Set NurseName=? Where rowid in(21,22,23,24,25,55,56,57,58,59,60)";
PreparedStatement e = conn.prepareStatement(group5);
e.setString(1,fifthName);

String group6 = "Update Calls Set NurseName=? Where rowid in(26,27,28,29,30,61,62,63,64,65,66)";
PreparedStatement f = conn.prepareStatement(group6);
f.setString(1,sixthName);

a.executeUpdate();

```

```

b.executeUpdate();
c.executeUpdate();
d.executeUpdate();
e.executeUpdate();
f.executeUpdate();
    }

    //This method confirms call list has been created
    public void callListCreation() throws Exception
    {
        main_frame.setVisible(false);
        JOptionPane.showMessageDialog(frame, "Call List created", "Call
        List", JOptionPane.ERROR_MESSAGE);
        new MainScreenAdmin();
    }

    //Return the date as a String
    public static String Get_date()
    {
        return date;
    }

    public static void main(String args[]) throws Exception
    {
        new AdminAllocate();
    }
}

```

'Search staff' main page - Administrator (303):-

```

public class SearchStaff
{
    static String nName, TownD, Band, pNumber;
    JOptionPane frame = new JOptionPane();
    JFrame staff_frame = new JFrame ("Bargoed Nurses");
    private JPanel heading;
    private ImageIcon image1;
    private JLabel label1;
    public static JPanel mainpage = new JPanel();
    JPanel rest = new JPanel(), sidebar = new JPanel();
    Color buttonBackground = new Color (234,234,234);
    Color blueBackground = new Color (7,32,157);
    Color PanelBackground = new Color (167, 165, 255);
    Font f = new Font("Dialog", Font.PLAIN, 20);
    static JTextField patientNameField = new JTextField();
}

```

```

static JList listtest = new JList();

public SearchStaff() throws Exception
{
    heading = new JPanel();
    heading.setMaximumSize(new Dimension(700,150));
    heading.setBackground(blueBackground);
    image1 = new ImageIcon(getClass().getResource("Bar.jpg"));
    label1 = new JLabel(image1);
    heading.add(label1);

    JButton search = new JButton("Search for Staff details");
    JButton back = new JButton("Back to main screen");
    staff_frame.setSize(780,570);
    staff_frame.setLayout(new FlowLayout());
    staff_frame.setBackground(Color.WHITE);

    mainpage.setPreferredSize(new Dimension(500,400));
    mainpage.removeAll();

    rest.setPreferredSize(new Dimension(800,600));
    rest.setBackground(blueBackground);

    staff_frame.add(heading); staff_frame.add(rest);

    //Inputting information into sidebar
    JPanel TestPanel = new JPanel();
    JLabel sidelabel = new JLabel("<html>* Enter known <br> patient details <br>into  
textboxes<br> provided ");
    JLabel sidelabelblank = new JLabel(" ");
    JLabel sidelabelblank2 = new JLabel(" ");
    JLabel sidelabelblank3 = new JLabel(" ");
    JLabel sidelabel2 = new JLabel("<html>* A first name for a patient needs to be  
entered");
    JLabel sidelabel3 = new JLabel("<html>* 2 details need <br> to be entered ");
    JLabel sidelabel4 = new JLabel("<html>* Once criteria has been entered then select  
appropriate button at bottom of the page");
    TestPanel.setBackground(PanelBackground);

    TestPanel.setPreferredSize(new Dimension(100,400));
    TestPanel.setLayout(new BorderLayout(TestPanel, BorderLayout.PAGE_AXIS));
    TestPanel.add(sidelabel);
    TestPanel.add(sidelabelblank);
    TestPanel.add(sidelabel2);
    TestPanel.add(sidelabelblank2);

```

```
TestPanel.add(sidelabel3);
TestPanel.add(sidelabelblank3);
TestPanel.add(sidelabel4);

rest.add(TestPanel); rest.add(mainpage);

JPanel moduleListPanel = new JPanel();
moduleListPanel.setLayout(new FlowLayout());
moduleListPanel.setPreferredSize(new Dimension(600,600));
JLabel welcome = new JLabel("<html>Enter known staff criteria:");
welcome.setFont(f);
welcome.setPreferredSize(new Dimension(600, 20));
welcome.setHorizontalTextPosition(JLabel.LEFT);
moduleListPanel.add(welcome);

JPanel txtPanel = new JPanel();
txtPanel.setLayout(null);
txtPanel.setLocation(20, 50);
txtPanel.setPreferredSize(new Dimension(200, 330));
txtPanel.setBackground(Color.WHITE);

moduleListPanel.add(txtPanel);
moduleListPanel.setBackground(Color.WHITE);

JPanel txtFieldPanel = new JPanel();
txtFieldPanel.setLayout(null);
txtFieldPanel.setLocation(150, 60);
txtFieldPanel.setPreferredSize(new Dimension(390, 330));
txtFieldPanel.setBackground(Color.WHITE);
moduleListPanel.add(txtFieldPanel);

JLabel nurseName = new JLabel("District Nurse name:");
nurseName.setLocation(0, 0);
nurseName.setSize(300, 50);
nurseName.setHorizontalTextPosition(JLabel.LEFT);
txtPanel.add(nurseName);

JLabel pinNumber = new JLabel("Pin Number:");
pinNumber.setLocation(0,50);
pinNumber.setSize(300,50);
pinNumber.setHorizontalTextPosition(JLabel.LEFT);
txtPanel.add(pinNumber);

JLabel Town = new JLabel("Town:");
Town.setLocation(0,100);
```

```
Town.setSize(300,50);
Town.setHorizontalTextPosition(JLabel.LEFT);
txtPanel.add(Town);

JLabel bandNumber = new JLabel("Band:");
bandNumber.setLocation(0,150);
bandNumber.setSize(300,50);
bandNumber.setHorizontalTextPosition(JLabel.LEFT);
txtPanel.add(bandNumber);

final JTextField nurseNameField = new JTextField();
nurseNameField.setEditable(true);
nurseNameField.setLocation(0, 0);
nurseNameField.setSize(390, 50);
nurseNameField.setBackground(PanelBackground);
txtFieldPanel.add(nurseNameField);

final JTextField pinNumberField = new JTextField();
pinNumberField.setEditable(true);
pinNumberField.setLocation(0, 50);
pinNumberField.setSize(390, 50);
pinNumberField.setBackground(PanelBackground);
txtFieldPanel.add(pinNumberField);

final JTextField TownField = new JTextField();
TownField.setEditable(true);
TownField.setLocation(0, 100);
TownField.setSize(390, 50);
TownField.setBackground(PanelBackground);
txtFieldPanel.add(TownField);

final JTextField BandField = new JTextField();
BandField.setEditable(true);
BandField.setLocation(0, 150);
BandField.setSize(390, 50);
BandField.setBackground(PanelBackground);
txtFieldPanel.add(BandField);

moduleListPanel.add(search);
moduleListPanel.add(back);

mainpage.add(moduleListPanel);
mainpage.setPreferredSize(new Dimension(650,400));
mainpage.setBackground(Color.WHITE);
```

```

back.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        try
        {
            //Redirect to the main administrator page
            staff_frame.setVisible(false);
            new MainScreenAdmin();
        }
        catch (Exception exc)
        {
            System.out.println(exc);
        }
    }
});

search.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        try
        {
            nName = nurseNameField.getText();
            pNumber = pinNumberField.getText();
            TownD = TownField.getText();
            Band = BandField.getText();

            String str = nurseNameField.getText();

            //Get the values entered from the textfields provided
            //Use these values to find information in the District Nurse
            table

            Class.forName("org.sqlite.JDBC");
            Connection conn =
            DriverManager.getConnection("jdbc:sqlite:NurseScheduling.sqlite");
            PreparedStatement stat = conn.prepareStatement("select * from
            DistrictNurse where NurseName=? and PinNumber=?");

            //SQL Injection Prevention
            stat.setString(1, nName);
            stat.setString(2, pNumber);

            ResultSet rs = stat.executeQuery();
            Class.forName("org.sqlite.JDBC");

```

```

Connection connt =
DriverManager.getConnection("jdbc:sqlite:NurseScheduling.sqlite");
PreparedStatement statt = conn.prepareStatement("select * from
DistrictNurse where NurseName=? and Town=?");

//SQL Injection Prevention
statt.setString(1, nName);
statt.setString(2, TownD);
ResultSet st = statt.executeQuery();
Class.forName("org.sqlite.JDBC");
Connection connr =
DriverManager.getConnection("jdbc:sqlite:NurseScheduling.sqlite");
PreparedStatement statr = conn.prepareStatement("select * from
DistrictNurse where NurseName=? and Band=?");

//SQL Injection Prevention
statr.setString(1, nName);
statr.setString(2, Band);
ResultSet sr = statr.executeQuery();

if (rs.next())
{
//Message to user detailing a match and redirect to
appropriate page
JOptionPane.showMessageDialog(frame, "Details have been
found to match the nurse name and pin number", "Staff
Criteria",JOptionPane.ERROR_MESSAGE);
new StaffDetailsPin();
staff_frame.setVisible(false);
}
else if (st.next())
{
//Message to user detailing a match and redirect to
appropriate page
JOptionPane.showMessageDialog(frame, "Details have been
found to match the nurse name and town", "Staff
Criteria",JOptionPane.ERROR_MESSAGE);
new SearchStaffDetailsTown();
staff_frame.setVisible(false);
}
else if (sr.next())
{
//Message to user detailing a match and redirect to
appropriate page
JOptionPane.showMessageDialog(frame, "Details have been

```

```

        found to match the nurse name and band", "Staff
Criteria", JOptionPane.ERROR_MESSAGE);
        new StaffDetailsBand();
        staff_frame.setVisible(false);
    }
    else if (str.equals(""))
    {
        //Message user detailing they need to enter a full name in
order to continue
        JOptionPane.showMessageDialog(frame, "A full length
nurse name needs to be entered", "Unentered
Details", JOptionPane.ERROR_MESSAGE);
    }
    else
    {
        //Message to user details there were no matches based on
criteria entered
        JOptionPane.showMessageDialog(frame, "No details found
for criteria entered", "Unfound District Nurse",
JOptionPane.ERROR_MESSAGE);
    }
}
}
catch (Exception exc)
{
    System.out.println(exc);
}
}
});
staff_frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
staff_frame.setVisible(true);
}

//Return nurse name
public static String Get_Nurse()
{
    return nName;
}

//Return pin number
public static String Get_Pin()
{
    return pNumber;
}
}

```

```

//Return town
public static String Get_Town()
{
    return TownD;
}

//Return band
public static String Get_Band()
{
    return Band;
}

public static void main(String args[]) throws Exception
{
    new SearchStaff();
}
}

```

'Search staff' pin and nurse name match- Administrator (248):-

```

public class StaffDetailsPin
{
    static String username;
    String nName, pinNumber, TownDa, Band;
    JFrame refreshFrame;
    JOptionPane frame = new JOptionPane();
    JFrame nurse_frame = new JFrame ("Bargoed Nurses");
    private JPanel heading;
    private ImageIcon image1;
    private JLabel label1;
    public static JPanel mainpage = new JPanel();
    JPanel rest = new JPanel(), sidebar = new JPanel();
    Color buttonBackground = new Color (234,234,234);
    Color blueBackground = new Color (7,32,157);
    Color PanelBackground = new Color (167, 165, 255);
    Font f = new Font("Dialog", Font.PLAIN, 20);

    public StaffDetailsPin() throws Exception
    {
        nName = SearchStaff.Get_Nurse();
        pinNumber = SearchStaff.Get_Pin();
        TownDa = SearchStaff.Get_Town();

        heading = new JPanel();
        heading.setMaximumSize(new Dimension(700,150));
    }
}

```

```

heading.setBackground(blueBackground);
image1 = new ImageIcon(getClass().getResource("Bar.jpg"));
label1 = new JLabel(image1);
heading.add(label1);

JButton print = new JButton("Print");
JButton searchStaff = new JButton("Search for another District nurse");
JButton back = new JButton("Back to Main");
nurse_frame.setSize(780,570);
nurse_frame.setLayout(new FlowLayout());
nurse_frame.setBackground(Color.WHITE);

mainpage.setPreferredSize(new Dimension(500,400));
mainpage.removeAll();

rest.setPreferredSize(new Dimension(800,600));
rest.setBackground(blueBackground);

nurse_frame.add(heading); nurse_frame.add(rest);

JPanel TestPanel = new JPanel();
TestPanel.setBackground(PanelBackground);
TestPanel.setPreferredSize(new Dimension(100,400));
TestPanel.setLayout(new BorderLayout(TestPanel, BorderLayout.PAGE_AXIS));

rest.add(TestPanel); rest.add(mainpage);
JPanel moduleListPanel = new JPanel();
moduleListPanel.setLayout(new FlowLayout());
moduleListPanel.setPreferredSize(new Dimension(600,600));
JLabel welcome = new JLabel("<html>Your personal details are:");
welcome.setPreferredSize(new Dimension(600, 20));
welcome.setHorizontalTextPosition(JLabel.LEFT);
moduleListPanel.add(welcome);

//Get details from the District Nurse table where matches are made for nurse name
and pin number
Class.forName("org.sqlite.JDBC");
Connection conn =
DriverManager.getConnection("jdbc:sqlite:NurseScheduling.sqlite");
Statement stat = conn.createStatement();
ResultSet rs = stat.executeQuery(String.format("select * from DistrictNurse where
NurseName='%s' and PinNumber='%s'", nName, pinNumber));

JPanel txtPanel = new JPanel();
txtPanel.setLayout(null);

```

```
txtPanel.setLocation(20, 50);
txtPanel.setPreferredSize(new Dimension(200, 330));
txtPanel.setBackground(Color.WHITE);

moduleListPanel.add(txtPanel);
moduleListPanel.setBackground(Color.WHITE);

JPanel txtFieldPanel = new JPanel();
txtFieldPanel.setLayout(null);
txtFieldPanel.setLocation(150, 60);
txtFieldPanel.setPreferredSize(new Dimension(390, 330));
txtFieldPanel.setBackground(Color.WHITE);
moduleListPanel.add(txtFieldPanel);

