HealthConnect

CSCI 430 Software Engineering
Fall Semester

Mireyahernandez57@yahoo.com

Table of Contents

Chapter 1: Overview

Introduction

Patients need efficient access to doctors regardless of financial situations or their location. HealthConnect is a web-based application that will provide a seamless and effective way to connect doctors and patients. Along with being effective and widely available, HealthConnect will be reliable, private, and secure. The application will allow patients to send messages to doctors and receive advice and recommendations quickly and adequately. Our first goal was to develop an application prototype for a future application intended to run on a mobile Android device. We focused on creating a GUI program that simulates the message interaction between doctors and patients.

In order to complete the goal of having a seamless and intuitive tool for doctor-patient communication, there are objectives the application must fulfill. Only registered patients will be able to send messages describing their symptoms. In return, doctors will reply to their questions with advice stating what the patient should do. Along with these objectives are high-level recommendations that are factored into the project before HealthConnect will be completed. With these goals, objectives, and high-level requirements, we can create an effective system for doctors and patients.

Justification

An advantage to creating the application prototype is using a program with a GUI tool. The GUI tool allows the prototype to be more user-friendly and attractive for all users. The tool generates GUI code for the user in comparison to coding the GUI by hand. The tool allows the creation of windows that enable users to view, create, control, and operate multiple tasks at one time which

is faster than a menu-driven program. Within, HealthConnect we can see how the user interacts with the application using GUI and analyze the success interface interaction.

Approach

With four people on the team, time conflicts will occur. We created and modified the requirements for the project during the semester. Although other project issues occurred, we also created tentative schedules and tasks to be completed. When all roles were assigned different tasks, development of the projects was started. When the prototype was completed, we tested each component of the application. After each component was checked, we then checked the app overall and verified its validation. That way, we didn't have to worry about patients and/or doctors sending in reports about components not performing the correct functions.

Tools and Technologies

The most efficient language for the application is Java. Java is an open source programming language which uses object-oriented and a class-based program system. Java is used for many mobile applications and software development. Also, we did not have to invest in software to start working on Java. We also used Netbeans, and SQLite. Netbeans is a free architecture application that contains developed packages containing everything needed to create an application. With the SQLite plug-in, we were able to connect the application to a database.

Roles and Responsibilities

The project manager will be X and he will be responsible for the work and effort of the group. The project manager is the motivator and the spokesman for the group as well. Along with the title the project manager, he also has numerous tasks. The first task of the project

manager is to design and create the user interface for the profiles of the application with the help of the architect. The final task of the manager will be to analyze the prototype's validation.

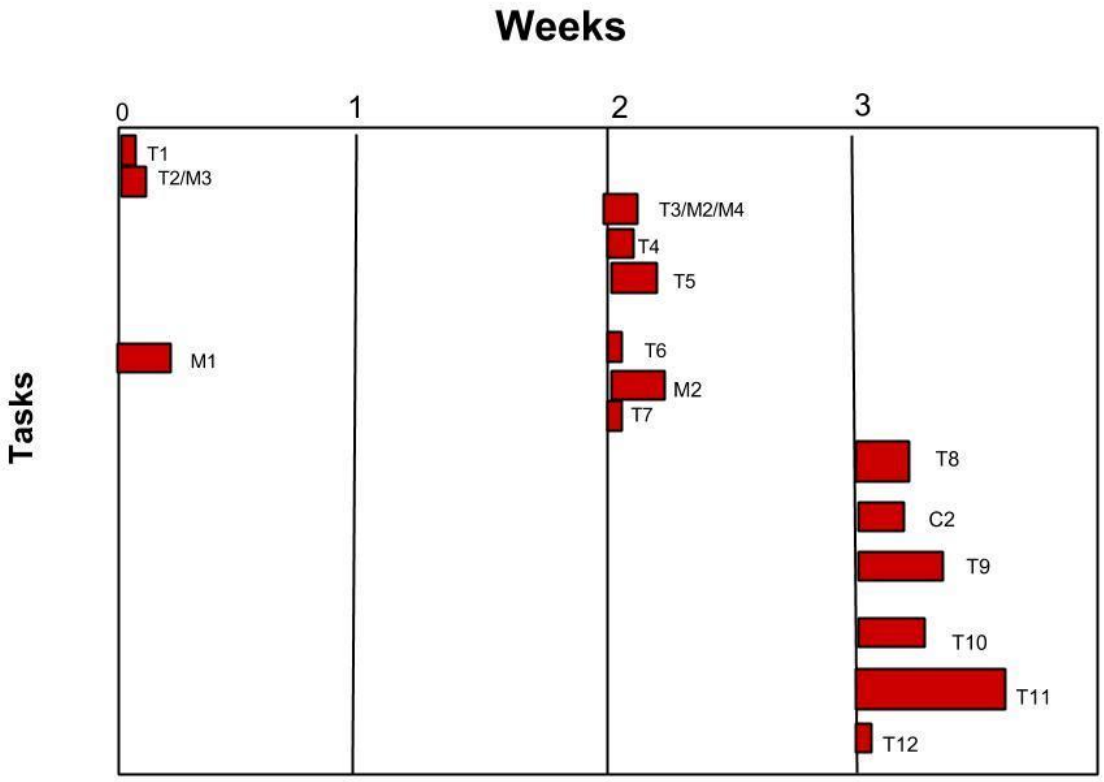
The project architect and designer is Y. She has control for the development of the design and documents the majority of the project. Along with the design, the architect also has a set of tasks. Y's first task is to design and create the user interfaces for the profiles along with the other members. She is in charge of creating a section in the interface for the requests. Another task is creating the function to assign the open, in progress, or closed status to each request.

The programmer, Z, is in charge of the coding portion of the project. The programmer converts requirements into code. The programmer and architect are responsible for entering the user information into a database and connecting database to the user interface. While working with the test manager for the next task, they are responsible for setting and testing permissions to check if the users could access too much. The programmer codes the timestamps and status into each request. The requests cannot be opened by anyone other than the doctor assigned to it. The programmer also debugs and solves technical problems in the coding.

The test manager of the project is XA and he is responsible for testing all aspects of the prototype. He performs various tests to ensure the program works properly in every aspect. If the project is not where it should be, he will notify the team members of the prototype vulnerabilities. Two very important tasks of the tester include testing the program's lack of security and incorrect permission within the users. The tests will check for SQL injection and other possible attacks. The test manager and programmer are responsible for troubleshooting unknown problems with security and coding.

Chapter 2: Requirements/Security and Defenses

Project Schedule



The project schedule above distributes the tasks and milestones within the timeline of the prototype development.

Security and Defenses

R1: Doctor and patient profiles must be created.

T1: Developers enter user information into a database.

T2: Design and create a user interface for the profiles. (Also a milestone)

r1: It could be confusing to the user,

T3: Set and test permissions.

r2: Incorrect permissions for users (too much or not enough permission)

T4: Allow doctor and patient to update a request

T5: Allow doctor and patient to view a request

M1: Create a SQLite database to store patient and doctor information.

M2: Test the application for incorrect user permissions.

C1: Prevent information from being stolen,

R2: Patients create requests and only one doctor may open it.

T6: Create request section in UI.

r3: They may input SQL queries

T7: Create settings where only doctors can see requests.

r4: Doctors may not see a request or pass over it.

r5: If setup incorrectly, patients may see other patient requests.

T8: Assign the requests to the doctor who opens it.

r6: A request may never be opened -

r7: Opened request may never be resolved and because it is open, no other doctor is assigned to it.

M3: Create intuitive and interactive user interface

C2: Add limitations and restrictions,

R3: Doctors reply to requests option

T9: Allow doctors to open request and respond

r8: May respond with a non-effective solution,

M4: Test for incorrect user permissions

R4: Requests have a timestamp and status.

T10: Create code to attach timestamps.

T11: Assign an open status showing the request which has not yet be answered. The progress status showing the request which is being answered as well as the closed status showing the request that has been closed.

r9: When do we determine if a request has been opened for too long? How is that request reassigned?

R5: Create troubleshooting process

T12: Problems with security

M5: Test the application for basic functionality (Can requests be sent, can users create a profile, etc.)

