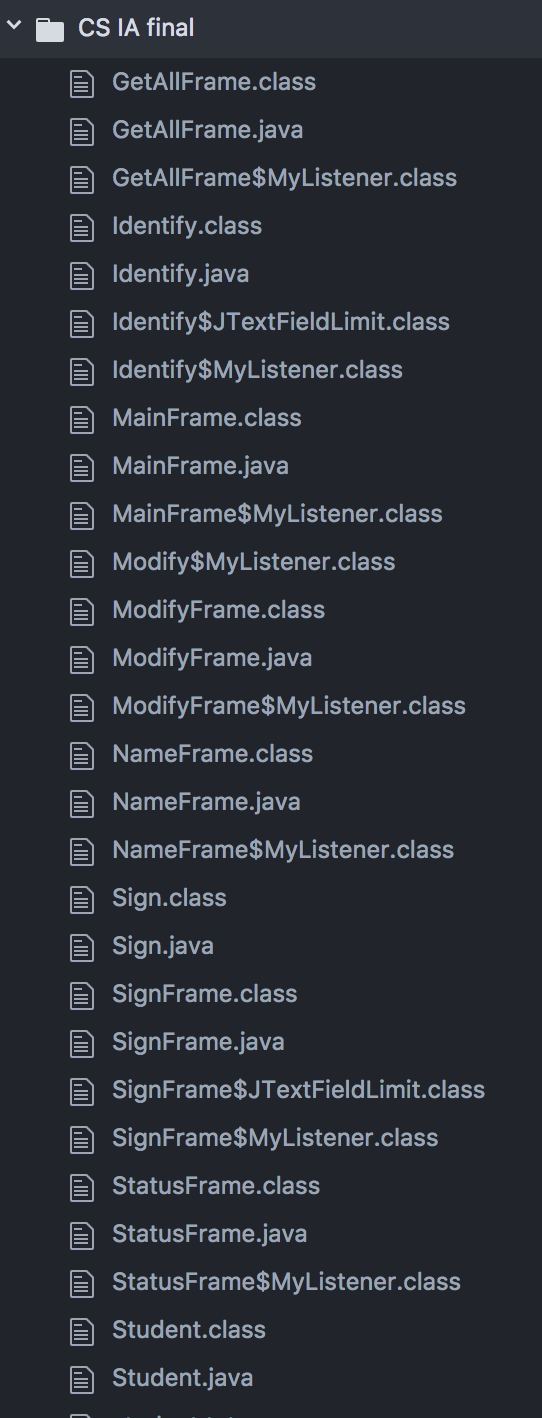
Criterion C: Development

**All classes:**



**List of the techniques:**

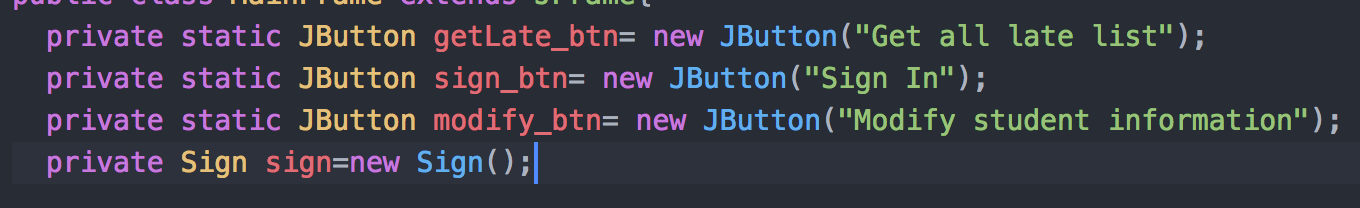
1. Each class extends JFrame to create controls and place them in the window
2. I created my own JFrame class and made my own methods in different frame classes for their different purposes
3. for loop
4. HashMap and ArrayList
5. JComponent and Panels
6. ActionListener
7. JTextFieldLimit
8. Scanner and File: student.txt
9. Static methods and variables

**Main Window:**

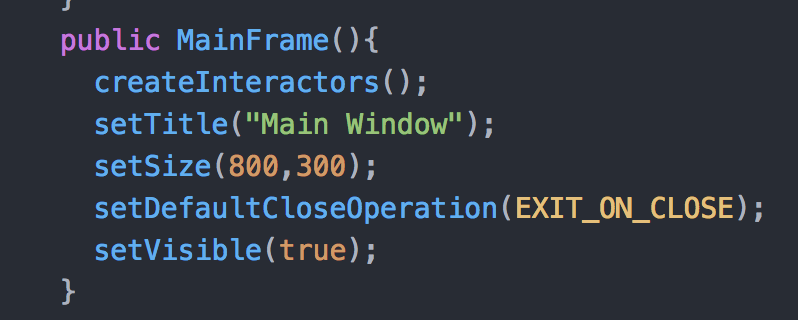


MainFrame class extends JFrame:

Use of instance variable:



Use of constructor:



createInteractors method:



MyListener class implements ActionListener:

* Add meaning and instructions to the buttons so that based on which button the user is able to decide what to do.