JLabel nurseName = new JLabel("District Nurse Name:");
nurseName.setLocation(0, 0);
nurseName.setSize(300, 30);
nurseName.setHorizontalTextPosition(JLabel.LEFT);
txtPanel.add(nurseName);

JLabel pinNum = new JLabel("Pin Number:");
pinNum.setLocation(0,40);
pinNum.setSize(300,30);
pinNum.setHorizontalTextPosition(JLabel.LEFT);
txtPanel.add(pinNum);

JLabel Address = new JLabel("Address:");
Address.setLocation(0,80);
Address.setSize(300,30);
Address.setHorizontalTextPosition(JLabel.LEFT);
txtPanel.add(Address);

JLabel Town = new JLabel("Town:");
Town.setLocation(0,120);
Town.setSize(300,30);
Town.setHorizontalTextPosition(JLabel.LEFT);
txtPanel.add(Town);

JLabel BandLevel = new JLabel("Band:");
BandLevel.setLocation(0,160);
BandLevel.setSize(300,30);
BandLevel.setHorizontalTextPosition(JLabel.LEFT);
txtPanel.add(BandLevel);

JLabel TelNumber = new JLabel("Telephone Number:");
```

```
TelNumber.setLocation(0,200);
TelNumber.setSize(300,30);
TelNumber.setHorizontalTextPosition(JLabel.LEFT);
txtPanel.add(TelNumber);

JTextField nurseNameField = new JTextField(rs.getString("NurseName"));
nurseNameField.setEditable(false);
nurseNameField.setLocation(0, 0);
nurseNameField.setSize(390, 30);
nurseNameField.setBackground(PanelBackground);
txtFieldPanel.add(nurseNameField);

JTextField pinNumField = new JTextField(rs.getString("PinNumber"));
pinNumField.setEditable(false);
pinNumField.setLocation(0, 40);
pinNumField.setSize(390, 30);
pinNumField.setBackground(PanelBackground);
txtFieldPanel.add(pinNumField);

final JTextArea addressArea = new JTextArea(rs.getString("Address"));
addressArea.setSize(390, 30);
addressArea.setEditable(false);
addressArea.setBackground(PanelBackground);
addressArea.setLineWrap(true);

final JScrollPane scrollPane = new JScrollPane(addressArea);
scrollPane.setLocation(0, 80);
scrollPane.setSize(390, 30);
txtFieldPanel.add(scrollPane);

JTextField TownField = new JTextField(rs.getString("Town"));
TownField.setEditable(false);
TownField.setLocation(0, 120);
TownField.setSize(390, 30);
TownField.setBackground(PanelBackground);
txtFieldPanel.add(TownField);

JTextField bandField = new JTextField(rs.getString("Band"));
bandField.setEditable(false);
bandField.setLocation(0, 160);
bandField.setSize(390, 30);
bandField.setBackground(PanelBackground);
txtFieldPanel.add(bandField);

JTextField TelNumField = new JTextField(rs.getString("TelNumberN"));
```

```

TelNumField.setEditable(false);
TelNumField.setLocation(0, 200);
TelNumField.setSize(390, 30);
TelNumField.setBackground(PanelBackground);
txtFieldPanel.add(TelNumField);

moduleListPanel.add(print);
moduleListPanel.add(searchStaff);
moduleListPanel.add(back);

mainpage.add(moduleListPanel);
mainpage.setPreferredSize(new Dimension(650,400));
mainpage.setBackground(Color.WHITE);

back.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        try
        {
            //Redirect back to the main administrator page
            nurse_frame.setVisible(false);
            new MainScreenAdmin();
        }
        catch (Exception exc)
        {
            System.out.println(exc);
        }
    }
});

print.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        try
        {
            //Message displayed to user indicating they have selected
            the print option
            JOptionPane.showMessageDialog(frame, "You have selected
            the print button", "Printing",
            JOptionPane.INFORMATION_MESSAGE);
        }
        catch (Exception exc)
        {

```

```

        System.out.println(exc);
    }
}
});

searchStaff.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        try
        {
            //Redirect back to main search staff page
            nurse_frame.setVisible(false);
            new SearchStaff();
        }
        catch (Exception exc)
        {
            System.out.println(exc);
        }
    }
});
nurse_frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
nurse_frame.setVisible(true);
}

public static void main(String args[]) throws Exception
{
    new StaffDetailsPin();
}
}

```

'Search staff' town and nurse name match- Administrator (248):-

```

public class SearchStaffDetailsTown
{
    static String username;
    String nName, pinNumber, TownDa, Band;
    JFrame refreshFrame;
    JOptionPane frame = new JOptionPane();
    JFrame nurse_frame = new JFrame ("Bargoed Nurses");
    private JPanel heading;
    private ImageIcon image1;
    private JLabel label1;
    public static JPanel mainpage = new JPanel();
    JPanel rest = new JPanel(), sidebar = new JPanel();
}

```

```

Color buttonBackground = new Color (234,234,234);
Color blueBackground = new Color (7,32,157);
Color PanelBackground = new Color (167, 165, 255);
Font f = new Font("Dialog", Font.PLAIN, 20);

public SearchStaffDetailsTown() throws Exception
{
    nName = SearchStaff.Get_Nurse();
    pinNumber = SearchStaff.Get_Pin();
    TownDa = SearchStaff.Get_Town();

    heading = new JPanel();
    heading.setMaximumSize(new Dimension(700,150));
    heading.setBackground(blueBackground);
    image1 = new ImageIcon(getClass().getResource("Bar.jpg"));
    label1 = new JLabel(image1);
    heading.add(label1);

    JButton print = new JButton("Print");
    JButton searchStaff = new JButton("Search for another District nurse");
    JButton back = new JButton("Back to Main");
    nurse_frame.setSize(780,570);
    nurse_frame.setLayout(new FlowLayout());
    nurse_frame.setBackground(Color.WHITE);

    mainpage.setPreferredSize(new Dimension(500,400));
    mainpage.removeAll();

    rest.setPreferredSize(new Dimension(800,600));
    rest.setBackground(blueBackground);
    nurse_frame.add(heading); nurse_frame.add(rest);

    JPanel TestPanel = new JPanel();

    TestPanel.setBackground(PanelBackground);
    TestPanel.setPreferredSize(new Dimension(100,400));
    TestPanel.setLayout(new BorderLayout(TestPanel, BorderLayout.PAGE_AXIS));
    rest.add(TestPanel); rest.add(mainpage);

    JPanel moduleListPanel = new JPanel();
    moduleListPanel.setLayout(new FlowLayout());
    moduleListPanel.setPreferredSize(new Dimension(600,600));
    JLabel welcome = new JLabel("<html>Your personal details are:");
    welcome.setPreferredSize(new Dimension(600, 20));
    welcome.setHorizontalTextPosition(JLabel.LEFT);

```

```
moduleListPanel.add(welcome);

//Get details from the District Nurse table where matches are made for nurse name
and town
Class.forName("org.sqlite.JDBC");
Connection conn =
DriverManager.getConnection("jdbc:sqlite:NurseScheduling.sqlite");
Statement stat = conn.createStatement();
ResultSet rs = stat.executeQuery(String.format("select * from DistrictNurse where
NurseName='%s' and Town='%s'", nName, TownDa));

JPanel txtPanel = new JPanel();
txtPanel.setLayout(null);
txtPanel.setLocation(20, 50);
txtPanel.setPreferredSize(new Dimension(200, 330));
txtPanel.setBackground(Color.WHITE);

moduleListPanel.add(txtPanel);
moduleListPanel.setBackground(Color.WHITE);

JPanel txtFieldPanel = new JPanel();
txtFieldPanel.setLayout(null);
txtFieldPanel.setLocation(150, 60);
txtFieldPanel.setPreferredSize(new Dimension(390, 330));
txtFieldPanel.setBackground(Color.WHITE);
moduleListPanel.add(txtFieldPanel);

JLabel nurseName = new JLabel("District Nurse Name:");
nurseName.setLocation(0, 0);
nurseName.setSize(300, 30);
nurseName.setHorizontalTextPosition(JLabel.LEFT);
txtPanel.add(nurseName);

JLabel pinNum = new JLabel("Pin Number:");
pinNum.setLocation(0,40);
pinNum.setSize(300,30);
pinNum.setHorizontalTextPosition(JLabel.LEFT);
txtPanel.add(pinNum);

JLabel Address = new JLabel("Address:");
Address.setLocation(0,80);
Address.setSize(300,30);
Address.setHorizontalTextPosition(JLabel.LEFT);
txtPanel.add(Address);
```

```
JLabel Town = new JLabel("Town: ");
Town.setLocation(0,120);
Town.setSize(300,30);
Town.setHorizontalTextPosition(JLabel.LEFT);
txtPanel.add(Town);

JLabel BandLevel = new JLabel("Band:");
BandLevel.setLocation(0,160);
BandLevel.setSize(300,30);
BandLevel.setHorizontalTextPosition(JLabel.LEFT);
txtPanel.add(BandLevel);

JLabel TelNumber = new JLabel("Telephone Number:");
TelNumber.setLocation(0,200);
TelNumber.setSize(300,30);
TelNumber.setHorizontalTextPosition(JLabel.LEFT);
txtPanel.add(TelNumber);

JTextField nurseNameField = new JTextField(rs.getString("NurseName"));
nurseNameField.setEditable(false);
nurseNameField.setLocation(0, 0);
nurseNameField.setSize(390, 30);
nurseNameField.setBackground(PanelBackground);
txtFieldPanel.add(nurseNameField);

JTextField pinNumField = new JTextField(rs.getString("PinNumber"));
pinNumField.setEditable(false);
pinNumField.setLocation(0, 40);
pinNumField.setSize(390, 30);
pinNumField.setBackground(PanelBackground);
txtFieldPanel.add(pinNumField);

final JTextArea addressArea = new JTextArea(rs.getString("Address"));
addressArea.setSize(390, 30);
addressArea.setEditable(false);
addressArea.setBackground(PanelBackground);
addressArea.setLineWrap(true);

final JScrollPane scrollPane = new JScrollPane(addressArea);
scrollPane.setLocation(0, 80);
scrollPane.setSize(390, 30);
txtFieldPanel.add(scrollPane);

JTextField TownField = new JTextField(rs.getString("Town"));
TownField.setEditable(false);
```

```

TownField.setLocation(0, 120);
TownField.setSize(390, 30);
TownField.setBackground(PanelBackground);
txtFieldPanel.add(TownField);

JTextField bandField = new JTextField(rs.getString("Band"));
bandField.setEditable(false);
bandField.setLocation(0, 160);
bandField.setSize(390, 30);
bandField.setBackground(PanelBackground);
txtFieldPanel.add(bandField);

JTextField TelNumField = new JTextField(rs.getString("TelNumberN"));
TelNumField.setEditable(false);
TelNumField.setLocation(0, 200);
TelNumField.setSize(390, 30);
TelNumField.setBackground(PanelBackground);
txtFieldPanel.add(TelNumField);

moduleListPanel.add(print);
moduleListPanel.add(searchStaff);
moduleListPanel.add(back);

mainpage.add(moduleListPanel);
mainpage.setPreferredSize(new Dimension(650,400));
mainpage.setBackground(Color.WHITE);

back.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        try
        {
            //Redirect back to the main administrator page
            nurse_frame.setVisible(false);
            new MainScreenAdmin();
        }
        catch (Exception exc)
        {
            System.out.println(exc);
        }
    }
});

print.addActionListener(new ActionListener()

```

```

    {
        public void actionPerformed(ActionEvent e)
        {
            try
            {
                //Message displayed indicating user has selected the print option
                JOptionPane.showMessageDialog(frame, "You have selected the
                print button", "Printing", JOptionPane.INFORMATION_MESSAGE);
            }
            catch (Exception exc)
            {
                System.out.println(exc);
            }
        }
    }
});

searchStaff.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        try
        {
            //Redirect back to the search staff main page
            nurse_frame.setVisible(false);
            new SearchStaff();
        }
        catch (Exception exc)
        {
            System.out.println(exc);
        }
    }
});

nurse_frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
nurse_frame.setVisible(true);
}

public static void main(String args[]) throws Exception
{
    new SearchStaffDetailsTown();
}
}

```

'Search staff' band and nurse name match- Administrator (248):-

```

public class StaffDetailsBand
{
    static String username;
    String nName, pinNumber, TownDa, Band;
    JFrame refreshFrame;
    JOptionPane frame = new JOptionPane();
    JFrame nurse_frame = new JFrame ("Bargoed Nurses");
    private JPanel heading;
    private ImageIcon image1;
    private JLabel label1;
    public static JPanel mainpage = new JPanel();
    JPanel rest = new JPanel(), sidebar = new JPanel();
    Color buttonBackground = new Color (234,234,234);
    Color blueBackground = new Color (7,32,157);
    Color PanelBackground = new Color (167, 165, 255);
    Font f = new Font("Dialog", Font.PLAIN, 20);

    public StaffDetailsBand() throws Exception
    {
        nName = SearchStaff.Get_Nurse();
        pinNumber = SearchStaff.Get_Pin();
        TownDa = SearchStaff.Get_Town();
        Band = SearchStaff.Get_Band();

        heading = new JPanel();
        heading.setMaximumSize(new Dimension(700,150));
        heading.setBackground(blueBackground);
        image1 = new ImageIcon(getClass().getResource("Bar.jpg"));
        label1 = new JLabel(image1);
        heading.add(label1);

        JButton print = new JButton("Print");
        JButton searchStaff = new JButton("Search for another District nurse");
        JButton back = new JButton("Back to Main");
        nurse_frame.setSize(780,570);
        nurse_frame.setLayout(new FlowLayout());
        nurse_frame.setBackground(Color.WHITE);

        mainpage.setPreferredSize(new Dimension(500,400));
        mainpage.removeAll();
        rest.setPreferredSize(new Dimension(800,600));
        rest.setBackground(blueBackground);

        nurse_frame.add(heading); nurse_frame.add(rest);
        JPanel TestPanel = new JPanel();
    }
}

```