Chapter 3: Design and Implementation

UML Diagrams

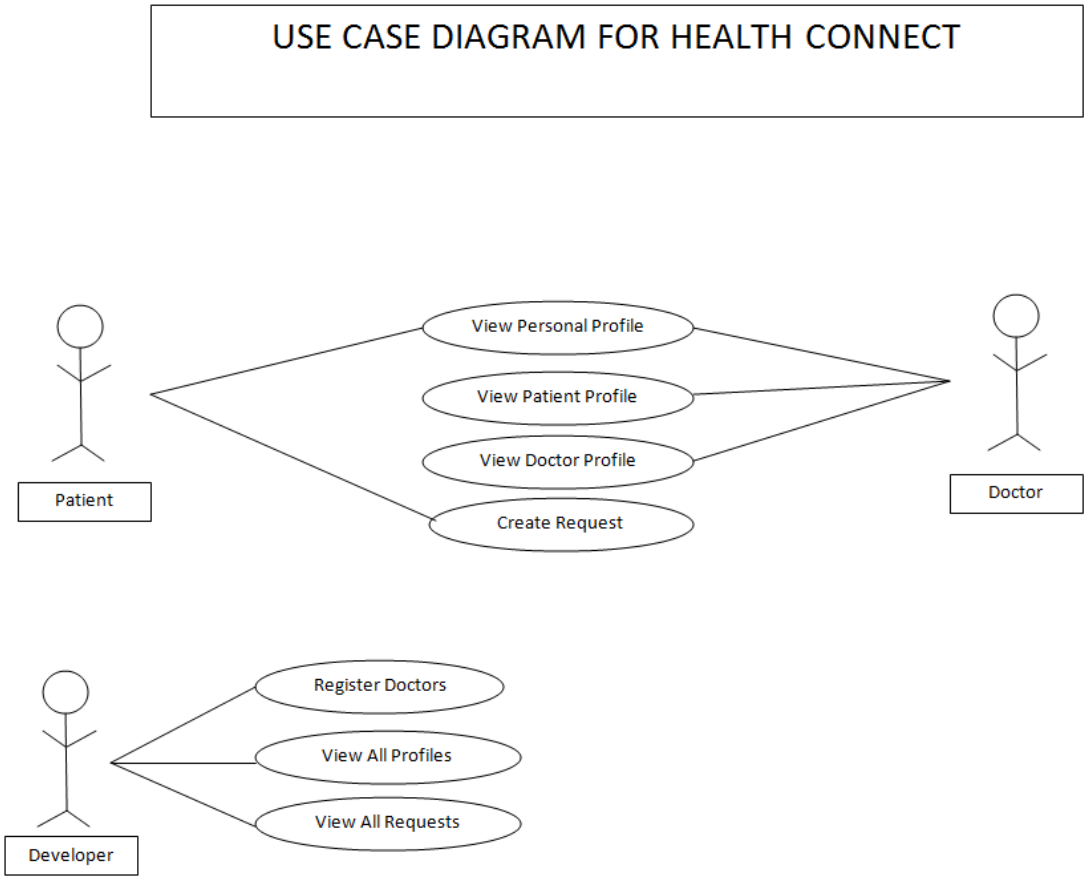


Figure 1: Use Case Diagram for the prototype

Class Diagram for Health Connect

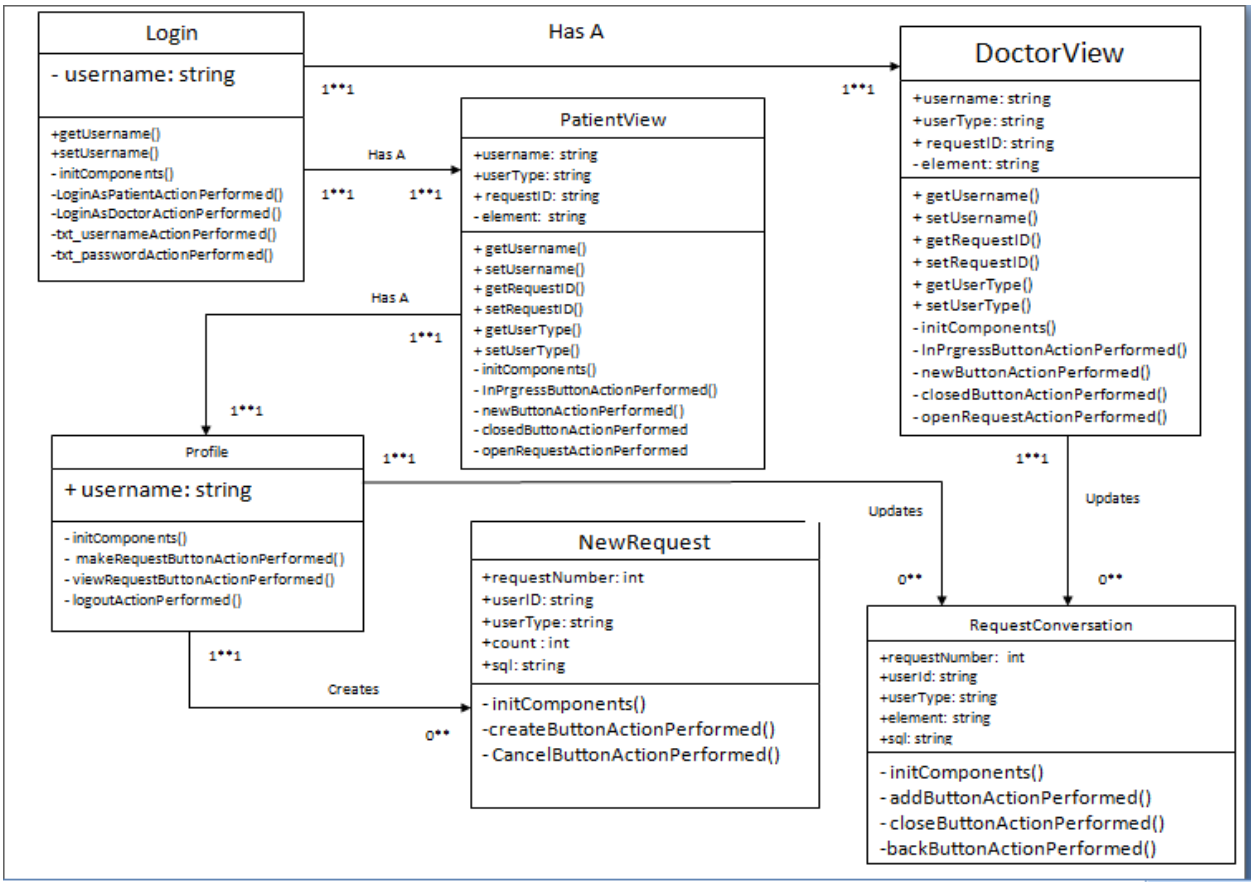


Figure 2: Class diagram for prototype

Sequence Diagrams for Health Connect

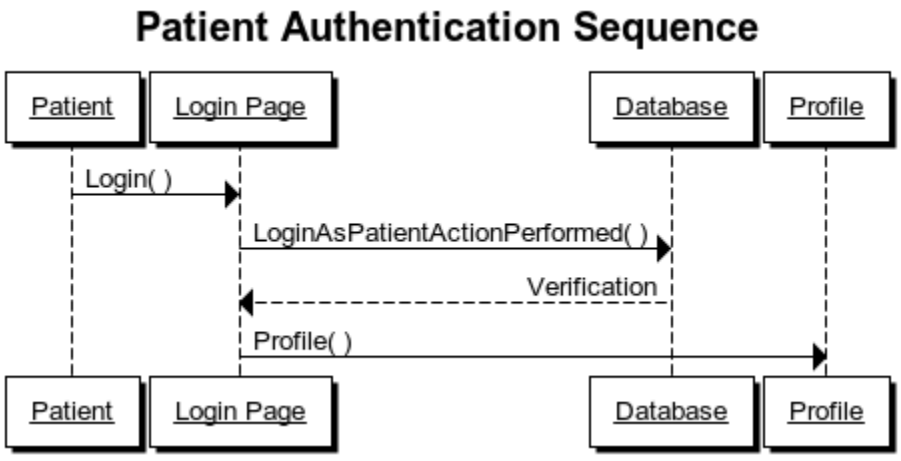


Figure 3: Sequence for patient to

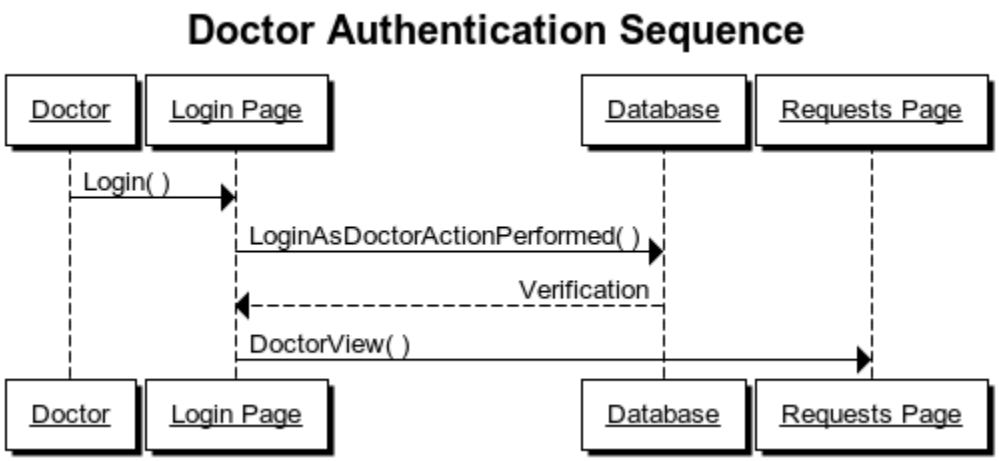


Figure 4: Sequence for doctor to login

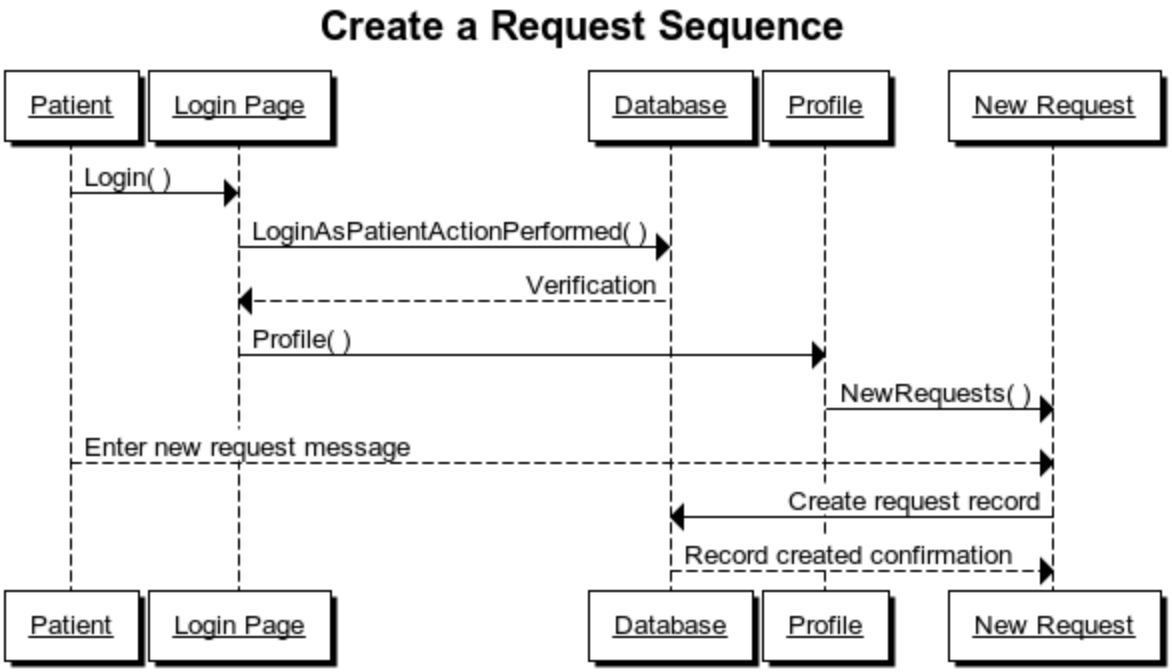


Figure 5: How a patient creates a request

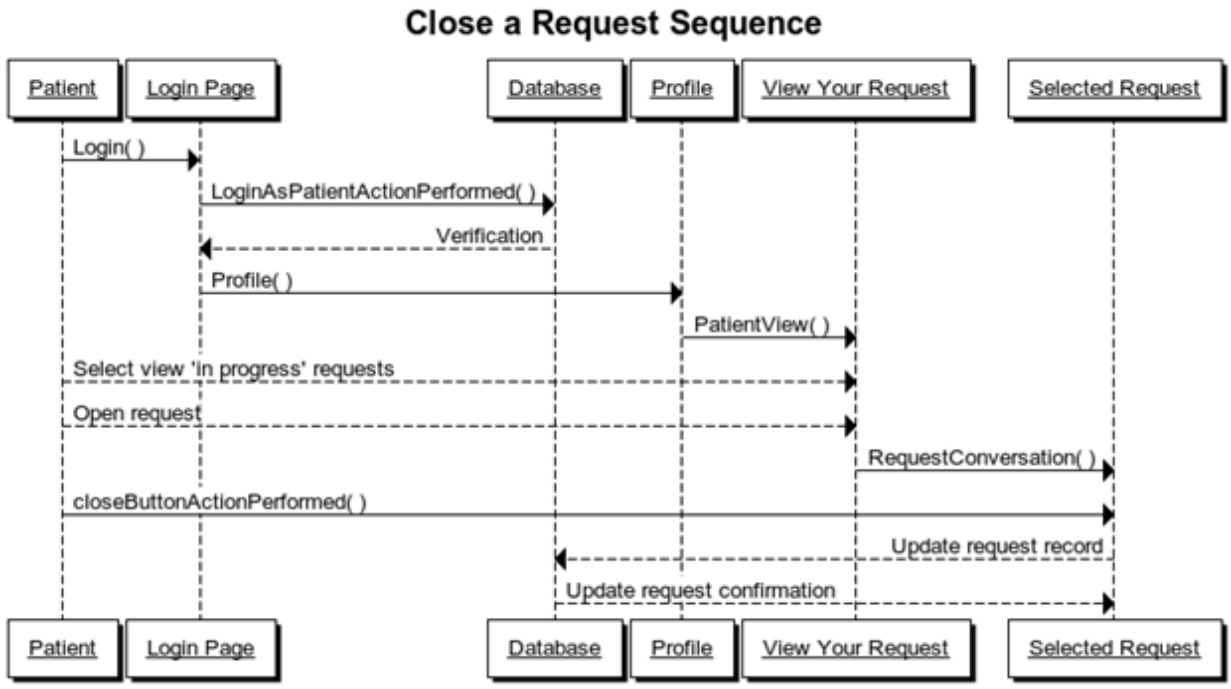


Figure 6: How a patient closes a request

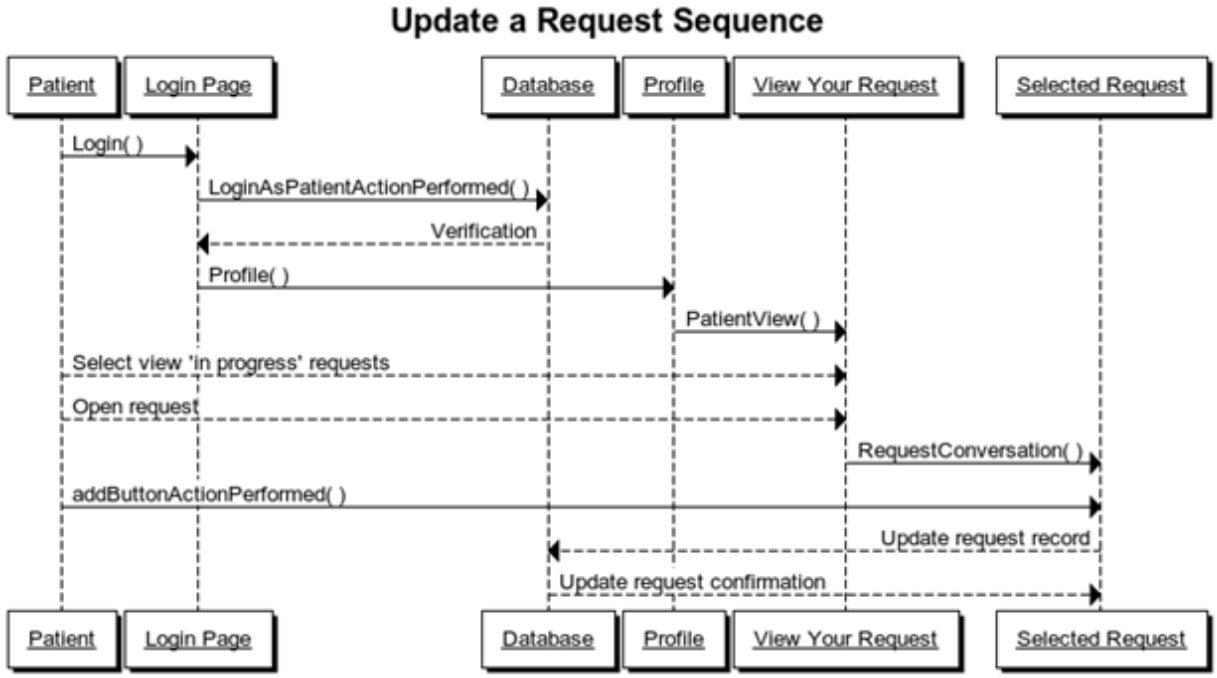


Figure 7: How a patient updates a request

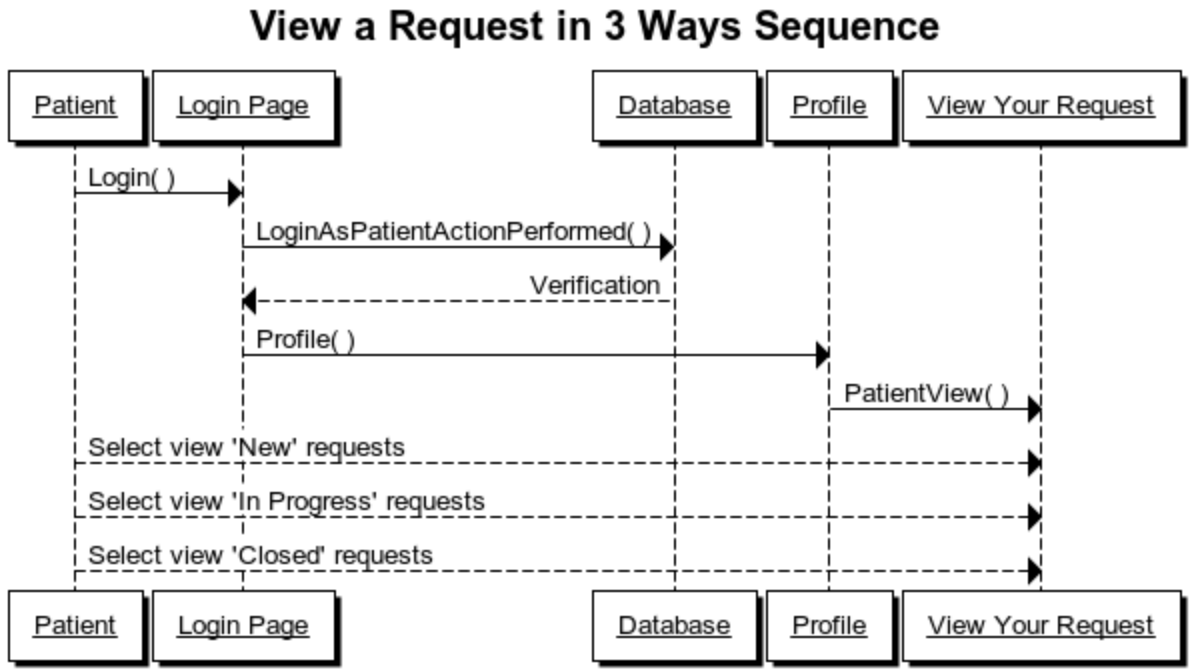


Figure 8: How a patient can view a request in three ways

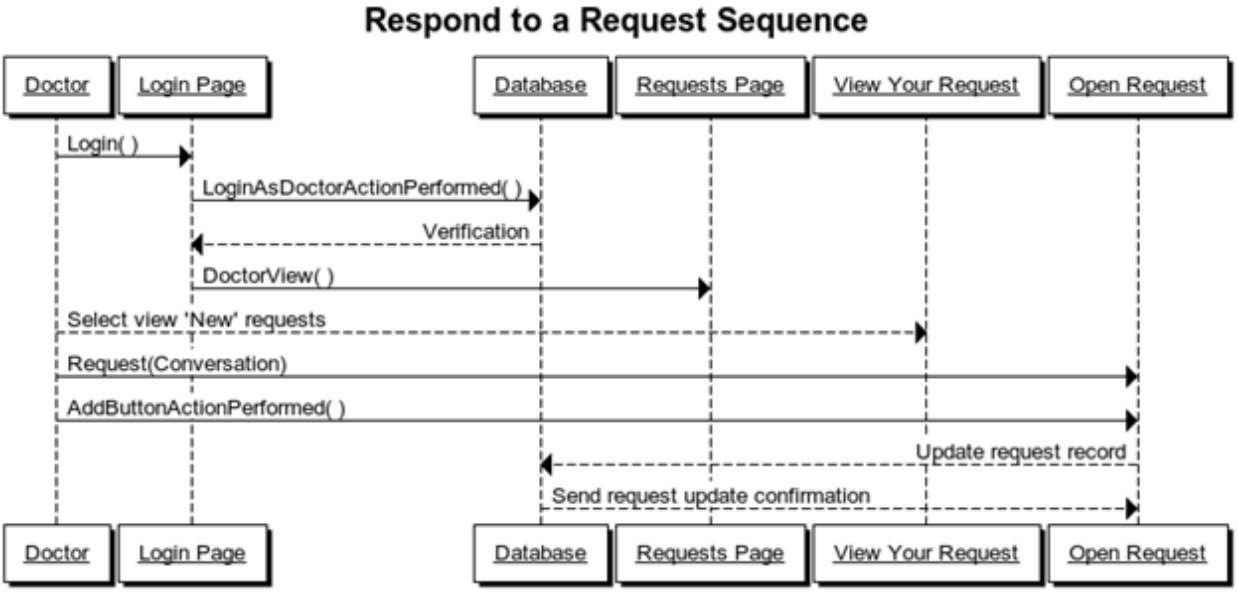


Figure 9: How a doctor can respond to a request

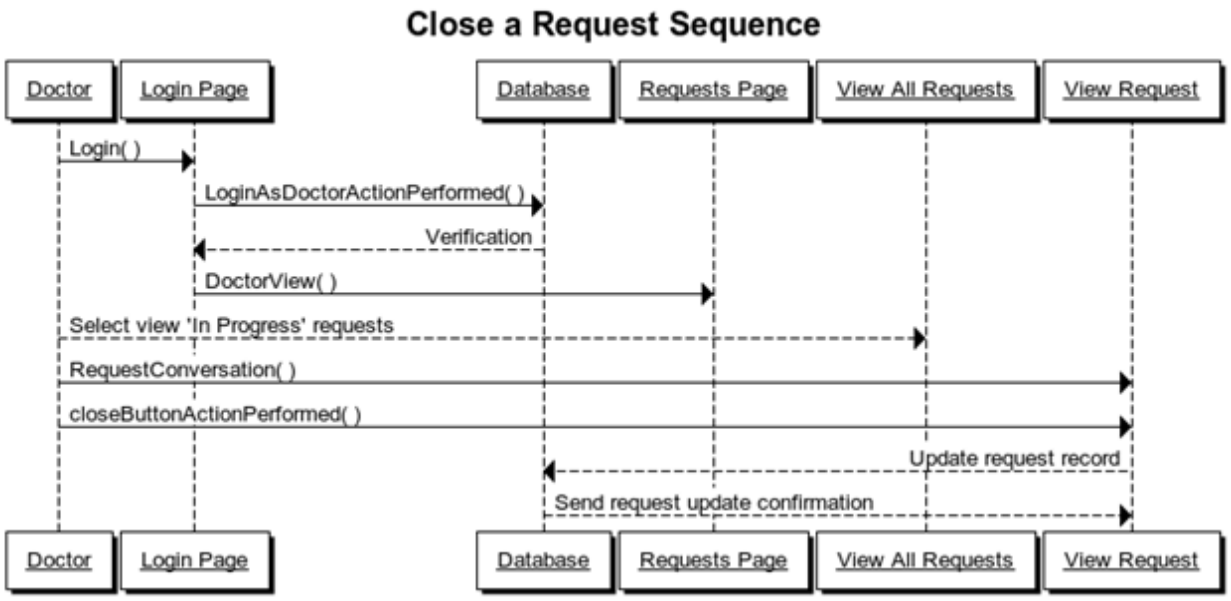


Figure 10: How a doctor closes a request

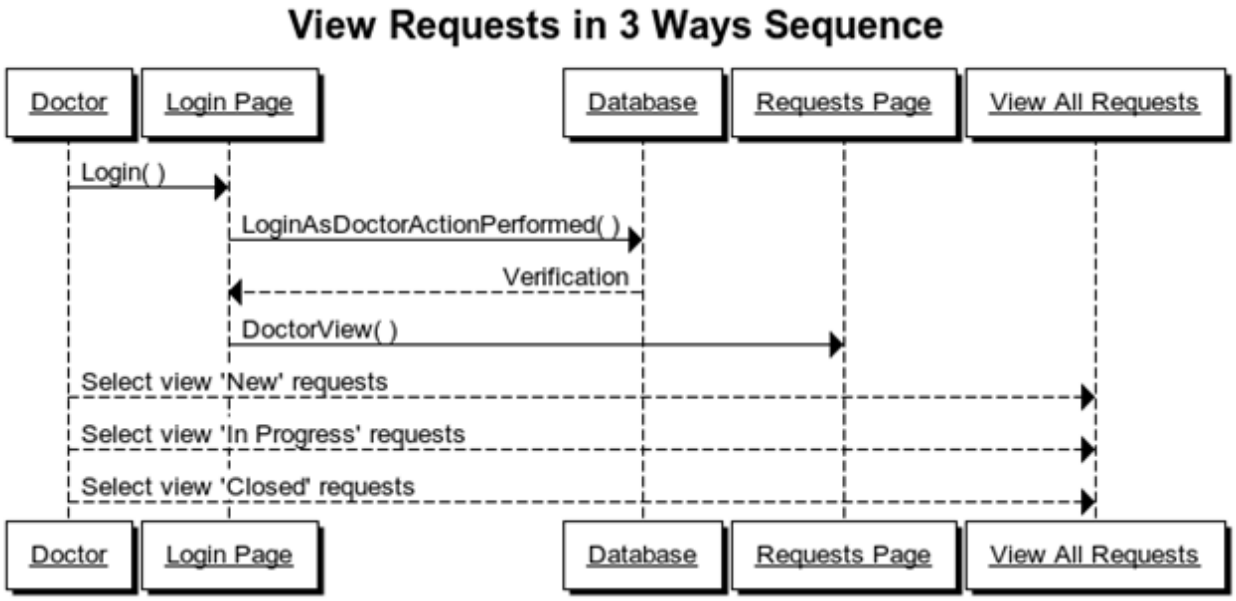
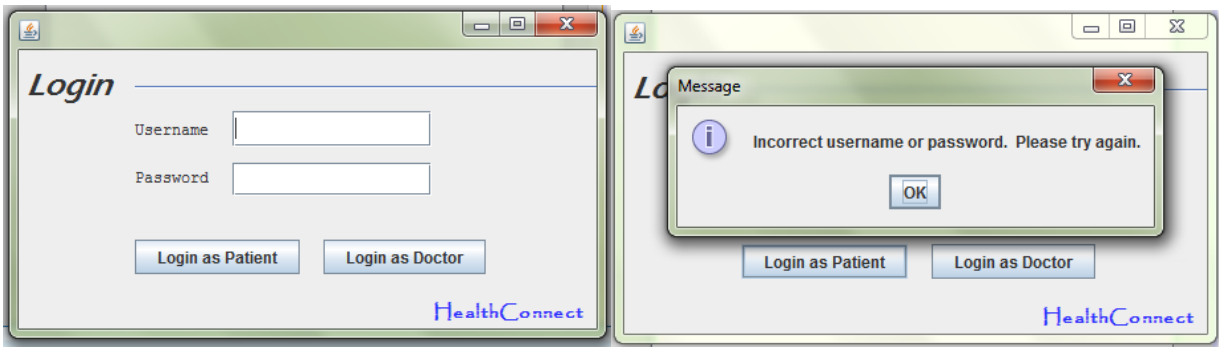


Figure 11: How a doctor can view a request in three ways

HealthConnect Prototype

The user interface was created with java GUI components. Because the interface was created in a windows theme, the interface buttons and options will look familiar to users of all experience. The interface was easy to create with the help of the interactive GUI tool within NetBeans. As you can see in

the image below, a normal login window appears when users first start the prototype, and dialog boxes appear with any extra information.



Source Code

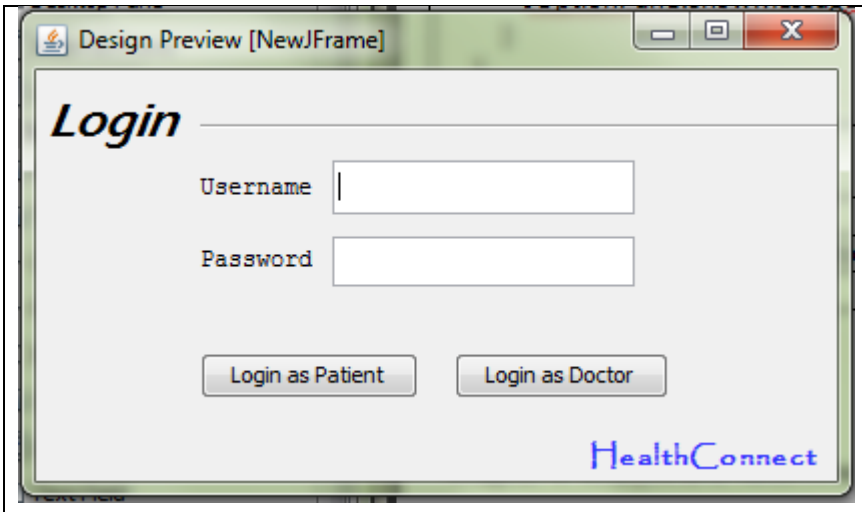
All of the source code below includes the name of the java file, the purpose of the file, and the code itself.

Health.java
This file tests the database connection and links to the login page.

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */

package health;

import java.sql.Connection;
import java.sql.DriverManager;
import javax.swing.JOptionPane;
import java.sql.Statement;
import java.sql.ResultSet;
import java.sql.SQLException;


public class Health {

   /**
    * @param args the command line arguments
    */
```

```
   public static void main(String[] args) {
      // TODO code application logic here

      @SuppressWarnings("UnusedAssignment")
      //test connection
      Connection conn = null;
      try {
         Class.forName("org.sqlite.JDBC");
         conn =
DriverManager.getConnection("jdbc:sqlite:C:\\Users\\Andrea\\Documents\\NetBeansProjects\\healt
h.sqlite");
         //JOptionPane.showMessageDialog (null, "Connected");
         Statement statement = conn.createStatement();
         ResultSet hc = statement.executeQuery("select * from Patient");

         while (hc.next()){
            System.out.println("Username = " + hc.getString("Username"));
            System.out.println("Password = " + hc.getString("Password"));
         }

         hc = statement.executeQuery("select * from Doctor");

         while (hc.next()){
            System.out.println("Username = " + hc.getString("Username"));
            System.out.println("Password = " + hc.getString("Password"));
         }
         //open login page
         NewJFrame s= new NewJFrame();
         s.setVisible(true);
      }
      catch(ClassNotFoundException | SQLException e){
         JOptionPane.showMessageDialog(null, e);
      }
   }
}
```

NewJFrame.java
This is the login form for the prototype. Users enter their username and password to begin.




```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */

package health;

import java.awt.Toolkit;
import java.awt.*;
import java.sql.Connection;
import java.sql.DriverManager;
