LONDON METROPOLITAN UNIVERSITY

Islington college
(इस्लिङ्टन कलेज)

Module Code & Module Title

CS4001NI Programming

Assessment Weightage & Type

30% Individual Coursework

Year and Semester

2018-19 Autumn

Student Name: Abhishek Gupta

London Met ID: 18029801

College ID: NP01CP4A180067

Assignment Due Date: 19th Apr. 2019

Assignment Submission Date: 19th Apr. 2019

I confirm that I understand my coursework needs to be submitted online via Google Classroom under the relevant module page before the deadline in order for my assignment to be accepted and marked. I am fully aware that late submissions will be treated as non-submission and a marks of zero will be awarded.

Table of Contents

List of Figures

Introduction

This is the second project of java which was given to us on the 20th week, which is to be submitted till the 24th week. The main purpose of performing this project is to add a class to the previous project that was done in the previous semester to make a GUI (Graphical User Interface) for the system that stores details of the developers, which are senior and junior developers. The purpose of this project is also to make main method which will be tested using the command prompt. There is a class called RigoTechnology which contains of the GUI coding as well as the other parts of the project. This class contains of various methods which accept the information by the user and also displays it. There are seven buttons used which have their own work. The RigoTechnology class also connects this project to the other three classes created previously and gathers information for various use. In this way, the whole project was completed using the various tasks that was assigned to us. (Sen, 2018-19)

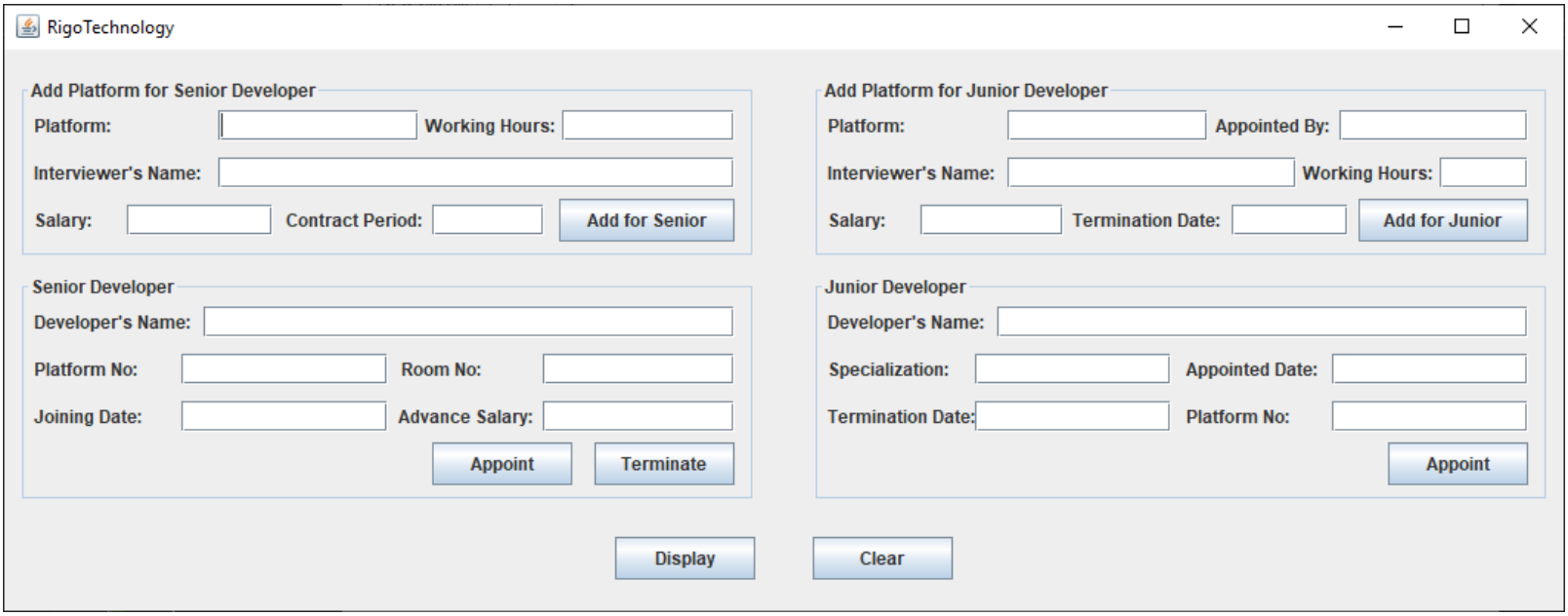

Figure 1: GUI representation of RigoTechnology

Class Diagram

Figure 2: Class Diagram of RigoTechnology

Pseudocode

The pseudocodes of the methods of class RigoTechnology are given below:

3.1 Method main():

```
CALL main()
     CREATE OBJECT rigoTechnology().gui()
```

3.2 Method gui():

```
CALL gui()

     ASSIGN JFrame for frame
     SET TITLE for frame
     SET OPAQUE for frame TO true
     SET EXIT ON CLOSE for frame

     ASSIGN JPanel for senDevPltPanel
     SET THE BOUND for senDevPltPanel
     SET OPAQUE for senDevPltPanel TO true
     ADD TO frame

     ASSIGN JLabel for lblSenPltfrm
     SET THE BOUND for lblSenPltfrm
     SET OPAQUE for lblSenPltfrm TO true
     ADD TO senDevPltPanel

     for lblSenIntrvwr, same as above
     ADD TO senDevPltPanel
```

```
for lblSenSalry, same as above
ADD TO senDevPltPanel

for lblSenWrkHr, same as above
ADD TO senDevPltPanel

for lblSenContPrd, same as above
ADD TO senDevPltPanel

ASSIGN JTextField for txtSenPltfrm
SET THE BOUND for txtSenPltfrm
SET OPAQUE for txtSenPltfrm TO true
ADD TO senDevPltPanel

for txtSenIntrvwr, same as above
ADD TO senDevPltPanel

for txtSenSalry, same as above
ADD TO senDevPltPanel

for txtSenWrkHr, same as above
ADD TO senDevPltPanel

for txtSenContPrd, same as above
ADD TO senDevPltPanel

ASSIGN JButton for btnAddS
SET THE BOUND for btnAddS
SET OPAQUE for btnAddS TO true
ADD TO senDevPltPanel
```

```
ASSIGN JPanel for senDev
SET THE BOUND for senDev
SET OPAQUE for senDev TO true
ADD TO frame

ASSIGN JLabel for lblSenDevName
SET THE BOUND for lblSenDevName
SET OPAQUE for lblSenDevName TO true
ADD TO senDev

for lblSenAdvsalry, same as above
ADD TO senDev

for lblSenPltfrmNum, same as above
ADD TO senDev

for lblSenStaffRoomNo, same as above
ADD TO senDev

for lblSenJoinDate, same as above
ADD TO senDev

ASSIGN JTextField for txtSenDevName
SET THE BOUND for txtSenDevName
SET OPAQUE for txtSenDevName TO true
ADD TO senDev

for txtSenAdvsalry, same as above
ADD TO senDev

for txtSenPltfrmNum, same as above
ADD TO senDev
```

```
for txtSenStaffRoomNo, same as above
ADD TO senDev

for txtSenJoinDate, same as above
ADD TO senDev

ASSIGN JButton for btnSenAppoint
SET THE BOUND for btnSenAppoint
SET OPAQUE for btnSenAppoint TO true
ADD TO senDev

for btnTerminate, same as above
ADD TO senDev


ASSIGN JPanel for junDevPltPanel
SET THE BOUND for junDevPltPanel
SET OPAQUE for junDevPltPanel TO true
ADD TO frame

ASSIGN JLabel for lblJunPltfrm
SET THE BOUND for lblJunPltfrm
SET OPAQUE for lblJunPltfrm TO true
ADD TO junDevPltPanel

for lblJunSalry, same as above
ADD TO junDevPltPanel

for lblJunIntrvwr, same as above
ADD TO junDevPltPanel
```

```
for lblJunWrkHr, same as above
ADD TO junDevPltPanel

for lblJunAppointBy, same as above
ADD TO junDevPltPanel

for lblJunTerminate, same as above
ADD TO junDevPltPanel

ASSIGN JTextField for txtJunPltfrm
SET THE BOUND for txtJunPltfrm
SET OPAQUE for txtJunPltfrm TO true
ADD TO junDevPltPanel

for txtJunSalry, same as above
ADD TO junDevPltPanel

for txtJunIntrvwr, same as above
ADD TO junDevPltPanel

for txtJunWrkHr, same as above
ADD TO junDevPltPanel

for txtJunAppointBy, same as above
ADD TO junDevPltPanel

for txtJunTerminate, same as above
ADD TO junDevPltPanel

ASSIGN JButton for btnAddJ
SET THE BOUND for btnAddJ
SET OPAQUE for btnAddJ TO true
```

```
ADD TO junDevPltPanel


ASSIGN JPanel for junDev
SET THE BOUND for junDev
SET OPAQUE for junDev TO true
ADD TO frame

ASSIGN JLabel for lblJunDevName
SET THE BOUND for lblJunDevName
SET OPAQUE for lblJunDevName TO true
ADD TO junDev

for lblJunAppointDate, same as above
ADD TO junDev

for lblJunSpecialization, same as above
ADD TO junDev

for lblJunTermDate, same as above
ADD TO junDev

for lblJunPltfrmNum, same as above
ADD TO junDev

ASSIGN JTextField for txtJunDevName
SET THE BOUND for txtJunDevName
SET OPAQUE for txtJunDevName TO true
ADD TO junDev

for txtJunDevName, same as above
ADD TO junDev
```

```
    for txtJunAppointDate, same as above
    ADD TO junDev

    for txtJunSpecialization, same as above
    ADD TO junDev

    for txtJunTermDate, same as above
    ADD TO junDev

    for txtJunPltfrmNum, same as above
    ADD TO junDev

    ASSIGN JButton for btnJunAppoint
    SET THE BOUND for btnJunAppoint
    SET OPAQUE for btnJunAppoint TO true
    ADD TO junDev

    ASSIGN JButton for btnDisplay
    SET THE BOUND for btnDisplay
    SET OPAQUE for btnDisplay TO true
    ADD TO frame

    for btnClear, same as above
    ADD TO frame
```