```

TestPanel.setBackground(PanelBackground);
TestPanel.setPreferredSize(new Dimension(100,400));
TestPanel.setLayout(new BorderLayout(TestPanel, BorderLayout.PAGE_AXIS));

rest.add(TestPanel); rest.add(mainpage);
JPanel moduleListPanel = new JPanel();
moduleListPanel.setLayout(new FlowLayout());
moduleListPanel.setPreferredSize(new Dimension(600,600));
JLabel welcome = new JLabel("<html>Your personal details are:");
welcome.setPreferredSize(new Dimension(600, 20));
welcome.setHorizontalTextPosition(JLabel.LEFT);
moduleListPanel.add(welcome);

//Get details from the District Nurse table where matches are made for nurse name
and band
Class.forName("org.sqlite.JDBC");
Connection conn =
DriverManager.getConnection("jdbc:sqlite:NurseScheduling.sqlite");
Statement stat = conn.createStatement();
ResultSet rs = stat.executeQuery(String.format("select * from DistrictNurse where
NurseName='%s' and Band='%s'", nName, Band));

JPanel txtPanel = new JPanel();
txtPanel.setLayout(null);
txtPanel.setLocation(20, 50);
txtPanel.setPreferredSize(new Dimension(200, 330));
txtPanel.setBackground(Color.WHITE);

moduleListPanel.add(txtPanel);
moduleListPanel.setBackground(Color.WHITE);

JPanel txtFieldPanel = new JPanel();
txtFieldPanel.setLayout(null);
txtFieldPanel.setLocation(150, 60);
txtFieldPanel.setPreferredSize(new Dimension(390, 330));
txtFieldPanel.setBackground(Color.WHITE);
moduleListPanel.add(txtFieldPanel);

JLabel nurseName = new JLabel("District Nurse Name:");
nurseName.setLocation(0, 0);
nurseName.setSize(300, 30);
nurseName.setHorizontalTextPosition(JLabel.LEFT);
txtPanel.add(nurseName);

JLabel pinNum = new JLabel("Pin Number:");

```

```
pinNum.setLocation(0,40);
pinNum.setSize(300,30);
pinNum.setHorizontalTextPosition(JLabel.LEFT);
txtPanel.add(pinNum);
```

```
JLabel Address = new JLabel("Address:");
Address.setLocation(0,80);
Address.setSize(300,30);
Address.setHorizontalTextPosition(JLabel.LEFT);
txtPanel.add(Address);
```

```
JLabel Town = new JLabel("Town: ");
Town.setLocation(0,120);
Town.setSize(300,30);
Town.setHorizontalTextPosition(JLabel.LEFT);
txtPanel.add(Town);
```

```
JLabel BandLevel = new JLabel("Band:");
BandLevel.setLocation(0,160);
BandLevel.setSize(300,30);
BandLevel.setHorizontalTextPosition(JLabel.LEFT);
txtPanel.add(BandLevel);
```

```
JLabel TelNumber = new JLabel("Telephone Number:");
TelNumber.setLocation(0,200);
TelNumber.setSize(300,30);
TelNumber.setHorizontalTextPosition(JLabel.LEFT);
txtPanel.add(TelNumber);
```

```
JTextField nurseNameField = new JTextField(rs.getString("NurseName"));
nurseNameField.setEditable(false);
nurseNameField.setLocation(0, 0);
nurseNameField.setSize(390, 30);
nurseNameField.setBackground(PanelBackground);
txtFieldPanel.add(nurseNameField);
```

```
JTextField pinNumField = new JTextField(rs.getString("PinNumber"));
pinNumField.setEditable(false);
pinNumField.setLocation(0, 40);
pinNumField.setSize(390, 30);
pinNumField.setBackground(PanelBackground);
txtFieldPanel.add(pinNumField);
```

```
final JTextArea addressArea = new JTextArea(rs.getString("Address"));
addressArea.setSize(390, 30);
```

```
addressArea.setEditable(false);
addressArea.setBackground(PanelBackground);
addressArea.setLinewrap(true);

final JScrollPane scrollPane = new JScrollPane(addressArea);
scrollPane.setLocation(0, 80);
scrollPane.setSize(390, 30);
txtFieldPanel.add(scrollPane);

JTextField TownField = new JTextField(rs.getString("Town"));
TownField.setEditable(false);
TownField.setLocation(0, 120);
TownField.setSize(390, 30);
TownField.setBackground(PanelBackground);
txtFieldPanel.add(TownField);

JTextField bandField = new JTextField(rs.getString("Band"));
bandField.setEditable(false);
bandField.setLocation(0, 160);
bandField.setSize(390, 30);
bandField.setBackground(PanelBackground);
txtFieldPanel.add(bandField);

JTextField TelNumField = new JTextField(rs.getString("TelNumberN"));
TelNumField.setEditable(false);
TelNumField.setLocation(0, 200);
TelNumField.setSize(390, 30);
TelNumField.setBackground(PanelBackground);
txtFieldPanel.add(TelNumField);

moduleListPanel.add(print);
moduleListPanel.add(searchStaff);
moduleListPanel.add(back);

mainpage.add(moduleListPanel);
mainpage.setPreferredSize(new Dimension(650,400));
mainpage.setBackground(Color.WHITE);

back.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        try
        {
            //Redirect back to the main administrator page
```

```

        nurse_frame.setVisible(false);
        new mainScreenAdmin();
    }
    catch (Exception exc)
    {
        System.out.println(exc);
    }
}
});

print.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        try
        {
            //Message displayed indicating user has selected the print option
            JOptionPane.showMessageDialog(frame, "You have selected the
            print button", "Printing", JOptionPane.INFORMATION_MESSAGE);
        }
        catch (Exception exc)
        {
            System.out.println(exc);
        }
    }
});

searchStaff.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        try
        {
            //Redirect back to the main search staff page
            nurse_frame.setVisible(false);
            new SearchStaff();
        }
        catch (Exception exc)
        {
            System.out.println(exc);
        }
    }
});
nurse_frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
nurse_frame.setVisible(true);

```

```

    }

    public static void main(String args[]) throws Exception
    {
        new StaffDetailsBand();
    }
}

```

'Search Patient' main page - Administrator (308):-

```

public class SearchPatient
{
    static String PName, FName, TownD, TreatmentD;

    JOptionPane frame = new JOptionPane();
    JFrame patient_frame = new JFrame ("Bargoed Nurses");
    private JPanel heading;
    private ImagemIcon image1;
    private JLabel label1;
    public static JPanel mainpage = new JPanel();
    JPanel rest = new JPanel(), sidebar = new JPanel();
    Color buttonBackground = new Color (234,234,234);
    Color blueBackground = new Color (7,32,157);
    Color PanelBackground = new Color (167, 165, 255);
    Font f = new Font("Dialog", Font.PLAIN, 26);
    static JTextField patientNameField = new JTextField();
    static JList listtest = new JList();

    public SearchPatient() throws Exception
    {
        heading = new JPanel();
        heading.setMaximumSize(new Dimension(700,150));
        heading.setBackground(blueBackground);
        image1 = new ImagemIcon(getClass().getResource("Bar.jpg"));
        label1 = new JLabel(image1);
        heading.add(label1);

        JButton search = new JButton("Search for Patient details");
        JButton back = new JButton("Back to main screen");
        patient_frame.setSize(780,570);
        patient_frame.setLayout(new FlowLayout());
        patient_frame.setBackground(Color.WHITE);

        mainpage.setPreferredSize(new Dimension(500,400));
    }
}

```

```

mainpage.removeAll();
rest.setPreferredSize(new Dimension(800,600));
rest.setBackground(blueBackground);

patient_frame.add(heading); patient_frame.add(rest);

//Inputting information into sidebar
JPanel TestPanel = new JPanel();
JLabel sidelabel = new JLabel("<html>* Enter known <br> patient details <br>into
textboxes<br> provided ");
JLabel sidelabelblank = new JLabel(" ");
JLabel sidelabelblank2 = new JLabel(" ");
JLabel sidelabelblank3 = new JLabel(" ");
JLabel sidelabel2 = new JLabel("<html>* A first name for a patient needs to be
entered");
JLabel sidelabel3 = new JLabel("<html>* 2 details need <br> to be entered ");
JLabel sidelabel4 = new JLabel("<html>* Once criteria has been entered then select
appropriate button at bottom of the page");
TestPanel.setBackground(PanelBackground);

TestPanel.setPreferredSize(new Dimension(100,400));
TestPanel.setLayout(new BorderLayout(TestPanel, BorderLayout.PAGE_AXIS));
TestPanel.add(sidelabel);
TestPanel.add(sidelabelblank);
TestPanel.add(sidelabel2);
TestPanel.add(sidelabelblank2);
TestPanel.add(sidelabel3);
TestPanel.add(sidelabelblank3);
TestPanel.add(sidelabel4);

rest.add(TestPanel); rest.add(mainpage);

JPanel moduleListPanel = new JPanel();
moduleListPanel.setLayout(new FlowLayout());
moduleListPanel.setPreferredSize(new Dimension(600,400));
JLabel welcome = new JLabel("<html>Enter known patient criteria:");
welcome.setFont(f);
welcome.setPreferredSize(new Dimension(600, 30));
welcome.setHorizontalTextPosition(JLabel.LEFT);

moduleListPanel.add(welcome);
JPanel txtPanel = new JPanel();
txtPanel.setLayout(null);
txtPanel.setLocation(20, 50);
txtPanel.setPreferredSize(new Dimension(200, 250));

```

```
txtPanel.setBackground(Color.WHITE);

moduleListPanel.add(txtPanel);
moduleListPanel.setBackground(Color.WHITE);

JPanel txtFieldPanel = new JPanel();
txtFieldPanel.setLayout(null);
txtFieldPanel.setLocation(150, 60);
txtFieldPanel.setPreferredSize(new Dimension(390, 250));
txtFieldPanel.setBackground(Color.WHITE);
moduleListPanel.add(txtFieldPanel);

JLabel patientName = new JLabel("Patient first name:");
patientName.setLocation(0, 0);
patientName.setSize(300, 50);
patientName.setHorizontalTextPosition(JLabel.LEFT);
txtPanel.add(patientName);

JLabel FamilyName = new JLabel("Family Name:");
FamilyName.setLocation(0,50);
FamilyName.setSize(300,50);
FamilyName.setHorizontalTextPosition(JLabel.LEFT);
txtPanel.add(FamilyName);

JLabel Town = new JLabel("Town:");
Town.setLocation(0,100);
Town.setSize(300,50);
Town.setHorizontalTextPosition(JLabel.LEFT);
txtPanel.add(Town);

JLabel treatment = new JLabel("Treatment:");
treatment.setLocation(0,150);
treatment.setSize(300,50);
treatment.setHorizontalTextPosition(JLabel.LEFT);
txtPanel.add(treatment);

final JTextField patientNameField = new JTextField();
patientNameField.setEditable(true);
patientNameField.setLocation(0, 0);
patientNameField.setSize(390, 50);
patientNameField.setBackground(PanelBackground);
txtFieldPanel.add(patientNameField);

final JTextField familyNameField = new JTextField();
familyNameField.setEditable(true);
```

```
familyNameField.setLocation(0, 50);
familyNameField.setSize(390, 50);
familyNameField.setBackground(PanelBackground);
txtFieldPanel.add(familyNameField);

final JTextField TownField = new JTextField();
TownField.setEditable(true);
TownField.setLocation(0, 100);
TownField.setSize(390, 50);
TownField.setBackground(PanelBackground);
txtFieldPanel.add(TownField);

final JTextArea Treatment = new JTextArea();
Treatment.setSize(390, 50);
Treatment.setEditable(true);
Treatment.setBackground(PanelBackground);
Treatment.setLineWrap(true);

final JScrollPane scrollPane = new JScrollPane(Treatment);
scrollPane.setLocation(0, 150);

scrollPane.setSize(390, 50);
txtFieldPanel.add(scrollPane);

moduleListPanel.add(search);
moduleListPanel.add(back);

mainpage.add(moduleListPanel);
mainpage.setPreferredSize(new Dimension(650,400));
mainpage.setBackground(Color.WHITE);

back.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        try
        {
            //Redirect back to main administrator page
            patient_frame.setVisible(false);
            new MainScreenAdmin();
        }
        catch (Exception exc)
        {
            System.out.println(exc);
        }
    }
}
```

```

    }
});

search.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        try
        {
            PName = patientNameField.getText();
            FName = familyNameField.getText();
            TownD = TownField.getText();
            TreatmentD = Treatment.getText();

            String str = patientNameField.getText();

            Class.forName("org.sqlite.JDBC");
            Connection conn =
            DriverManager.getConnection("jdbc:sqlite:NurseScheduling.sqlite");
            PreparedStatement stat = conn.prepareStatement("select * from
            Patients2 where PName=? and FName=?");

            //SQL Injection Prevention
            stat.setString(1, PName);
            stat.setString(2, FName);

            ResultSet rs = stat.executeQuery();

            Class.forName("org.sqlite.JDBC");
            Connection connnt =
            DriverManager.getConnection("jdbc:sqlite:NurseScheduling.sqlite");
            PreparedStatement statt = conn.prepareStatement("select * from
            Patients2 where PName=? and PTown=?");

            //SQL Injection Prevention
            statt.setString(1, PName);
            statt.setString(2, TownD);
            ResultSet st = statt.executeQuery();

            Class.forName("org.sqlite.JDBC");
            Connection connr =
            DriverManager.getConnection("jdbc:sqlite:NurseScheduling.sqlite");
            PreparedStatement statr = conn.prepareStatement("select * from
            Patients2 where PName=? and Treatment=?");

```