import javax.swing.JOptionPane;
import java.sql.*;
import java.sql.Statement;
import java.sql.ResultSet;

public class NewJFrame extends javax.swing.JFrame {
   Connection conn=null;
   ResultSet rs=null;
   PreparedStatement pst=null;
   int curRow=0;
   private String username;
   /**
    * Creates new form NewJFrame
    */
   public NewJFrame() {
      initComponents();
      try {
         Class.forName("org.sqlite.JDBC");
         conn =
DriverManager.getConnection("jdbc:sqlite:C:\\Users\\Andrea\\Documents\\NetBeansProjects\\healt
h.sqlite");
```

```
        //JOptionPane.showMessageDialog (null, "Connected");
        Statement statement = conn.createStatement();
    }
    catch(ClassNotFoundException | SQLException e){
        JOptionPane.showMessageDialog(null, e);
    }
    Toolkit toolkit = getToolkit();
    Dimension size = toolkit.getScreenSize();
    setLocation(size.width/2 - getWidth()/2,
    size.height/2 - getHeight()/2);
}


public String getUsername(){
    return this.username;
}

public void setUsername(String username){
    this.username = this.txt_username.getText();
}

/**
 * This method is called from within the constructor to initialize the form.
 * WARNING: Do NOT modify this code. The content of this method is always
 * regenerated by the Form Editor.
 */
@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated Code">
private void initComponents() {

    jLabel1 = new javax.swing.JLabel();
    txt_username = new javax.swing.JTextField();
    jLabel2 = new javax.swing.JLabel();
    LoginAsPatient = new javax.swing.JButton();
    LoginAsDoctor = new javax.swing.JButton();
    jLabel3 = new javax.swing.JLabel();
    txt_password = new javax.swing.JPasswordField();
    jSeparator1 = new javax.swing.JSeparator();
    jLabel4 = new javax.swing.JLabel();

    setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

    jLabel1.setFont(new java.awt.Font("Courier New", 0, 12)); // NOI18N
    jLabel1.setText("Username");

    txt_username.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            txt_usernameActionPerformed(evt);
```

```
        }
    });

    jLabel2.setFont(new java.awt.Font("Courier New", 0, 12)); // NOI18N
    jLabel2.setText("Password");

    LoginAsPatient.setText("Login as Patient");
    LoginAsPatient.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            LoginAsPatientActionPerformed(evt);
        }
    });

    LoginAsDoctor.setText("Login as Doctor");
    LoginAsDoctor.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            LoginAsDoctorActionPerformed(evt);
        }
    });

    jLabel3.setFont(new java.awt.Font("Eras Demi ITC", 2, 24)); // NOI18N
    jLabel3.setText("Login");

    txt_password.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            txt_passwordActionPerformed(evt);
        }
    });

    jLabel4.setFont(new java.awt.Font("Papyrus", 1, 14)); // NOI18N
    jLabel4.setForeground(new java.awt.Color(51, 51, 255));
    jLabel4.setText("HealthConnect");

    javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
    getContentPane().setLayout(layout);
    layout.setHorizontalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
        .addGroup(layout.createSequentialGroup()
            .addGap(6, 6, 6)
            .addComponent(jLabel3, javax.swing.GroupLayout.PREFERRED_SIZE, 70,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addComponent(jSeparator1)
                .addGroup(layout.createSequentialGroup()
                    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                        .addGroup(layout.createSequentialGroup()
                            .addComponent(LoginAsPatient)
```