3.3 Method actionPerformed:

```
CALL actionPerformed (actionEvent e)
    DO
        IF e IS instance of btnAddS THEN
            TRY
```

```
            DECLARE seniorPlatform AS txtSenPltfrm as String
            DECLARE seniorInterviewer AS txtSenIntrvwr as
            String
            DECLARE seniorWorkingHr AS txtSenWrkHr as int
            DECLARE seniorSalary AS txtSenSalry as int
            DECLARE seniorContractPrd AS txtSenContPrd as int

           IF seniorWorkingHr IS LESS THAN 12 OR seniorSalary
           IS LESS THAN 5000 OR seniorContractPrd IS LESS
           THAN 3 THEN
                THROW EXCEPTION
                 OUTPUT message: Working Hours, Salary and
                 Contract Period should be more than 12
                 hours, Rs. 5000 and 3 months respectively.
            ENDIF

            DECLARE seniorPltfrmInfo AND CALL SeniorDeveloper
            (input seniorPlatform, seniorInterviewer,
            seniorSalary, seniorWorkingHr, seniorContractPrd)
            ADD seniorPltfrmInfo TO list
            OUTPUT message: Senior developer platform added.
       CATCH NumberFormatException AS emptyNumExc
            OUTPUT message: Please fill all the text fields.
            Salary, Working Hours and Contract Period must be
            in number.
       CATCH Exception AS allExc
            OUTPUT above thrown exception
       ENDIF
  ENDDO

  DO
       IF e IS instance of btnSenAppoint THEN
```

```
TRY
    DECLARE seniorDevName AS txtSenDevName as
    String
    DECLARE seniorJoinDate AS txtSenJoinDate as
    String
    DECLARE seniorAdvSalary AS txtSenAdvsalry as
    int
    DECLARE seniorStaffRoomNo AS
    txtSenStaffRoomNo as String
    DECLARE seniorPltfrmNum AS txtSenPltfrmNum
    as int

    IF seniorPltfrmNum IS GREATER THAN 0 OR
    seniorPltfrmNum IS LESS THAN OR EQUALS TO
    the size of list THEN
        IF     list.get(seniorPltfrmNum)     is
        INSTANCE OF SeniorDeveloper THEN
            DECLARE        seniorDev        AS
            list.get(seniorPltfrmNum)
            IF  seniorDev.getSalary()  is  less
            than seniorAdvSalary THEN
                THROW EXCEPTION
                OUTPUT    message:    Advance
                salary cannot be more than
                Salary.
            ENDIF
            CALL hireSeniorDeveloper (input
            seniorDevName, seniorJoinDate,
            seniorAdvSalary,
            seniorStaffRoomNo)
            OUTPUT message: Senior developer
added appointed.
```

```
                    ENDIF
               ENDIF
          CATCH NumberFormatException AS emptyNumExc
               OUTPUT message: Please fill all the text
               fields. Advance Salary and Platform Number
               must be in number.
          CATCH IndexOutOfBoundsException pltfrmExc
               OUTPUT message: Platform Number not
               available. Please enter a valid Platform
               Number.
          CATCH Exception AS allExc
               OUTPUT above thrown exception
     ENDIF
ENDDO

DO
     IF e IS instance of btnTerminate THEN
          TRY
               DECLARE seniorPltfrmNum AS txtSenPltfrmNum
               as int
               DECLARE seniorDevObj AS
               list.get(seniorPltfrmNum)
               IF seniorDevObj.getAppointed() IS EQUALS TO
               true THEN
                    CALL contractTermination
                    OUTPUT message: Senior Developer has
                    been terminated successfully.
               ELSE
                    OUTPUT message: No developer appointed
                    in this Platform. Please enter a valid
                    Platform Number.
               ENDIF
```

```
            CATCH IndexOutOfBoundsException pltfrmExc
                OUTPUT message: Platform Number not
                available. Please enter a valid Platform
                Number.
            CATCH Exception AS allExc
                OUTPUT message: Please fill in all of the
                fields. Valid Platform No. is required for
                termination.
        ENDIF
    ENDDO

    DO
        IF e IS instance of btnAddJ THEN
            TRY
                DECLARE juniorPlatform AS txtJunPltfrm as
                String
                DECLARE juniorInterviewer AS txtJunIntrvwr
                as String
                DECLARE juniorWorkingHr AS txtJunWrkHr as
                int
                DECLARE juniorSalary AS txtJunSalry as int
                DECLARE juniorAppointed AS txtJunAppointBy
                as String
                DECLARE juniorTermination AS txtJunTerminate
                as String

                IF juniorWorkingHr IS LESS THAN 12 OR
                juniorSalary IS LESS THAN 5000 THEN
                    THROW EXCEPTION
                    OUTPUT message: Working Hours and
                    Salary should be more than 12 hours and
                    Rs. 5000 respectively.
```

```
                    ENDIF

                    DECLARE juniorPltfrmInfo AND CALL
                    JuniorDeveloper (input juniorPlatform,
                    juniorInterviewer, juniorWorkingHr,
                    juniorSalary, juniorAppointed,
                    juniorTermination)
                    ADD juniorPltfrmInfo TO list
                    OUTPUT message: Junior developer platform
                    added.
               CATCH NumberFormatException AS emptyNumExc
                    OUTPUT message: Please fill all the text
                    fields. Salary and Working Hours must be in
                    number.
               CATCH Exception AS allExc
                    OUTPUT above thrown exception
          ENDIF
     ENDDO

     DO
          IF e IS instance of btnJunAppoint THEN
               TRY
                    DECLARE juniorDevName AS txtJunDevName as
                    String
                    DECLARE juniorAppointDate AS
                    txtJunAppointDate as String
                    DECLARE juniorSpecialization AS
                    txtJunSpecialization as String
                    DECLARE juniorTermDate AS     txtJunTermDate
                    as String
                    DECLARE juniorPltfrmNum AS txtJunPltfrmNum
                    as int
```

```
                IF juniorPltfrmNum IS GREATER THAN 0 OR
                juniorPltfrmNum IS LESS THAN OR EQUALS TO
                the size of list THEN
                     IF     list.get(juniorPltfrmNum)     is
                     INSTANCE OF JuniorDeveloper THEN
                          DECLARE        juniorDev        AS
                          list.get(juniorPltfrmNum)
                          CALL hireJuniorDeveloper (input
                          juniorDevName, juniorAppointDate,
                          juniorTermDate,
                          juniorSpecialization)
                          OUTPUT message: Junior developer
                          added appointed.
                     ENDIF
                ENDIF
           CATCH NumberFormatException AS emptyNumExc
                OUTPUT message: Please fill all the text
                fields. Advance Salary and Platform Number
                must be in number.
           CATCH IndexOutOfBoundsException pltfrmExc
                OUTPUT   message:   Platform   Number   not
                available. Please enter a valid Platform
                Number.
     ENDIF
ENDDO

DO
     IF e IS instance of btnDisplay THEN
          IF list IS NOT empty THEN
                FOR Developer dev in list
                     IF dev instance of SeniorDeveloper THEN
```

```
                              DECLARE obj AS
                              (SeniorDeveloper)dev
                              CALL obj.display()
                         ENDIF
                         IF dev instance of JuniorDeveloper THEN
                              DECLARE obj1 AS
                              (JuniorDeveloper)dev
                              CALL obj1.display()
                         ENDIF
                    ENDFOR
               ELSE
                    OUTPUT message: Nothing to Display!
               ENDIF
          ENDIF
     ENDDO

     DO
          IF e IS instance of btnClear THEN
               SET txtSenPltfrm AS NULL
               SET txtSenIntrvwr AS NULL
               SET txtSenSalry AS NULL
               SET txtSenWrkHr AS NULL
               SET txtSenContPrd AS NULL
               SET txtSenDevName AS NULL
               SET txtSenAdvsalry AS NULL
               SET txtSenPltfrmNum AS NULL
               SET txtSenStaffRoomNo AS NULL
               SET txtSenJoinDate AS NULL
               SET txtJunSalry AS NULL
               SET txtJunPltfrm AS NULL
               SET txtJunIntrvwr AS NULL
               SET txtJunWrkHr AS NULL
```

```
            SET txtJunAppointBy AS NULL
            SET txtJunTerminate AS NULL
            SET txtJunDevName AS NULL
            SET txtJunAppointDate AS NULL
            SET txtJunSpecialization AS NULL
            SET txtJunTermDate AS NULL
            SET txtJunPltfrmNum AS NULL
            OUTPUT message: All Field(s) Cleared.
        ENDIF
    ENDDO
```

Description

Description of each methods of class RigoTechnology are given below:

4.1 Main Method:

The main() method is the main declaration part of the class RigoTechnology. This method is always public and static as it needs to be called without the instantiation of the class. In this method, the class RigoTechnology is instantiated with creation of its object rigoTechnology(). And then, this object is used to call the method gui().

4.2 GUI Method:

The GUI (Graphical User Interface) of this class is defined in the gui() method. An application form is created in this method in order to access the different methods and the attributes of the developer, senior and junior developer classes. We create the graphical components like the frame, labels, text fields, buttons and panel here, and label and configured them according to their location and much more. These components are designed and added to the main window frame, which is the form. To trigger the action in the event of any user interaction with the components of the form, we use the ActionListener and add them through its interface.

4.3 Action Performed Method:

The method actionPerformed() is the method of the ActionListener interface. This method helps with the interaction of the methods and actions of superclass Developer and the child classes Senior and Junior Developer. This method is overridden for the interaction of each of the components, to feature different actions to each of them. Each component has its own action and interaction. We use try and catch to catch different exception to handle the errors whilst the program is running in this method. With the comparison of the label, text field or name, actions to be triggered on interaction, is joined to the respective components.