```

//SQL Injection Prevention
statr.setString(1, PName);
statr.setString(2, TreatmentD);
ResultSet sr = statr.executeQuery();

if (rs.next())
{
    JOptionPane.showMessageDialog(frame, "Details have been
found to match the patient first name and family name" ,
"Patient Criteria",JOptionPane.ERROR_MESSAGE);
    new SearchPatientDetailsTest();
    patient_frame.setVisible(false);
}
else if (st.next())
{
    String PatientFound = st.getString("PName");
    JOptionPane.showMessageDialog(frame, "Details have been
found to match the patient first name and town","Patient
Criteria",JOptionPane.ERROR_MESSAGE);
    new SearchPatientDetailsTownTest();
    patient_frame.setVisible(false);
}
else if (sr.next())
{
    JOptionPane.showMessageDialog(frame, "Details have been
found to match the patient first name and
treatment", "Patient Criteria",
JOptionPane.ERROR_MESSAGE);
    new SearchPatientDetailsTreatment();
    patient_frame.setVisible(false);
}
else if (str.equals(""))
{
    JOptionPane.showMessageDialog(frame, "A first name for
the patient needs to be entered", "Unentered
Details",JOptionPane.ERROR_MESSAGE);
}
else
{
    JOptionPane.showMessageDialog(frame, "No details found
for criteria entered", "Unfound Patient",
JOptionPane.ERROR_MESSAGE);
}
}

```

```

        rs.close();
        st.close();
        sr.close();
        conn.close();
        connt.close();
        connr.close();
    }
    catch (Exception exc)
    {
        System.out.println(exc);
    }
}
});
patient_frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
patient_frame.setVisible(true);
}

public static String Get_Patient()
{
    return PName;
}

public static String Get_Family()
{
    return FName;
}

public static String Get_Town()
{
    return TownD;
}

public static String Get_Treatment()
{
    return TreatmentD;
}

public static JList Get_allocation()
{
    return listtest;
}

public static void main(String args[]) throws Exception
{
    new SearchPatient();
}

```

```
}  
}
```

'Search Patient' patient name and family name match - Administrator (290):-

```
public class SearchPatientDetailsTest  
{  
    String PaName, FaName, TownDa, TreatmentDa;  
  
    JOptionPane frame = new JOptionPane();  
    private static JFrame main_frame = new JFrame ("Bargoed Nurses");  
    JPanel main = new JPanel();  
    private JPanel heading, footer;  
    private ImageIcon image1;  
    private JLabel label1, ViewLabel, StaffName;  
    public static JPanel mainpage = new JPanel();  
    JScrollPane vertical = new JScrollPane();  
    JPanel rest = new JPanel(), sidebar = new JPanel();  
    Color buttonBackground = new Color (234,234,234);  
    Color blueBackground = new Color (7,32,157);  
    Font f = new Font("Dialog", Font.PLAIN, 26);  
    Color PanelBackground = new Color (167, 165, 255);  
    Font t = new Font("Dialog", Font.PLAIN, 20);  
    DefaultListModel data;  
    final JList TestList = new JList();  
  
    static String selectedLeftUsername;  
    static String lastone;  
    private static String ending;  
    private static String PaID;  
    private static int index;  
    private static String PaIDTest;  
    private static String fileName;  
  
    public SearchPatientDetailsTest() throws Exception  
    {  
        PaName = SearchPatient.Get_Patient();  
        FaName = SearchPatient.Get_Family();  
  
        heading = new JPanel();  
        heading.setMaximumSize(new Dimension(700,150));  
        heading.setBackground(blueBackground);  
        image1 = new ImageIcon(getClass().getResource("Bar.jpg"));  
        label1 = new JLabel(image1);
```

```

heading.add(label1);

footer = new JPanel();
footer.setMaximumSize(new Dimension(800, 300));
footer.setBackground(Color.GREEN);
JButton Print = new JButton("Print Details");
JButton AnotherPatient = new JButton("Another Patient");
JButton BackMain = new JButton("Back to Main Screen");
footer.add(Print); footer.add(AnotherPatient); footer.add(BackMain);
footer.setBackground(Color.WHITE);

ViewLabel = new JLabel("Here are the patients' details found:-");
ViewLabel.setFont(f);
main_frame.setSize(780,570);
main_frame.setLayout(new FlowLayout());
main_frame.setBackground(Color.WHITE);
main.setLayout(new BorderLayout(main, BorderLayout.PAGE_AXIS));
main.setPreferredSize(new Dimension(600,350));

final JList TestList = new JList(listPatients());
TestList.setSelectionMode(ListSelectionModel.SINGLE_SELECTION);
TestList.setSize(200, 500);
TestList.setSelectedIndex(0);
TestList.setVisibleRowCount(6);

JScrollPane vertical = new JScrollPane(TestList);
vertical.setVerticalScrollBarPolicy(JScrollPane.VERTICAL_SCROLLBAR_ALWAYS);
vertical.getViewPort().setBackground(Color.white);
mainpage.setPreferredSize(new Dimension(500,400));
mainpage.removeAll();
rest.setPreferredSize(new Dimension(800,600));

//Inputting information into sidebar
JPanel TestPanel = new JPanel();
JLabel sidelabel = new JLabel("<html>* You can double click each name to view
more details on each patient");
TestPanel.setBackground(PanelBackground);

TestPanel.setPreferredSize(new Dimension(100,400));
TestPanel.setLayout(new BorderLayout(TestPanel, BorderLayout.PAGE_AXIS));
TestPanel.add(sidelabel);
rest.add(TestPanel);
rest.add(mainpage);
rest.setBackground(blueBackground);

```

```
main_frame.add(heading); main_frame.add(rest);
main.add(ViewLabel);
main.add(vertical);
vertical.setMaximumSize(new Dimension(1200, 250));
mainpage.add(main);
mainpage.add(footer);
mainpage.setPreferredSize(new Dimension(650,400));
mainpage.setBackground(Color.WHITE);
main.setBackground(Color.WHITE);
```

```
MouseListener mouseListener = new MouseAdapter()
{
    public void mouseClicked(MouseEvent e)
    {
        try
        {
            //Double click and item in the list
            //Collect information from the item selected
            //Redirect to page for details on selected item
            if (e.getClickCount() == 2)
            {
                selectedLeftUsername = TestList.getSelectedValue().toString();
                index = TestList.locationToIndex(e.getPoint());
                PaIDTest = PaID + index;
                int test = selectedLeftUsername.indexOf(".");
                fileName = selectedLeftUsername.substring(0, test);

                String[] dot = selectedLeftUsername.split(".");
                lastone = dot[dot.length-1];
                ending = lastone;
                new PatientLastName();
                main_frame.setVisible(false);
            }
            else
            {
            }
        }
        catch (Exception exc)
        {
            System.out.println(exc);
        }
    }
};
```

```

TestList.addMouseListener(mouseListener);

Print.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        try
        {
            //Message displayed indicating user has selected the print option
            JOptionPane.showMessageDialog(frame, "You have selected the
            print button", "Printing", JOptionPane.INFORMATION_MESSAGE);
        }
        catch (Exception exc)
        {
            System.out.println(exc);
        }
    }
});

AnotherPatient.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        try
        {
            //Redirect back to main search patient page
            main_frame.setVisible(false);
            new SearchPatient();
        }
        catch (Exception exc)
        {
            System.out.println(exc);
        }
    }
});

BackMain.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        try
        {
            //Redirect back to main administrator
            main_frame.setVisible(false);
            new MainScreenAdmin();
        }
    }
});

```

```

        }
        catch (Exception exc)
        {
            System.out.println(exc);
        }
    }
});
main_frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
main_frame.setVisible(true);
}

public String[] listPatients() throws Exception
{
    //Get details on patients based on a match from the family name and patient name
    Class.forName("org.sqlite.JDBC");
    Connection conn =
    DriverManager.getConnection("jdbc:sqlite:NurseScheduling.sqlite");
    Statement stat = conn.createStatement();
    ResultSet rs = stat.executeQuery(String.format("select * from Patients2 where
    FName='%s' and PName='%s'", FaName, PaName));

    String[] rawModList = new String[200];
    int index = 0;

    while (rs.next())
    {
        rawModList[index] = rs.getString("PID") + " " + rs.getString("PName") + " " +
        rs.getString("FName");
        FaName = rs.getString("FName");
        PaID = rs.getString("PID");
        index++;
    }

    rs.close();
    conn.close();

    int numElements = 0;
    for (int i = 0; i < 200; i++)
    {
        if (rawModList[i] != null)
        {
            numElements++;
        }
    }
    String[] modList = new String[numElements];

```

```

        System.arraycopy(rawModList, 0, modList, 0, numElements);
        return modList;
    }

    public static String Get_Patient()
    {
        return ending;
    }

    public static String Get_ID()
    {
        return fileName;
    }

    public static JFrame test_frame()
    {
        return main_frame;
    }

    public static void main(String args[]) throws Exception
    {
        new SearchPatientDetailsTest();
    }
}

```

'Search Patient' view details on patient based on selection - Administrator (318):-

```

public class PatientLastName
{
    static String username;
    String PaName, FaName, TownDa, TreatmentDa, PatientTr, lastone, PaID;
    JFrame refreshFrame;
    JOptionPane frame = new JOptionPane();
    JFrame nurse_frame = new JFrame ("Bargoed Nurses");
    private JPanel heading;
    private ImageIcon image1;
    private JLabel label1;
    public static JPanel mainpage = new JPanel();
    JPanel rest = new JPanel(), sidebar = new JPanel();
    Color buttonBackground = new Color (234,234,234);
    Color blueBackground = new Color (7,32,157);
    Color PanelBackground = new Color (167, 165, 255);
    Font f = new Font("Dialog", Font.PLAIN, 20);
}

```

```

public PatientLastName() throws Exception
{
    PaName = SearchPatient.Get_Patient();
    FaName = SearchPatient.Get_Family();
    TownDa = SearchPatient.Get_Town();
    lastone = SearchPatientDetailsTest.Get_Patient();
    PaID = SearchPatientDetailsTest.Get_ID();
    refreshFrame = SearchPatientDetailsTest.test_frame();

    heading = new JPanel();
    heading.setMaximumSize(new Dimension(700,150));
    heading.setBackground(blueBackground);
    image1 = new ImageIcon(getClass().getResource("Bar.jpg"));
    label1 = new JLabel(image1);
    heading.add(label1);

    JButton print = new JButton("Print");
    JButton back = new JButton("Back to list of Patients");
    nurse_frame.setSize(780,725);
    nurse_frame.setLayout(new FlowLayout());
    nurse_frame.setBackground(Color.WHITE);

    mainpage.setPreferredSize(new Dimension(500,700));
    mainpage.removeAll();

    rest.setPreferredSize(new Dimension(800,580));
    rest.setBackground(blueBackground);
    nurse_frame.add(heading); nurse_frame.add(rest);

    JPanel TestPanel = new JPanel();
    TestPanel.setBackground(PanelBackground);
    TestPanel.setPreferredSize(new Dimension(100,560));
    TestPanel.setLayout(new BoxLayout(TestPanel, BoxLayout.PAGE_AXIS));

    rest.add(TestPanel); rest.add(mainpage);
    JPanel moduleListPanel = new JPanel();
    moduleListPanel.setLayout(new FlowLayout());
    moduleListPanel.setPreferredSize(new Dimension(600,550));
    JLabel welcome = new JLabel("<html>Your personal details are:");
    welcome.setPreferredSize(new Dimension(600, 20));
    welcome.setHorizontalAlignment(JLabel.LEFT);
    moduleListPanel.add(welcome);

    //Get patient details based on matched details
    Class.forName("org.sqlite.JDBC");

```

```
Connection conn =
DriverManager.getConnection("jdbc:sqlite:NurseScheduling.sqlite");
Statement stat = conn.createStatement();
ResultSet rs = stat.executeQuery(String.format("select * from Patients2 where
PName='%s' and FName='%s' and PID ='%s'", PaName, lastone, PaID));

JPanel txtPanel = new JPanel();
txtPanel.setLayout(null);
txtPanel.setLocation(20, 50);
txtPanel.setPreferredSize(new Dimension(200, 480));
txtPanel.setBackground(Color.WHITE);

moduleListPanel.add(txtPanel);
moduleListPanel.setBackground(Color.WHITE);

JPanel txtFieldPanel = new JPanel();
txtFieldPanel.setLayout(null);
txtFieldPanel.setLocation(150, 60);
txtFieldPanel.setPreferredSize(new Dimension(390, 480));
txtFieldPanel.setBackground(Color.WHITE);
moduleListPanel.add(txtFieldPanel);

JLabel patientName = new JLabel("Patient Name:");
patientName.setLocation(0, 0);
patientName.setSize(300, 30);
patientName.setHorizontalTextPosition(JLabel.LEFT);
txtPanel.add(patientName);

JLabel familyName = new JLabel("Family Name:");
familyName.setLocation(0,40);
familyName.setSize(300,30);
familyName.setHorizontalTextPosition(JLabel.LEFT);
txtPanel.add(familyName);

JLabel Address = new JLabel("Address:");
Address.setLocation(0,80);
Address.setSize(300,30);
Address.setHorizontalTextPosition(JLabel.LEFT);
txtPanel.add(Address);

JLabel DTreatment = new JLabel("Detailed Treatment:");
DTreatment.setLocation(0,160);
DTreatment.setSize(300,30);
DTreatment.setHorizontalTextPosition(JLabel.LEFT);
txtPanel.add(DTreatment);
```

```
JLabel Town = new JLabel("Town:");  
Town.setLocation(0,120);  
Town.setSize(300,30);  
Town.setHorizontalTextPosition(JLabel.LEFT);  
txtPanel.add(Town);
```

```
JLabel Allergy = new JLabel("Allergy:");  
Allergy.setLocation(0,200);  
Allergy.setSize(300,30);  
Allergy.setHorizontalTextPosition(JLabel.LEFT);  
txtPanel.add(Allergy);
```

```
JLabel ExplainAllergy = new JLabel("ExplainAllergy:");  
ExplainAllergy.setLocation(0,240);  
ExplainAllergy.setSize(300,30);  
ExplainAllergy.setHorizontalTextPosition(JLabel.LEFT);  
txtPanel.add(ExplainAllergy);
```