```
                        .addGap(18, 18, 18)
                        .addComponent(LoginAsDoctor))
                    .addGroup(layout.createSequentialGroup()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
                        .addComponent(jLabel2, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                        .addComponent(jLabel1, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
                    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
                        .addComponent(txt_username)
                        .addComponent(txt_password, javax.swing.GroupLayout.DEFAULT_SIZE, 151,
Short.MAX_VALUE))))
                .addContainerGap(84, Short.MAX_VALUE))))
        .addGroup(layout.createSequentialGroup()
           .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
           .addComponent(jLabel4, javax.swing.GroupLayout.PREFERRED_SIZE, 124,
javax.swing.GroupLayout.PREFERRED_SIZE))
      );
      layout.setVerticalGroup(
         layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
         .addGroup(layout.createSequentialGroup()
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
               .addGroup(layout.createSequentialGroup()
                  .addGap(27, 27, 27)
                  .addComponent(jSeparator1, javax.swing.GroupLayout.PREFERRED_SIZE, 10,
javax.swing.GroupLayout.PREFERRED_SIZE))
               .addGroup(layout.createSequentialGroup()
                  .addContainerGap()
                  .addComponent(jLabel3)))
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
               .addComponent(txt_username, javax.swing.GroupLayout.PREFERRED_SIZE, 27,
javax.swing.GroupLayout.PREFERRED_SIZE)
               .addComponent(jLabel1, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
               .addComponent(jLabel2, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
               .addComponent(txt_password, javax.swing.GroupLayout.DEFAULT_SIZE, 25,
Short.MAX_VALUE))
            .addGap(33, 33, 33)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
               .addGroup(layout.createSequentialGroup()
                  .addComponent(LoginAsDoctor)
```

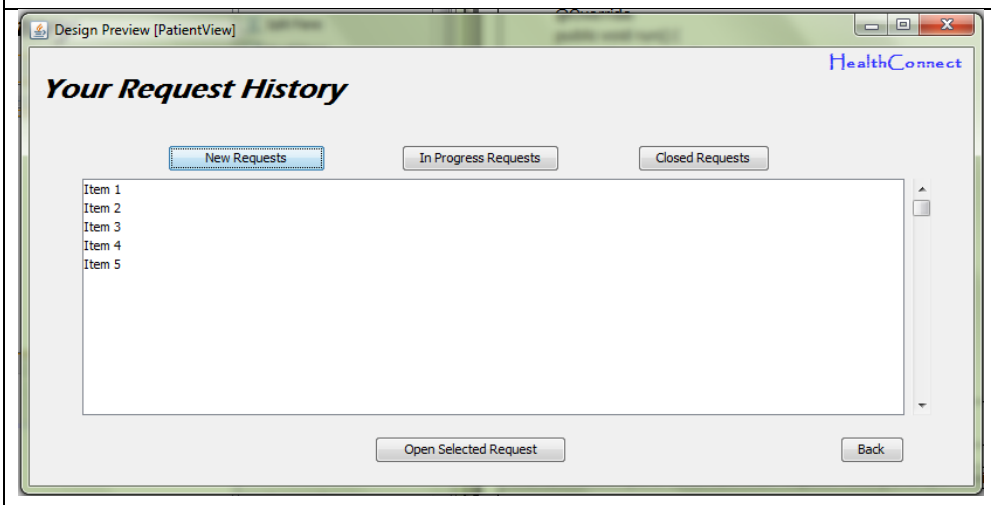
```
                .addGap(18, 18, Short.MAX_VALUE)
                .addComponent(jLabel4))
            .addComponent(LoginAsPatient)))
    );

    pack();
}// </editor-fold>

private void LoginAsPatientActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
   String sql ="select * from Patient where username=? and password=?";
   try{
      pst=conn.prepareStatement(sql);
      pst.setString(1, txt_username.getText());
      pst.setString(2, txt_password.getText());
      username = txt_username.getText();
      setUsername(username);
      //JOptionPane.showMessageDialog (null, "Username = " + username);
      rs = pst.executeQuery();


      if(rs.next()){
         //JOptionPane.showMessageDialog(null, "Username and Password is correct");
         Profile s= new Profile(username);
         s.setVisible(true);
         dispose();
      }
      else{
         JOptionPane.showMessageDialog(null, "Incorrect username or password.  Please try again.");

      }
   }

   catch(HeadlessException | SQLException e){
      JOptionPane.showMessageDialog(null, e);
   }finally{
       try{
          rs.close();
             pst.close();
       }
       catch(SQLException e){
          JOptionPane.showMessageDialog(null, e);
       }
    }


}
```

```
private void LoginAsDoctorActionPerformed(java.awt.event.ActionEvent evt) {
   // TODO add your handling code here:
  String sql ="select * from Doctor where username=? and password=?";
  try{
     pst=conn.prepareStatement(sql);
     pst.setString(1, txt_username.getText());
     pst.setString(2, txt_password.getText());
     username = txt_username.getText();
     setUsername(username);

     rs = pst.executeQuery();
     if(rs.next()){
        //JOptionPane.showMessageDialog(null, "Username and Password is correct");
        DoctorView d= new DoctorView(username);
        d.setVisible(true);
        dispose();
     }
     else{
        JOptionPane.showMessageDialog(null, "Incorrect username or password.  Please try again.");

     }
  }

  catch(HeadlessException | SQLException e){
     JOptionPane.showMessageDialog(null, e);
  }finally{
      try{
         rs.close();
            pst.close();
      }
      catch(SQLException e){
         JOptionPane.showMessageDialog(null, e);
      }
   }


}

private void txt_usernameActionPerformed(java.awt.event.ActionEvent evt) {
   // TODO add your handling code here:
}

private void txt_passwordActionPerformed(java.awt.event.ActionEvent evt) {
   // TODO add your handling code here:
}

/**
 * @param args the command line arguments
```

```
    */
   public static void main(String args[]) {
      /* Set the Nimbus look and feel */
      //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
      /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
       * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
       */
      try {
         for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
            if ("Nimbus".equals(info.getName())) {
               javax.swing.UIManager.setLookAndFeel(info.getClassName());
               break;
            }
         }
      } catch (ClassNotFoundException | InstantiationException | IllegalAccessException |
javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(NewJFrame.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
      }
      //</editor-fold>

      /* Create and display the form */
      java.awt.EventQueue.invokeLater(new Runnable() {
         @Override
         public void run() {
            new NewJFrame().setVisible(true);
         }
      });
   }

   // Variables declaration - do not modify
   private javax.swing.JButton LoginAsDoctor;
   private javax.swing.JButton LoginAsPatient;
   private javax.swing.JLabel jLabel1;
   private javax.swing.JLabel jLabel2;
   private javax.swing.JLabel jLabel3;
   private javax.swing.JLabel jLabel4;
   private javax.swing.JSeparator jSeparator1;
   private javax.swing.JPasswordField txt_password;
   public javax.swing.JTextField txt_username;
   // End of variables declaration
}
```

PatientView.java

This form allows patients to view their request history and open their requests.




```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */

package health;

import java.awt.HeadlessException;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import javax.swing.DefaultListModel;
import javax.swing.JOptionPane;

public final class PatientView extends javax.swing.JFrame {
   Connection conn=null;
   ResultSet rs=null;
   PreparedStatement pst=null;
   int curRow=0;
   String username, userType;
   DefaultListModel model = new DefaultListModel();
   int index;
   int requestID;
   /**
    * Creates new form PatientView
    * @param patient
    */
```

```
    public PatientView(String patient) {
        initComponents();
        try {
            Class.forName("org.sqlite.JDBC");
            conn =
DriverManager.getConnection("jdbc:sqlite:C:\\Users\\Andrea\\Documents\\NetBeansProjects\\healt
h.sqlite");
            //JOptionPane.showMessageDialog (null, "Connected");
            Statement statement = conn.createStatement();
        }
        catch(ClassNotFoundException | SQLException e){
            JOptionPane.showMessageDialog(null, e);
        }
        username = patient;
        setUsername(patient);
        userType = "Patient";
        setUserType(userType);
        jList1.setVisible(false);

    }

    public String getUsername(){
        return this.username;
    }

    public void setUsername(String username){
        this.username = username;
    }

    public int getRequestID(){
        return this.requestID;
    }

    public void setRequestID(int requestID){
        this.requestID = requestID;
    }

    public String getUserType(){
        return this.userType;
    }

    public void setUserType(String userType){
        this.userType = userType;
    }

    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
```

```
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        InProgressButton = new javax.swing.JButton();
        closedButton = new javax.swing.JButton();
        jScrollPane1 = new javax.swing.JScrollPane();
        jList1 = new javax.swing.JList();
        jScrollBar1 = new javax.swing.JScrollBar();
        jLabel1 = new javax.swing.JLabel();
        openRequest = new javax.swing.JButton();
        jLabel7 = new javax.swing.JLabel();
        backButton = new javax.swing.JButton();
        newButton = new javax.swing.JButton();

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

        InProgressButton.setText("In Progress Requests");
        InProgressButton.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                InProgressButtonActionPerformed(evt);
            }
        });

        closedButton.setText("Closed Requests");
        closedButton.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                closedButtonActionPerformed(evt);
            }
        });

        jList1.setModel(new javax.swing.AbstractListModel() {
            String[] strings = { "Item 1", "Item 2", "Item 3", "Item 4", "Item 5" };
            public int getSize() { return strings.length; }
            public Object getElementAt(int i) { return strings[i]; }
        });
        jScrollPane1.setViewportView(jList1);

        jLabel1.setFont(new java.awt.Font("Eras Demi ITC", 3, 24)); // NOI18N
        jLabel1.setText("Your Request History");

        openRequest.setText("Open Selected Request");
        openRequest.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                openRequestActionPerformed(evt);
            }
```

```
    });

    jLabel7.setFont(new java.awt.Font("Papyrus", 1, 14)); // NOI18N
    jLabel7.setForeground(new java.awt.Color(51, 51, 255));
    jLabel7.setText("HealthConnect");

    backButton.setText("Back");
    backButton.addActionListener(new java.awt.event.ActionListener() {
      public void actionPerformed(java.awt.event.ActionEvent evt) {
        backButtonActionPerformed(evt);
      }
    });

    newButton.setText("New Requests");
    newButton.addActionListener(new java.awt.event.ActionListener() {
      public void actionPerformed(java.awt.event.ActionEvent evt) {
        newButtonActionPerformed(evt);
      }
    });

    javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
    getContentPane().setLayout(layout);
    layout.setHorizontalGroup(
      layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
      .addGroup(layout.createSequentialGroup()
        .addContainerGap()
        .addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED_SIZE, 513,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
        .addComponent(jLabel7, javax.swing.GroupLayout.PREFERRED_SIZE, 124,
javax.swing.GroupLayout.PREFERRED_SIZE))
      .addGroup(layout.createSequentialGroup()
        .addGap(45, 45, 45)
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
          .addGroup(layout.createSequentialGroup()
            .addComponent(openRequest, javax.swing.GroupLayout.PREFERRED_SIZE, 164,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(234, 234, 234)
            .addComponent(backButton))
          .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 703,
javax.swing.GroupLayout.PREFERRED_SIZE)
          .addGroup(layout.createSequentialGroup()
            .addComponent(newButton, javax.swing.GroupLayout.PREFERRED_SIZE, 135,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(65, 65, 65)
            .addComponent(InProgressButton)
            .addGap(67, 67, 67)
```

```
                    .addComponent(closedButton)
                    .addGap(115, 115, 115)))
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(jScrollBar1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addContainerGap(36, Short.MAX_VALUE))
        );
        layout.setVerticalGroup(
           layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
           .addGroup(layout.createSequentialGroup()
              .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                 .addGroup(layout.createSequentialGroup()
                    .addGap(2, 2, 2)
                    .addComponent(jLabel7))
                 .addGroup(layout.createSequentialGroup()
                    .addContainerGap()
                    .addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED_SIZE, 48,
javax.swing.GroupLayout.PREFERRED_SIZE)))
              .addGap(24, 24, 24)
              .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                 .addComponent(InProgressButton)
                 .addComponent(closedButton)
                 .addComponent(newButton))
              .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
              .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
                 .addComponent(jScrollBar1, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                 .addComponent(jScrollPane1, javax.swing.GroupLayout.DEFAULT_SIZE, 202,
Short.MAX_VALUE))
              .addGap(18, 18, 18)
              .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                 .addComponent(openRequest)
                 .addComponent(backButton))
              .addGap(19, 19, 19))
        );

        pack();
    }// </editor-fold>

    private void InProgressButtonActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        jLabel1.setText("Your Opened Requests");
        jList1.setVisible(true);
        String element;
        String sql ="select RID,Date from Request where Status=? and PUsername=?";
        model.removeAllElements();
        element = "RID       Date";
```

```
    model.addElement(element);
   try{
      pst=conn.prepareStatement(sql);
      pst.setString(1, "In Progress");
      pst.setString(2, username);
      rs = pst.executeQuery();
      if(rs.next()){
         //JOptionPane.showMessageDialog(null, "Username and Password is correct");
         element = rs.getString("RID") + "        " + rs.getString("Date");
         model.addElement(element);
         while (rs.next()){
          element = rs.getString("RID") + "        " + rs.getString("Date");
          model.addElement(element);
      }
      jList1.setModel(model);
      }
      else{
         JOptionPane.showMessageDialog(null, "No requests are in progress.");