Testing

5.1 Test 1

a) Compiling the program using Command Prompt

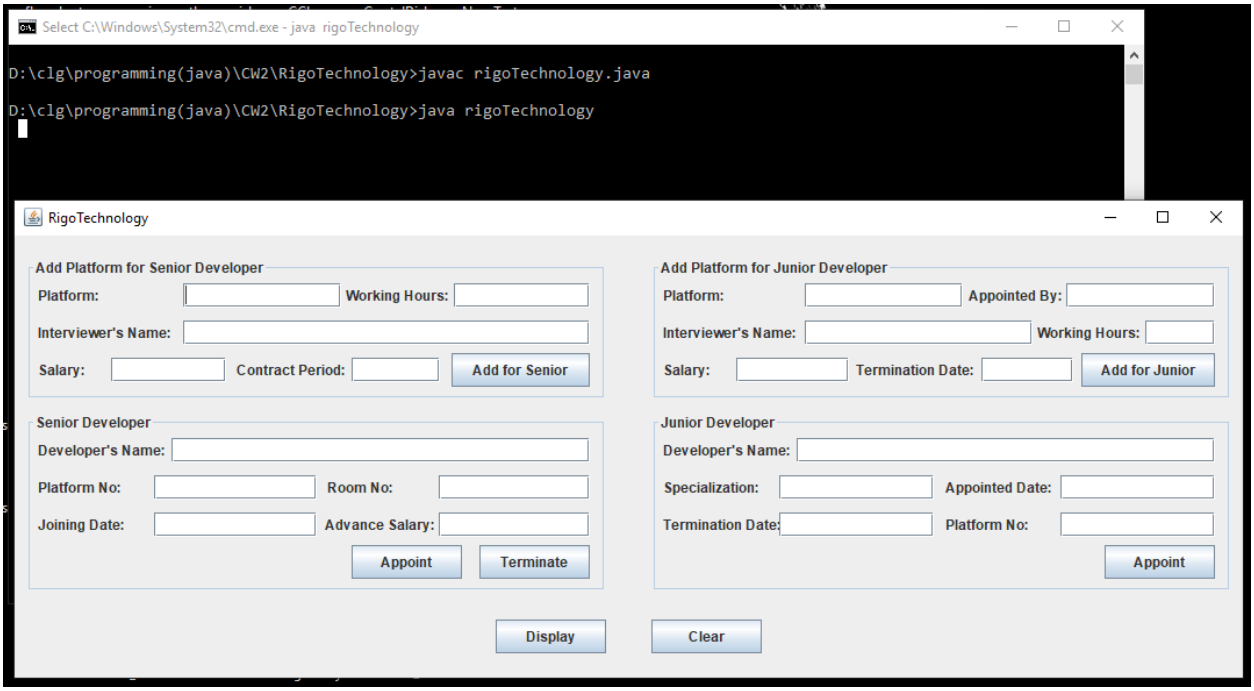

Figure 3: Testing 1

5.2 Test 2

a) Add SeniorDeveloper Platform

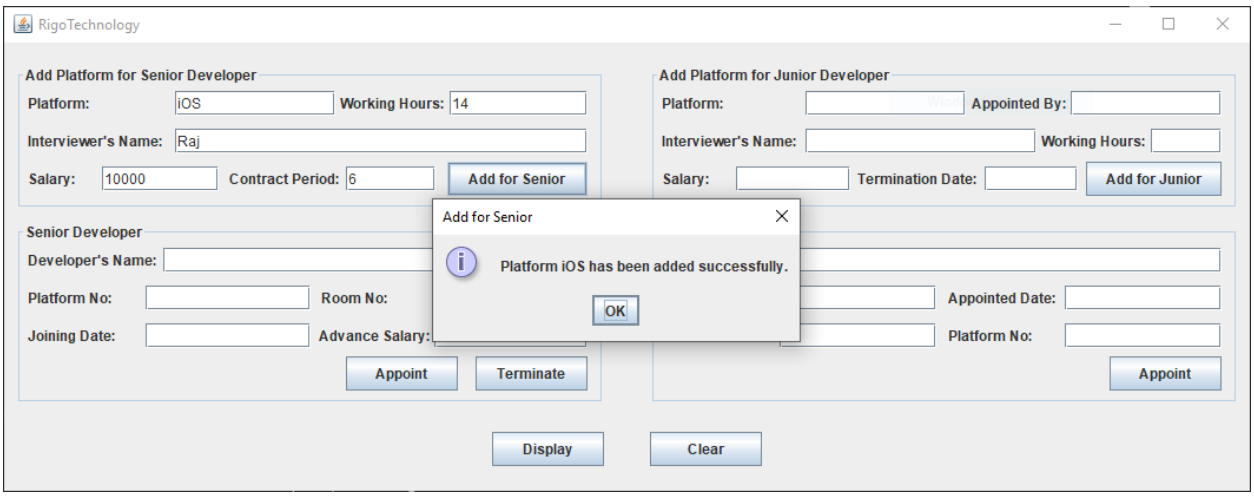

Figure 4: Testing 2.1

b) Add JuniorDeveloper Platform

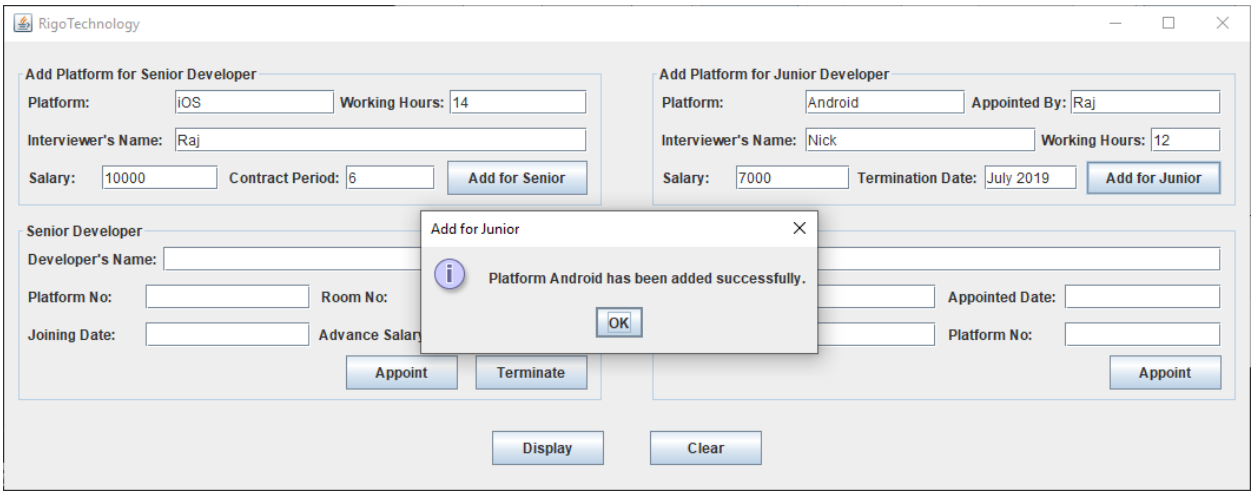

Figure 5: Testing 2.2

c) Appoint a SeniorDeveloper

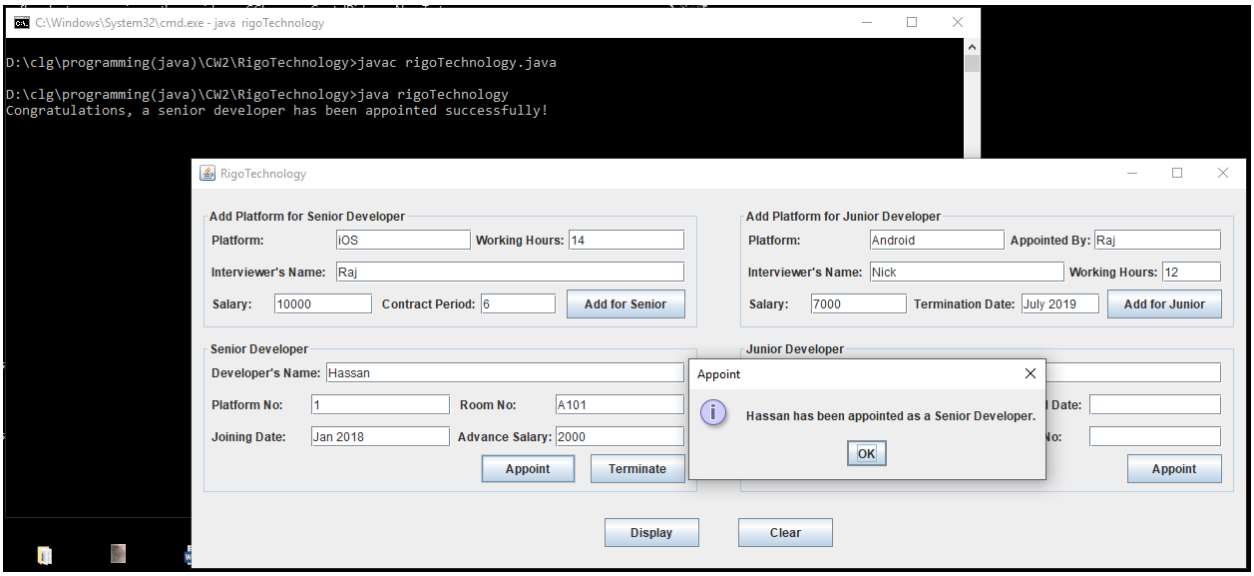

Figure 6: Testing 2.3

d) Appoint a JuniorDeveloper

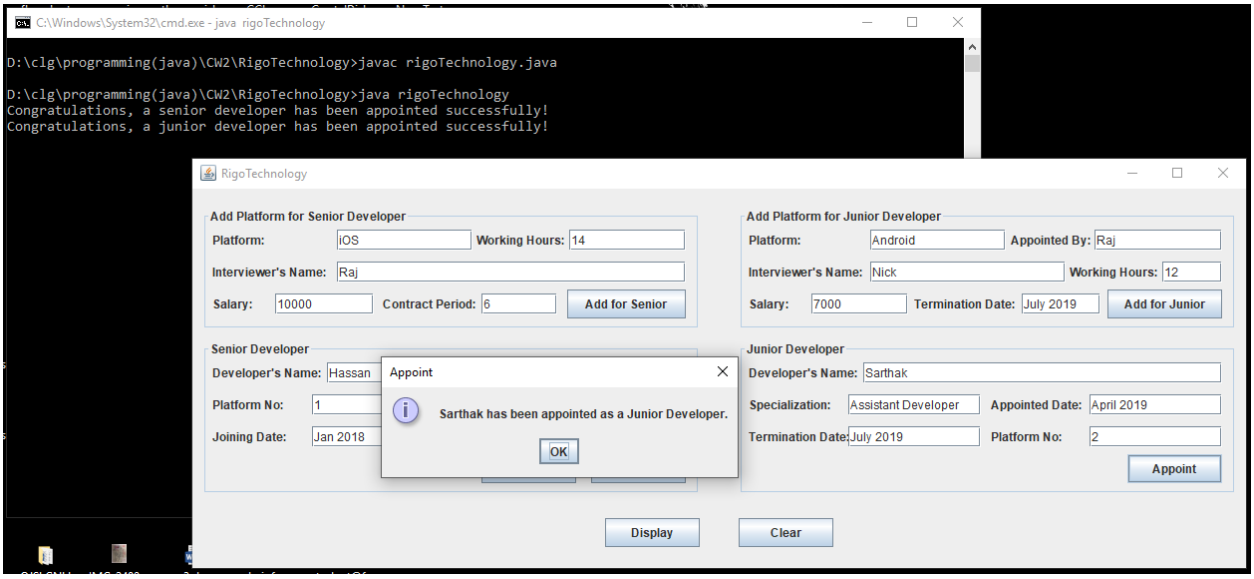

Figure 7: Testing 2.4

e) Displaying all the data and Terminating a SeniorDeveloper

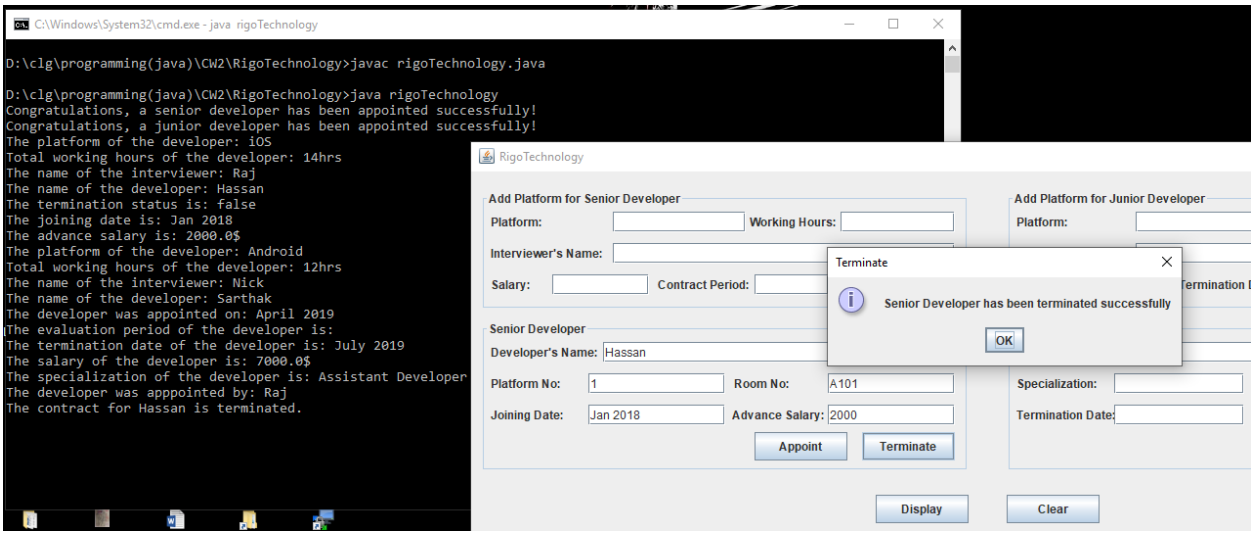

Figure 8: Testing 2.5

5.3 Test 3

a) Fill all the text fields and Salary, Working Hours and Contract Period must be in number.

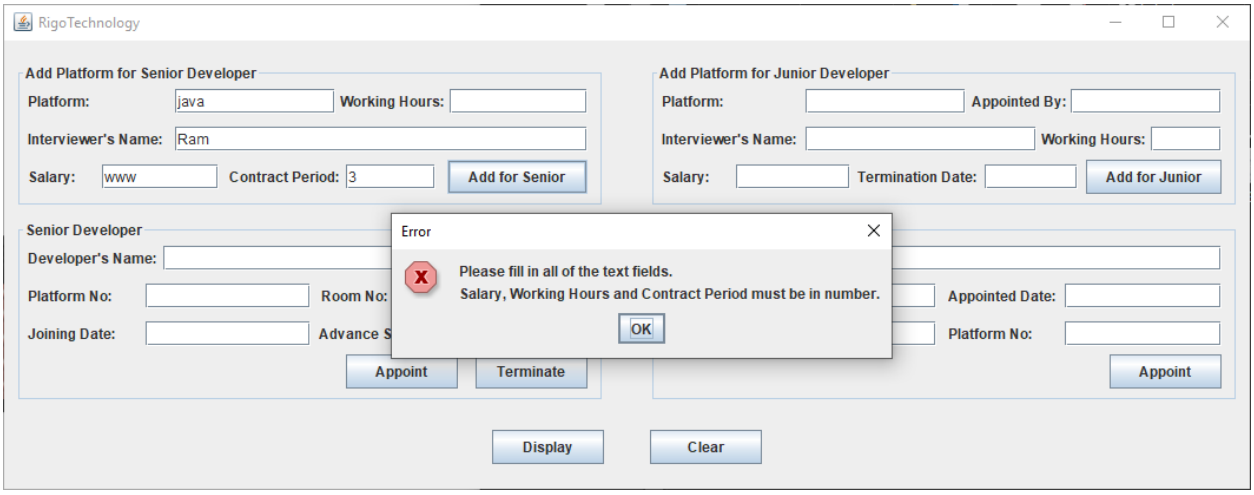

Figure 9: Testing 3.1

b) Working Hours, Salary and Contract Period should be more than 12 hours, Rs. 5000 and 3 months respectively.