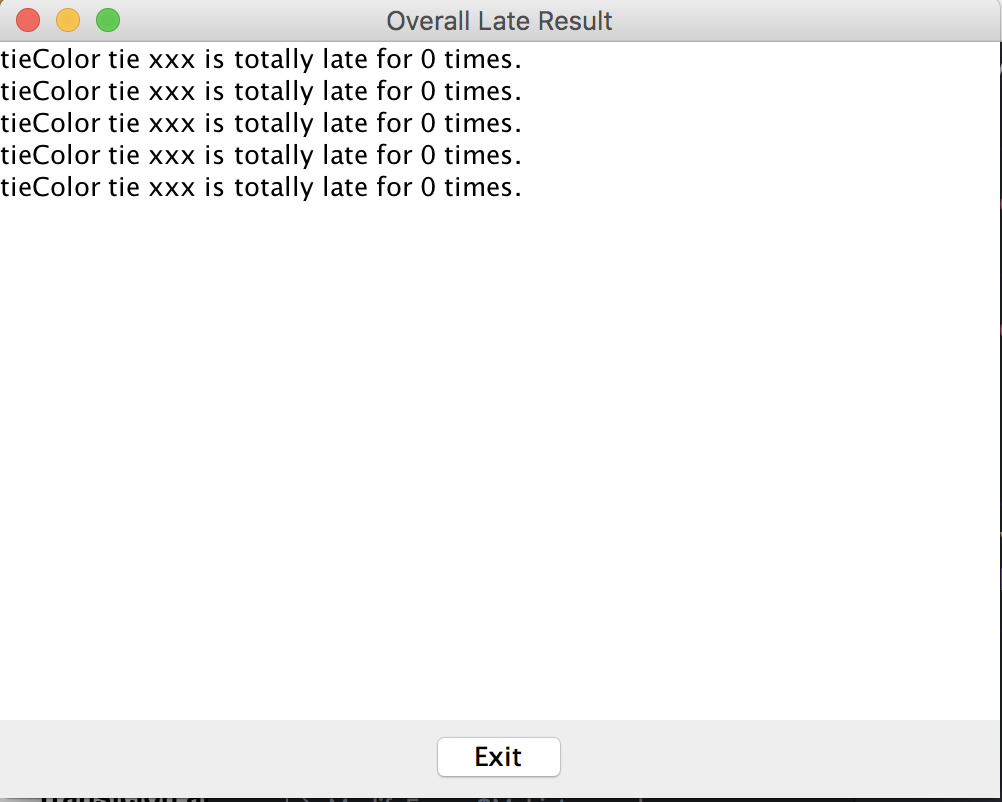


actionPerformed method:

* Clarifies what action to be taken for each button

**Overall Late Result Window:**

When clicked “Get all late list”, a GetLateFrame object variable is declared and create the following window with an “Exit” button:



GetLateFrame class extends JFrame:

* The window shows the total times a student has been late for dinner in a period of time. The “Exit” button will allow users to go back to the Main Window.



MyListener class implements ActionListener:

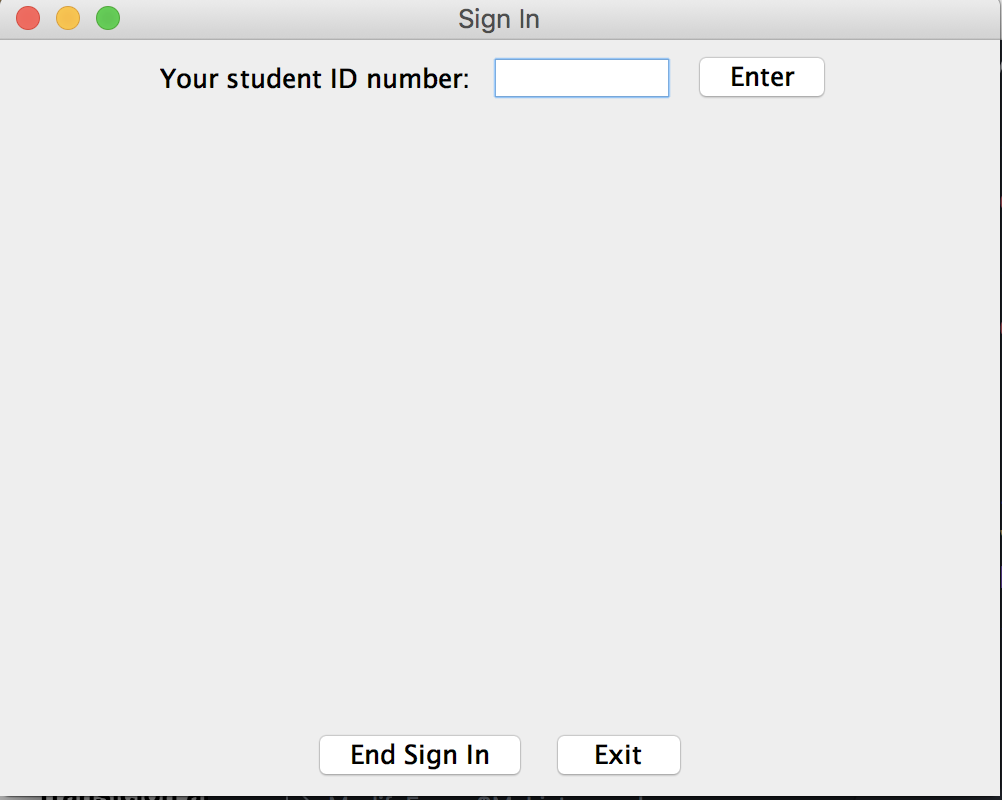
actionPerformed method:

* Allow users to go back to the Main Window



**Sign In Window:**

When click “Sign In,” a SignFrame object variable is declared to display the following interface window with a text field that the user can type numbers and three buttons: “Enter,” “End Sign In,” “Exit.”



SignFrame class extends JFrame:

The use of private variables for the JComponents in the interface:



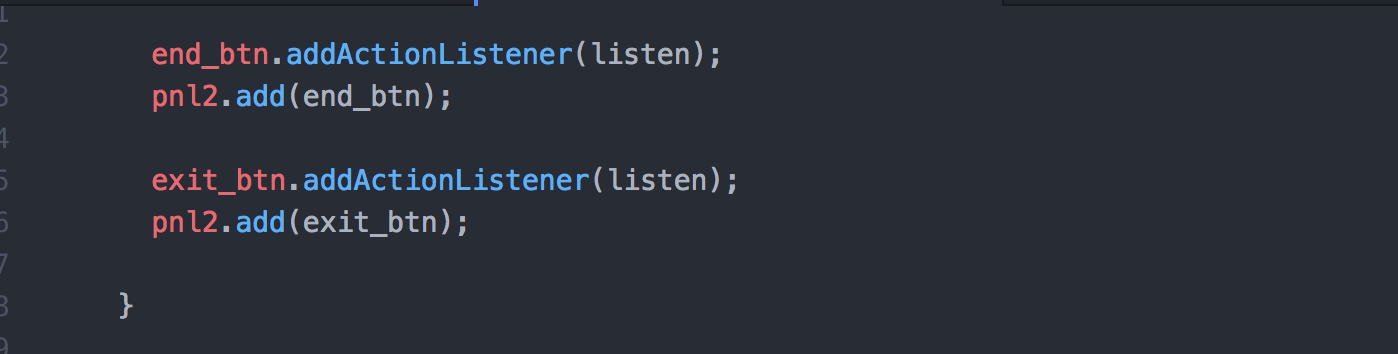
The use of constructor to set the interface size and to be visible:



createInteractors method:

* Set up the JComponents in the window and add ActionListener to add meaning and instruction to each button