      }
   }

   catch(SQLException | HeadlessException e){
      JOptionPane.showMessageDialog(null, e);}finally{
       try{
          rs.close();
             pst.close();
       }
       catch(SQLException e){
          JOptionPane.showMessageDialog(null, e);
       }
    }
}

private void newButtonActionPerformed(java.awt.event.ActionEvent evt) {
   // TODO add your handling code here:
   jLabel1.setText("Your New Requests");
   jList1.setVisible(true);
   String element;
   String sql ="select RID,Date from Request where Status=? and PUsername=?";
   model.removeAllElements();
   element = "RID       Date";
   model.addElement(element);
   try{
      pst=conn.prepareStatement(sql);
      pst.setString(1, "New");
      pst.setString(2, username);
      rs = pst.executeQuery();
```

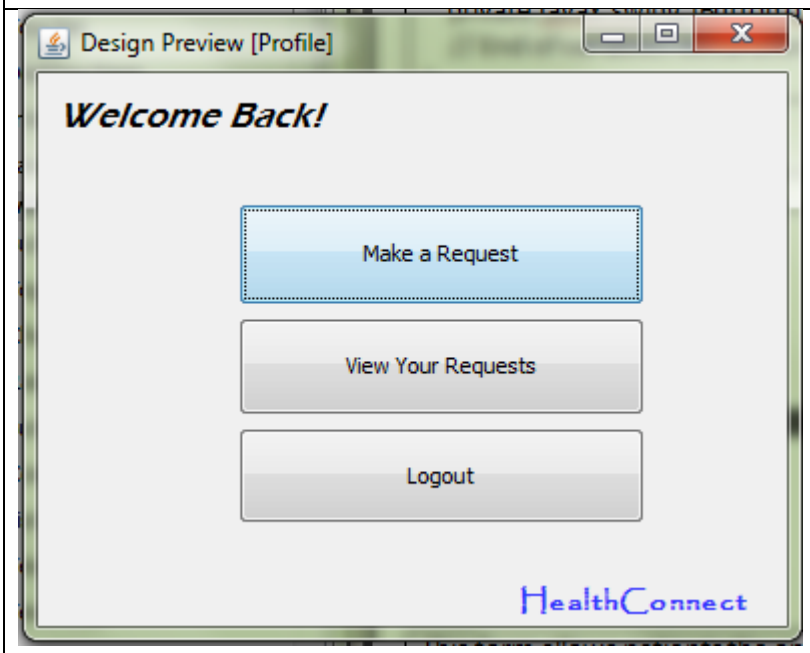
```
        if(rs.next()){
           //JOptionPane.showMessageDialog(null, "Username and Password is correct");
           element = rs.getString("RID") + "        " + rs.getString("Date");
           model.addElement(element);
           while (rs.next()){
            element = rs.getString("RID") + "        " + rs.getString("Date");
            model.addElement(element);
        }
        jList1.setModel(model);
        }
        else{
           JOptionPane.showMessageDialog(null, "No new requests.");

        }
     }

     catch(SQLException | HeadlessException e){
        JOptionPane.showMessageDialog(null, e);}finally{
         try{
            rs.close();
               pst.close();
         }
         catch(SQLException e){
            JOptionPane.showMessageDialog(null, e);
         }
     }
}

private void closedButtonActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    jLabel1.setText("Your Closed Requests");
    jList1.setVisible(true);
    String element;
    String sql ="select RID,Date from Request where Status=? and PUsername=?";
    model.removeAllElements();
    element = "RID       Date";
    model.addElement(element);
    try{
       pst=conn.prepareStatement(sql);
       pst.setString(1, "Closed");
       pst.setString(2, username);
       rs = pst.executeQuery();
       if(rs.next()){
          //JOptionPane.showMessageDialog(null, "Username and Password is correct");
          element = rs.getString("RID") + "        " + rs.getString("Date");
          model.addElement(element);
          while (rs.next()){
           element = rs.getString("RID") + "        " + rs.getString("Date");
```

```
            model.addElement(element);
        }
        jList1.setModel(model);
        }
        else{
           JOptionPane.showMessageDialog(null, "No requests have been closed.");

        }
    }

    catch(SQLException | HeadlessException e){
        JOptionPane.showMessageDialog(null, e);}finally{
        try{
            rs.close();
               pst.close();
        }
        catch(SQLException e){
            JOptionPane.showMessageDialog(null, e);
        }
    }
}

private void backButtonActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    dispose();
    Profile p = new Profile(username);
    p.setVisible(true);
}

private void openRequestActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    if(jList1.getSelectedIndex() != -1)
    {
    String temp_requestID = jList1.getSelectedValue().toString();
    temp_requestID = temp_requestID.substring(0,3);
    requestID = Integer.parseInt(temp_requestID);
    setRequestID(requestID);
    RequestConversation r = new RequestConversation(requestID, username, userType);
    dispose();
    r.setVisible(true);
    }
    else
        JOptionPane.showMessageDialog(null, "Please select a request");
}

/**
 * @param args the command line arguments
 */
```

```
    public static void main(String args[]) {
        /* Set the Nimbus look and feel */
        //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
        /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
         * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
         */
        try {
            for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
                if ("Nimbus".equals(info.getName())) {
                    javax.swing.UIManager.setLookAndFeel(info.getClassName());
                    break;
                }
            }
        } catch (ClassNotFoundException | InstantiationException | IllegalAccessException |
javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(PatientView.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
        }
        //</editor-fold>
        final NewJFrame s = new NewJFrame();
        /* Create and display the form */
        java.awt.EventQueue.invokeLater(new Runnable() {
            @Override
            public void run() {
                new PatientView(s.getUsername()).setVisible(true);
            }
        });
    }

    // Variables declaration - do not modify
    private javax.swing.JButton InProgressButton;
    private javax.swing.JButton backButton;
    private javax.swing.JButton closedButton;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel7;
    private javax.swing.JList jList1;
    private javax.swing.JScrollBar jScrollBar1;
    private javax.swing.JScrollPane jScrollPane1;
    private javax.swing.JButton newButton;
    private javax.swing.JButton openRequest;
    // End of variables declaration
}
```

Profile.java

This form allows patients the option to either view their history or make a request.




```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */

package health;

import java.awt.Dimension;
import java.awt.Toolkit;
import java.sql.Connection;
import javax.swing.JOptionPane;
import java.sql.*;
import java.sql.ResultSet;

public class Profile extends javax.swing.JFrame {
   Connection conn=null;
   ResultSet rs=null;
   PreparedStatement pst=null;
   int curRow=0;
   String username;
   /**
    * Creates new form Profile
    * @param patient
    */
   public Profile(String patient) {
      initComponents();
```

```
    Toolkit toolkit = getToolkit();
    Dimension size = toolkit.getScreenSize();
    setLocation(size.width/2 - getWidth()/2,
    size.height/2 - getHeight()/2);
    username = patient;
    welcome.setText("Welcome Back, " + username + "!");
}

/**
 * This method is called from within the constructor to initialize the form.
 * WARNING: Do NOT modify this code. The content of this method is always
 * regenerated by the Form Editor.
 */
@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated Code">
private void initComponents() {

    jLabel4 = new javax.swing.JLabel();
    welcome = new javax.swing.JLabel();
    makeRequestButton = new javax.swing.JButton();
    viewRequestButton = new javax.swing.JButton();
    jLabel5 = new javax.swing.JLabel();
    logout = new javax.swing.JButton();

    jLabel4.setFont(new java.awt.Font("Papyrus", 1, 14)); // NOI18N
    jLabel4.setForeground(new java.awt.Color(51, 51, 255));
    jLabel4.setText("HealthConnect");

    setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

    welcome.setFont(new java.awt.Font("Eras Demi ITC", 2, 18)); // NOI18N
    welcome.setText("Welcome Back!");

    makeRequestButton.setText("Make a Request");
    makeRequestButton.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            makeRequestButtonActionPerformed(evt);
        }
    });

    viewRequestButton.setText("View Your Requests");
    viewRequestButton.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            viewRequestButtonActionPerformed(evt);
        }
    });

    jLabel5.setFont(new java.awt.Font("Papyrus", 1, 14)); // NOI18N
```

```
        jLabel5.setForeground(new java.awt.Color(51, 51, 255));
        jLabel5.setText("HealthConnect");

        logout.setText("Logout");
        logout.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                logoutActionPerformed(evt);
            }
        });

        javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
        getContentPane().setLayout(layout);
        layout.setHorizontalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()
                        .addGap(0, 0, Short.MAX_VALUE)
                        .addComponent(jLabel5, javax.swing.GroupLayout.PREFERRED_SIZE, 124,
javax.swing.GroupLayout.PREFERRED_SIZE))
                    .addGroup(layout.createSequentialGroup()
                        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                            .addGroup(layout.createSequentialGroup()
                                .addGap(100, 100, 100)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
                                .addComponent(logout, javax.swing.GroupLayout.PREFERRED_SIZE, 203,
javax.swing.GroupLayout.PREFERRED_SIZE)
                                .addComponent(viewRequestButton, javax.swing.GroupLayout.PREFERRED_SIZE,
203, javax.swing.GroupLayout.PREFERRED_SIZE)
                                .addComponent(makeRequestButton, javax.swing.GroupLayout.PREFERRED_SIZE,
203, javax.swing.GroupLayout.PREFERRED_SIZE)))
                            .addGroup(layout.createSequentialGroup()
                                .addContainerGap()
                                .addComponent(welcome, javax.swing.GroupLayout.PREFERRED_SIZE, 342,
javax.swing.GroupLayout.PREFERRED_SIZE)))
                        .addGap(0, 12, Short.MAX_VALUE)))
                .addContainerGap())
        );
        layout.setVerticalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addComponent(welcome, javax.swing.GroupLayout.PREFERRED_SIZE, 42,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(23, 23, 23)
                .addComponent(makeRequestButton, javax.swing.GroupLayout.PREFERRED_SIZE, 51,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
```

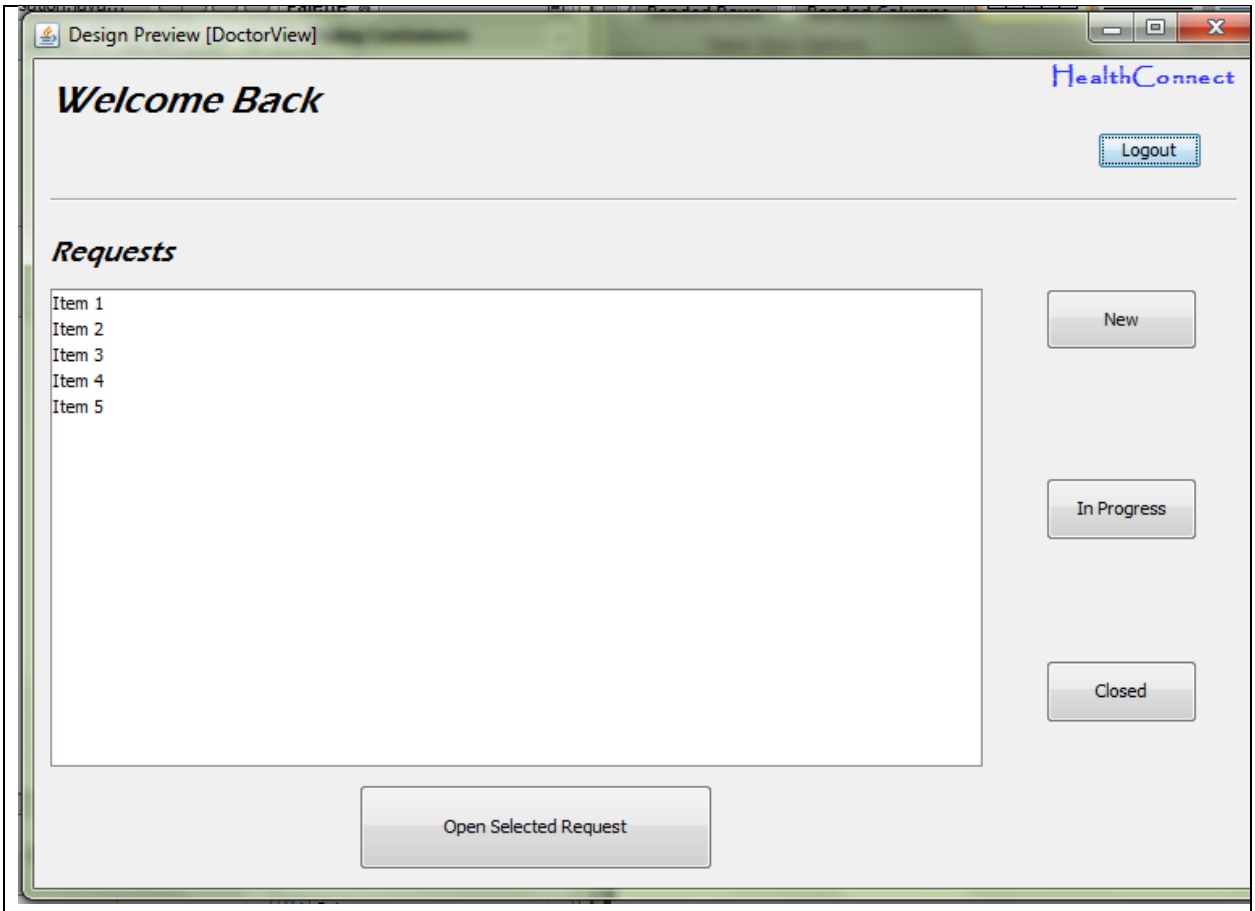
```
            .addComponent(viewRequestButton, javax.swing.GroupLayout.PREFERRED_SIZE, 49,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(logout, javax.swing.GroupLayout.PREFERRED_SIZE, 48,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 28,
Short.MAX_VALUE)
            .addComponent(jLabel5))
        );

        pack();
    }// </editor-fold>

    private void makeRequestButtonActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        dispose();
        NewRequests n = new NewRequests(username);
        n.setVisible(true);
    }

    private void viewRequestButtonActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        dispose();
        PatientView p = new PatientView(username);
        p.setVisible(true);
    }

    private void logoutActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        int pane = JOptionPane.showConfirmDialog(null, "Are you sure you want to logout?", "Logout",
JOptionPane.YES_NO_OPTION);
        if(pane==0){
        dispose();
        NewJFrame n = new NewJFrame();
        n.setVisible(true);}
    }

    /**
     * @param args the command line arguments
     */
    public static void main(String args[]) {
        /* Set the Nimbus look and feel */
        //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
        /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
         * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
         */
        try {
```

```
        for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
            if ("Nimbus".equals(info.getName())) {
                javax.swing.UIManager.setLookAndFeel(info.getClassName());
                break;
            }
        }
    } catch (ClassNotFoundException | InstantiationException | IllegalAccessException |
javax.swing.UnsupportedLookAndFeelException ex) {
        java.util.logging.Logger.getLogger(Profile.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
    }
    //</editor-fold>
    final NewJFrame s = new NewJFrame();
    /* Create and display the form */
    java.awt.EventQueue.invokeLater(new Runnable() {
        @Override
        public void run() {
            new Profile(s.getUsername()).setVisible(true);
        }
    });
  }

  // Variables declaration - do not modify
  private javax.swing.JLabel jLabel4;
  private javax.swing.JLabel jLabel5;
  private javax.swing.JButton logout;
  private javax.swing.JButton makeRequestButton;
  private javax.swing.JButton viewRequestButton;
  private javax.swing.JLabel welcome;
  // End of variables declaration
}
```

DoctorView.java

This form is how doctors view all new requests and only requests they have opened if it is in progress

or closed.