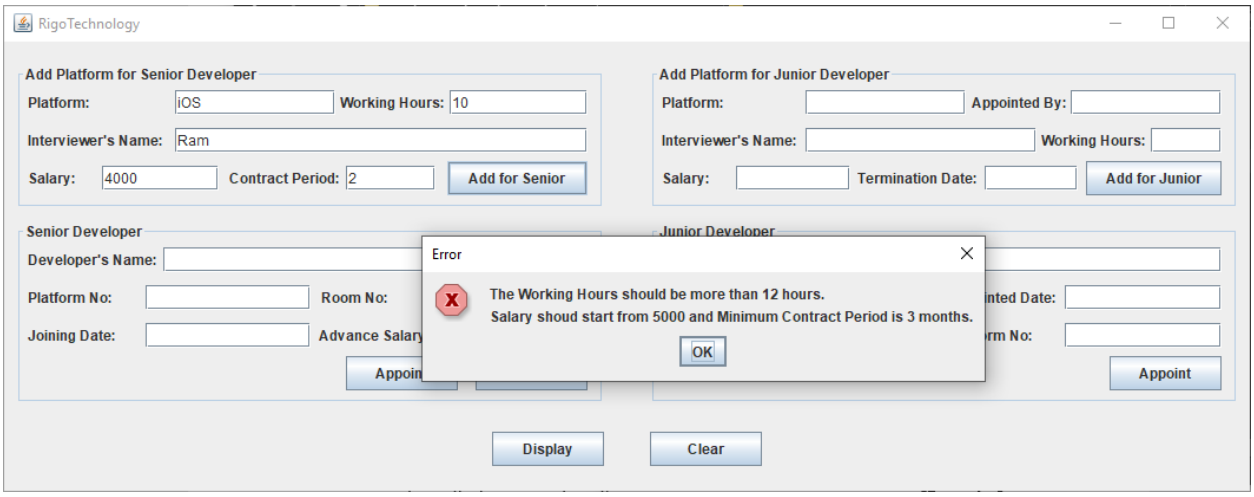

Figure 10: Testing 3.2

c) Fill all the text fields and Advance Salary and Platform Number must be in number.

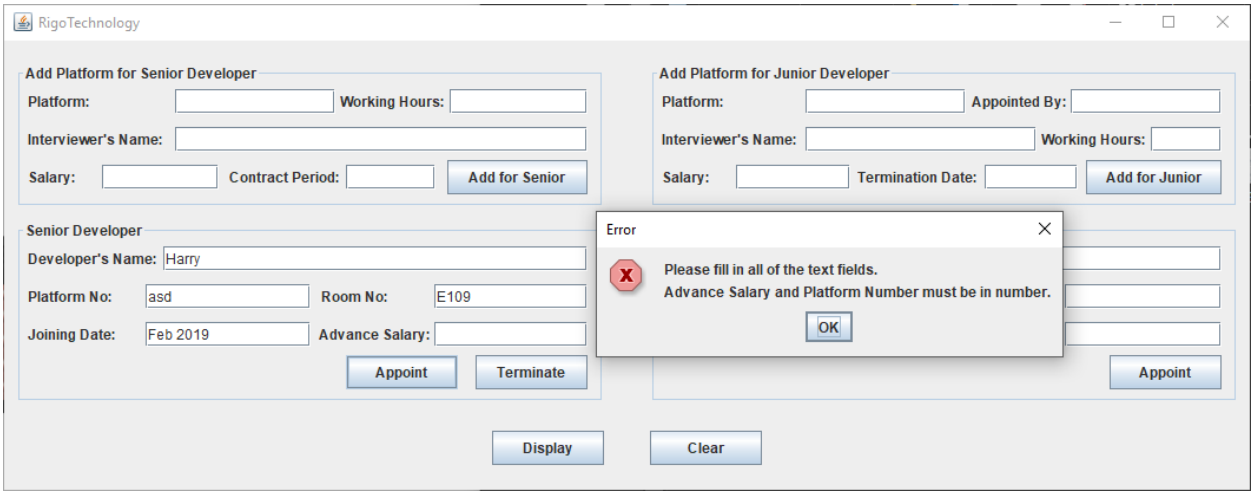

Figure 11: Testing 3.3

d) To enter a valid Platform Number.

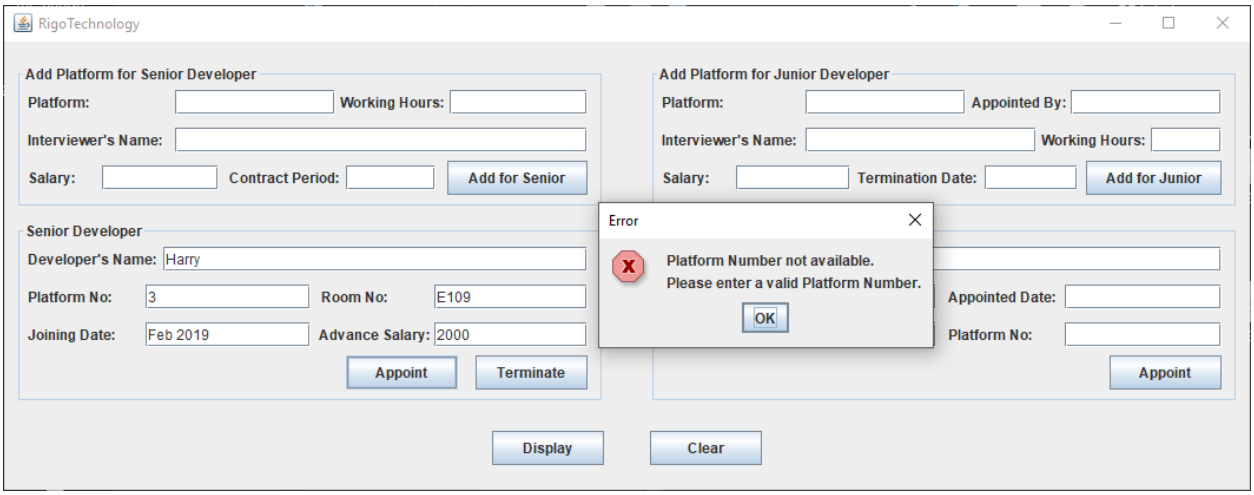

Figure 12: Testing 3.4

e) Advance salary must be less than Salary.

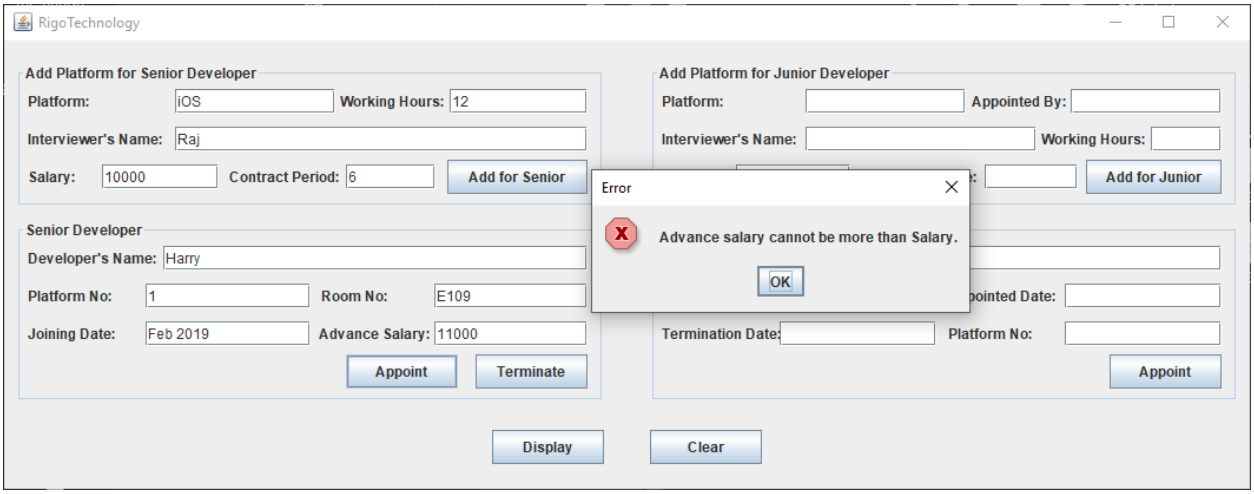

Figure 13: Testing 3.5

f) To enter a valid Platform Number as no developer appointed in that Platform to terminate.

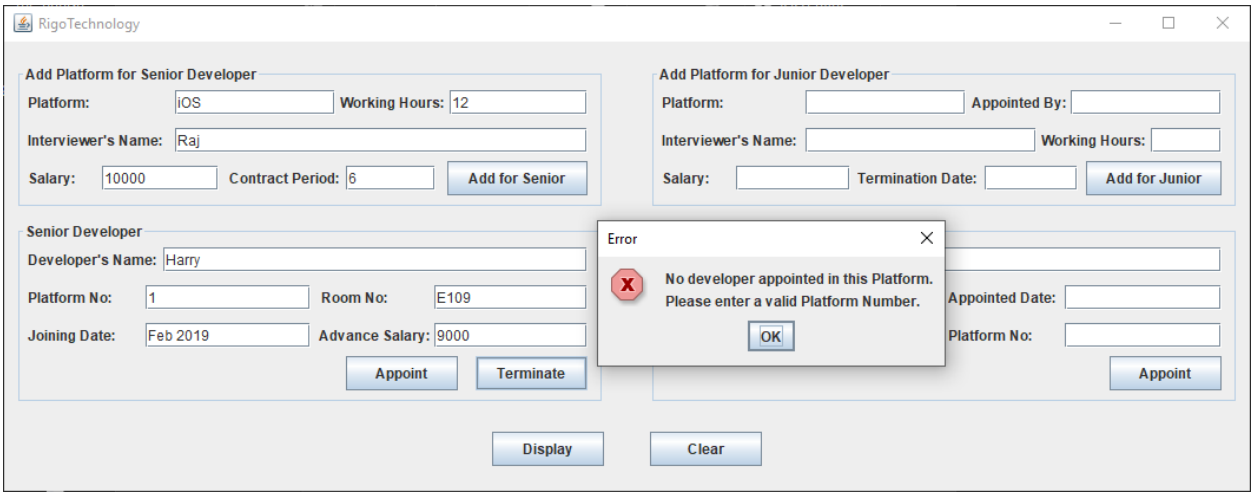

Figure 14: Testing 3.6

g) To enter a valid Platform Number

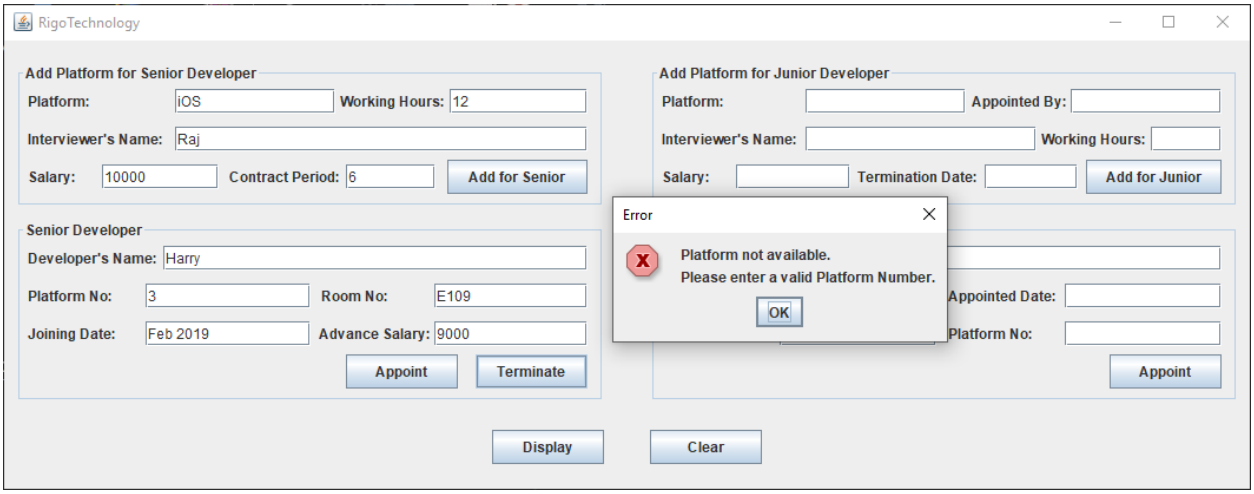

Figure 15: Testing 3.7

h) Fill all the text fields and valid Platform Number is required.