```
JLabel KeyCode = new JLabel("Key Code:");  
KeyCode.setLocation(0,280);  
KeyCode.setSize(300,30);  
KeyCode.setHorizontalTextPosition(JLabel.LEFT);  
txtPanel.add(KeyCode);
```

```
JLabel Equipment = new JLabel("Equipment:");  
Equipment.setLocation(0,320);  
Equipment.setSize(300,30);  
Equipment.setHorizontalTextPosition(JLabel.LEFT);  
txtPanel.add(Equipment);
```

```
JLabel Telephone = new JLabel("Telephone:");  
Telephone.setLocation(0,360);  
Telephone.setSize(300,30);  
Telephone.setHorizontalTextPosition(JLabel.LEFT);  
txtPanel.add(Telephone);
```

```
JLabel Doctor = new JLabel("Doctor:");  
Doctor.setLocation(0,400);  
Doctor.setSize(300,30);  
Doctor.setHorizontalTextPosition(JLabel.LEFT);  
txtPanel.add(Doctor);
```

```
JLabel Bloods = new JLabel("Bloods:");  
Bloods.setLocation(0,440);
```

```
Bloods.setSize(300,30);
Bloods.setHorizontalTextPosition(JLabel.LEFT);
txtPanel.add(Bloods);

JTextField UsernameField = new JTextField(rs.getString("PName"));
UsernameField.setEditable(false);
UsernameField.setLocation(0, 0);
UsernameField.setSize(390, 30);
UsernameField.setBackground(PanelBackground);
txtFieldPanel.add(UsernameField);

JTextField NurseNameField = new JTextField(rs.getString("FName"));
NurseNameField.setEditable(false);
NurseNameField.setLocation(0, 40);
NurseNameField.setSize(390, 30);
NurseNameField.setBackground(PanelBackground);
txtFieldPanel.add(NurseNameField);

final JTextArea addressArea = new JTextArea(rs.getString("PAddress"));
addressArea.setSize(390, 30);
addressArea.setEditable(false);
addressArea.setBackground(PanelBackground);
addressArea.setLineWrap(true);

final JScrollPane scrollPane = new JScrollPane(addressArea);
scrollPane.setLocation(0, 80);
scrollPane.setSize(390, 30);
txtFieldPanel.add(scrollPane);

JTextField TownField = new JTextField(rs.getString("PTown"));
TownField.setEditable(false);
TownField.setLocation(0, 120);
TownField.setSize(390, 30);
TownField.setBackground(PanelBackground);
txtFieldPanel.add(TownField);

final JTextArea addressArea2 = new JTextArea(rs.getString("DTreatment"));
addressArea2.setSize(390, 30);
addressArea2.setEditable(false);
addressArea2.setBackground(PanelBackground);
addressArea2.setLineWrap(true);

final JScrollPane scrollPane2 = new JScrollPane(addressArea2);
scrollPane2.setLocation(0, 160);
scrollPane2.setSize(390, 30);
```

```
txtFieldPanel.add(scrollPane2);
```

```
JTextField AllergyField = new JTextField(rs.getString("Allergy"));  
AllergyField.setEditable(false);  
AllergyField.setLocation(0, 200);  
AllergyField.setSize(390, 30);  
AllergyField.setBackground(PanelBackground);  
txtFieldPanel.add(AllergyField);
```

```
final JTextArea addressAreaAllergy = new JTextArea(rs.getString("ExplainAllergy"));  
addressAreaAllergy.setSize(390, 30);  
addressAreaAllergy.setEditable(false);  
addressAreaAllergy.setBackground(PanelBackground);  
addressAreaAllergy.setLineWrap(true);
```

```
final JScrollPane scrollPane3 = new JScrollPane(addressAreaAllergy);  
scrollPane3.setLocation(0, 240);  
scrollPane3.setSize(390, 30);  
txtFieldPanel.add(scrollPane3);
```

```
JTextField KeyCodeField = new JTextField(rs.getString("Kcode"));  
KeyCodeField.setEditable(false);  
KeyCodeField.setLocation(0, 280);  
KeyCodeField.setSize(390, 30);  
KeyCodeField.setBackground(PanelBackground);  
txtFieldPanel.add(KeyCodeField);
```

```
JTextField EquipmentField = new JTextField(rs.getString("Equipment"));  
EquipmentField.setEditable(false);  
EquipmentField.setLocation(0, 320);  
EquipmentField.setSize(390, 30);  
EquipmentField.setBackground(PanelBackground);  
txtFieldPanel.add(EquipmentField);
```

```
JTextField TelephoneField = new JTextField(rs.getString("PTelNumber"));  
TelephoneField.setEditable(false);  
TelephoneField.setLocation(0, 360);  
TelephoneField.setSize(390, 30);  
TelephoneField.setBackground(PanelBackground);  
txtFieldPanel.add(TelephoneField);
```

```
JTextField DoctorField = new JTextField(rs.getString("Doctor"));  
DoctorField.setEditable(false);  
DoctorField.setLocation(0, 400);  
DoctorField.setSize(390, 30);
```

```

DoctorField.setBackground(PanelBackground);
txtFieldPanel.add(DoctorField);

JTextField BloodsField = new JTextField(rs.getString("Bloods"));
BloodsField.setEditable(false);
BloodsField.setLocation(0, 440);
BloodsField.setSize(390, 30);
BloodsField.setBackground(PanelBackground);
txtFieldPanel.add(BloodsField);

moduleListPanel.add(print);
moduleListPanel.add(back);

mainpage.add(moduleListPanel);
mainpage.setPreferredSize(new Dimension(650,560));
mainpage.setBackground(Color.WHITE);

back.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        try
        {
            //Redirect back to list of matched patients
            nurse_frame.setVisible(false);
            refreshFrame.setVisible(true);
        }
        catch (Exception exc)
        {
            System.out.println(exc);
        }
    }
});

print.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        try
        {
            //Message displayed indicating user has selected the print option
            JOptionPane.showMessageDialog(frame, "You have selected the
            print button", "Printing", JOptionPane.INFORMATION_MESSAGE);
        }
        catch (Exception exc)

```

```

        {
            System.out.println(exc);
        }
    }
});
nurse_frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
nurse_frame.setVisible(true);
}

public static void main(String args[]) throws Exception
{
    new PatientLastName();
}
}

```

'Search Patient' patient name and town match - Administrator (308):-

```

public class SearchPatientDetailsTownTest
{
    String PaName, FaName, TownDa, TreatmentDa;

    JOptionPane frame = new JOptionPane();
    private static JFrame main_frame = new JFrame ("Bargoed Nurses");
    JPanel main = new JPanel();
    private JPanel heading, footer;
    private ImageIcon image1;
    private JLabel label1, ViewLabel, StaffName;
    public static JPanel mainpage = new JPanel();
    JScrollPane vertical = new JScrollPane();
    JPanel rest = new JPanel(), sidebar = new JPanel();
    Color buttonBackground = new Color (234,234,234);
    Color blueBackground = new Color (7,32,157);
    Font f = new Font("Dialog", Font.PLAIN, 26);
    Color PanelBackground = new Color (167, 165, 255);
    Font t = new Font("Dialog", Font.PLAIN, 20);
    DefaultListModel data;
    final JList TestList = new JList();

    static String selectedLeftUsername;
    static String lastone;
    private static String ending;
    private static String PaID;
    private static String fileName;
}

```

```

public SearchPatientDetailsTownTest() throws Exception
{
    PaName = SearchPatient.Get_Patient();
    FaName = SearchPatient.Get_Family();
    TownDa = SearchPatient.Get_Town();
    TreatmentDa = SearchPatient.Get_Treatment();

    heading = new JPanel();
    heading.setMaximumSize(new Dimension(700,150));
    heading.setBackground(blueBackground);
    image1 = new ImageIcon(getClass().getResource("Bar.jpg"));
    label1 = new JLabel(image1);
    heading.add(label1);

    footer = new JPanel();
    footer.setMaximumSize(new Dimension(800, 300));
    footer.setBackground(Color.GREEN);
    JButton Print = new JButton("Print Details");
    JButton AnotherPatient = new JButton("Another Patient");
    JButton BackMain = new JButton("Back to Main Screen");
    footer.add(Print); footer.add(AnotherPatient); footer.add(BackMain);
    footer.setBackground(Color.WHITE);

    ViewLabel = new JLabel("Here are the patients' details found:-");
    ViewLabel.setFont(f);

    main_frame.setSize(780,570);
    main_frame.setLayout(new FlowLayout());
    main_frame.setBackground(Color.WHITE);

    main.setLayout(new BorderLayout(main, BorderLayout.PAGE_AXIS));
    main.setPreferredSize(new Dimension(600,350));

    final JList TestList = new JList(listPatients());
    TestList.setSelectionMode(ListSelectionModel.SINGLE_SELECTION);
    TestList.setSize(200, 500);
    TestList.setSelectedIndex(0);
    TestList.setVisibleRowCount(6);

    JScrollPane vertical = new JScrollPane(TestList);

    vertical.setVerticalScrollBarPolicy(JScrollPane.VERTICAL_SCROLLBAR_ALWAYS);
    vertical.getViewport().setBackground(Color.white);
    mainpage.setPreferredSize(new Dimension(500,400));
    mainpage.removeAll();
}

```

```

rest.setPreferredSize(new Dimension(800,600));

//Inputting information into sidebar
JPanel TestPanel = new JPanel();
JLabel sidelabel = new JLabel("<html>* You can double click each name to view
more details on each patient");
TestPanel.setBackground(PanelBackground);
TestPanel.setPreferredSize(new Dimension(100,400));
TestPanel.setLayout(new BorderLayout(TestPanel, BorderLayout.PAGE_AXIS));
TestPanel.add(sidelabel);
rest.add(TestPanel);
rest.add(mainpage);
rest.setBackground(blueBackground);

main_frame.add(heading); main_frame.add(rest);

main.add(ViewLabel);
main.add(vertical);
vertical.setMaximumSize(new Dimension(1200, 250));
mainpage.add(main);
mainpage.add(footer);
mainpage.setPreferredSize(new Dimension(650,400));
mainpage.setBackground(Color.WHITE);
main.setBackground(Color.WHITE);

MouseListener mouseListener = new MouseAdapter()
{
    public void mouseClicked(MouseEvent e)
    {
        try
        {
            //Double click and item in the list
            //Collect information from the item selected
            //Redirect to page for details on selected item
            if (e.getClickCount() == 2)
            {
                selectedLeftUsername = TestList.getSelectedValue().toString();
                final int index = TestList.locationToIndex(e.getPoint());
                int test = selectedLeftUsername.indexOf(" ");
                fileName = selectedLeftUsername.substring(0, test);
                String[] dot = selectedLeftUsername.split(" ");
                lastone = dot[dot.length-1];
                ending = lastone;
            }
        }
    }
}

```

```

        new PatientTown();
        main_frame.setVisible(false);
    }
    else
    {
    }
    }
    catch(Exception exc)
    {
        System.out.println(exc);
    }
}
};

TestList.addMouseListener(mouseListener);

Print.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        try
        {
            //Message displayed indicating user has selected the print option
            JOptionPane.showMessageDialog(frame, "You have selected the
            print button", "Printing", JOptionPane.INFORMATION_MESSAGE);
        }
        catch (Exception exc)
        {
            System.out.println(exc);
        }
    }
});

AnotherPatient.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        try
        {
            //Redirect back to the main search patient page
            main_frame.setVisible(false);
            new SearchPatient();
        }
        catch (Exception exc)
        {

```

```

        System.out.println(exc);
    }
}
});

BackMain.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        try
        {
            //Redirect back to the main administrator page
            main_frame.setVisible(false);
            new MainScreenAdmin();
        }
        catch (Exception exc)
        {
            System.out.println(exc);
        }
    }
});
main_frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
main_frame.setVisible(true);
}

public String[] listPatients() throws Exception
{
    //Get details on patients based on a match from the town and patient name
    Class.forName("org.sqlite.JDBC");
    Connection conn =
    DriverManager.getConnection("jdbc:sqlite:NurseScheduling.sqlite");
    Statement stat = conn.createStatement();
    ResultSet rs = stat.executeQuery(String.format("select * from Patients2 where
    PTown='%s' and PName='%s'", TownDa, PaName));

    String[] rawModList = new String[200];
    int index = 0;

    while (rs.next())
    {
        rawModList[index] = rs.getString("PID") + " " + rs.getString("PName") + " " +
        rs.getString("FName");
        PaID = rs.getString("PID");
        index++;
    }
}

```

```

rs.close();
conn.close();

int numElements = 0;
for (int i = 0; i < 200; i++)
{
    if (rawModList[i] != null)
    {
        numElements++;
    }
}
String[] modList = new String[numElements];
System.arraycopy(rawModList, 0, modList, 0, numElements);
return modList;
}

public static String Get_Patient()
{
    return ending;
}

public static String Get_ID()
{
    return fileName;
}

public static JFrame test_frame()
{
    return main_frame;
}

public static void main(String args[]) throws Exception
{
    new SearchPatientDetailsTownTest();
}
}

```

'Search Patient' view details on patient based on selection - Administrator (316):-

```

public class PatientTown
{
    static String username;

    String PaName, FaName, TownDa, TreatmentDa, PatientTr, lastone, PaID;
}

```

```

JFrame refreshFrame;

JOptionPane frame = new JOptionPane();
JFrame nurse_frame = new JFrame ("Bargoed Nurses");
private JPanel heading;
private ImageIcon image1;
private JLabel label1;
public static JPanel mainpage = new JPanel();
JPanel rest = new JPanel(), sidebar = new JPanel();
Color buttonBackground = new Color (234,234,234);
Color blueBackground = new Color (7,32,157);
Color PanelBackground = new Color (167, 165, 255);
Font f = new Font("Dialog", Font.PLAIN, 20);

public PatientTown() throws Exception
{
    PaName = SearchPatient.Get_Patient();
    FaName = SearchPatient.Get_Family();
    TownDa = SearchPatient.Get_Town();
    lastone = SearchPatientDetailsTownTest.Get_Patient();
    PaID = SearchPatientDetailsTownTest.Get_ID();
    refreshFrame = SearchPatientDetailsTownTest.test_frame();

    heading = new JPanel();
    heading.setMaximumSize(new Dimension(700,150));
    heading.setBackground(blueBackground);
    image1 = new ImageIcon(getClass().getResource("Bar.jpg"));
    label1 = new JLabel(image1);
    heading.add(label1);