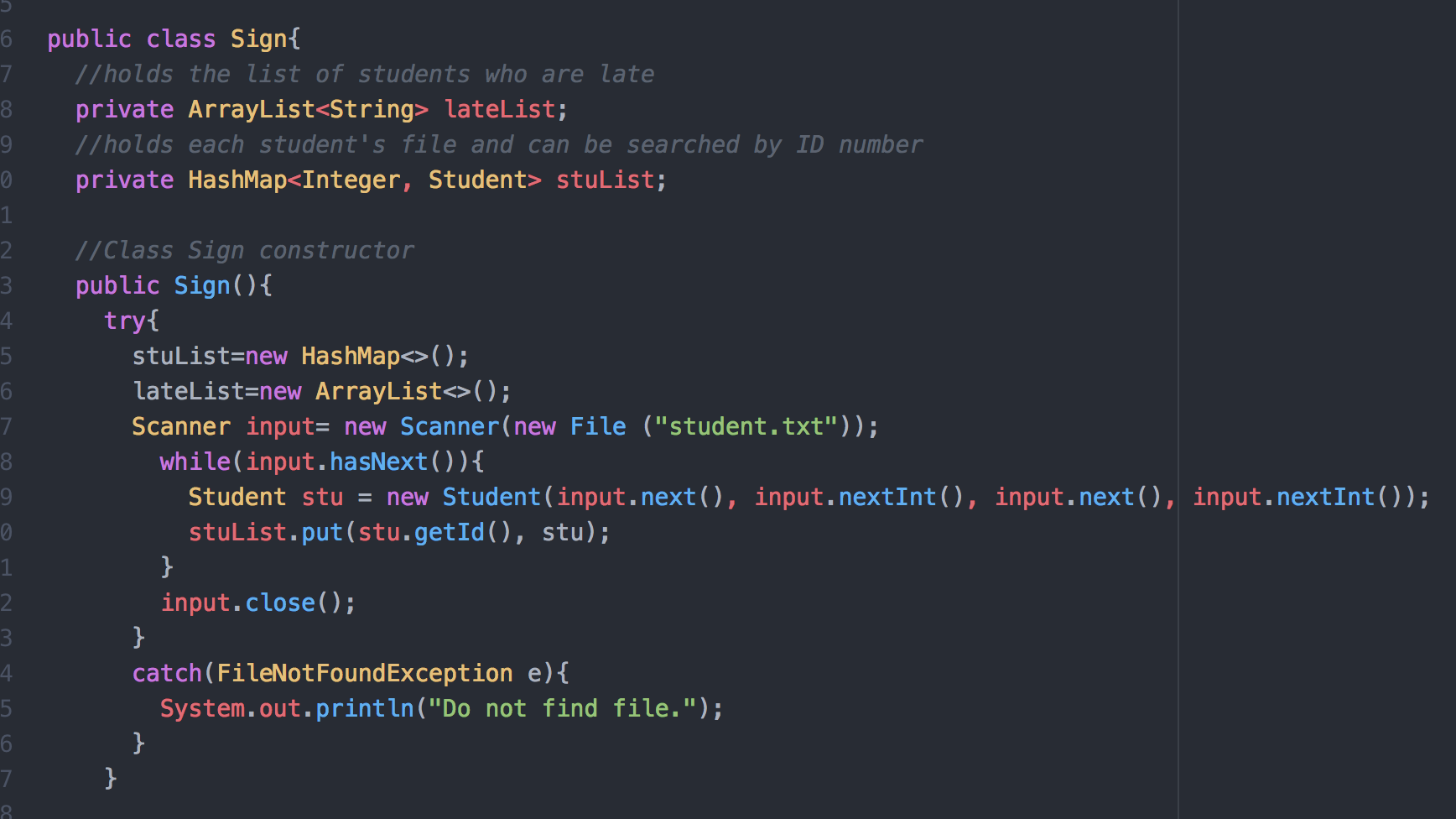
MyListener class implements ActionListener:

* implements from ActionListener add instructors for each JButton when click.



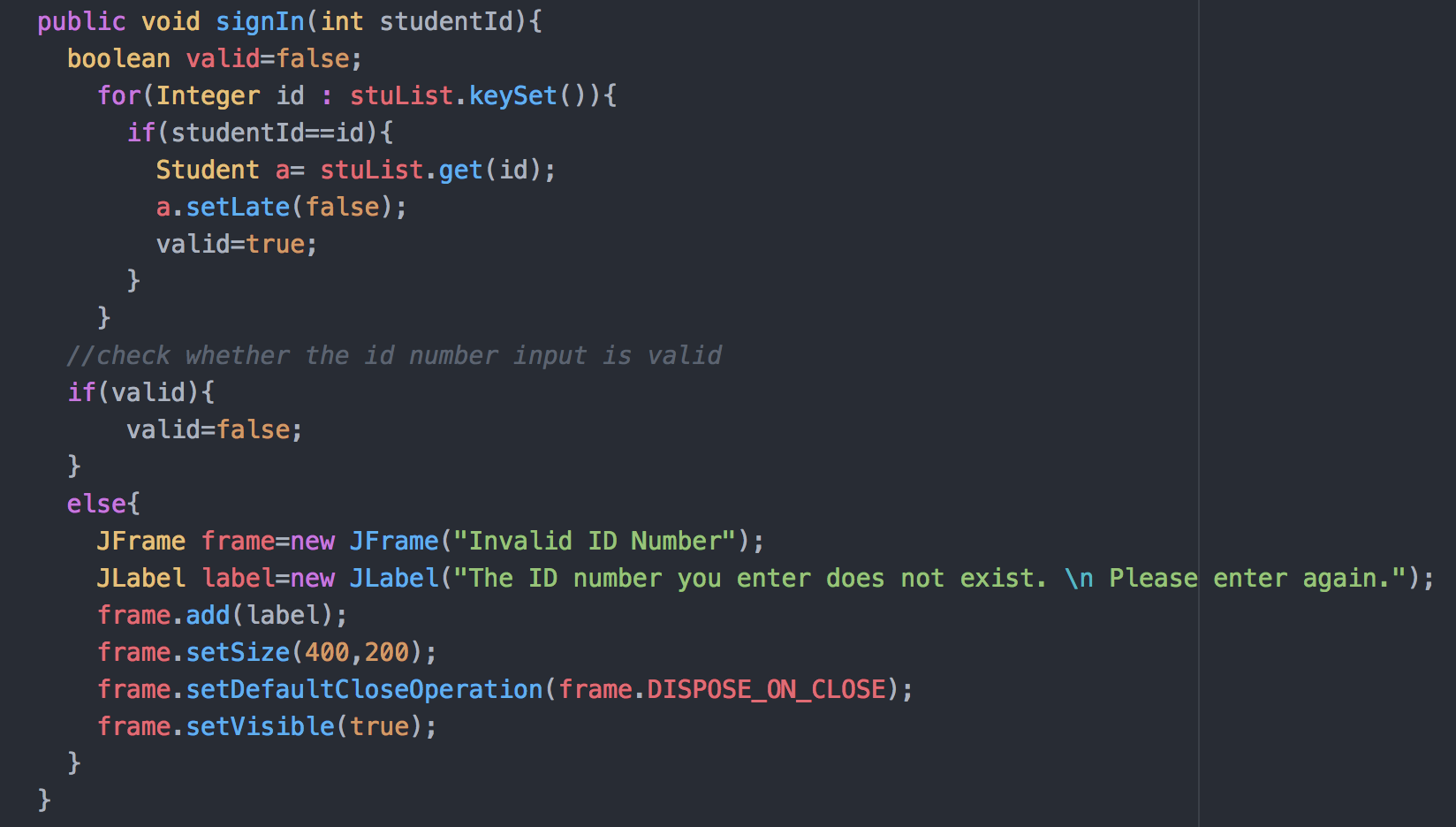
Sign class:

* I create a Sign class to handle student sign in when the user input a student number in the JTextField I have declared.
* The constructor catches when the text file does not exit by printing out “Do not find file.”

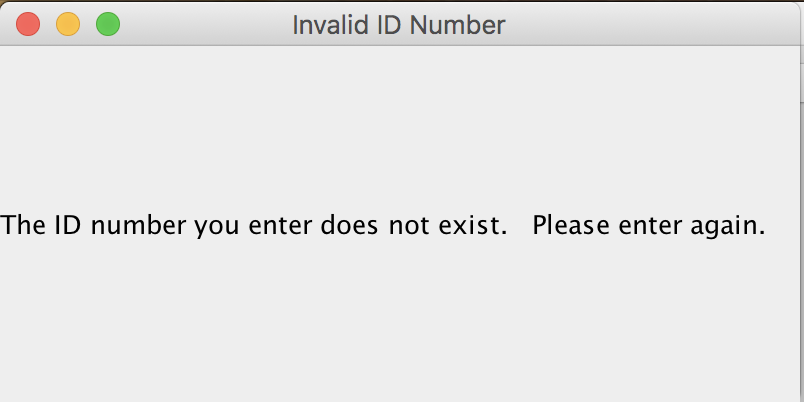


signIn method:

* By clicking “Enter” button or the return key on the keyboard will allow a method in Sign class called signIn to sign in students.

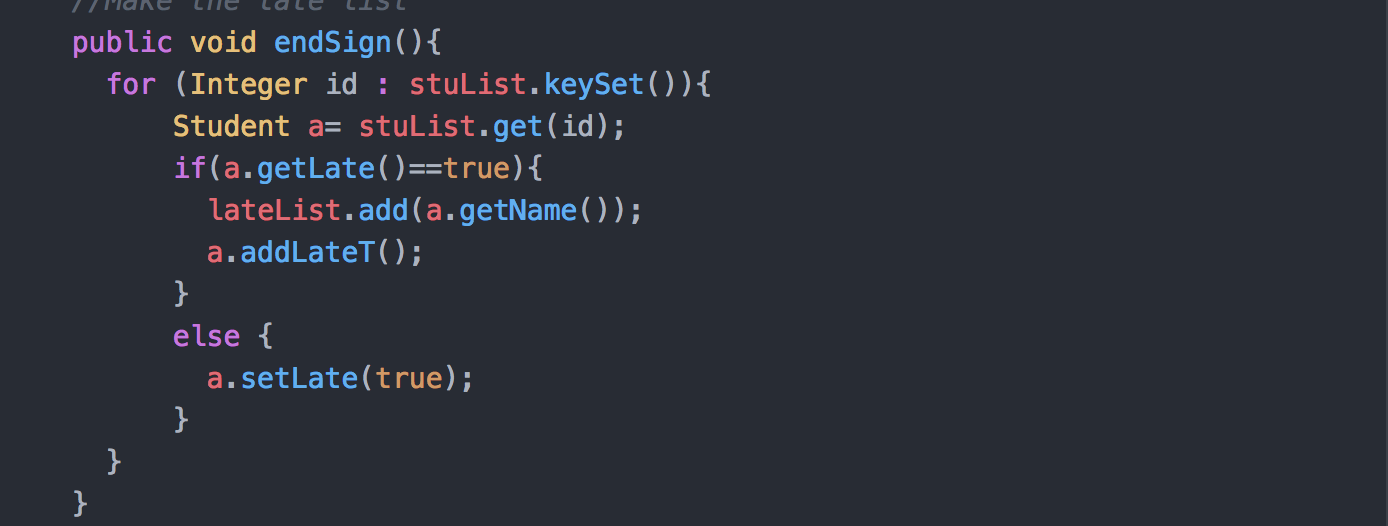


* The method catches error entry from the user by prompting a new window saying the number entered is invalid.



endSign method:

* it is called when the user clicks “End Sign In” button in Sign In Window, so a lateList is made and stored as an instance variable in Sign class



getLateList method:

* Then, SignFrame use getLateList method in Sign class to display the late list in a new window as following



If the user clicks any “Exit” button, it will bring the user to the previous displayed window.

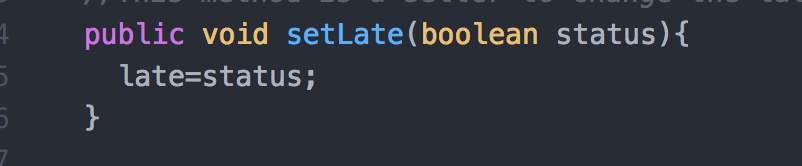
Student class:

* I created a Student class so that the user is able to get and modify student information stored in the Student object with its method