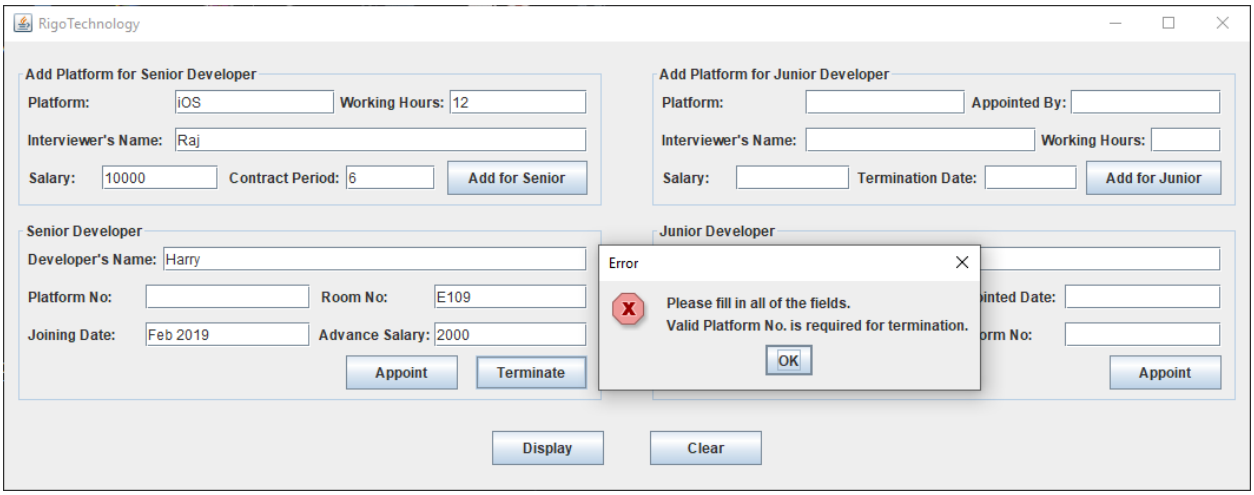

Figure 16: Testing 3.8

i) Fill all the text fields and Salary, Working Hours and Contract Period must be in number.

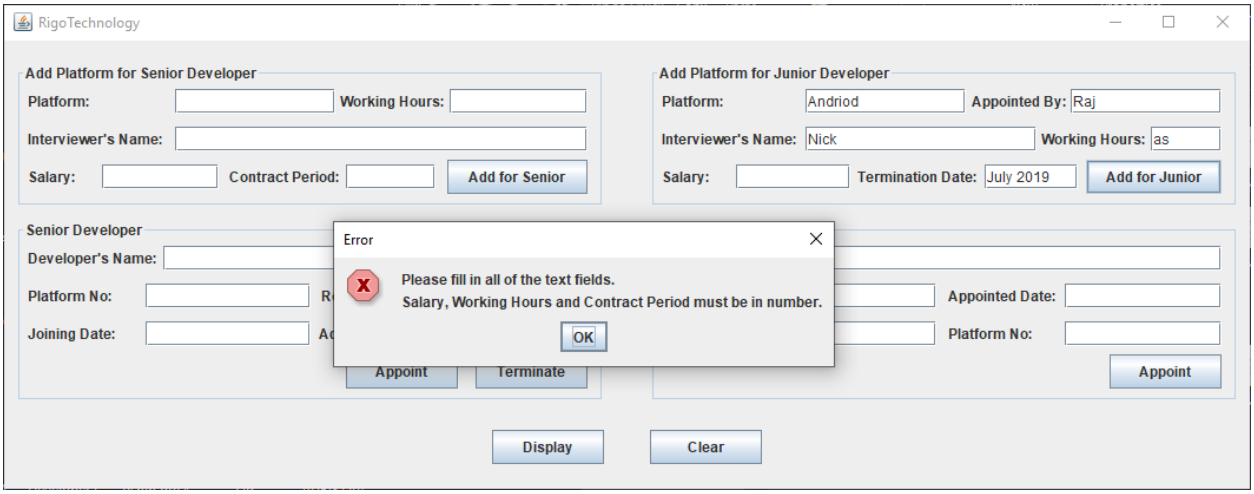

Figure 17: Testing 3.9

j) Working Hours and Salary should be more than 12 hours and Rs. 5000 respectively

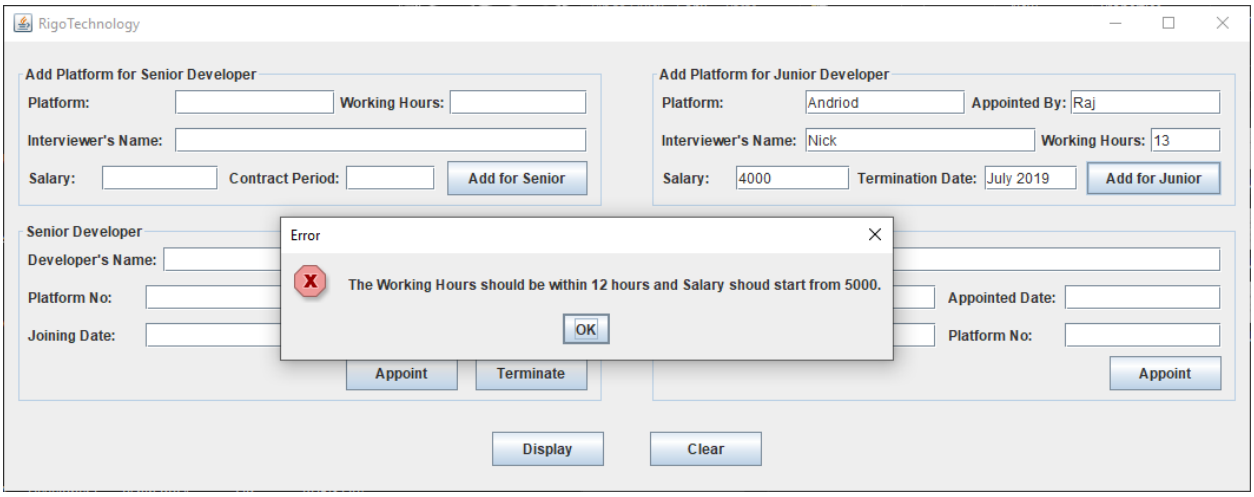

Figure 18: Testing 3.10

k) Fill all the text fields and Platform Number must be in number.

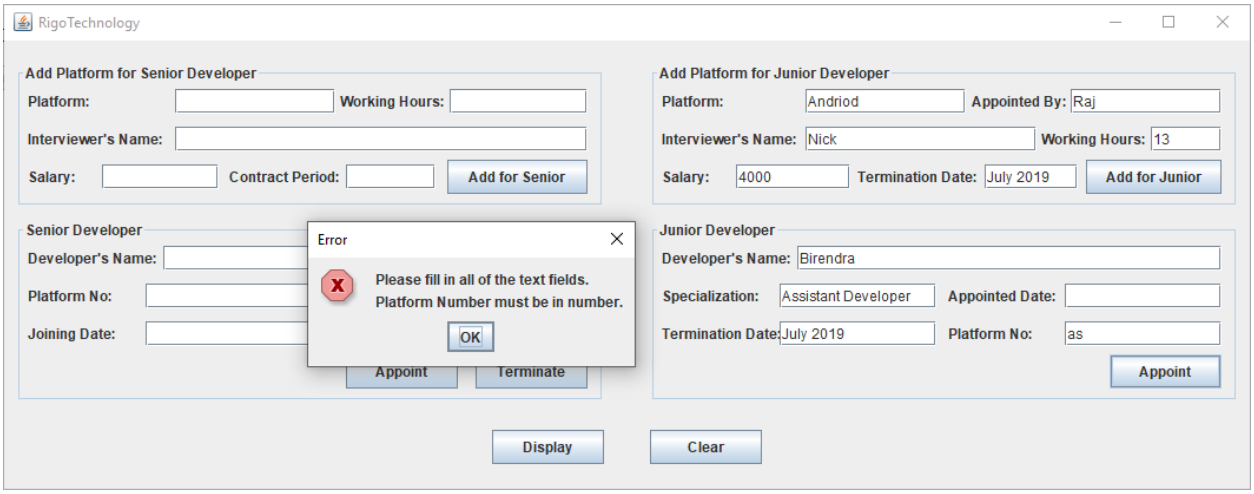

Figure 19: Testing 3.11

l) To enter a valid Platform Number.

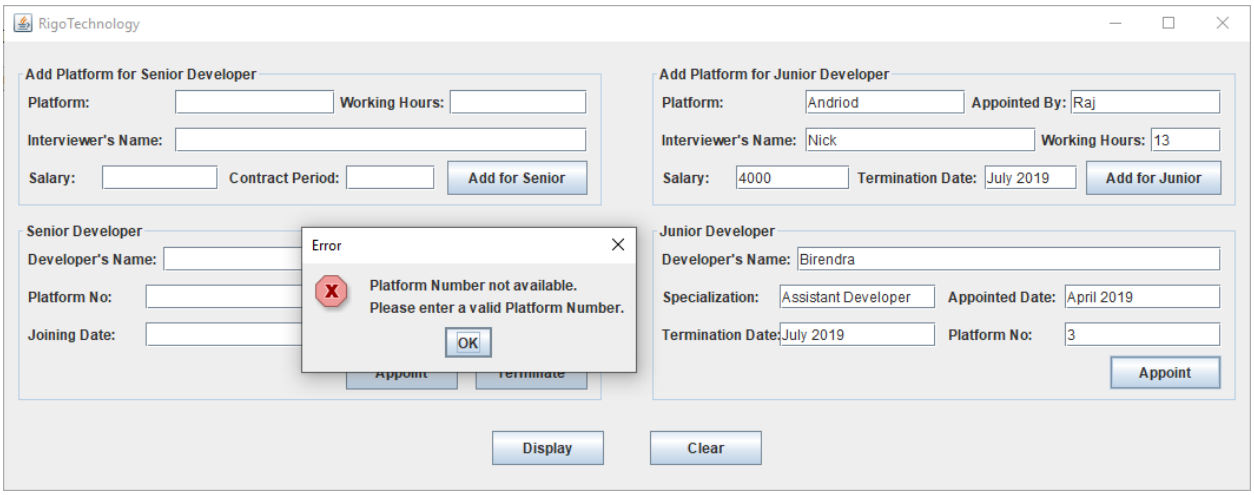

Figure 20: Testing 3.12

m) Nothing to Display.

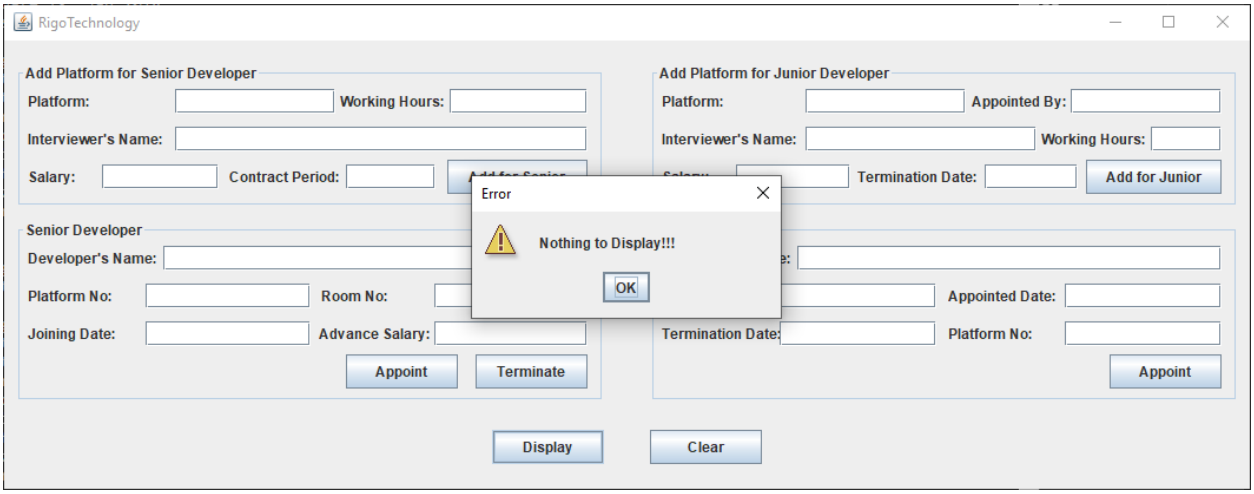

Figure 21: Testing 3.13

Error Detection and Correction

In Java Programming Language, there are 3 types of errors:

- Syntax Error
- Run-time Error
- Logical Error

6.1 Syntax Error

Syntax errors are errors that are found in the syntax of the program such as missing a semicolon at the end of a statement or due to missing braces, class not found, etc. and prevents the code from compiling. These errors will be detected by java compiler and displays the error onto the screen while compiling.