    JButton print = new JButton("Print");
    JButton back = new JButton("Back to list of Patients");
    nurse_frame.setSize(780,725);
    nurse_frame.setLayout(new FlowLayout());
    nurse_frame.setBackground(Color.WHITE);

    mainpage.setPreferredSize(new Dimension(500,700));
    mainpage.removeAll();

    rest.setPreferredSize(new Dimension(800,580));
    rest.setBackground(blueBackground);

    nurse_frame.add(heading); nurse_frame.add(rest);

    JPanel TestPanel = new JPanel();

```

```

TestPanel.setBackground(PanelBackground);
TestPanel.setPreferredSize(new Dimension(100,560));
TestPanel.setLayout(new BorderLayout(TestPanel, BorderLayout.PAGE_AXIS));
rest.add(TestPanel); rest.add(mainpage);
JPanel moduleListPanel = new JPanel();
moduleListPanel.setLayout(new FlowLayout());
moduleListPanel.setPreferredSize(new Dimension(600,550));
JLabel welcome = new JLabel("<html>Your personal details are:");
welcome.setPreferredSize(new Dimension(600, 20));
welcome.setHorizontalTextPosition(JLabel.LEFT);
moduleListPanel.add(welcome);

//Get details based on the selected item from matches list
Class.forName("org.sqlite.JDBC");
Connection conn =
DriverManager.getConnection("jdbc:sqlite:NurseScheduling.sqlite");
Statement stat = conn.createStatement();
ResultSet rs = stat.executeQuery(String.format("select * from Patients2 where
PName='%s' and FName='%s' and PID='%s'", PaName, lastone, PaID));

JPanel txtPanel = new JPanel();
txtPanel.setLayout(null);
txtPanel.setLocation(20, 50);
txtPanel.setPreferredSize(new Dimension(200, 480));
txtPanel.setBackground(Color.WHITE);

moduleListPanel.add(txtPanel);
moduleListPanel.setBackground(Color.WHITE);

JPanel txtFieldPanel = new JPanel();
txtFieldPanel.setLayout(null);
txtFieldPanel.setLocation(150, 60);
txtFieldPanel.setPreferredSize(new Dimension(390, 480));
txtFieldPanel.setBackground(Color.WHITE);
moduleListPanel.add(txtFieldPanel);

JLabel patientName = new JLabel("Patient Name:");
patientName.setLocation(0, 0);
patientName.setSize(300, 30);
patientName.setHorizontalTextPosition(JLabel.LEFT);
txtPanel.add(patientName);

JLabel familyName = new JLabel("Family Name: ");
familyName.setLocation(0,40);
familyName.setSize(300,30);

```

```
familyName.setHorizontalTextPosition(JLabel.LEFT);  
txtPanel.add(familyName);
```

```
JLabel Address = new JLabel("Address: ");  
Address.setLocation(0,80);  
Address.setSize(300,30);  
Address.setHorizontalTextPosition(JLabel.LEFT);  
txtPanel.add(Address);
```

```
JLabel DTreatment = new JLabel("Detailed Treatment: ");  
DTreatment.setLocation(0,160);  
DTreatment.setSize(300,30);  
DTreatment.setHorizontalTextPosition(JLabel.LEFT);  
txtPanel.add(DTreatment);
```

```
JLabel Town = new JLabel("Town: ");  
Town.setLocation(0,120);  
Town.setSize(300,30);  
Town.setHorizontalTextPosition(JLabel.LEFT);  
txtPanel.add(Town);
```

```
JLabel Allergy = new JLabel("Allergy: ");  
Allergy.setLocation(0,200);  
Allergy.setSize(300,30);  
Allergy.setHorizontalTextPosition(JLabel.LEFT);  
txtPanel.add(Allergy);
```

```
JLabel ExplainAllergy = new JLabel("ExplainAllergy:");  
ExplainAllergy.setLocation(0,240);  
ExplainAllergy.setSize(300,30);  
ExplainAllergy.setHorizontalTextPosition(JLabel.LEFT);  
txtPanel.add(ExplainAllergy);
```

```
JLabel KeyCode = new JLabel("Key Code:");  
KeyCode.setLocation(0,280);  
KeyCode.setSize(300,30);  
KeyCode.setHorizontalTextPosition(JLabel.LEFT);  
txtPanel.add(KeyCode);
```

```
JLabel Equipment = new JLabel("Equipment:");  
Equipment.setLocation(0,320);  
Equipment.setSize(300,30);  
Equipment.setHorizontalTextPosition(JLabel.LEFT);  
txtPanel.add(Equipment);
```

```
JLabel Telephone = new JLabel("Telephone:");
Telephone.setLocation(0,360);
Telephone.setSize(300,30);
Telephone.setHorizontalTextPosition(JLabel.LEFT);
txtPanel.add(Telephone);

JLabel Doctor = new JLabel("Doctor:");
Doctor.setLocation(0,400);
Doctor.setSize(300,30);
Doctor.setHorizontalTextPosition(JLabel.LEFT);
txtPanel.add(Doctor);

JLabel Bloods = new JLabel("Bloods:");
Bloods.setLocation(0,440);
Bloods.setSize(300,30);
Bloods.setHorizontalTextPosition(JLabel.LEFT);
txtPanel.add(Bloods);

JTextField UsernameField = new JTextField(rs.getString("PName"));
UsernameField.setEditable(false);
UsernameField.setLocation(0, 0);
UsernameField.setSize(390, 30);
UsernameField.setBackground(PanelBackground);
txtFieldPanel.add(UsernameField);

JTextField NurseNameField = new JTextField(rs.getString("FName"));
NurseNameField.setEditable(false);
NurseNameField.setLocation(0, 40);
NurseNameField.setSize(390, 30);
NurseNameField.setBackground(PanelBackground);
txtFieldPanel.add(NurseNameField);

final JTextArea addressArea = new JTextArea(rs.getString("PAddress"));
addressArea.setSize(390, 30);
addressArea.setEditable(false);
addressArea.setBackground(PanelBackground);
addressArea.setLineWrap(true);

final JScrollPane scrollPane = new JScrollPane(addressArea);
scrollPane.setLocation(0, 80);
scrollPane.setSize(390, 30);
txtFieldPanel.add(scrollPane);

JTextField TownField = new JTextField(rs.getString("PTown"));
TownField.setEditable(false);
```

```
TownField.setLocation(0, 120);
TownField.setSize(390, 30);
TownField.setBackground(PanelBackground);
txtFieldPanel.add(TownField);

final JTextArea addressArea2 = new JTextArea(rs.getString("DTreatment"));
addressArea2.setSize(390, 30);
addressArea2.setEditable(false);
addressArea2.setBackground(PanelBackground);
addressArea2.setLineWrap(true);

final JScrollPane scrollPane2 = new JScrollPane(addressArea2);
scrollPane2.setLocation(0, 160);
scrollPane2.setSize(390, 30);
txtFieldPanel.add(scrollPane2);

JTextField AllergyField = new JTextField(rs.getString("Allergy"));
AllergyField.setEditable(false);
AllergyField.setLocation(0, 200);
AllergyField.setSize(390, 30);
AllergyField.setBackground(PanelBackground);
txtFieldPanel.add(AllergyField);

final JTextArea addressAreaAllergy = new JTextArea(rs.getString("ExplainAllergy"));
addressAreaAllergy.setSize(390, 30);
addressAreaAllergy.setEditable(false);
addressAreaAllergy.setBackground(PanelBackground);
addressAreaAllergy.setLineWrap(true);

final JScrollPane scrollPane3 = new JScrollPane(addressAreaAllergy);
scrollPane3.setLocation(0, 240);
scrollPane3.setSize(390, 30);
txtFieldPanel.add(scrollPane3);

JTextField KeyCodeField = new JTextField(rs.getString("Kcode"));
KeyCodeField.setEditable(false);
KeyCodeField.setLocation(0, 280);
KeyCodeField.setSize(390, 30);
KeyCodeField.setBackground(PanelBackground);
txtFieldPanel.add(KeyCodeField);

JTextField EquipmentField = new JTextField(rs.getString("Equipment"));
EquipmentField.setEditable(false);
EquipmentField.setLocation(0, 320);
EquipmentField.setSize(390, 30);
```

```

EquipmentField.setBackground(PanelBackground);
txtFieldPanel.add(EquipmentField);

JTextField TelephoneField = new JTextField(rs.getString("PTelNumber"));
TelephoneField.setEditable(false);
TelephoneField.setLocation(0, 360);
TelephoneField.setSize(390, 30);
TelephoneField.setBackground(PanelBackground);
txtFieldPanel.add(TelephoneField);

JTextField DoctorField = new JTextField(rs.getString("Doctor"));
DoctorField.setEditable(false);
DoctorField.setLocation(0, 400);
DoctorField.setSize(390, 30);
DoctorField.setBackground(PanelBackground);
txtFieldPanel.add(DoctorField);

JTextField BloodsField = new JTextField(rs.getString("Bloods"));
BloodsField.setEditable(false);
BloodsField.setLocation(0, 440);
BloodsField.setSize(390, 30);
BloodsField.setBackground(PanelBackground);
txtFieldPanel.add(BloodsField);

moduleListPanel.add(print);
moduleListPanel.add(back);

mainpage.add(moduleListPanel);
mainpage.setPreferredSize(new Dimension(650,560));
mainpage.setBackground(Color.WHITE);

back.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        try
        {
            //Redirect back to list of matches
            nurse_frame.setVisible(false);
            refreshFrame.setVisible(true);
        }
        catch (Exception exc)
        {
            System.out.println(exc);
        }
    }
}

```

```

    }
    });

    print.addActionListener(new ActionListener()
    {
        public void actionPerformed(ActionEvent e)
        {
            try
            {
                //Message indicating the user has selected the print option
                JOptionPane.showMessageDialog(frame, "You have selected the
                print button", "Printing", JOptionPane.INFORMATION_MESSAGE);
            }
            catch (Exception exc)
            {
                System.out.println(exc);
            }
        }
    });
    nurse_frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    nurse_frame.setVisible(true);
}

public static void main(String args[]) throws Exception
{
    new PatientTown();
}
}

```

'Search Patient' patient name and treatment match - Administrator (284):-

```

public class SearchPatientDetailsTreatment
{
    String PaName, FaName, TownDa, TreatmentDa;
    JOptionPane frame = new JOptionPane();
    private static JFrame main_frame = new JFrame ("Bargoed Nurses");
    JPanel main = new JPanel();
    private JPanel heading, footer;
    private ImageIcon image1;
    private JLabel label1, ViewLabel, StaffName;
    public static JPanel mainpage = new JPanel();
    JScrollPane vertical = new JScrollPane();
    JPanel rest = new JPanel(), sidebar = new JPanel();
    Color buttonBackground = new Color (234,234,234);
    Color blueBackground = new Color (7,32,157);
}

```

```

Font f = new Font("Dialog", Font.PLAIN, 26);
Color PanelBackground = new Color (167, 165, 255);
DefaultListModel data;
final JList TestList = new JList();

static String selectedLeftUsername;
static String lastone;
private static String ending;
private static String PaID;
private static String fileName;

public SearchPatientDetailsTreatment() throws Exception
{
    PaName = SearchPatient.Get_Patient();
    FaName = SearchPatient.Get_Family();
    TownDa = SearchPatient.Get_Town();
    TreatmentDa = SearchPatient.Get_Treatment();

    heading = new JPanel();
    heading.setMaximumSize(new Dimension(700,150));
    heading.setBackground(blueBackground);
    image1 = new ImageIcon(getClass().getResource("Bar.jpg"));
    label1 = new JLabel(image1);
    heading.add(label1);

    footer = new JPanel();
    footer.setMaximumSize(new Dimension(800, 300));
    footer.setBackground(Color.GREEN);
    JButton Print = new JButton("Print Details");
    JButton AnotherPatient = new JButton("Another Patient");
    JButton BackMain = new JButton("Back to Main Screen");
    footer.add(Print); footer.add(AnotherPatient); footer.add(BackMain);
    footer.setBackground(Color.WHITE);

    ViewLabel = new JLabel("Here are the patients' details found:-");
    ViewLabel.setFont(f);

    main_frame.setSize(780,570);
    main_frame.setLayout(new FlowLayout());
    main_frame.setBackground(Color.WHITE);
    main.setLayout(new BorderLayout(main, BorderLayout.PAGE_AXIS));
    main.setPreferredSize(new Dimension(600,350));

    final JList TestList = new JList(listPatients());
    TestList.setSelectionMode(ListSelectionModel.SINGLE_SELECTION);
}

```

```

TestList.setSize(200, 500);
TestList.setSelectedIndex(0);
TestList.setVisibleRowCount(6);

JScrollPane vertical = new JScrollPane(TestList);

vertical.setVerticalScrollBarPolicy(JScrollPane.VERTICAL_SCROLLBAR_ALWAYS);
vertical.getViewport().setBackground(Color.white);
mainpage.setPreferredSize(new Dimension(500,400));
mainpage.removeAll();
rest.setPreferredSize(new Dimension(800,600));

//Inputting information into sidebar
JPanel TestPanel = new JPanel();
JLabel sidelabel = new JLabel("<html>* You can double click each name to view
more details on each patient");
TestPanel.setBackground(PanelBackground);
TestPanel.setPreferredSize(new Dimension(100,400));
TestPanel.setLayout(new BorderLayout(TestPanel, BorderLayout.PAGE_AXIS));
TestPanel.add(sidelabel);
rest.add(TestPanel);
rest.add(mainpage);
rest.setBackground(blueBackground);

main_frame.add(heading); main_frame.add(rest);
main.add(ViewLabel);
main.add(vertical);
vertical.setMaximumSize(new Dimension(1200, 250));
mainpage.add(main);
mainpage.add(footer);
mainpage.setPreferredSize(new Dimension(650,400));
mainpage.setBackground(Color.WHITE);
main.setBackground(Color.WHITE);