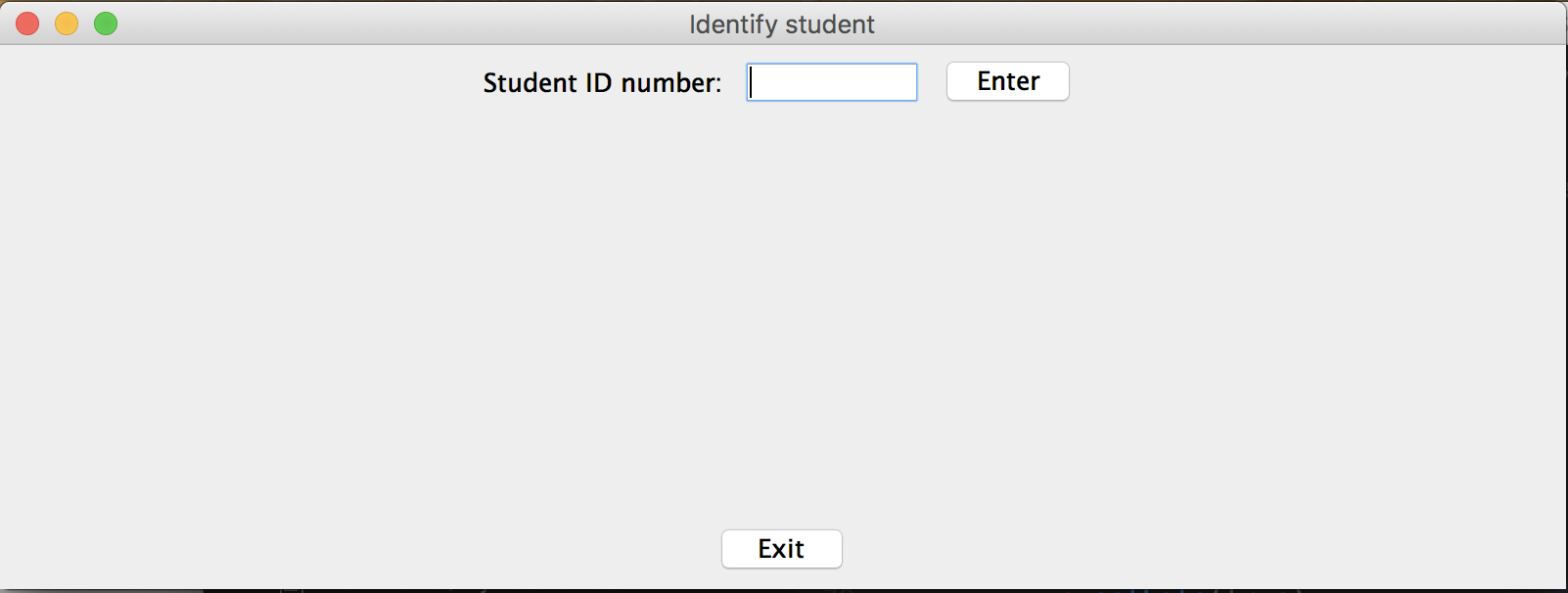
setLate method:

* Let the user to change the late status of the student with the correlated ID number to fasle (meaning not late)



**Modify Window:**

* When the user clicks the “Modify student information” button, Identify class will be declared to display the following new interface window for users to input the student ID number to identify which student’s information they wish to change.



Identify class:

* Sets up what the interface should look like

****

createInteractor method:

* Add the components into the window frame

****

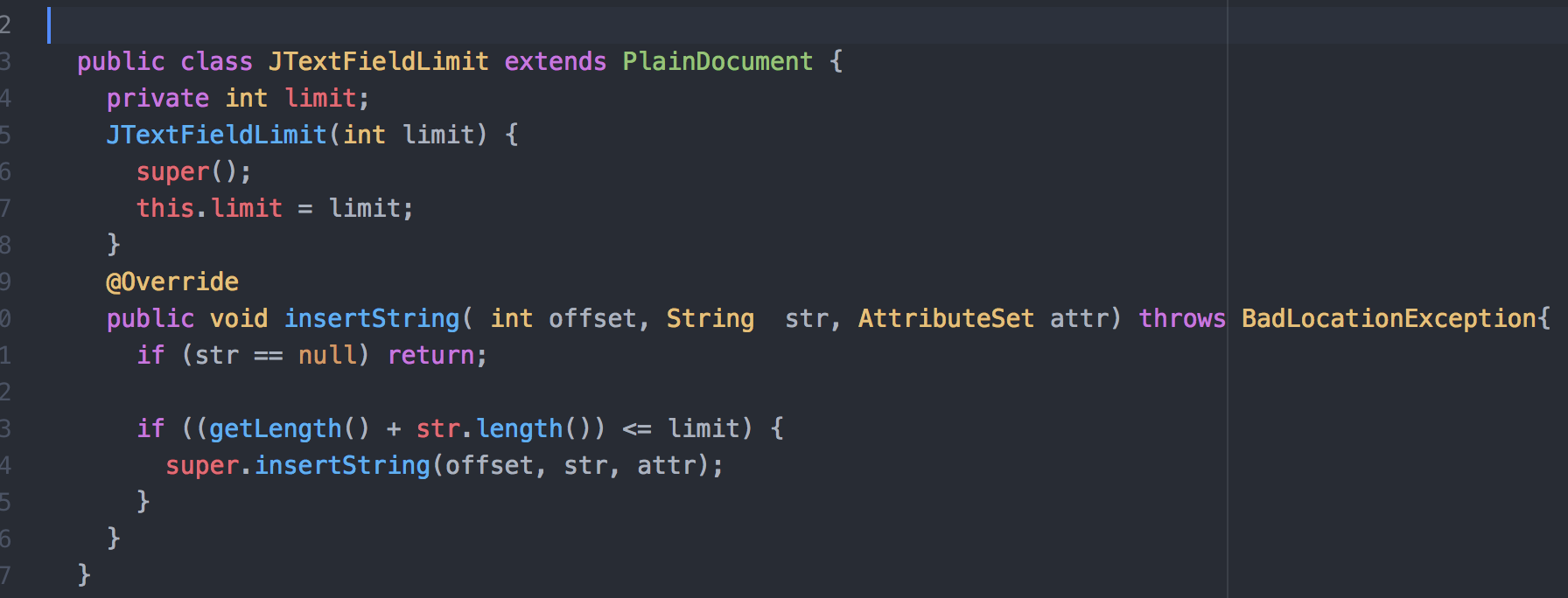
MyListener class:

* Allows the buttons to have meaning so that depending on the button that the user has clicked different method will be called.