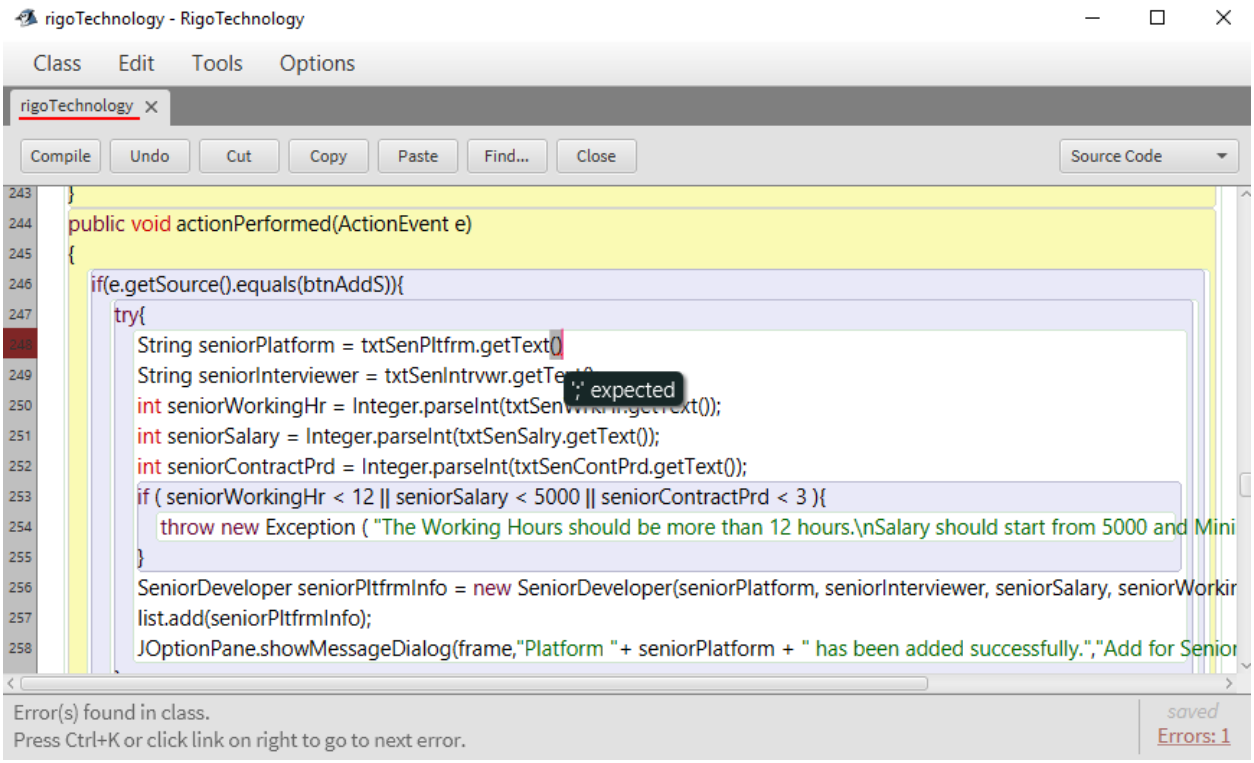

Figure 22: Example of a syntax error found whilst coding

In the above example, the semicolon was missing and was detected by the compiler and was debugged.

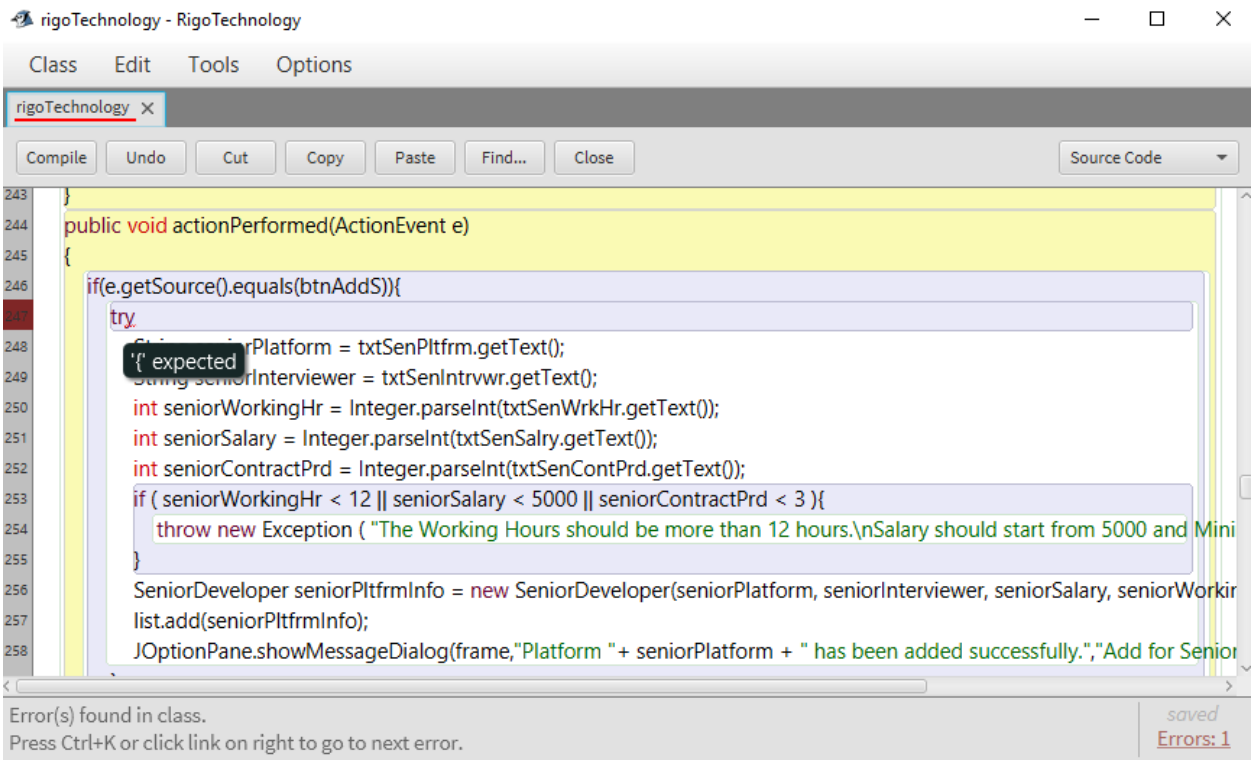

Figure 23: Another example of a syntax error found whilst coding

In this above example, the bracket was missing and was detected by the compiler and was debugged. Due to a single missing bracket, many errors arrived in the program.

6.2 Run-time Error

Run-time errors are errors that occur when the program is running. Run time errors are not detected by the java compiler. The JVM detects this type of error while the program is running. Example: trying to convert a string "hello" into an integer is not possible. The compiler is unable to see this problem but an error is given out whilst the program is running.

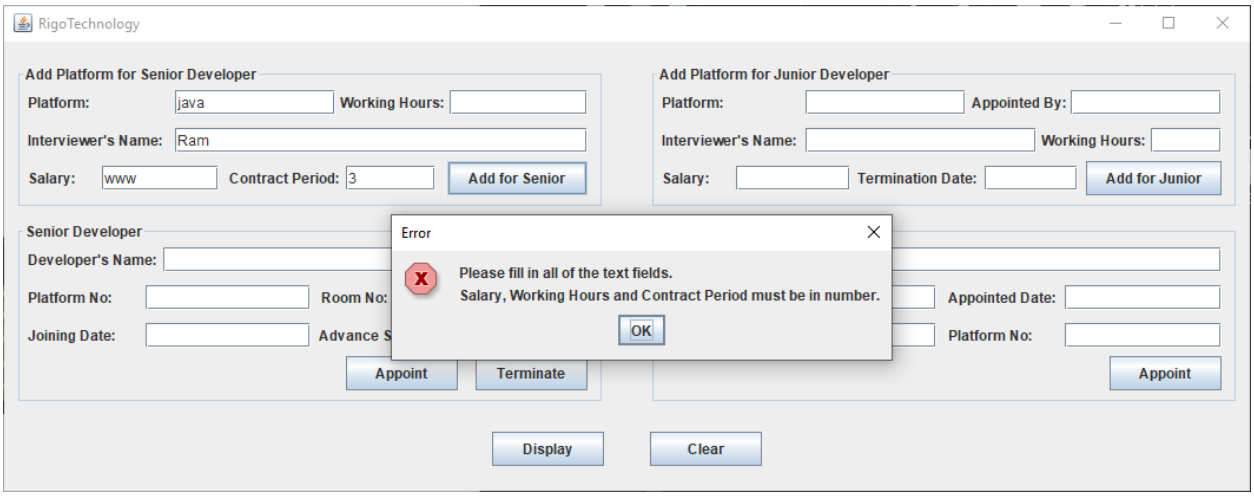

Figure 24: Example of a Run-time error found whilst coding

In this above example, the datatype of salary is number and was in string in the above error also working hours was missing and was detected while running the program. Proper input according to the datatype should be given to solve this type of error.

6.3 Logical Error

Logical errors are the errors that occur when there is a mistake or mistakes made by the programmer. It is neither detected by a compiler nor by the JVM. This type of error gives out unexpected output. The errors may be there due to a wrong idea or a concept of the programmer whilst coding.

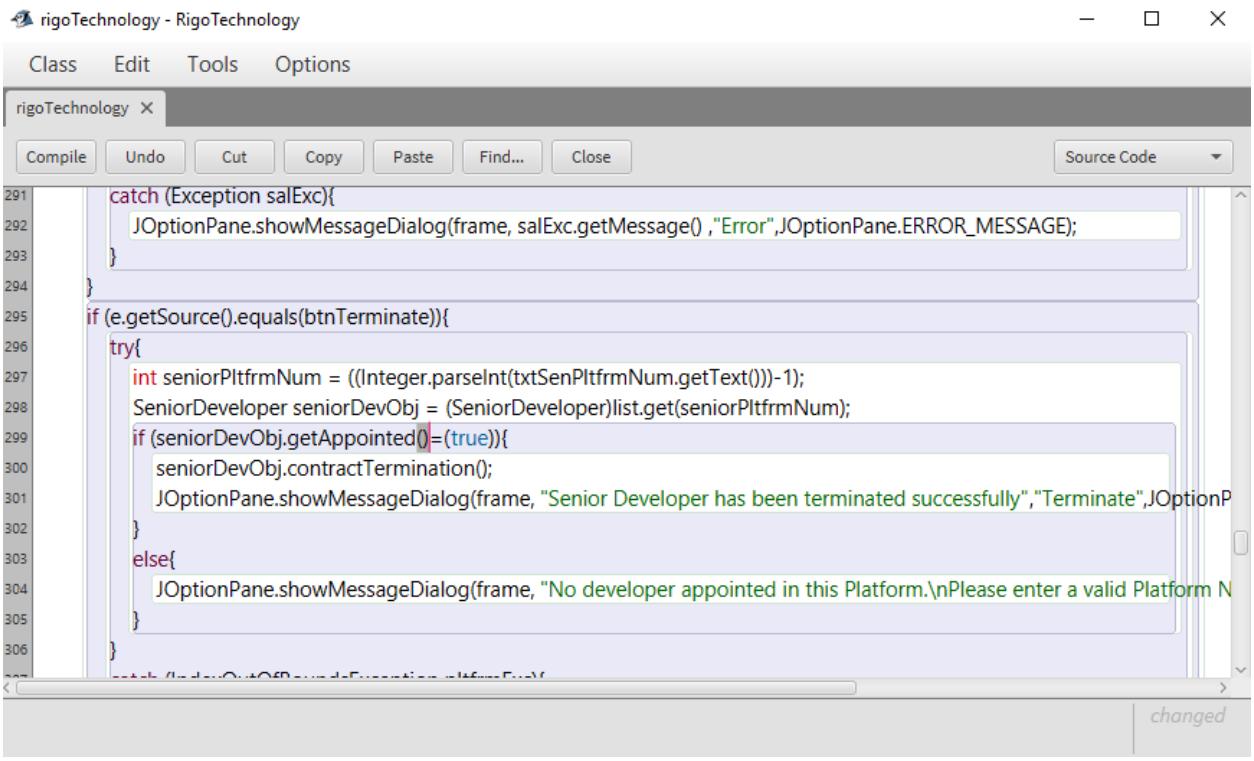

Figure 25: Example of a Logical error found whilst coding

In the highlighted area mentioned in the above example, there is a logical error. In java, "==" means "is equal to," while "=" means to assign a value in that variable. So, the incorrect if statement will always return the value “true” in getAppointed variable. If the code is corrected, the if statement executes only if the value of getAppointed is equal to “true”. However, since the syntax of the incorrect code is acceptable, it will not produce any syntax error and the code will compile with success. The error is seen only after the program has run. So, the programmer should write the program thinking properly and with ease. This type of error is usually hard to find and debug.