MouseListener mouseListener = new MouseAdapter()
{
    public void mouseClicked(MouseEvent e)
    {
        try
        {
            //Double click and item in the list
            //Collect information from the item selected
            //Redirect to page for details on selected item
            if (e.getClickCount() == 2)
            {

```

```

        selectedLeftUsername = TestList.getSelectedValue().toString();
        final int index = TestList.locationToIndex(e.getPoint());
        int test = selectedLeftUsername.indexOf(".");
        fileName = selectedLeftUsername.substring(0, test);
        String[] dot = selectedLeftUsername.split(".");
        lastone = dot[dot.length-1];
        ending = lastone;
        new PatientTreatment();
        main_frame.setVisible(false);
    }
    else
    {
    }
    }
    catch (Exception exc)
    {
        System.out.println(exc);
    }
}
};

TestList.addMouseListener(mouseListener);

Print.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        try
        {
            //Message displayed indicating the user has selected the print option
            JOptionPane.showMessageDialog(frame, "You have selected the
            print button", "Printing", JOptionPane.INFORMATION_MESSAGE);
        }
        catch (Exception exc)
        {
            System.out.println(exc);
        }
    }
});

AnotherPatient.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        try

```

```

        {
            //Redirect back to the main search patient page
            main_frame.setVisible(false);
            new SearchPatient();
        }
        catch (Exception exc)
        {
            System.out.println(exc);
        }
    }
});

BackMain.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        try
        {
            //Redirect back to the main administrator page
            main_frame.setVisible(false);
            new MainScreenAdmin();
        }
        catch (Exception exc)
        {
            System.out.println(exc);
        }
    }
});

main_frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
main_frame.setVisible(true);
}

public String[] listPatients() throws Exception
{
    //Get details on patients based on a match from the treatment and patient name
    Class.forName("org.sqlite.JDBC");
    Connection conn =
    DriverManager.getConnection("jdbc:sqlite:NurseScheduling.sqlite");
    Statement stat = conn.createStatement();
    ResultSet rs = stat.executeQuery(String.format("select * from Patients2 where
    Treatment='%s' and PName='%s'", TreatmentDa, PaName));

    String[] rawModList = new String[200];
    int index = 0;
    while (rs.next())

```

```

        {
            rawModList[index] = rs.getString("PID") + " " + rs.getString("PName") + " " +
            rs.getString("FName");
            FaName = rs.getString("FName");
            index++;
        }
        rs.close();
        conn.close();

        int numElements = 0;
        for (int i = 0; i < 200; i++)
        {
            if (rawModList[i] != null)
            {
                numElements++;
            }
        }
        String[] modList = new String[numElements];
        System.arraycopy(rawModList, 0, modList, 0, numElements);
        return modList;
    }

    public static String Get_Patient()
    {
        return ending;
    }

    public static String Get_ID()
    {
        return fileName;
    }

    public static JFrame test_frame()
    {
        return main_frame;
    }

    public static void main(String args[]) throws Exception
    {
        new SearchPatientDetailsTreatment();
    }
}

```

'Search Patient' view details on patient based on selection - Administrator (319):-

```

public class PatientTreatment
{
    static String username;

    String PaName, FaName, TownDa, TreatmentDa, PatientTr, lastone, PaID;
    JFrame testFrame;

    JOptionPane frame = new JOptionPane();
    JFrame nurse_frame = new JFrame ("Bargoed Nurses");
    private JPanel heading;
    private ImageIcon image1;
    private JLabel label1;
    public static JPanel mainpage = new JPanel();
    JPanel rest = new JPanel(), sidebar = new JPanel();
    Color buttonBackground = new Color (234,234,234);
    Color blueBackground = new Color (7,32,157);
    Color PanelBackground = new Color (167, 165, 255);
    Font f = new Font("Dialog", Font.PLAIN, 20);

    public PatientTreatment() throws Exception
    {
        PaName = SearchPatient.Get_Patient();
        FaName = SearchPatient.Get_Family();
        TownDa = SearchPatient.Get_Town();
        TreatmentDa = SearchPatient.Get_Treatment();
        PatientTr = SearchPatientDetailsTreatment.Get_Patient();
        lastone = SearchPatientDetailsTreatment.Get_Patient();
        PaID = SearchPatientDetailsTreatment.Get_ID();
        testFrame = SearchPatientDetailsTreatment.test_frame();

        heading = new JPanel();
        heading.setMaximumSize(new Dimension(700,150));
        heading.setBackground(blueBackground);
        image1 = new ImageIcon(getClass().getResource("Bar.jpg"));
        label1 = new JLabel(image1);
        heading.add(label1);

        JButton print = new JButton("Print");
        JButton back = new JButton("Back to list of Patients");
        nurse_frame.setSize(780,725);
        nurse_frame.setLayout(new FlowLayout());
        nurse_frame.setBackground(Color.WHITE);

        mainpage.setPreferredSize(new Dimension(500,700));
        mainpage.removeAll();
    }
}

```

```

rest.setPreferredSize(new Dimension(800,580));
rest.setBackground(blueBackground);
nurse_frame.add(heading); nurse_frame.add(rest);

JPanel TestPanel = new JPanel();
TestPanel.setBackground(PanelBackground);
TestPanel.setPreferredSize(new Dimension(100,560));
TestPanel.setLayout(new BorderLayout(TestPanel, BorderLayout.PAGE_AXIS));
rest.add(TestPanel); rest.add(mainpage);

JPanel moduleListPanel = new JPanel();
moduleListPanel.setLayout(new FlowLayout());
moduleListPanel.setPreferredSize(new Dimension(600,550));
JLabel welcome = new JLabel("<html>Your personal details are:");
welcome.setPreferredSize(new Dimension(600, 20));
welcome.setHorizontalTextPosition(JLabel.LEFT);
moduleListPanel.add(welcome);

//Get details based on the selected item from the list of matches
Class.forName("org.sqlite.JDBC");
Connection conn =
DriverManager.getConnection("jdbc:sqlite:NurseScheduling.sqlite");
Statement stat = conn.createStatement();
ResultSet rs = stat.executeQuery(String.format("select * from Patients2 where
PName='%s' and FName='%s' and PID='%s'", PaName, lastone, PaID));

JPanel txtPanel = new JPanel();
txtPanel.setLayout(null);
txtPanel.setLocation(20, 50);
txtPanel.setPreferredSize(new Dimension(200, 480));
txtPanel.setBackground(Color.WHITE);

moduleListPanel.add(txtPanel);
moduleListPanel.setBackground(Color.WHITE);

JPanel txtFieldPanel = new JPanel();
txtFieldPanel.setLayout(null);
txtFieldPanel.setLocation(150, 60);
txtFieldPanel.setPreferredSize(new Dimension(390, 480));
txtFieldPanel.setBackground(Color.WHITE);
moduleListPanel.add(txtFieldPanel);

JLabel patientName = new JLabel("Patient Name:");
patientName.setLocation(0, 0);

```

```
patientName.setSize(300, 30);
patientName.setHorizontalTextPosition(JLabel.LEFT);
txtPanel.add(patientName);
```

```
JLabel familyName = new JLabel("Family Name: ");
familyName.setLocation(0,40);
familyName.setSize(300,30);
familyName.setHorizontalTextPosition(JLabel.LEFT);
txtPanel.add(familyName);
```

```
JLabel Address = new JLabel("Address: ");
Address.setLocation(0,80);
Address.setSize(300,30);
Address.setHorizontalTextPosition(JLabel.LEFT);
txtPanel.add(Address);
```

```
JLabel DTreatment = new JLabel("Detailed Treatment: ");
DTreatment.setLocation(0,160);
DTreatment.setSize(300,30);
DTreatment.setHorizontalTextPosition(JLabel.LEFT);
txtPanel.add(DTreatment);
```

```
JLabel Town = new JLabel("Town: ");
Town.setLocation(0,120);
Town.setSize(300,30);
Town.setHorizontalTextPosition(JLabel.LEFT);
txtPanel.add(Town);
```

```
JLabel Allergy = new JLabel("Allergy: ");
Allergy.setLocation(0,200);
Allergy.setSize(300,30);
Allergy.setHorizontalTextPosition(JLabel.LEFT);
txtPanel.add(Allergy);
```

```
JLabel ExplainAllergy = new JLabel("ExplainAllergy:");
ExplainAllergy.setLocation(0,240);
ExplainAllergy.setSize(300,30);
ExplainAllergy.setHorizontalTextPosition(JLabel.LEFT);
txtPanel.add(ExplainAllergy);
```

```
JLabel KeyCode = new JLabel("Key Code:");
KeyCode.setLocation(0,280);
KeyCode.setSize(300,30);
KeyCode.setHorizontalTextPosition(JLabel.LEFT);
txtPanel.add(KeyCode);
```

```
JLabel Equipment = new JLabel("Equipment:");
Equipment.setLocation(0,320);
Equipment.setSize(300,30);
Equipment.setHorizontalTextPosition(JLabel.LEFT);
txtPanel.add(Equipment);
```

```
JLabel Telephone = new JLabel("Telephone:");
Telephone.setLocation(0,360);
Telephone.setSize(300,30);
Telephone.setHorizontalTextPosition(JLabel.LEFT);
txtPanel.add(Telephone);
```

```
JLabel Doctor = new JLabel("Doctor:");
Doctor.setLocation(0,400);
Doctor.setSize(300,30);
Doctor.setHorizontalTextPosition(JLabel.LEFT);
txtPanel.add(Doctor);
```

```
JLabel Bloods = new JLabel("Bloods:");
Bloods.setLocation(0,440);
Bloods.setSize(300,30);
Bloods.setHorizontalTextPosition(JLabel.LEFT);
txtPanel.add(Bloods);
```

```
JTextField UsernameField = new JTextField(rs.getString("PName"));
UsernameField.setEditable(false);
UsernameField.setLocation(0, 0);
UsernameField.setSize(390, 30);
UsernameField.setBackground(PanelBackground);
txtFieldPanel.add(UsernameField);
```

```
JTextField NurseNameField = new JTextField(rs.getString("FName"));
NurseNameField.setEditable(false);
NurseNameField.setLocation(0, 40);
NurseNameField.setSize(390, 30);
NurseNameField.setBackground(PanelBackground);
txtFieldPanel.add(NurseNameField);
```

```
final JTextArea addressArea = new JTextArea(rs.getString("PAddress"));
addressArea.setSize(390, 30);
addressArea.setEditable(false);
addressArea.setBackground(PanelBackground);
addressArea.setLineWrap(true);
```

```
final JScrollPane scrollPane = new JScrollPane(addressArea);
scrollPane.setLocation(0, 80);
scrollPane.setSize(390, 30);
txtFieldPanel.add(scrollPane);

JTextField TownField = new JTextField(rs.getString("PTown"));
TownField.setEditable(false);
TownField.setLocation(0, 120);
TownField.setSize(390, 30);
TownField.setBackground(PanelBackground);
txtFieldPanel.add(TownField);

final JTextArea addressArea2 = new JTextArea(rs.getString("DTreatment"));
addressArea2.setSize(390, 30);
addressArea2.setEditable(false);
addressArea2.setBackground(PanelBackground);
addressArea2.setLineWrap(true);

final JScrollPane scrollPane2 = new JScrollPane(addressArea2);
scrollPane2.setLocation(0, 160);
scrollPane2.setSize(390, 30);
txtFieldPanel.add(scrollPane2);

JTextField AllergyField = new JTextField(rs.getString("Allergy"));
AllergyField.setEditable(false);
AllergyField.setLocation(0, 200);
AllergyField.setSize(390, 30);
AllergyField.setBackground(PanelBackground);
txtFieldPanel.add(AllergyField);

final JTextArea addressAreaAllergy = new JTextArea(rs.getString("ExplainAllergy"));
addressAreaAllergy.setSize(390, 30);
addressAreaAllergy.setEditable(false);
addressAreaAllergy.setBackground(PanelBackground);
addressAreaAllergy.setLineWrap(true);

final JScrollPane scrollPane3 = new JScrollPane(addressAreaAllergy);
scrollPane3.setLocation(0, 240);
scrollPane3.setSize(390, 30);
txtFieldPanel.add(scrollPane3);

JTextField KeyCodeField = new JTextField(rs.getString("Kcode"));
KeyCodeField.setEditable(false);
KeyCodeField.setLocation(0, 280);
```

```
KeyCodeField.setSize(390, 30);
KeyCodeField.setBackground(PanelBackground);
txtFieldPanel.add(KeyCodeField);

JTextField EquipmentField = new JTextField(rs.getString("Equipment"));
EquipmentField.setEditable(false);
EquipmentField.setLocation(0, 320);
EquipmentField.setSize(390, 30);
EquipmentField.setBackground(PanelBackground);
txtFieldPanel.add(EquipmentField);

JTextField TelephoneField = new JTextField(rs.getString("PTelNumber"));
TelephoneField.setEditable(false);
TelephoneField.setLocation(0, 360);
TelephoneField.setSize(390, 30);
TelephoneField.setBackground(PanelBackground);
txtFieldPanel.add(TelephoneField);

JTextField DoctorField = new JTextField(rs.getString("Doctor"));
DoctorField.setEditable(false);
DoctorField.setLocation(0, 400);
DoctorField.setSize(390, 30);
DoctorField.setBackground(PanelBackground);
txtFieldPanel.add(DoctorField);

JTextField BloodsField = new JTextField(rs.getString("Bloods"));
BloodsField.setEditable(false);
BloodsField.setLocation(0, 440);
BloodsField.setSize(390, 30);
BloodsField.setBackground(PanelBackground);
txtFieldPanel.add(BloodsField);

moduleListPanel.add(print);
moduleListPanel.add(back);

mainpage.add(moduleListPanel);
mainpage.setPreferredSize(new Dimension(650,560));
mainpage.setBackground(Color.WHITE);

back.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        try
        {
```