```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */

package health;

import java.awt.HeadlessException;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import javax.swing.DefaultListModel;
import javax.swing.JOptionPane;

public final class DoctorView extends javax.swing.JFrame {
   Connection conn=null;
   ResultSet rs=null;
   PreparedStatement pst=null;
```

```
    int curRow=0;
    String username, userType;
    DefaultListModel model = new DefaultListModel();
    int index;
    int requestID;
    /**
     * Creates new form DoctorView
     * @param doctor
     */
    public DoctorView(String doctor) {
        initComponents();
        try {
            Class.forName("org.sqlite.JDBC");
            conn =
DriverManager.getConnection("jdbc:sqlite:C:\\Users\\Andrea\\Documents\\NetBeansProjects\\healt
h.sqlite");
            //JOptionPane.showMessageDialog (null, "Connected");
            Statement statement = conn.createStatement();
        }
        catch(ClassNotFoundException | SQLException e){
            JOptionPane.showMessageDialog(null, e);
        }
        setUsername(doctor);
        setUserType("Doctor");
        welcome.setText("Welcome Back, " + username + "!");
        requestsList.setVisible(false);
    }
     public String getUsername(){
        return this.username;
    }

    public void setUsername(String username){
        this.username = username;
    }

    public int getRequestID(){
        return this.requestID;
    }

    public void setRequestID(int requestID){
        this.requestID = requestID;
    }

    public String getUserType(){
        return this.userType;
    }

    public void setUserType(String userType){
```

```
    this.userType = userType;
}

/**
 * This method is called from within the constructor to initialize the form.
 * WARNING: Do NOT modify this code. The content of this method is always
 * regenerated by the Form Editor.
 */
@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated Code">
private void initComponents() {

    jLabel5 = new javax.swing.JLabel();
    jSeparator1 = new javax.swing.JSeparator();
    jScrollPane1 = new javax.swing.JScrollPane();
    requestsList = new javax.swing.JList();
    viewedRequests = new javax.swing.JLabel();
    welcome = new javax.swing.JLabel();
    jLabel6 = new javax.swing.JLabel();
    newRequestButton = new javax.swing.JButton();
    inProgressButton = new javax.swing.JButton();
    closeRequestButton = new javax.swing.JButton();
    openSelectedButton = new javax.swing.JButton();
    logout = new javax.swing.JButton();

    jLabel5.setFont(new java.awt.Font("Papyrus", 1, 14)); // NOI18N
    jLabel5.setForeground(new java.awt.Color(51, 51, 255));
    jLabel5.setText("HealthConnect");

    setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

    requestsList.setModel(new javax.swing.AbstractListModel() {
        String[] strings = { "Item 1", "Item 2", "Item 3", "Item 4", "Item 5" };
        public int getSize() { return strings.length; }
        public Object getElementAt(int i) { return strings[i]; }
    });
    requestsList.addListSelectionListener(new javax.swing.event.ListSelectionListener() {
        public void valueChanged(javax.swing.event.ListSelectionEvent evt) {
            requestsListValueChanged(evt);
        }
    });
    jScrollPane1.setViewportView(requestsList);

    viewedRequests.setFont(new java.awt.Font("Eras Demi ITC", 2, 18)); // NOI18N
    viewedRequests.setText("Requests");

    welcome.setFont(new java.awt.Font("Eras Demi ITC", 2, 24)); // NOI18N
    welcome.setText("Welcome Back");
```

```
jLabel6.setFont(new java.awt.Font("Papyrus", 1, 14)); // NOI18N
jLabel6.setForeground(new java.awt.Color(51, 51, 255));
jLabel6.setText("HealthConnect");

newRequestButton.setText("New");
newRequestButton.addActionListener(new java.awt.event.ActionListener() {
   public void actionPerformed(java.awt.event.ActionEvent evt) {
      newRequestButtonActionPerformed(evt);
   }
});

inProgressButton.setText("In Progress");
inProgressButton.addActionListener(new java.awt.event.ActionListener() {
   public void actionPerformed(java.awt.event.ActionEvent evt) {
      inProgressButtonActionPerformed(evt);
   }
});

closeRequestButton.setText("Closed");
closeRequestButton.addActionListener(new java.awt.event.ActionListener() {
   public void actionPerformed(java.awt.event.ActionEvent evt) {
      closeRequestButtonActionPerformed(evt);
   }
});

openSelectedButton.setText("Open Selected Request");
openSelectedButton.addActionListener(new java.awt.event.ActionListener() {
   public void actionPerformed(java.awt.event.ActionEvent evt) {
      openSelectedButtonActionPerformed(evt);
   }
});

logout.setText("Logout");
logout.addActionListener(new java.awt.event.ActionListener() {
   public void actionPerformed(java.awt.event.ActionEvent evt) {
      logoutActionPerformed(evt);
   }
});

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
   layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
   .addGroup(layout.createSequentialGroup()
      .addContainerGap()
      .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
         .addGroup(layout.createSequentialGroup()
```

```
                .addComponent(jSeparator1)
                .addContainerGap())
            .addGroup(layout.createSequentialGroup()
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .addComponent(viewedRequests, javax.swing.GroupLayout.PREFERRED_SIZE, 201,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addGroup(layout.createSequentialGroup()
                        .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 577,
javax.swing.GroupLayout.PREFERRED_SIZE)
                        .addGap(39, 39, 39)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
                            .addComponent(closeRequestButton, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                            .addComponent(inProgressButton, javax.swing.GroupLayout.DEFAULT_SIZE, 94,
Short.MAX_VALUE)
                            .addComponent(newRequestButton, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))))
                .addGap(0, 34, Short.MAX_VALUE))
            .addGroup(layout.createSequentialGroup()
                .addComponent(welcome, javax.swing.GroupLayout.PREFERRED_SIZE, 593,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                .addComponent(jLabel6, javax.swing.GroupLayout.PREFERRED_SIZE, 124,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(1, 1, 1))))
        .addGroup(layout.createSequentialGroup()
            .addGap(201, 201, 201)
            .addComponent(openSelectedButton, javax.swing.GroupLayout.PREFERRED_SIZE, 219,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
        .addGroup(layout.createSequentialGroup()
            .addGap(289, 658, Short.MAX_VALUE)
            .addComponent(logout)
            .addGap(31, 31, 31))
    );
    layout.setVerticalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(layout.createSequentialGroup()
                    .addContainerGap()
                    .addComponent(welcome))
                .addComponent(jLabel6))
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(logout)
            .addGap(18, 18, 18)
```

```
            .addComponent(jSeparator1, javax.swing.GroupLayout.PREFERRED_SIZE, 10,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
            .addComponent(viewedRequests, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
               .addGroup(layout.createSequentialGroup()
                  .addComponent(newRequestButton, javax.swing.GroupLayout.PREFERRED_SIZE, 38,
javax.swing.GroupLayout.PREFERRED_SIZE)
                  .addGap(79, 79, 79)
                  .addComponent(inProgressButton, javax.swing.GroupLayout.PREFERRED_SIZE, 39,
javax.swing.GroupLayout.PREFERRED_SIZE)
                  .addGap(74, 74, 74)
                  .addComponent(closeRequestButton, javax.swing.GroupLayout.PREFERRED_SIZE, 39,
javax.swing.GroupLayout.PREFERRED_SIZE))
               .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 296,
javax.swing.GroupLayout.PREFERRED_SIZE))
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
            .addComponent(openSelectedButton, javax.swing.GroupLayout.PREFERRED_SIZE, 53,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addContainerGap())
      );

      pack();
   }// </editor-fold>

   private void newRequestButtonActionPerformed(java.awt.event.ActionEvent evt) {
      // TODO add your handling code here:
      viewedRequests.setText("New Requests");
      requestsList.setVisible(true);
      String element;
      String sql ="select * from Request where Status=?";
      model.removeAllElements();
      element = "RID       Date                              Patient Username";
      model.addElement(element);
     try{
         pst=conn.prepareStatement(sql);
         pst.setString(1, "New");
         rs = pst.executeQuery();
         if(rs.next()){
            //JOptionPane.showMessageDialog(null, "Username and Password is correct");
            element = rs.getString("RID") + "       " + rs.getString("Date") + "          " +
rs.getString("PUsername");
            model.addElement(element);
            while (rs.next()){
```

```
            element = rs.getString("RID") + "       " + rs.getString("Date") + "          " +
rs.getString("PUsername");
            model.addElement(element);
        }
        requestsList.setModel(model);
        }
        else{
          JOptionPane.showMessageDialog(null, "No new requests created.");

        }
     }

     catch(SQLException | HeadlessException e){
        JOptionPane.showMessageDialog(null, e);}finally{
         try{
            rs.close();
               pst.close();
         }
         catch(SQLException e){
            JOptionPane.showMessageDialog(null, e);
         }
      }
   }

   private void inProgressButtonActionPerformed(java.awt.event.ActionEvent evt) {
      // TODO add your handling code here:
      viewedRequests.setText("In Progress Requests");
      requestsList.setVisible(true);
      String element;
      String sql ="select distinct Request.RID, Date, PUsername from Request, Message where
Request.RID = Message.RID and Request.Status=? and Message.DUsername=?";
      model.removeAllElements();
      element = "RID       Date                              Patient Username";
      model.addElement(element);
     try{
        pst=conn.prepareStatement(sql);
        pst.setString(1, "In Progress");
        pst.setString(2, username);
        rs = pst.executeQuery();
        if(rs.next()){
          //JOptionPane.showMessageDialog(null, "Username and Password is correct");
          element = rs.getString("RID") + "       " + rs.getString("Date") + "          " +
rs.getString("PUsername");
          model.addElement(element);
          while (rs.next()){
           element = rs.getString("RID") + "       " + rs.getString("Date") + "          " +
rs.getString("PUsername");
           model.addElement(element);
```

```
        }
        requestsList.setModel(model);
        }
        else{
           JOptionPane.showMessageDialog(null, "No requests are in progress");

        }
     }

     catch(SQLException | HeadlessException e){
        JOptionPane.showMessageDialog(null, e);}finally{
        try{
           rs.close();
              pst.close();
        }
        catch(SQLException e){
           JOptionPane.showMessageDialog(null, e);
        }
     }
  }

  private void openSelectedButtonActionPerformed(java.awt.event.ActionEvent evt) {
     // TODO add your handling code here:
     if(requestsList.getSelectedIndex() != -1)
     {
     String temp_requestID = requestsList.getSelectedValue().toString();
     temp_requestID = temp_requestID.substring(0,3);
     requestID = Integer.parseInt(temp_requestID);
     setRequestID(requestID);
        try{
           rs.close();
              pst.close();
        }
        catch(SQLException e){
           JOptionPane.showMessageDialog(null, e);
        }
     RequestConversation r = new RequestConversation(requestID, username, userType);
     dispose();
     r.setVisible(true);
     }
     else
        JOptionPane.showMessageDialog(null, "Please select a request");
  }

  private void closeRequestButtonActionPerformed(java.awt.event.ActionEvent evt) {
     // TODO add your handling code here:
     viewedRequests.setText("Closed Requests");
     requestsList.setVisible(true);
```

```
      String element;
      String sql ="select Distinct Request.RID, Date, PUsername from Request, Message where
Request.RID = Message.RID and Request.Status=? and Message.DUsername=?";
            model.removeAllElements();
            element = "RID       Date                              Patient Username";
     model.addElement(element);
     try{
        pst=conn.prepareStatement(sql);
        pst.setString(1, "Closed");
        pst.setString(2, username);
        rs = pst.executeQuery();
        if(rs.next()){
          //JOptionPane.showMessageDialog(null, "Username and Password is correct");
           element = rs.getString("RID") + "       " + rs.getString("Date") + "          " +
rs.getString("PUsername");
           model.addElement(element);
           while (rs.next()){
           element = rs.getString("RID") + "       " + rs.getString("Date") + "          " +
rs.getString("PUsername");
           model.addElement(element);
        }
        requestsList.setModel(model);
        }
        else{
          JOptionPane.showMessageDialog(null, "No requests have been closed.");

        }
     }

     catch(SQLException | HeadlessException e){
        JOptionPane.showMessageDialog(null, e);}finally{
         try{
            rs.close();
               pst.close();
         }
         catch(SQLException e){
           JOptionPane.showMessageDialog(null, e);
         }
      }
   }

   private void requestsListValueChanged(javax.swing.event.ListSelectionEvent evt) {
      // TODO add your handling code here:
      if (requestsList.getSelectedIndex() == -1){
        //Nothing was selected.  Do nothing
      }
      else
         index = requestsList.getSelectedIndex();
```

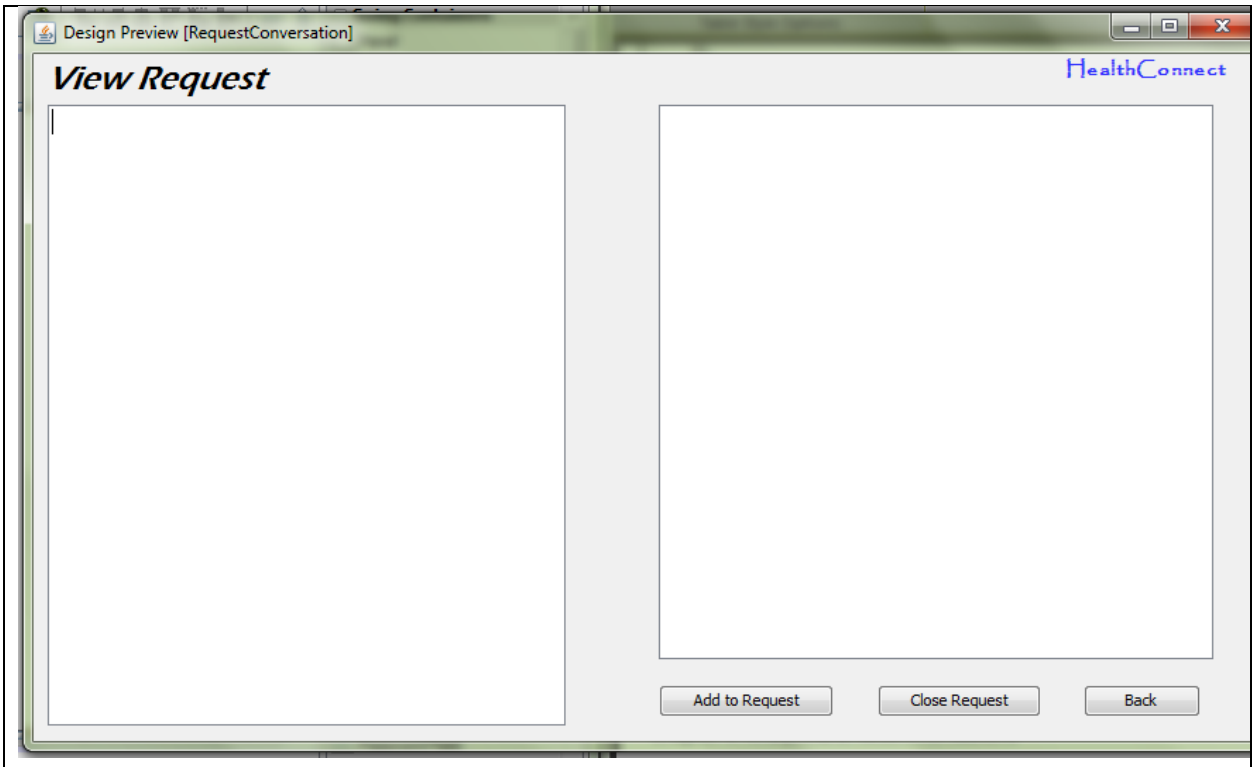
```
    }

    private void logoutActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
            try{
                rs.close();
                    pst.close();
            }
            catch(SQLException e){
                JOptionPane.showMessageDialog(null, e);
            }
        int pane = JOptionPane.showConfirmDialog(null, "Are you sure you want to logout?", "Logout",
JOptionPane.YES_NO_OPTION);
        if(pane==0){
        dispose();
        NewJFrame n = new NewJFrame();
        n.setVisible(true);}
    }