Conclusion

The coursework given was the second coursework which was assigned to is to add a class to the previous project that was done in the previous semester to make a GUI (Graphical User Interface) for the system that stores details of all the Developers, which are Senior Developer and Junior Developer in a Company. The coursework had to be done by creating a class RigoTechnology which consists of the GUI of this program. All the methods made in this program were successfully running. All the tasks assigned in the coursework was finally completed. For the program to run successfully, hard work was done.

Lots of research was done related to the contents and tasks of the coursework. Though the project seemed easy, it was a bit tough to do. The necessity of taking the help from the module leader was also required. It was hard to find the information related to the java programming language. Necessary java codes were revised again from the lecture slides that benefited in developing this project. A lot of time was given for the programming part which was quite tough and it took a lot of time for it to develop. The module teachers were very helpful to us if we had any issues with our program.

While researching on the report, we got knowledge about writing the pseudocodes, making the class diagram, testing and debugging any error that we encountered. We also got knowledge that errors were encountered and sometimes can be related to the platform rather than the code itself. So, the version of both JDK and IDEs are supposed to be same which are very important for better output. Whilst debugging the program, we got knowledge about different types of errors and are able to handle those errors. Overall, this project helped us increase our knowledge about the java programming language. This project gave us a drastic change in our knowledge of java programming which will be very helpful for our future need.

To conclude, I would like to thank my module leaders for giving us such a productive and advantageous coursework and for all the helps that they have been providing us. I have learned so many things by doing this project. I had faced many errors while doing this program and hence can address to it if it occurs again in the near future.