```

        //Redirect beck to list of matches
        nurse_frame.setVisible(false);
        testFrame.setVisible(true);
    }
    catch (Exception exc)
    {
        System.out.println(exc);
    }
}
});

print.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        try
        {
            //Message indicating the user has selected the print option
            JOptionPane.showMessageDialog(frame, "You have selected the
            print button", "Printing", JOptionPane.INFORMATION_MESSAGE);
        }
        catch (Exception exc)
        {
            System.out.println(exc);
        }
    }
});
nurse_frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
nurse_frame.setVisible(true);
}

public static void main(String args[]) throws Exception
{
    new PatientTreatment();
}
}

```

'View calls' view list of working staff - Administrator (275):-

```

public class SearchStaffWorking
{
    //String PaName, FaName, TownDa, TreatmentDa;
    JFrame refresh;

    JOptionPane frame = new JOptionPane();
}

```

```

private static JFrame main_frame = new JFrame ("Bargoed Nurses");
JPanel main = new JPanel();
private JPanel heading, footer;
private ImageIcon image1;
private JLabel label1, ViewLabel, StaffName;
public static JPanel mainpage = new JPanel();
JScrollPane vertical = new JScrollPane();
JPanel rest = new JPanel(), sidebar = new JPanel();
Color buttonBackground = new Color (234,234,234);
Color blueBackground = new Color (7,32,157);
Font f = new Font("Dialog", Font.PLAIN, 36);
Color PanelBackground = new Color (167, 165, 255);
Font t = new Font("Dialog", Font.PLAIN, 20);
DefaultListModel data;
final JList TestList = new JList();

static String selectedStaff;
static String lastone;
private static String staffSelected;
private static String PaID;
private static int index;
private static String PaIDTest;
private static String fileName;

public SearchStaffWorking() throws Exception
{
    refresh = SearchSelectedStaff.test_frame();
    heading = new JPanel();
    heading.setMaximumSize(new Dimension(700,150));
    heading.setBackground(blueBackground);
    image1 = new ImageIcon(getClass().getResource("Bar.jpg"));
    label1 = new JLabel(image1);
    heading.add(label1);

    footer = new JPanel();
    footer.setMaximumSize(new Dimension(800, 300));
    footer.setBackground(Color.GREEN);
    JButton Print = new JButton("Print Details");
    JButton BackMain = new JButton("Back to Main Screen");
    footer.add(Print); footer.add(BackMain);
    footer.setBackground(Color.WHITE);

    ViewLabel = new JLabel("Here are staff that are working:-");
    ViewLabel.setFont(t);

```

```

main_frame.setSize(780,570);
main_frame.setLayout(new FlowLayout());
main_frame.setBackground(Color.WHITE);

main.setLayout(new BorderLayout(main, BorderLayout.PAGE_AXIS));
main.setPreferredSize(new Dimension(600,350));

final JList TestList = new JList(listWorkingStaff());
TestList.setSelectionMode(ListSelectionModel.SINGLE_SELECTION);
TestList.setSize(200, 500);
TestList.setSelectedIndex(0);
TestList.setVisibleRowCount(6);

JScrollPane vertical = new JScrollPane(TestList);
vertical.setVerticalScrollBarPolicy(JScrollPane.VERTICAL_SCROLLBAR_ALWAYS);
vertical.getViewport().setBackground(Color.white);
mainpage.setPreferredSize(new Dimension(500,400));
mainpage.removeAll();

rest.setPreferredSize(new Dimension(800,600));

//Inputting information into sidebar
JPanel TestPanel = new JPanel();
JLabel sidelabel = new JLabel("<html>* Here is the list of staff working");
JLabel sidelabelblank = new JLabel(" ");
JLabel sidelabelblank2 = new JLabel(" ");
JLabel sidelabel2 = new JLabel("<html>* You can double click each member of staff
in order to review each call list");
TestPanel.setBackground(PanelBackground);
TestPanel.setPreferredSize(new Dimension(100,400));
TestPanel.setLayout(new BorderLayout(TestPanel, BorderLayout.PAGE_AXIS));
TestPanel.add(sidelabel);
TestPanel.add(sidelabelblank);
TestPanel.add(sidelabel2);
TestPanel.add(sidelabelblank2);
rest.add(TestPanel); rest.add(mainpage);
rest.setBackground(blueBackground);

main_frame.add(heading); main_frame.add(rest);
main.add(ViewLabel);
main.add(vertical);
vertical.setMaximumSize(new Dimension(1200, 250));
mainpage.add(main);
mainpage.add(footer);
mainpage.setPreferredSize(new Dimension(650,400));

```

```
mainpage.setBackground(Color.WHITE);
```

```
main.setBackground(Color.WHITE);
```

```
MouseListener mouseListener = new MouseAdapter()
```

```
{
```

```
    public void mouseClicked(MouseEvent e)
```

```
    {
```

```
        try
```

```
        {
```

```
            //User selects an item (member of staff) from the list by clicking it twice
```

```
            //Then information is gathered from the item selected
```

```
            //Then user is directed to view the individual call list for the selected  
            member of staff
```

```
                if (e.getClickCount() == 2)
```

```
                {
```

```
                    selectedStaff = TestList.getSelectedValue().toString();
```

```
                    String[] space = selectedStaff.split(" ");
```

```
                    staffSelected = space[1] + " " + space[2];
```

```
                    main_frame.setVisible(false);
```

```
                    new SearchSelectedStaff();
```

```
                }
```

```
                else
```

```
                {
```

```
                }
```

```
                }
```

```
                catch(Exception exc)
```

```
                {
```

```
                    System.out.println(exc);
```

```
                }
```

```
            }
```

```
};
```

```
TestList.addMouseListener(mouseListener);
```

```
Print.addActionListener(new ActionListener()
```

```
{
```

```
    public void actionPerformed(ActionEvent e)
```

```
    {
```

```
        try
```

```
        {
```

```
            //Message displayed indicating the user has selected the  
            print option
```

```
            JOptionPane.showMessageDialog(frame, "You have selected  
            the print button", "Printing",
```

```
            JOptionPane.INFORMATION_MESSAGE);
```

```

        }
        catch (Exception exc)
        {
            System.out.println(exc);
        }
    }
});

BackMain.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        try
        {
            //Redirect back to the main administrator page
            main_frame.setVisible(false);
            new MainScreenAdmin();
        }
        catch (Exception exc)
        {
            System.out.println(exc);
        }
    }
});

//If there are no calls stored in the database then give a message to the user to
notify them
if (index == 0)
{
    JOptionPane.showMessageDialog(frame, "There are no calls allocated", "No
calls found", JOptionPane.ERROR_MESSAGE);
    main_frame.setVisible(false);
    new MainScreenAdmin();
}
else
{
    main_frame.setVisible(true);
}

main_frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
}

public String[] listWorkingStaff() throws Exception
{

```

```

//Get the list of staff that are working from the database
Class.forName("org.sqlite.JDBC");
Connection conn =
DriverManager.getConnection("jdbc:sqlite:NurseScheduling.sqlite");
Statement stat = conn.createStatement();
ResultSet rs = stat.executeQuery(String.format("select * from CallLists2"));

String[] rawModList = new String[200];
index = 0;
while (rs.next())
{
    rawModList[index] = rs.getString("Date") + " " + rs.getString("NurseName");
    index++;
}
rs.close();
conn.close();
int numElements = 0;
for (int i = 0; i < 200; i++)
{
    if (rawModList[i] != null)
    {
        numElements++;
    }
}

String[] modList = new String[numElements];
System.arraycopy(rawModList, 0, modList, 0, numElements);
return modList;
}

public static JFrame test_frame()
{
    return main_frame;
}

public static String Get_selectedStaff()
{
    return staffSelected;
}

public static void main(String args[]) throws Exception
{
    new SearchStaffWorking();
}

```

```
}
```

'View calls' view individual list for staff member - Administrator (258):-

```
public class SearchSelectedStaff
{
    String selectedStaff;
    JFrame refreshFrame;
    JOptionPane frame = new JOptionPane();
    private static JFrame main_frame = new JFrame ("Bargood Nurses");
    JPanel main = new JPanel();
    private JPanel heading, footer;
    private ImageIcon image1;
    private JLabel label1, ViewLabel, StaffName;
    public static JPanel mainpage = new JPanel();
    JScrollPane vertical = new JScrollPane();
    JPanel rest = new JPanel(), sidebar = new JPanel();
    Color buttonBackground = new Color (234,234,234);
    Color blueBackground = new Color (7,32,157);
    Font f = new Font("Dialog", Font.PLAIN, 26);
    Color PanelBackground = new Color (167, 165, 255);
    Font t = new Font("Dialog", Font.PLAIN, 20);
    DefaultListModel data;
    final JList TestList = new JList();

    static String lastone;
    private static String staffSelected;
    private static String PaID;
    private static int index;
    private static String PaIDTest;
    private static String fileName;

    public SearchSelectedStaff() throws Exception
    {
        refreshFrame = SearchStaffWorking.test_frame();
        selectedStaff = SearchStaffWorking.Get_selectedStaff();

        heading = new JPanel();
        heading.setMaximumSize(new Dimension(700,150));
        heading.setBackground(blueBackground);
        image1 = new ImageIcon(getClass().getResource("Bar.jpg"));
        label1 = new JLabel(image1);
        heading.add(label1);

        footer = new JPanel();
        footer.setMaximumSize(new Dimension(800, 300));
```

```
footer.setBackground(Color.GREEN);
JButton Print = new JButton("Print Details");
JButton StaffList = new JButton("back to staff working list");
JButton BackMain = new JButton("Back to Main Screen");
footer.add(Print); footer.add(StaffList); footer.add(BackMain);
footer.setBackground(Color.WHITE);

ViewLabel = new JLabel("Here are the lists of calls for" + selectedStaff + ":-");
ViewLabel.setFont(f);

main_frame.setSize(780,570);
main_frame.setLayout(new FlowLayout());
main_frame.setBackground(Color.WHITE);

main.setLayout(new BorderLayout(main, BorderLayout.PAGE_AXIS));
main.setPreferredSize(new Dimension(600,350));

final JList TestList = new JList(listWorkingStaff());
TestList.setSelectionMode(ListSelectionModel.SINGLE_SELECTION);
TestList.setSize(200, 500);
TestList.setSelectedIndex(0);
TestList.setVisibleRowCount(6);

JScrollPane vertical = new JScrollPane(TestList);
vertical.setVerticalScrollBarPolicy(JScrollPane.VERTICAL_SCROLLBAR_ALWAYS);
vertical.getViewport().setBackground(Color.white);
mainpage.setPreferredSize(new Dimension(500,400));
mainpage.removeAll();
rest.setPreferredSize(new Dimension(800,600));

JPanel TestPanel = new JPanel();
TestPanel.setBackground(PanelBackground);
TestPanel.setPreferredSize(new Dimension(100,400));
TestPanel.setLayout(new BorderLayout(TestPanel, BorderLayout.PAGE_AXIS));
rest.add(TestPanel);
rest.add(mainpage);
rest.setBackground(blueBackground);

main_frame.add(heading); main_frame.add(rest);
main.add(ViewLabel);
main.add(vertical);
vertical.setMaximumSize(new Dimension(1200, 250));
mainpage.add(main);
mainpage.add(footer);
mainpage.setPreferredSize(new Dimension(650,400));
```

```

mainpage.setBackground(Color.WHITE);
main.setBackground(Color.WHITE);

MouseListener mouseListener = new MouseAdapter()
{
    public void mouseClicked(MouseEvent e)
    {
        try
        {
        }
        catch (Exception exc)
        {
            System.out.println(exc);
        }
    }
};

TestList.addMouseListener(mouseListener);

Print.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        try
        {
            //Message indicating the user has selected the print option
            JOptionPane.showMessageDialog(frame, "You have selected
            the print button", "Printing",
            JOptionPane.INFORMATION_MESSAGE);
        }
        catch (Exception exc)
        {
            System.out.println(exc);
        }
    }
});

StaffList.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        try
        {
            //Redirect back to list of staff working
            main_frame.setVisible(false);
        }
    }
});

```

```

        refreshFrame.setVisible(true);
    }
    catch (Exception exc)
    {
        System.out.println(exc);
    }
}
});

BackMain.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        try
        {
            //Redirect back to main administrator page
            main_frame.setVisible(false);
            new MainScreenAdmin();
        }
        catch (Exception exc)
        {
            System.out.println(exc);
        }
    }
});
main_frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
main_frame.setVisible(true);
}

public String[] listWorkingStaff() throws Exception
{
    //Get the individual list of calls based on the selected staff member
    Class.forName("org.sqlite.JDBC");
    Connection conn =
    DriverManager.getConnection("jdbc:sqlite:NurseScheduling.sqlite");
    Statement stat = conn.createStatement();
    ResultSet rs = stat.executeQuery(String.format("SELECT * FROM Calls, CallLists2
    where Calls.NurseName = CallLists2.NurseName and CallLists2.NurseName = '%s'",
    selectedStaff));

    String[] rawModList = new String[200];
    int index = 0;
    while (rs.next())
    {
        rawModList[index] = rs.getString("Date") + " " + rs.getString("PName") + " "

```

```

        + rs.getString("FName") + " " + rs.getString("PAddress") + " " +
        rs.getString("PTown") + " " + rs.getString("Treatment") + " " +
        rs.getString("Kcode");
        index++;
    }
    rs.close();
    conn.close();
    int numElements = 0;
    for (int i = 0; i < 200; i++)
    {
        if (rawModList[i] != null)
        {
            numElements++;
        }
    }

    String[] modList = new String[numElements];
    System.arraycopy(rawModList, 0, modList, 0, numElements);
    return modList;
}

public static JFrame test_frame()
{
    return main_frame;
}

public static String Get_selectedStaff()
{
    return staffSelected;
}

public static void main(String args[]) throws Exception
{
    new SearchSelectedStaff();
}
}

```