    /**
     * @param args the command line arguments
     */
    public static void main(String args[]) {
        /* Set the Nimbus look and feel */
        //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
        /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
         * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
         */
        try {
            for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
                if ("Nimbus".equals(info.getName())) {
                    javax.swing.UIManager.setLookAndFeel(info.getClassName());
                    break;
                }
            }
        } catch (ClassNotFoundException | InstantiationException | IllegalAccessException |
javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(DoctorView.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
        }
        //</editor-fold>
        final NewJFrame s = new NewJFrame();
        /* Create and display the form */
        java.awt.EventQueue.invokeLater(new Runnable() {
            @Override
            public void run() {
```

```
            new DoctorView(s.getUsername()).setVisible(true);
        }
    });
}

// Variables declaration - do not modify
private javax.swing.JButton closeRequestButton;
private javax.swing.JButton inProgressButton;
private javax.swing.JLabel jLabel5;
private javax.swing.JLabel jLabel6;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JSeparator jSeparator1;
private javax.swing.JButton logout;
private javax.swing.JButton newRequestButton;
private javax.swing.JButton openSelectedButton;
private javax.swing.JList requestsList;
private javax.swing.JLabel viewedRequests;
private javax.swing.JLabel welcome;
// End of variables declaration
}
```

RequestConversation.java

When a user opens a previously made request, this form appears and allows users to add to a request

if applicable.




```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */

package health;

import java.awt.HeadlessException;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.text.SimpleDateFormat;
import java.util.Date;
import javax.swing.JOptionPane;

public class RequestConversation extends javax.swing.JFrame {
   int requestNumber;
   String userID;
   String userType;
   Connection conn=null;
   ResultSet rs=null;
   PreparedStatement pst=null;
```

```
    String element;
    /**
     * Creates new form RequestConversation
     * @param new_requestID
     * @param new_userID
     * @param new_userType
     */
    public RequestConversation(int new_requestID, String new_userID, String new_userType) {
        initComponents();
        requestNumber = new_requestID;
        userID = new_userID;
        userType = new_userType;
        try {
            Class.forName("org.sqlite.JDBC");
            conn =
DriverManager.getConnection("jdbc:sqlite:C:\\Users\\Andrea\\Documents\\NetBeansProjects\\healt
h.sqlite");
            //JOptionPane.showMessageDialog (null, "Connected");
            Statement statement = conn.createStatement();
        }
        catch(ClassNotFoundException | SQLException e){
            JOptionPane.showMessageDialog(null, e);
        }
        String sql ="select * from Message where RID=?";
        try{
          pst=conn.prepareStatement(sql);
          String temp = Integer.toString(requestNumber);
          pst.setString(1, temp);
          rs = pst.executeQuery();
          currentRequest.setLineWrap(true);
          currentRequest.setWrapStyleWord(true);
          addToRequest.setLineWrap(true);
          addToRequest.setWrapStyleWord(true);
          if(rs.next()){
              element = rs.getString("TimeStamp");
              currentRequest.append(element + "\n");
              element = rs.getString("Message");
              currentRequest.append(element + "\n");
              while (rs.next()){
                  element = rs.getString("TimeStamp");
                  currentRequest.append(element + "\n");
                  element = rs.getString("Message");
                  currentRequest.append(element + "\n");
               }
          }
          else{
              JOptionPane.showMessageDialog(null, "No message added");
          }
```

```
        if("Doctor".equals(userType))
        {
        sql = "update Message set DUsername=? where RID =?";
        pst=conn.prepareStatement(sql);
        pst.setString(1, userID);
        pst.setString(2, temp);
        pst.execute();
        }

    }

    catch(HeadlessException | SQLException e){
        JOptionPane.showMessageDialog(null, e);}finally{
        try{
            rs.close();
                pst.close();
        }
        catch(SQLException e){
            JOptionPane.showMessageDialog(null, e);
        }
    }

    sql = "select Status from Request where RID =?";
    try{
        pst=conn.prepareStatement(sql);
        String temp = Integer.toString(requestNumber);
        pst.setString(1, temp);
        rs = pst.executeQuery();
        if("Closed".equals(rs.getString("Status")))
        {
            closeButton.setEnabled(false);
            addButton.setEnabled(false);
        }
    }
    catch(SQLException e){
        JOptionPane.showMessageDialog(null, e);
    }finally{
        try{
            rs.close();
                pst.close();
        }
        catch(SQLException e){
            JOptionPane.showMessageDialog(null, e);
        }
    }
}

/**
```

```
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        jScrollPane1 = new javax.swing.JScrollPane();
        currentRequest = new javax.swing.JTextArea();
        jScrollPane2 = new javax.swing.JScrollPane();
        addToRequest = new javax.swing.JTextArea();
        addButton = new javax.swing.JButton();
        closeButton = new javax.swing.JButton();
        backButton = new javax.swing.JButton();
        jLabel7 = new javax.swing.JLabel();
        jLabel1 = new javax.swing.JLabel();

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

        currentRequest.setColumns(20);
        currentRequest.setRows(5);
        jScrollPane1.setViewportView(currentRequest);

        addToRequest.setColumns(20);
        addToRequest.setRows(5);
        jScrollPane2.setViewportView(addToRequest);

        addButton.setText("Add to Request");
        addButton.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                addButtonActionPerformed(evt);
            }
        });

        closeButton.setText("Close Request");
        closeButton.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                closeButtonActionPerformed(evt);
            }
        });

        backButton.setText("Back");
        backButton.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                backButtonActionPerformed(evt);
            }
        });
```

```
    jLabel7.setFont(new java.awt.Font("Papyrus", 1, 14)); // NOI18N
    jLabel7.setForeground(new java.awt.Color(51, 51, 255));
    jLabel7.setText("HealthConnect");

    jLabel1.setFont(new java.awt.Font("Eras Demi ITC", 2, 24)); // NOI18N
    jLabel1.setText("View Request");

    javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
    getContentPane().setLayout(layout);
    layout.setHorizontalGroup(
       layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
       .addGroup(layout.createSequentialGroup()
          .addContainerGap()
          .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
             .addGroup(layout.createSequentialGroup()
                .addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED_SIZE, 462,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                .addComponent(jLabel7, javax.swing.GroupLayout.PREFERRED_SIZE, 124,
javax.swing.GroupLayout.PREFERRED_SIZE))
             .addGroup(layout.createSequentialGroup()
                .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 367,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(66, 66, 66)
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                   .addGroup(layout.createSequentialGroup()
                      .addComponent(addButton, javax.swing.GroupLayout.PREFERRED_SIZE, 124,
javax.swing.GroupLayout.PREFERRED_SIZE)
                      .addGap(31, 31, 31)
                      .addComponent(closeButton, javax.swing.GroupLayout.PREFERRED_SIZE, 116,
javax.swing.GroupLayout.PREFERRED_SIZE)
                      .addGap(30, 30, 30)
                      .addComponent(backButton, javax.swing.GroupLayout.PREFERRED_SIZE, 83,
javax.swing.GroupLayout.PREFERRED_SIZE))
                   .addComponent(jScrollPane2, javax.swing.GroupLayout.PREFERRED_SIZE, 393,
javax.swing.GroupLayout.PREFERRED_SIZE))
                .addGap(0, 19, Short.MAX_VALUE)))
          .addContainerGap())
    );
    layout.setVerticalGroup(
       layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
       .addGroup(layout.createSequentialGroup()
          .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
             .addComponent(jLabel7)
             .addGroup(layout.createSequentialGroup()
                .addGap(4, 4, 4)
```

```
                .addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED_SIZE, 26,
javax.swing.GroupLayout.PREFERRED_SIZE)))
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
              .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()
                .addComponent(jScrollPane2, javax.swing.GroupLayout.PREFERRED_SIZE, 393,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(18, 18, 18)
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                  .addComponent(closeButton)
                  .addComponent(addButton, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                  .addComponent(backButton))
                .addGap(6, 6, 6))
              .addComponent(jScrollPane1, javax.swing.GroupLayout.Alignment.TRAILING))
            .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
        );

        pack();
    }// </editor-fold>

    private void addButtonActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        int pane = JOptionPane.showConfirmDialog(null, "Are you sure you want to add your message to
the request?", "Add To Request", JOptionPane.YES_NO_OPTION);
        if(pane==0){
        String sql ="insert into Message (RID, DUsername, TimeStamp, Message) values (?, ?, ?, ?)";

        try{
          pst=conn.prepareStatement(sql);
          String temp = Integer.toString(requestNumber);
          pst.setString(1, temp);
          pst.setString(2, userID);
          Date date = new Date();
          String timestamp = new SimpleDateFormat("yyyy/MM/dd HH:mm:ss").format(date);
          pst.setString(3, timestamp);
          StringBuilder stringBuilder = new StringBuilder();
          stringBuilder.append("\n");
          stringBuilder.append(addToRequest.getText());
          stringBuilder.append("\n Added by ").append(userType).append(" ").append(userID);
          String finalString = stringBuilder.toString();
          pst.setString(4, finalString);
          pst.execute();
          JOptionPane.showMessageDialog(null, "Message added");
          sql = "update Request set Status='In Progress' where RID =?";
          pst=conn.prepareStatement(sql);
          temp = Integer.toString(requestNumber);
          pst.setString(1, temp);
```

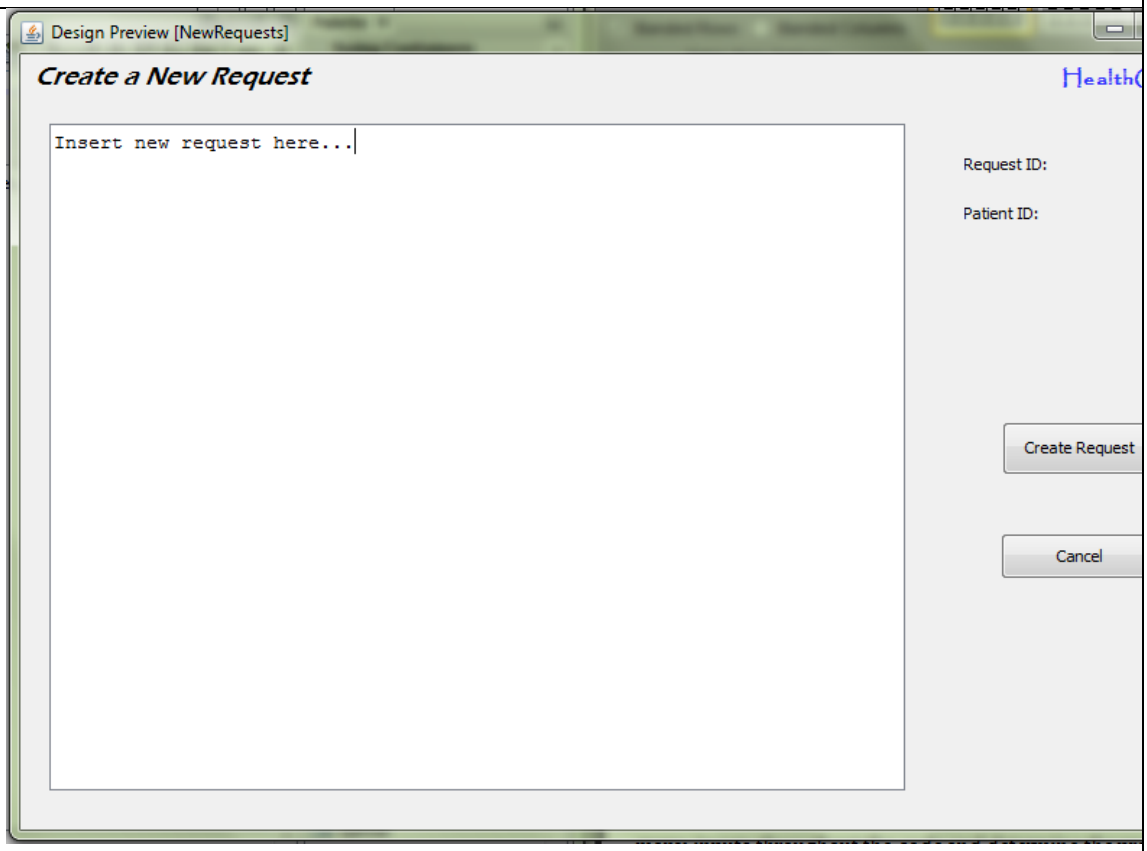
```
        pst.execute();
        currentRequest.append("\n");
        currentRequest.append(timestamp);
        currentRequest.append("\n");
        currentRequest.append(finalString);
        addToRequest.setText("");
        sql = "update Message set DUsername=? where RID =?";
        pst=conn.prepareStatement(sql);
        pst.setString(1, userID);
        pst.setString(2, temp);
        pst.execute();
        }

      catch(SQLException | HeadlessException e){
        JOptionPane.showMessageDialog(null, e);}finally{
         try{
            rs.close();
               pst.close();
         }
         catch(SQLException e){
            JOptionPane.showMessageDialog(null, e);
         }
      }
      }

   }

   private void closeButtonActionPerformed(java.awt.event.ActionEvent evt) {
      // TODO add your handling code here:
      int pane = JOptionPane.showConfirmDialog(null, "Are you sure you want to close the request?",
"Close Request", JOptionPane.YES_NO_OPTION);
      if(pane==0){
      String sql = "update Request set Status='Closed' where RID =?";
      try{
         pst=conn.prepareStatement(sql);
         String temp = Integer.toString(requestNumber);
         pst.setString(1, temp);
         pst.execute();
         JOptionPane.showMessageDialog(null, "Request has been closed.");
      }
      catch(SQLException | HeadlessException e){
         JOptionPane.showMessageDialog(null, e);
      }finally{
         try{
            rs.close();
               pst.close();
         }
         catch(SQLException e){
```