Appendix

```
/**
 *This is the GUI interface.
 * @author Abhishek Gupta
 * @version 1.1
 */
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JPanel;
import javax.swing.JTextField;
import javax.swing.JButton;
import javax.swing.border.TitledBorder;
import javax.swing.JOptionPane;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.util.ArrayList;
import java.util.List;
public class rigoTechnology implements ActionListener
{
    JFrame frame;

    JPanel senDevPltPanel, senDev, junDevPltPanel, junDev;

    JLabel     lblSenPltfrm,     lblSenIntrvwr,     lblSenSalry,
lblSenWrkHr, lblSenContPrd, lblSenStaffRoomNo;
    JLabel   lblSenDevName,   lblSenAdvsalry,   lblSenPltfrmNum,
lblSenJoinDate;
    JLabel     lblJunSalry,     lblJunPltfrm,     lblJunIntrvwr,
lblJunWrkHr, lblJunAppointBy, lblJunTerminate;
    JLabel           lblJunDevName,           lblJunAppointDate,
lblJunSpecialization, lblJunTermDate, lblJunPltfrmNum;
```

```
    JTextField txtSenPltfrm, txtSenIntrvwr, txtSenSalry,
txtSenWrkHr, txtSenContPrd;
    JTextField txtSenDevName, txtSenAdvsalry, txtSenPltfrmNum,
txtSenStaffRoomNo, txtSenJoinDate;
    JTextField txtJunSalry, txtJunPltfrm, txtJunIntrvwr,
txtJunWrkHr, txtJunAppointBy, txtJunTerminate;
    JTextField txtJunDevName, txtJunAppointDate,
txtJunSpecialization, txtJunTermDate, txtJunPltfrmNum;

    JButton btnClear, btnDisplay, btnAddS, btnSenAppoint,
btnTerminate, btnAddJ, btnJunAppoint;

    List<Developer> list = new ArrayList<Developer>();
    public static void main(String[] args)
    {
        new rigoTechnology().gui();
    }
    public void gui()
    {
        frame = new JFrame("Rigo Technology");
        frame.setTitle("RigoTechnology");
        frame.getContentPane().setLayout(null);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        lblSenPltfrm=new JLabel("Platform:");
        lblSenIntrvwr=new JLabel("Interviewer's Name:");
        lblSenSalry=new JLabel("Salary:");
        lblSenWrkHr=new JLabel("Working Hours:");
        lblSenContPrd = new JLabel("Contract Period:");
        lblSenStaffRoomNo = new JLabel("Room No:");
        lblSenDevName=new JLabel("Developer's Name:");
```

```
        lblSenAdvsalry=new JLabel("Advance Salary:");
        lblSenPltfrmNum=new JLabel("Platform No:");
        lblSenJoinDate = new JLabel("Joining Date:");
        lblJunSalry = new JLabel("Salary:");
        lblJunPltfrm = new JLabel("Platform:");
        lblJunIntrvwr = new JLabel("Interviewer's Name:");
        lblJunWrkHr = new JLabel("Working Hours:");
        lblJunAppointBy = new JLabel("Appointed By:");
        lblJunTerminate = new JLabel("Termination Date:");
        lblJunDevName = new JLabel("Developer's Name:");
        lblJunAppointDate = new JLabel("Appointed Date:");
        lblJunSpecialization = new JLabel("Specialization:");
        lblJunTermDate = new JLabel("Termination Date:");
        lblJunPltfrmNum = new JLabel("Platform No:");

        txtSenPltfrm=new JTextField();
        txtSenIntrvwr=new JTextField();
        txtSenSalry=new JTextField();
        txtSenWrkHr=new JTextField();
        txtSenContPrd = new JTextField();
        txtSenDevName=new JTextField();
        txtSenAdvsalry=new JTextField();
        txtSenPltfrmNum=new JTextField();
        txtSenStaffRoomNo = new JTextField();
        txtSenJoinDate = new JTextField();
        txtJunSalry = new JTextField();
        txtJunPltfrm = new JTextField();
        txtJunIntrvwr = new JTextField();
        txtJunWrkHr = new JTextField();
        txtJunAppointBy = new JTextField();
        txtJunTerminate = new JTextField();
        txtJunDevName = new JTextField();
```

```
        txtJunAppointDate = new JTextField();
        txtJunSpecialization = new JTextField();
        txtJunTermDate = new JTextField();
        txtJunPltfrmNum = new JTextField();

        btnClear=new JButton("Clear");
        btnDisplay = new JButton("Display");
        btnAddS=new JButton("Add for Senior");
        btnSenAppoint=new JButton("Appoint");
        btnTerminate = new JButton("Terminate");
        btnAddJ = new JButton("Add for Junior");
        btnJunAppoint = new JButton("Appoint");

        senDevPltPanel = new JPanel();
        senDevPltPanel.setBorder(new   TitledBorder(null,   "Add
Platform    for    Senior    Developer",    TitledBorder.LEADING,
TitledBorder.TOP, null, null));
        senDevPltPanel.setLayout(null);

        senDevPltPanel.add(lblSenPltfrm);
        senDevPltPanel.add(txtSenPltfrm);
        senDevPltPanel.add(lblSenIntrvwr);
        senDevPltPanel.add(txtSenIntrvwr);
        senDevPltPanel.add(lblSenSalry);
        senDevPltPanel.add(txtSenSalry);
        senDevPltPanel.add(lblSenWrkHr);
        senDevPltPanel.add(txtSenWrkHr);
        senDevPltPanel.add(lblSenContPrd);
        senDevPltPanel.add(txtSenContPrd);
        senDevPltPanel.add(btnAddS);

        senDev = new JPanel();
```

```
        senDev.setBorder(new      TitledBorder(null,      "Senior
Developer",    TitledBorder.LEADING,    TitledBorder.TOP,    null,
null));
        senDev.setLayout(null);

        senDev.add(lblSenDevName);
        senDev.add(txtSenDevName);
        senDev.add(lblSenAdvsalry);
        senDev.add(txtSenAdvsalry);
        senDev.add(lblSenPltfrmNum);
        senDev.add(txtSenPltfrmNum);
        senDev.add(lblSenStaffRoomNo);
        senDev.add(txtSenStaffRoomNo);
        senDev.add(lblSenJoinDate);
        senDev.add(txtSenJoinDate);
        senDev.add(btnSenAppoint);
        senDev.add(btnTerminate);

        junDevPltPanel = new JPanel();
        junDevPltPanel.setBorder(new   TitledBorder(null,   "Add
Platform    for    Junior    Developer",    TitledBorder.LEADING,
TitledBorder.TOP, null, null));
        junDevPltPanel.setLayout(null);

        junDevPltPanel.add(lblJunSalry);
        junDevPltPanel.add(txtJunSalry);
        junDevPltPanel.add(lblJunPltfrm);
        junDevPltPanel.add(txtJunPltfrm);
        junDevPltPanel.add(lblJunIntrvwr);
        junDevPltPanel.add(txtJunIntrvwr);
        junDevPltPanel.add(lblJunWrkHr);
        junDevPltPanel.add(txtJunWrkHr);
```

```
        junDevPltPanel.add(txtJunAppointBy);
        junDevPltPanel.add(lblJunAppointBy);
        junDevPltPanel.add(lblJunTerminate);
        junDevPltPanel.add(txtJunTerminate);
        junDevPltPanel.add(btnAddJ);

        junDev = new JPanel();
        junDev.setBorder(new      TitledBorder(null,      "Junior
Developer",    TitledBorder.LEADING,    TitledBorder.TOP,    null,
null));
        junDev.setLayout(null);

        junDev.add(lblJunDevName);
        junDev.add(txtJunDevName);
        junDev.add(lblJunAppointDate);
        junDev.add(txtJunAppointDate);
        junDev.add(lblJunSpecialization);
        junDev.add(txtJunSpecialization);
        junDev.add(lblJunTermDate);
        junDev.add(txtJunTermDate);
        junDev.add(lblJunPltfrmNum);
        junDev.add(txtJunPltfrmNum);
        junDev.add(btnJunAppoint);

        senDevPltPanel.setBounds(10, 19, 499, 122);
        senDev.setBounds(10, 152, 499, 154);
        junDev.setBounds(548, 152, 499, 154);
        junDevPltPanel.setBounds(548, 19, 499, 122);

        lblSenWrkHr.setBounds(275, 22, 90, 21);
        lblSenContPrd.setBounds(181, 86, 95, 21);
        lblSenSalry.setBounds(10, 86, 60, 21);
```

```
        lblSenIntrvwr.setBounds(10, 54, 117, 21);
        lblSenPltfrm.setBounds(10, 22, 70, 21);
        lblSenDevName.setBounds(10, 22, 110, 21);
        lblSenAdvsalry.setBounds(257, 86, 98, 21);
        lblSenPltfrmNum.setBounds(10, 54, 85, 21);
        lblSenStaffRoomNo.setBounds(259, 54, 63, 21);
        lblSenJoinDate.setBounds(10, 86, 80, 21);
        lblJunSalry.setBounds(10, 86, 60, 21);
        lblJunPltfrm.setBounds(10, 22, 70, 21);
        lblJunIntrvwr.setBounds(10, 54, 117, 21);
        lblJunWrkHr.setBounds(332, 54, 90, 21);
        lblJunAppointBy.setBounds(273, 22, 83, 21);
        lblJunTerminate.setBounds(176, 86, 104, 21);
        lblJunDevName.setBounds(10, 22, 110, 21);
        lblJunAppointDate.setBounds(253, 54, 94, 21);
        lblJunSpecialization.setBounds(10, 54, 84, 21);
        lblJunTermDate.setBounds(10, 86, 120, 21);
        lblJunPltfrmNum.setBounds(253, 86, 79, 21);

        txtSenIntrvwr.setBounds(135, 54, 350, 21);
        txtSenPltfrm.setBounds(135, 22, 136, 21);
        txtSenWrkHr.setBounds(368, 22, 117, 21);
        txtSenSalry.setBounds(73, 86, 99, 21);
        txtSenContPrd.setBounds(280, 86, 76, 21);
        txtSenDevName.setBounds(125, 22, 360, 21);
        txtSenAdvsalry.setBounds(355, 86, 130, 21);
        txtSenPltfrmNum.setBounds(110, 54, 140, 21);
        txtSenStaffRoomNo.setBounds(355, 54, 130, 21);
        txtSenJoinDate.setBounds(110, 86, 140, 21);
        txtJunSalry.setBounds(73, 86, 97, 21);
        txtJunPltfrm.setBounds(132, 22, 136, 21);
        txtJunIntrvwr.setBounds(132, 54, 196, 21);
```

```
        txtJunWrkHr.setBounds(425, 54, 60, 21);
        txtJunAppointBy.setBounds(357, 22, 128, 21);
        txtJunTerminate.setBounds(284, 86, 79, 21);
        txtJunDevName.setBounds(125, 22, 360, 21);
        txtJunAppointDate.setBounds(352, 54, 133, 21);
        txtJunSpecialization.setBounds(110, 54, 133, 21);
        txtJunTermDate.setBounds(110, 86, 133, 21);
        txtJunPltfrmNum.setBounds(352, 86, 133, 21);

        btnAddS.setBounds(366, 82, 119, 29);
        btnDisplay.setBounds(414, 330, 95, 29);
        btnClear.setBounds(548,330,95,29);
        btnSenAppoint.setBounds(280, 114, 95, 29);
        btnTerminate.setBounds(390, 114, 95, 29);
        btnAddJ.setBounds(370, 82, 115, 29);
        btnJunAppoint.setBounds(390, 114, 95, 29);

        frame.getContentPane().add(senDev);
        frame.getContentPane().add(senDevPltPanel);
        frame.getContentPane().add(junDev);
        frame.getContentPane().add(junDevPltPanel);
        frame.getContentPane().add(btnDisplay);
        frame.getContentPane().add(btnClear);

        btnAddS.addActionListener(this);
        btnSenAppoint.addActionListener(this);
        btnTerminate.addActionListener(this);
        btnAddJ.addActionListener(this);
        btnJunAppoint.addActionListener(this);
        btnDisplay.addActionListener(this);
        btnClear.addActionListener(this);
```

```
        frame.setSize(1073,419);
        frame.setVisible(true);
    }
    public void actionPerformed(ActionEvent e)
    {
        if(e.getSource().equals(btnAddS)){
            try{
                String seniorPlatform = txtSenPltfrm.getText();
                String             seniorInterviewer             =
txtSenIntrvwr.getText();
                int               seniorWorkingHr               =
Integer.parseInt(txtSenWrkHr.getText());
                int                 seniorSalary                =
Integer.parseInt(txtSenSalry.getText());
                int              seniorContractPrd              =
Integer.parseInt(txtSenContPrd.getText());
                if ( seniorWorkingHr < 12 || seniorSalary < 5000
|| seniorContractPrd < 3 ){
                    throw new Exception ( "The Working Hours
should be more than 12 hours.\nSalary should start from 5000 and
Minimum Contract Period is 3 months." );
                }
                SeniorDeveloper    seniorPltfrmInfo    =    new
SeniorDeveloper(seniorPlatform, seniorInterviewer, seniorSalary,
seniorWorkingHr, seniorContractPrd);
                list.add(seniorPltfrmInfo);
                JOptionPane.showMessageDialog(frame,"Platform  "+
seniorPlatform  +  "  has  been  added  successfully.","Add  for
Senior",JOptionPane.INFORMATION_MESSAGE);
            }
            catch(NumberFormatException emptyNumExc){
```

```
                JOptionPane.showMessageDialog(frame,"Please  fill
in all of the text fields.\nSalary, Working Hours and Contract
Period must be in number.","Error",JOptionPane.ERROR_MESSAGE);
            }
            catch (Exception allExc){
                JOptionPane.showMessageDialog(frame,
allExc.getMessage(),"Error",JOptionPane.ERROR_MESSAGE);
            }
        }
        if (e.getSource().equals(btnSenAppoint)){
            try{
                String seniorDevName = txtSenDevName.getText();
                String              seniorJoinDate              =
txtSenJoinDate.getText();
                int               seniorAdvSalary               =
Integer.parseInt(txtSenAdvsalry.getText());
                String            seniorStaffRoomNo             =
txtSenStaffRoomNo.getText();
                int               seniorPltfrmNum               =
((Integer.parseInt(txtSenPltfrmNum.getText()))-1);
                if(seniorPltfrmNum>0                            ||
seniorPltfrmNum<=list.size()){
                    if(list.get(seniorPltfrmNum)       instanceof
SeniorDeveloper){
                        SeniorDeveloper        seniorDev        =
(SeniorDeveloper)list.get(seniorPltfrmNum);
                        if     (     seniorDev.getSalary()     <
seniorAdvSalary ){
                            throw  new  Exception("Advance  salary
cannot be more than Salary.");
                        }
```

```
seniorDev.hireSeniorDeveloper(seniorDevName, seniorJoinDate, seniorAdvSalary, seniorStaffRoomNo);
                        JOptionPane.showMessageDialog(frame, seniorDevName+" has been appointed as a Senior Developer.","Appoint",JOptionPane.INFORMATION_MESSAGE);
                    }
                }
            }
            catch(NumberFormatException emptyNumExc){
                JOptionPane.showMessageDialog(frame,"Please fill in all of the text fields.\nAdvance Salary and Platform Number must be in number.","Error",JOptionPane.ERROR_MESSAGE);
            }
            catch (IndexOutOfBoundsException pltfrmExc){
                JOptionPane.showMessageDialog(frame, "Platform Number not available.\nPlease enter a valid Platform Number.","Error",JOptionPane.ERROR_MESSAGE);
            }
            catch (Exception salExc){
                JOptionPane.showMessageDialog(frame, salExc.getMessage() ,"Error",JOptionPane.ERROR_MESSAGE);
            }
        }
        if (e.getSource().equals(btnTerminate)){
            try{
                int seniorPltfrmNum = ((Integer.parseInt(txtSenPltfrmNum.getText()))-1);
                SeniorDeveloper seniorDevObj = (SeniorDeveloper)list.get(seniorPltfrmNum);
                if (seniorDevObj.getAppointed()==(true)){
                    seniorDevObj.contractTermination();
```

```
                    JOptionPane.showMessageDialog(frame, "Senior
Developer has been terminated
successfully","Terminate",JOptionPane.INFORMATION_MESSAGE);
                }
                else{
                    JOptionPane.showMessageDialog(frame, "No
developer appointed in this Platform.\nPlease enter a valid
Platform Number.","Error",JOptionPane.ERROR_MESSAGE);
                }
            }
            catch (IndexOutOfBoundsException pltfrmExc){
                JOptionPane.showMessageDialog(frame, "Platform
not available.\nPlease enter a valid Platform
Number.","Error",JOptionPane.ERROR_MESSAGE);
            }
            catch(Exception emptyExc){
                JOptionPane.showMessageDialog(frame,"Please fill
in all of the fields.\nValid Platform No. is required for
termination.","Error",JOptionPane.ERROR_MESSAGE);
            }
        }
        if (e.getSource().equals(btnAddJ)){
            try{
                String juniorPlatform = txtJunPltfrm.getText();
                String juniorInterviewer =
txtJunIntrvwr.getText();
                int juniorWorkingHr =
Integer.parseInt(txtJunWrkHr.getText());
                int juniorSalary =
Integer.parseInt(txtJunSalry.getText());
                String juniorAppointed =
txtJunAppointBy.getText();
```

```
                String            juniorTermination            =
txtJunTerminate.getText();
                if  (juniorWorkingHr  <  12  ||  juniorSalary  <
5000){
                    throw  new  Exception  (  "The  Working  Hours
should be within 12 hours and Salary should start from 5000." );
                }
                JuniorDeveloper    juniorPltfrmInfo    =    new
JuniorDeveloper(juniorPlatform,               juniorInterviewer,
juniorWorkingHr,          juniorSalary,          juniorAppointed,
juniorTermination);
                list.add(juniorPltfrmInfo);
                JOptionPane.showMessageDialog(frame,"Platform  "+
juniorPlatform  +  "  has  been  added  successfully.","Add  for
Junior",JOptionPane.INFORMATION_MESSAGE);
            }
            catch (NumberFormatException emptyNumExc){
                JOptionPane.showMessageDialog(frame,"Please  fill
in all of the text fields.\nSalary and Working Hours must be in
number.","Error",JOptionPane.ERROR_MESSAGE);
            }
            catch (Exception allExc){
                JOptionPane.showMessageDialog(frame,
allExc.getMessage(),"Error",JOptionPane.ERROR_MESSAGE);
            }
        }
        if (e.getSource().equals(btnJunAppoint)){
            try{
                String juniorDevName = txtJunDevName.getText();
                String            juniorAppointDate            =
txtJunAppointDate.getText();
```

```
                String          juniorSpecialization          =
txtJunSpecialization.getText();
                String             juniorTermDate             =
txtJunTermDate.getText();
                int               juniorPltfrmNum             =
((Integer.parseInt(txtJunPltfrmNum.getText()))-1);
                if(juniorPltfrmNum>0                          ||
juniorPltfrmNum<=list.size()){
                    if(list.get(juniorPltfrmNum)      instanceof
JuniorDeveloper){
                        JuniorDeveloper       juniorDev       =
(JuniorDeveloper) list.get(juniorPltfrmNum);
                        juniorDev.hireJuniorDeveloper
(juniorDevName,       juniorAppointDate,       juniorTermDate,
juniorSpecialization);
                        JOptionPane.showMessageDialog(frame,
juniorDevName+"    has    been    appointed    as    a    Junior
Developer.","Appoint",JOptionPane.INFORMATION_MESSAGE);
                    }
                }
            }
            catch(NumberFormatException emptyNumExc){
                JOptionPane.showMessageDialog(frame,"Please  fill
in  all  of  the  text  fields.\nPlatform  Number  must  be  in
number.","Error",JOptionPane.ERROR_MESSAGE);
            }
            catch (IndexOutOfBoundsException pltfrmExc){
                JOptionPane.showMessageDialog(frame,    "Platform
Number   not   available.\nPlease   enter   a   valid   Platform
Number.","Error",JOptionPane.ERROR_MESSAGE);
            }
        }
```

```
        if(e.getSource().equals(btnDisplay)){
            if (!list.isEmpty()){
                for (Developer dev: list){
                    if(dev instanceof SeniorDeveloper){
                        SeniorDeveloper
obj=(SeniorDeveloper)dev;
                        obj.display();
                    }
                    if(dev instanceof JuniorDeveloper){
                        JuniorDeveloper
obj1=(JuniorDeveloper)dev;
                        obj1.display();
                    }
                }
            }
            else{
                JOptionPane.showMessageDialog(frame,"Nothing   to
Display!!!","Error",JOptionPane.WARNING_MESSAGE);
            }
        }
        if (e.getSource().equals(btnClear)){
            txtSenPltfrm.setText("");
            txtSenIntrvwr.setText("");
            txtSenSalry.setText("");
            txtSenWrkHr.setText("");
            txtSenContPrd.setText("");
            txtSenDevName.setText("");
            txtSenAdvsalry.setText("");
            txtSenPltfrmNum.setText("");
            txtSenStaffRoomNo.setText("");
            txtSenJoinDate.setText("");
            txtJunSalry.setText("");
```

```
            txtJunPltfrm.setText("");
            txtJunIntrvwr.setText("");
            txtJunWrkHr.setText("");
            txtJunAppointBy.setText("");
            txtJunTerminate.setText("");
            txtJunDevName.setText("");
            txtJunAppointDate.setText("");
            txtJunSpecialization.setText("");
            txtJunTermDate.setText("");
            txtJunPltfrmNum.setText("");
            JOptionPane.showMessageDialog(frame,"All     Field(s)
Cleared.","Clear",JOptionPane.INFORMATION_MESSAGE);
        }
    }
}
```

References

Sen, D. (2018-19) *Programming*. Lecture Slides. Kathmandu: Dhruba Sen London Metropolitan University.