Team 18 Unit (White Box) Testing Report

Enter pain data use case:

Actors: Patient

Pre-conditions (Entry Conditions): Patient login password entered and verified. Accesses the "Symptoms" tab to send information to the physician.

Post-conditions (Exit Conditions): Results are documented and saved to the database and/or sent to physician for immediate response. Prompted with a message verifying the information was sent.

Scenario: The patient opens the application on any device that has access to the internet. The patient is prompted to login and enter credential (this scenario does not involve the emergency option as that option is an exception case where login credentials are overlooked). Once login credentials are validated, the patient may choose the option to send symptoms to their physician that they are feeling. Sending this information would also send the severity of the symptom based on a simple scale for understanding. Patient may also send a note to the physician that is related to the symptom for further understanding. Patient then verifies the information is as correct to their knowledge and sends the information. A dialogue of "Information Sent" will be presented to the patient to verify the physician has received the information.

Exceptions:

Invalid Login/password: An error will pop up to the patient and prompt for the information be re-entered into the system.

Exiting mid-entry/computer malfunction: Survey may not be saved properly.

Emergency option chosen: If the emergency option is chosen an ambulance will be summoned to get the patient to the nearest hospital.

Use-Case Relationships: Includes "Login" and "Access Patient conditions".

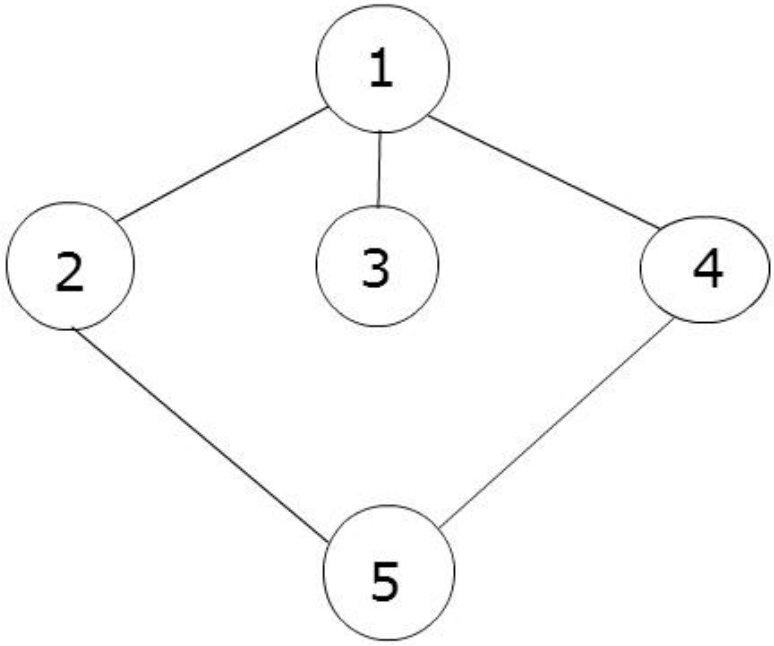
Path Test:

Actual code

```
/************************** Controls Events1 ***********************************/
     //When Changing Slider 1 Cursor...do this
1.     slider1.addChangeListener(new ChangeListener() {
         @Override
         public void stateChanged(ChangeEvent e) {
           txt1.setText(String.valueOf(slider1.getValue()));
         }
       });
       //When Press Enter After Change...do this
2.    txt1.addActionListener(new ActionListener() {
         @Override
         public void actionPerformed(ActionEvent e) {
           try
           {
             slider1.setValue(Integer.parseInt(txt1.getText()));
           }
           catch(Exception ex)
           {
3.             txt1.setText("ERROR");
            txt1.setToolTipText("Set Value in Range between 0 - 10 ") ;
           }
         }
       });
4.     this.addFocusListener(new FocusListener() {
         @Override
         public void focusLost(FocusEvent e){
         }

         @Override
         public void focusGained(FocusEvent e) {
5.           txt1.setText(String.valueOf(slider1.getValue()));

         }
       });
       /*****************************************************************/
```



Cyclomatic complexity = 5 – 5 + 2 = 2
Path 1: 1, 2, 5 test case (patient uses jslider)
Path 2: 1,3 test case (patient hovers over tool tip for help)
Path 3: 1,4, 5 test case (patient uses text box)

Viewing patient pain data use case:

Actors: Doctor

Pre-conditions (Entry Conditions): Doctor login and password entered and verified. Chooses a patient to view symptoms and information.

Post-conditions (Exit Conditions):

Scenario: Doctor accesses information for each patient that has set an appointment. Doctor is able to write notes and additional information with the information presented during the appointment.

Exceptions:

Invalid login/password: An error will show to the Doctor prompting the information be re-entered into the system.