```
            JOptionPane.showMessageDialog(null, e);
        }
    }

    if("Doctor".equals(userType))
    {
        NewJFrame n = new NewJFrame();
        DoctorView d = new DoctorView(userID);
        d.setVisible(true);
        dispose();
    }
    else
    {
        NewJFrame n = new NewJFrame();
        PatientView p = new PatientView(userID);
        p.setVisible(true);
        dispose();
    }
    }
}

private void backButtonActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
        try{
            rs.close();
                pst.close();
        }
        catch(SQLException e){
            JOptionPane.showMessageDialog(null, e);
        }
    if("Doctor".equals(userType))
    {
        NewJFrame n = new NewJFrame();
        DoctorView d = new DoctorView(userID);
        d.setVisible(true);
        dispose();
    }
    else
    {
        NewJFrame n = new NewJFrame();
        PatientView p = new PatientView(userID);
        p.setVisible(true);
        dispose();
    }
}

/**
 * @param args the command line arguments
```

```
    */
   public static void main(String args[]) {
      /* Set the Nimbus look and feel */
      //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
      /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
       * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
       */
      try {
         for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
            if ("Nimbus".equals(info.getName())) {
               javax.swing.UIManager.setLookAndFeel(info.getClassName());
               break;
            }
         }
      } catch (ClassNotFoundException | InstantiationException | IllegalAccessException |
javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(RequestConversation.class.getName()).log(java.util.logging.Level.SE
VERE, null, ex);
      }
      //</editor-fold>
      NewJFrame n = new NewJFrame();
      final DoctorView d = new DoctorView(n.getUsername());
      /* Create and display the form */
      java.awt.EventQueue.invokeLater(new Runnable() {
         @Override
         public void run() {
            new RequestConversation(d.getRequestID(), d.getUsername(),
d.getUserType()).setVisible(true);
         }
      });
   }

   // Variables declaration - do not modify
   private javax.swing.JButton addButton;
   private javax.swing.JTextArea addToRequest;
   private javax.swing.JButton backButton;
   private javax.swing.JButton closeButton;
   private javax.swing.JTextArea currentRequest;
   private javax.swing.JLabel jLabel1;
   private javax.swing.JLabel jLabel7;
   private javax.swing.JScrollPane jScrollPane1;
   private javax.swing.JScrollPane jScrollPane2;
   // End of variables declaration
}
```

NewRequests.java

When a patient wants creates a new request, this form appears.




```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */

package health;

import java.awt.HeadlessException;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.text.SimpleDateFormat;
import java.util.Date;
import javax.swing.JOptionPane;
```

```
public class NewRequests extends javax.swing.JFrame {
   int requestNumber;
   String userID;
   String userType;
   Connection conn=null;
   ResultSet rs=null;
   PreparedStatement pst=null;
   int count = 100;

   /**
    * Creates new form NewRequests
    * @param new_userID
    */
   public NewRequests(String new_userID) {
      initComponents();
      userID = new_userID;
      try {
         Class.forName("org.sqlite.JDBC");
         conn =
DriverManager.getConnection("jdbc:sqlite:C:\\Users\\Andrea\\Documents\\NetBeansProjects\\healt
h.sqlite");
         //JOptionPane.showMessageDialog (null, "Connected");
         Statement statement = conn.createStatement();
         String sql ="select RID from Request";
         rs = statement.executeQuery(sql);
         while(rs.next())
            count++;
      }
      catch(ClassNotFoundException | SQLException e){
         JOptionPane.showMessageDialog(null, e);
      }finally{
         try{
            rs.close();
         }
         catch(SQLException e){
            JOptionPane.showMessageDialog(null, e);
         }
      }

      RequestID.setText("RequestID: "+ count);
      PatientID.setText("PatientID: "+ userID);
      jTextArea1.setLineWrap(true);
      jTextArea1.setWrapStyleWord(true);

   }

   /**
    * This method is called from within the constructor to initialize the form.
```

```
 * WARNING: Do NOT modify this code. The content of this method is always
 * regenerated by the Form Editor.
 */
@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated Code">
private void initComponents() {

    jLabel6 = new javax.swing.JLabel();
    Title = new javax.swing.JLabel();
    jScrollPane1 = new javax.swing.JScrollPane();
    jTextArea1 = new javax.swing.JTextArea();
    createButton = new javax.swing.JButton();
    CancelButton = new javax.swing.JButton();
    RequestID = new javax.swing.JLabel();
    PatientID = new javax.swing.JLabel();
    jLabel7 = new javax.swing.JLabel();

    jLabel6.setFont(new java.awt.Font("Papyrus", 1, 14)); // NOI18N
    jLabel6.setForeground(new java.awt.Color(51, 51, 255));
    jLabel6.setText("HealthConnect");

    setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

    Title.setFont(new java.awt.Font("Eras Demi ITC", 2, 18)); // NOI18N
    Title.setText("Create a New Request");

    jTextArea1.setColumns(20);
    jTextArea1.setRows(5);
    jTextArea1.setText("Insert new request here...");
    jScrollPane1.setViewportView(jTextArea1);

    createButton.setText("Create Request");
    createButton.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            createButtonActionPerformed(evt);
        }
    });

    CancelButton.setText("Cancel");
    CancelButton.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            CancelButtonActionPerformed(evt);
        }
    });

    RequestID.setText("Request ID:");

    PatientID.setText("Patient ID:");
```

```
    jLabel7.setFont(new java.awt.Font("Papyrus", 1, 14)); // NOI18N
    jLabel7.setForeground(new java.awt.Color(51, 51, 255));
    jLabel7.setText("HealthConnect");

    javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
    getContentPane().setLayout(layout);
    layout.setHorizontalGroup(
       layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
       .addGroup(layout.createSequentialGroup()
          .addGap(20, 20, 20)
          .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 592,
javax.swing.GroupLayout.PREFERRED_SIZE)
          .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
             .addGroup(layout.createSequentialGroup()
                .addGap(41, 41, 41)
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                   .addComponent(RequestID, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                   .addComponent(PatientID, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
                .addContainerGap())
             .addGroup(layout.createSequentialGroup()
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 66,
Short.MAX_VALUE)
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
                   .addComponent(CancelButton, javax.swing.GroupLayout.PREFERRED_SIZE, 110,
javax.swing.GroupLayout.PREFERRED_SIZE)
                   .addComponent(createButton))
                .addContainerGap(56, Short.MAX_VALUE))))
       .addGroup(layout.createSequentialGroup()
          .addContainerGap()
          .addComponent(Title, javax.swing.GroupLayout.PREFERRED_SIZE, 633,
javax.swing.GroupLayout.PREFERRED_SIZE)
          .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
          .addComponent(jLabel7, javax.swing.GroupLayout.PREFERRED_SIZE, 124,
javax.swing.GroupLayout.PREFERRED_SIZE))
    );
    layout.setVerticalGroup(
       layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
       .addGroup(layout.createSequentialGroup()
          .addGap(63, 63, 63)
          .addComponent(RequestID, javax.swing.GroupLayout.PREFERRED_SIZE, 27,
javax.swing.GroupLayout.PREFERRED_SIZE)
          .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
          .addComponent(PatientID, javax.swing.GroupLayout.PREFERRED_SIZE, 28,
javax.swing.GroupLayout.PREFERRED_SIZE)
```

```
            .addGap(130, 130, 130)
            .addComponent(createButton, javax.swing.GroupLayout.PREFERRED_SIZE, 37,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(40, 40, 40)
            .addComponent(CancelButton, javax.swing.GroupLayout.DEFAULT_SIZE, 32,
Short.MAX_VALUE)
            .addGap(173, 173, 173))
         .addGroup(layout.createSequentialGroup()
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
               .addComponent(Title, javax.swing.GroupLayout.PREFERRED_SIZE, 30,
javax.swing.GroupLayout.PREFERRED_SIZE)
               .addComponent(jLabel7))
            .addGap(18, 18, 18)
            .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 461,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
      );

      pack();
   }// </editor-fold>

   private void createButtonActionPerformed(java.awt.event.ActionEvent evt) {
      // TODO add your handling code here:
      int pane = JOptionPane.showConfirmDialog(null, "Are you sure you want to create the request?",
"Create Request", JOptionPane.YES_NO_OPTION);
      if(pane==0){
      String sql ="insert into Message (RID, DUsername, TimeStamp, Message) values (?, ?, ?, ?)";

      try{
         pst=conn.prepareStatement(sql);
         String temp = Integer.toString(count);
         pst.setString(1, temp);
         pst.setString(2, null);
         Date date = new Date();
         String timestamp = new SimpleDateFormat("yyyy/MM/dd HH:mm:ss").format(date);
         pst.setString(3, timestamp);
         StringBuilder stringBuilder = new StringBuilder();
         stringBuilder.append("\n");
         stringBuilder.append(jTextArea1.getText());
         stringBuilder.append("\n Added by ").append("Patient").append(" ").append(userID);
         String finalString = stringBuilder.toString();
         pst.setString(4, finalString);
         pst.execute();
         JOptionPane.showMessageDialog(null, "Message created");
         sql = "insert into Request (RID, PUsername, Date, Status) values (?, ?, ?, ?)";
         pst=conn.prepareStatement(sql);
         pst.setString(1, temp);
         pst.setString(2, userID);
```

```
            pst.setString(3, timestamp);
            pst.setString(4, "New");
            pst.execute();
            }

        catch(SQLException | HeadlessException e){
            JOptionPane.showMessageDialog(null, e);}finally{
            try{
                rs.close();
                    pst.close();
            }
            catch(SQLException e){
                JOptionPane.showMessageDialog(null, e);
            }
        }
        PatientView j = new PatientView(userID);
        j.setVisible(true);
        dispose();
        }
    }

    private void CancelButtonActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        Profile p = new Profile(userID);
        p.setVisible(true);
        dispose();
    }

    /**
     * @param args the command line arguments
     */
    public static void main(String args[]) {
        /* Set the Nimbus look and feel */
        //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
        /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
         * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
         */
        try {
            for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
                if ("Nimbus".equals(info.getName())) {
                    javax.swing.UIManager.setLookAndFeel(info.getClassName());
                    break;
                }
            }
        } catch (ClassNotFoundException | InstantiationException | IllegalAccessException |
javax.swing.UnsupportedLookAndFeelException ex) {
```

```
java.util.logging.Logger.getLogger(NewRequests.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
        }
        //</editor-fold>
        NewJFrame n = new NewJFrame();
        final PatientView v = new PatientView(n.getUsername());
        /* Create and display the form */
        java.awt.EventQueue.invokeLater(new Runnable() {
            @Override
            public void run() {
                new NewRequests(v.getUsername()).setVisible(true);
            }
        });
    }

    // Variables declaration - do not modify
    private javax.swing.JButton CancelButton;
    private javax.swing.JLabel PatientID;
    private javax.swing.JLabel RequestID;
    private javax.swing.JLabel Title;
    private javax.swing.JButton createButton;
    private javax.swing.JLabel jLabel6;
    private javax.swing.JLabel jLabel7;
    private javax.swing.JScrollPane jScrollPane1;
    private javax.swing.JTextArea jTextArea1;
    // End of variables declaration
}
```

.

Chapter 4: Analysis and Recommendations

Validation and Testing

When performing unit testing, we implemented structural techniques where we test many inputs throughout the code and determine the proper outputs. Some of the components are statement coverage, which is testing that covers only the true conditions. Furthermore, condition testing is the process that we should follow throughout the rest of the application. An example, we will be testing to see if usernames and passwords are valid. We also used expression testing, in which the application is tested for different values with regular expression. An example includes testing case insensitive, and multiline to see if we run into possible SQL injections.

When using component testing we considered components from the project test plan, and system requirements. Project test plan included testing values in determine how many test cases to build. System requirements use a detailed and structured use case model which is used to determine the require classes.

When system testing, we thought about the components and methods in the prototype. Some of the components we used within HealthConnect are included in the requirements testing technique and manual support testing technique. Requirements testing techniques checks to ensure that the system performs correctly. For example, we used a user profile to verify if the user is a doctor or patient. Manual support testing technique tests all of the functions performed by the user while updating the database.

Results

Through a lot of hard work and time spent, we created a working prototype of a HealthConnect application. We initialize the user accounts whether doctor or patient. From there, patients are able to create or view their own requests. Doctors are also able to open and respond to requests. The prepareStatement functions prevent SQL injection and dialogue boxes are used for human error.

Space and Time Complexity Evaluation

The database itself is currently only 384KB and the Netbeans complete project is 560KB. However, Netbeans does use a little over 600,000K on memory to run. Because there are only a few 'while' statements within the program, the time is varied based on the size of the database. For the purpose of a working prototype, the program does not take a lot of space or time. All unnecessary computations were avoided in order to do the minimum requirements. Efficient programs avoid creating and copying data when it is unneeded.

Recommendations and Improvements

The database changes should be well documented. We would not recommend SQLite as it does not handle advance SQL features well. For example, it does not allow multiple primary keys in one table and foreign keys are hard to manipulate. Instead, we recommend the database already built into netbeans. We also suggest better documentation of the requirements to maintain project growth.

Chapter 5: Conclusion

Deliverables

All goals of the working prototype were met. Patients and doctors can log into the system. Patients are able to view their own requests or create a new one. Doctors are also able to respond to requests and everything is stored in the database. We were able to complete the project by the deadline. Some of the challenges were limited experience. We have not had a lot of experience in GUI. The working HealthConnect app could not be completed but the prototype is available. The prototype is a Netbeans java projects and the database uses SQLite available from Mozilla Firefox.

References

"5 Step Procedure for Android Deployment with NetBeans IDE (Geertjan's Blog)." *5 Step Procedure for Android Deployment with NetBeans IDE (Geertjan's Blog)*. N.p., n.d. Web. 10 Dec. 2013.

"NetBeans IDE." *Welcome to NetBeans*. N.p., n.d. Web. 10 Dec. 2013.

"Preventing SQL Injection in Java." - *OWASP*. N.p., n.d. Web. 10 Dec. 2013.

"SQLite Home Page." *SQLite Home Page*. N.p., n.d. Web. 10 Dec. 2013.

"Top Free in Health & Fitness - Android Apps on Google Play." *Top Free in Health & Fitness - Android Apps on Google Play*. N.p., n.d. Web. 10 Dec. 2013.