Exiting mid-view/ computer malfunction: Current view location will be saved, this will allow immediate access to the patient's information once the connection has been restored to the application.

Use-Case Relationships: Includes "Login", "Suggest Treatment", and "Access Patient Conditions",

Path Test:
Actual code

```
    /****************** Labels Properties *********************************/
1.  lbl1 = new JLabel("Pain");
    lbl2 = new JLabel("Tiredness");
    lbl3 = new JLabel("Nausea");
    lbl4 = new JLabel("Depression");
    lbl5 = new JLabel("Anxiety");
    lbl6 = new JLabel("Drowsiness");
    lbl7 = new JLabel("Appetite");
    lbl8 = new JLabel("Wellbeing");
    lbl9 = new JLabel("Shortness of breath");
    lbl10 = new JTextField(4);
    lbl11 = new JLabel("Other");


2.      static class Action implements ActionListener {
          public void actionPerformed(ActionEvent e) {
             String nameData = field.getText();
             @SuppressWarnings("deprecation")
                        String passwordData = p.getText();
3.            if (nameData.equals(""))
                JOptionPane.showMessageDialog(null,"Please enter User Name in the Text Box.");
4.            if (passwordData.equals(""))
                JOptionPane.showMessageDialog(null,"Please enter Password in the Text Box.");
              else{



5.              switch (nameData) {


6.          case "Doctor1": if (passwordData.equals("Doctor1password")){
```

7. JOptionPane.showMessageDialog(null,"Success1");

frame.dispose();

doctor1();

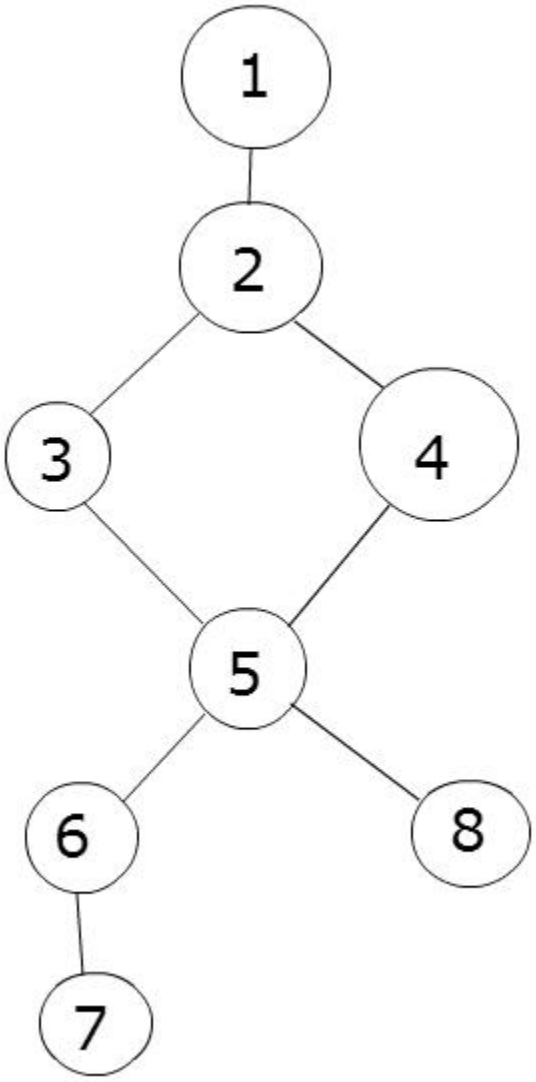
}

8. else{

JOptionPane.showMessageDialog(null,"Failed");

}

break;



Cyclomatic complexity = 8 – 8 + 2 = 2
Path 1: 1, 2, 3, 5, 6, 7 test case (Login needs user name)
Path 2: 1, 2, 3, 5, 8 test case (Doctor logs in to patient screen)
Path 3: 1, 2, 4, 5, 6, 7 test case (Login needs password)
Path 4: 1, 2, 4, 5, 8 test case (Doctor fails to login to patient screen)

Help button use case:

Actors: Patient and Doctor

Pre-conditions (Entry Conditions): User needs access to the program. The user can click the help button to view additional guides to the program.

Post-Conditions (Exit Conditions): Patient is able to close the help screen and continue inputting information or log out of the program. The help screen will not interfere with the information previously added to the survey.

Scenario: The user opens the application on any devices that has access to the internet. The user is then able to access the help button which will open in a new window and provide basic guides to access the application. The guide will contain useful information for both patients and doctors. The help guide is completely independent from the application and will not interfere with the surveys being done. User can exit out of the help guide whenever necessary.

Exceptions:

Help Directory Error: An error will display to the user the help button is not properly displaying, and will prompt user to restart the program.

Exiting mid-view/computer malfunction: The help screen will close when the program closes.

Use-Case Relationships: Includes “Login” and “Help” screen display.

Path Test:
Actual code

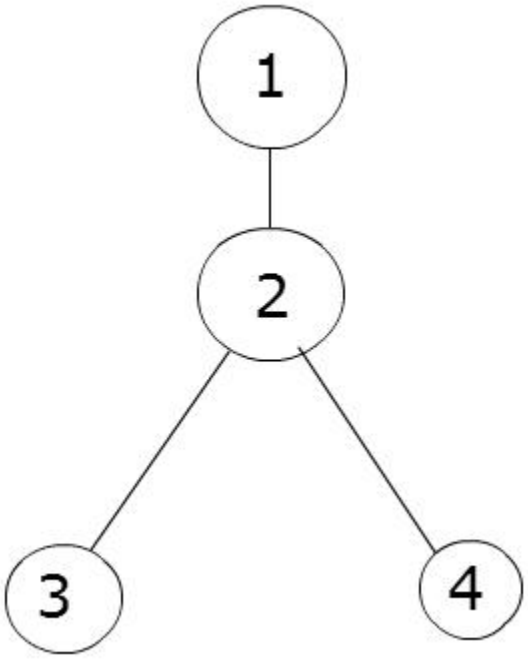
```
    /**************************************Help Button*******************************************/
1.      panel11 = new JPanel();
        c.add(panel11);
        b = new JButton("Help");
        b.setSize(b.getPreferredSize());
2.      b.addActionListener(new Action());
        panel11.add(b);
```

```
3.      b.addActionListener(new ActionListener() {
         public void actionPerformed(ActionEvent e) {
         panel12 = new JPanel();
4.         c.add(panel12);
           panel12.add(txthelp)
          }
        });
      /*************************************************************************************************/
```

Cyclomatic complexity = 4 – 4 + 2 = 2

Path 1: 1, 2, 3 test case (button isn't clicked)

Path 2: 1, 2, 4 test case (button is clicked for help box)



Emergency button use case:

Actors: Patient and Doctor

Pre-conditions (Entry Conditions): Patient username and password logged in and verified.

Post-conditions (Exit Conditions): Doctor and ambulance contacted with patient information.

Scenario: Patient is in critical need of help. They enter their username and password to access the screen of the interface that allows them to press the Emergency Button. After pressing this button, there is a Confirm and a Cancel Button. The Cancel Button is available if it was

accidental. However, if the Patient needs help, they the Confirm Button, and their Doctor and an ambulance are contacted with the Patient's information.

Exceptions:

Invalid login/password: An error stating that the password or the username entered was incorrect and/or not found.

Incomplete Submission/computer malfunction: Submission advice will be temporarily saved for the Doctor to access and edit. Otherwise user will be prompted they are leaving the document without it being completed.

Use-Case Relationships: Includes "Login" and aggregate the "Cancel" Button and "Confirm" Button.

Path Test:
Actual code

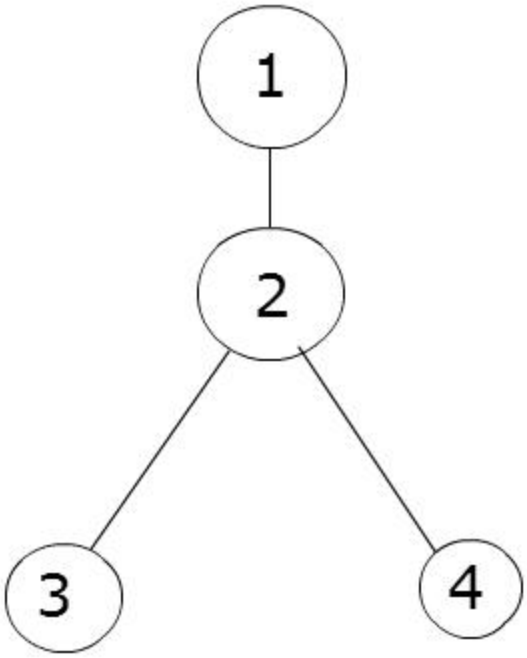
```
    /*************************************Emergency Button*****************************************/
1.   panel113 = new JPanel();
     c.add(panel13);
     b = new JButton("Emergency");
     b.setSize(b.getPreferredSize());
2.   b.addActionListener(new Action());
     panel13.add(b);

3.     b.addActionListener(new ActionListener() {
         public void actionPerformed(ActionEvent e) {
4.       panel14 = new JPanel();
         panel14.add(txtemergency)
         }
      });
     /*************************************************************************************************/
```

Cyclomatic complexity = 4 – 4 + 2 = 2

Path 1: 1, 2, 3 test case (button isn't clicked)

Path 2: 1, 2, 4 test case (button is clicked for emergency services)


1
2
3
4