****

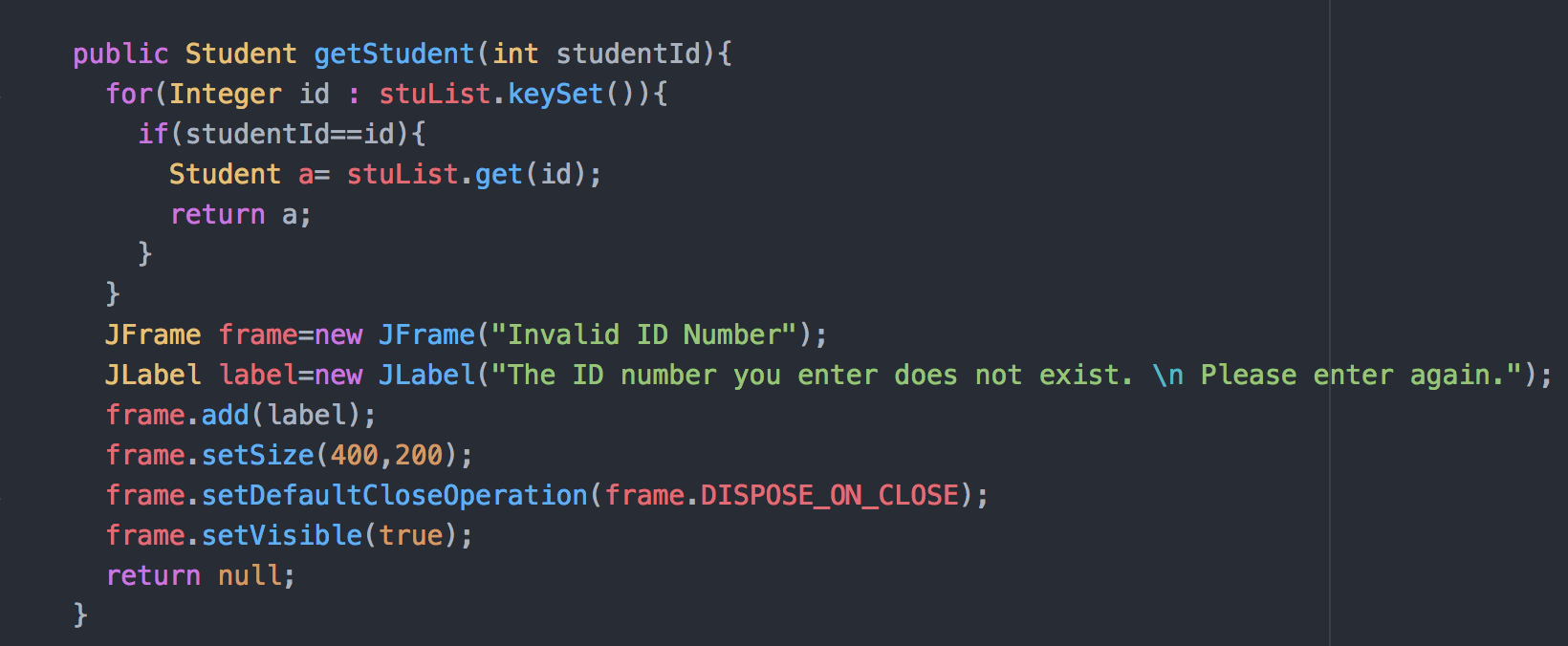
JTextFieldLimit class:

* extends from PlainDocument class to prevent user from inputting more than seven digits number (as client has requested)

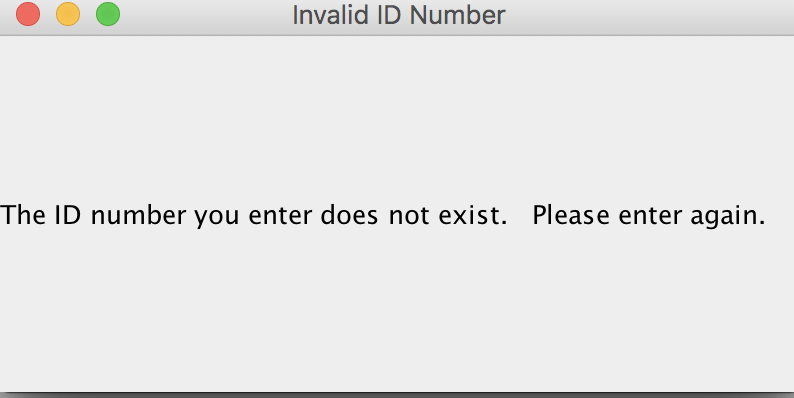
****

getStudent method:

* After inputting the student number, getStudent method in Sign class will catch error entry if the user types in a number that is not valid.

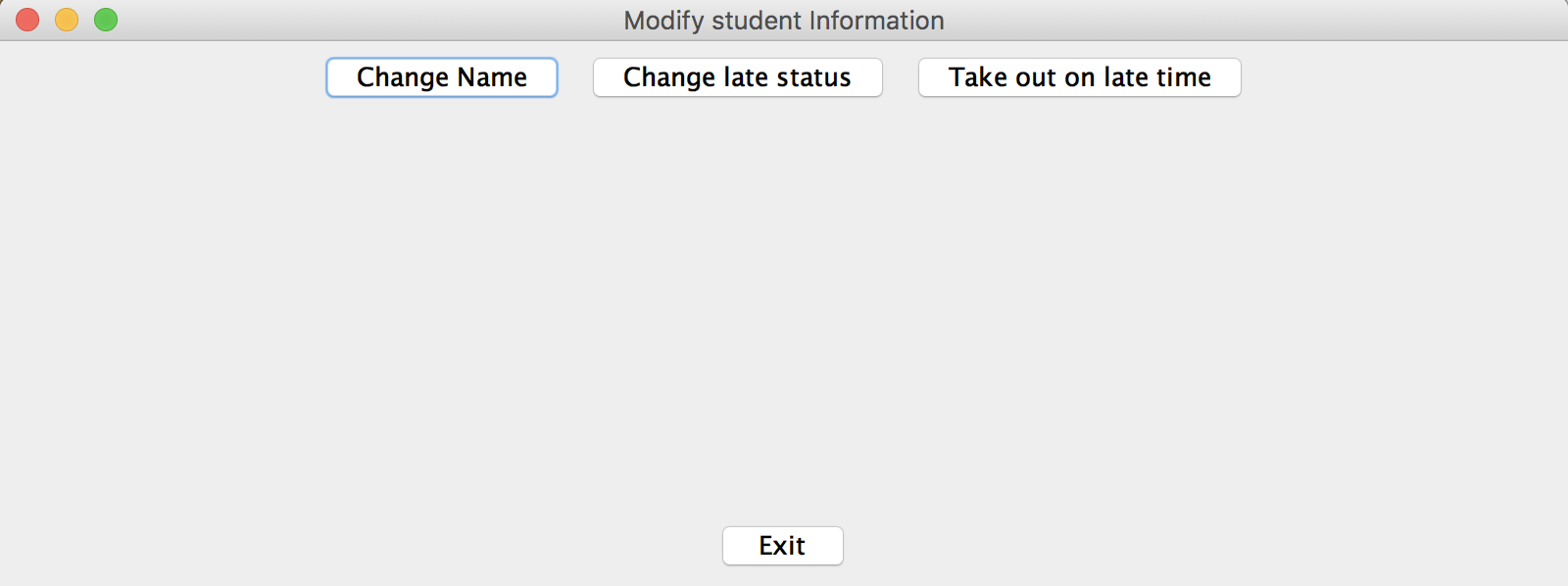
****

* When an error entry is inputted, it will show as following:

****

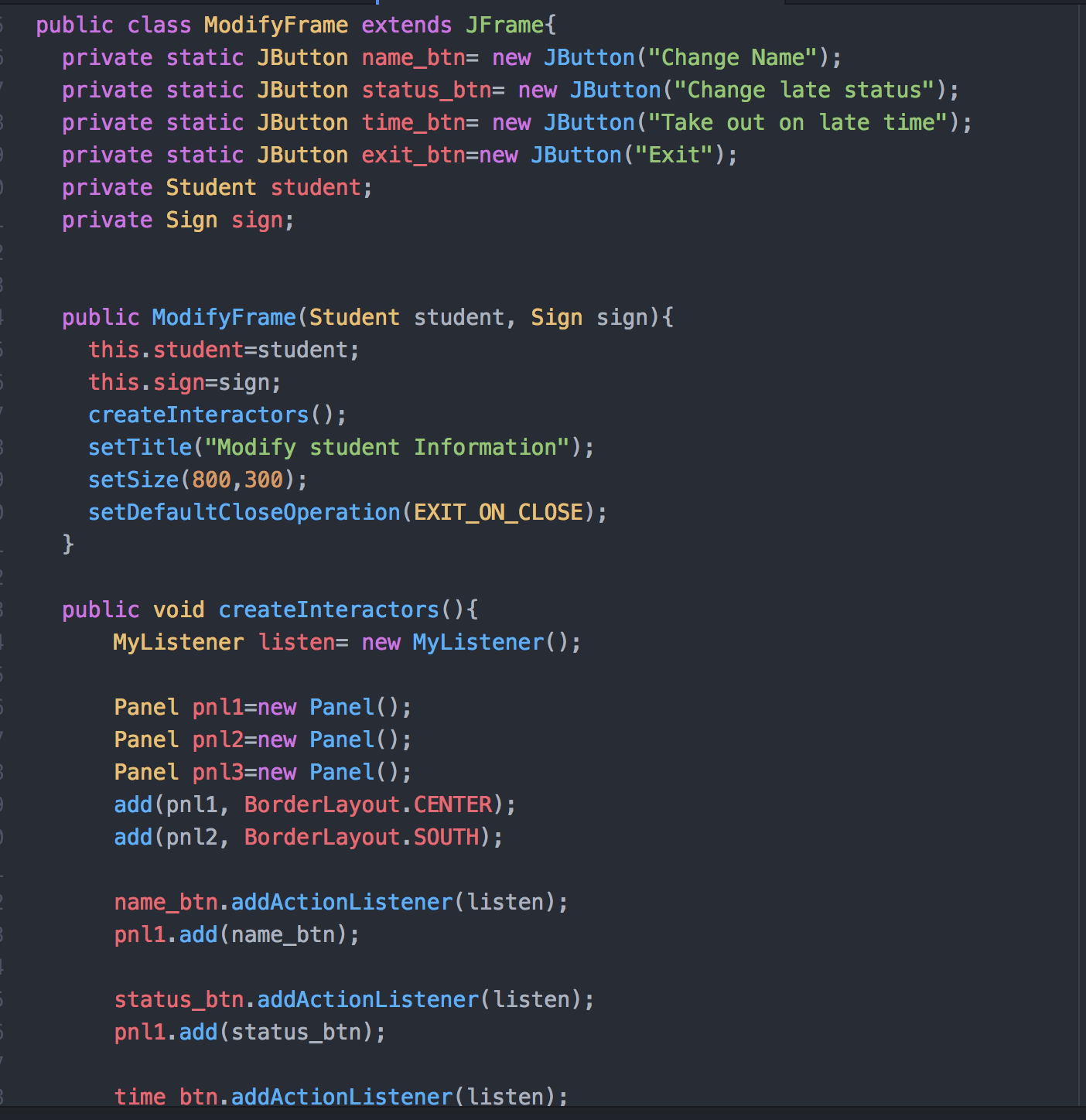
**Modify Window:**

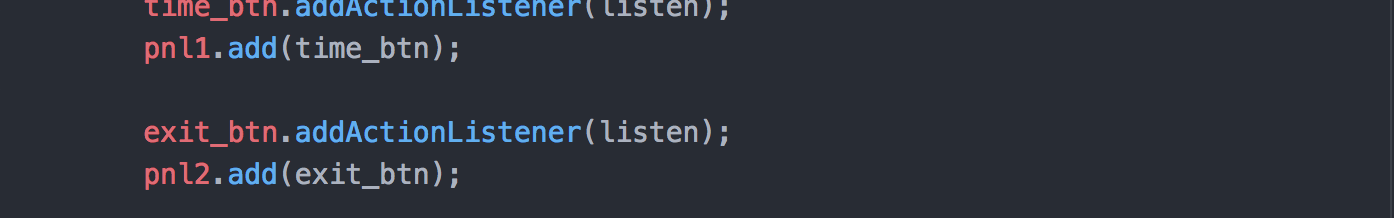
After entering the valid number, a new window will display to ask the user what to change by creating a ModifyFrame class:



ModifyFrame class:

* Sets up the interface window



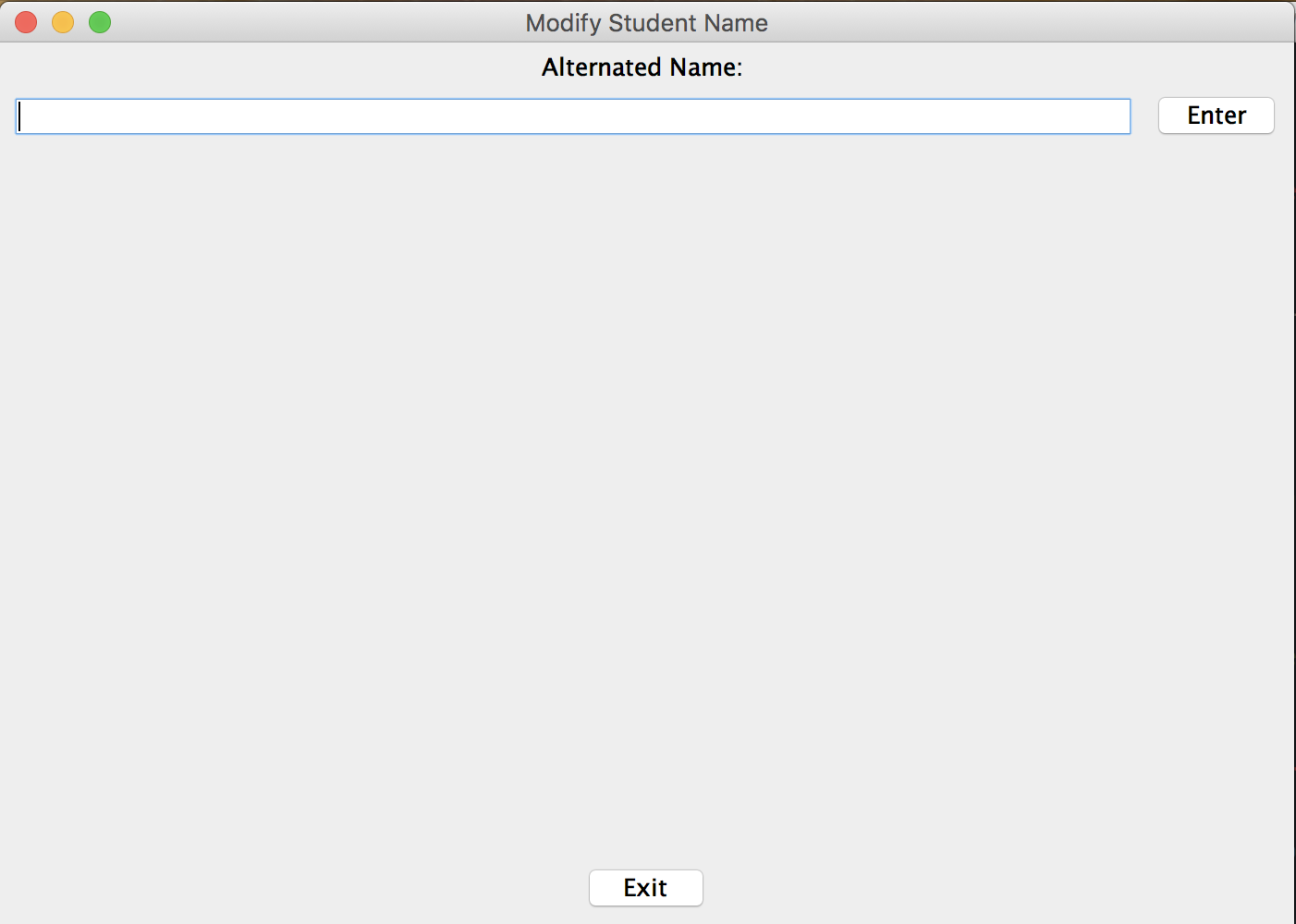
****

MyListener class:

* Allows the buttons to have meaning so that depending on the button that the user has clicked different method will be called.

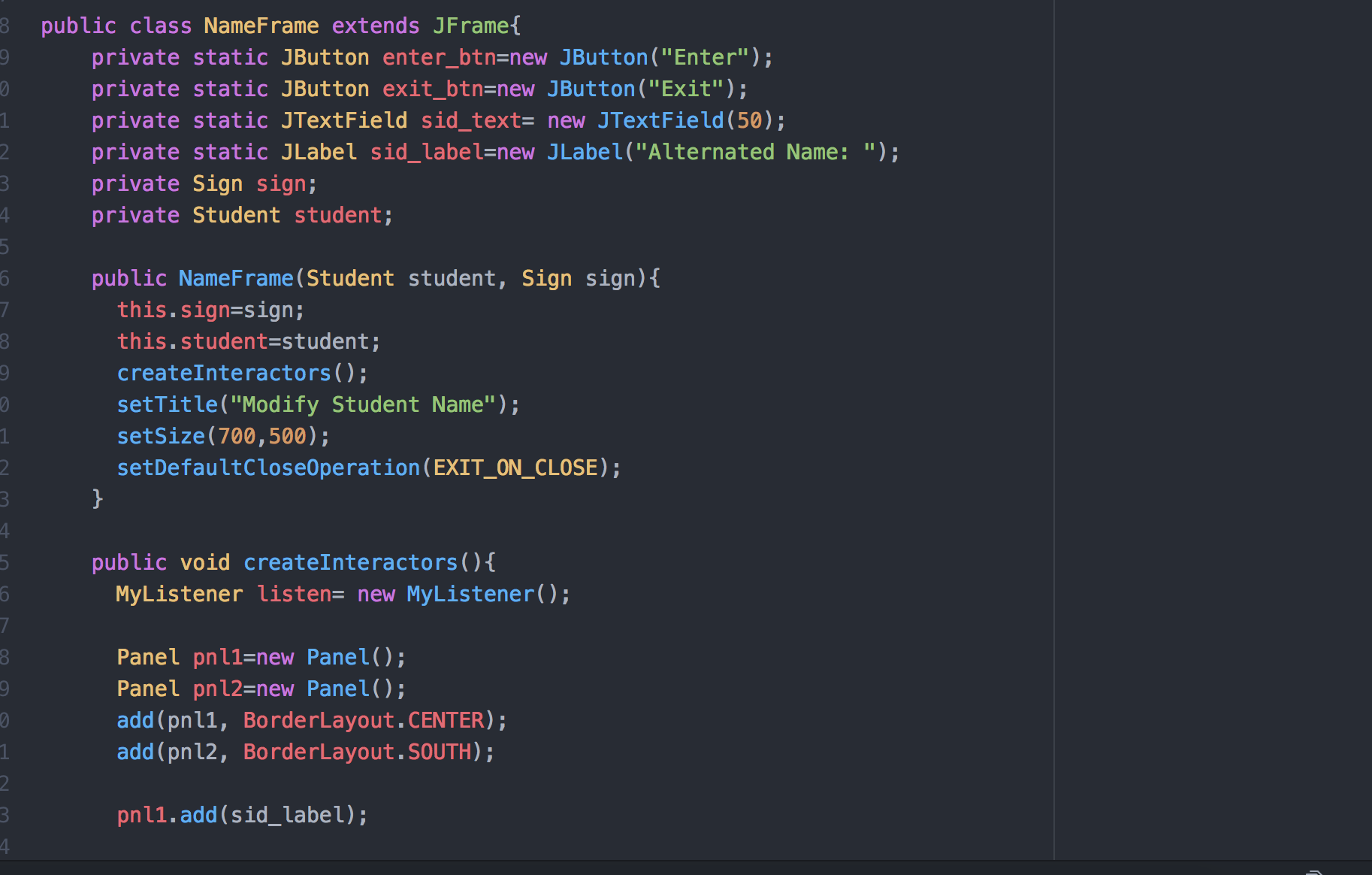


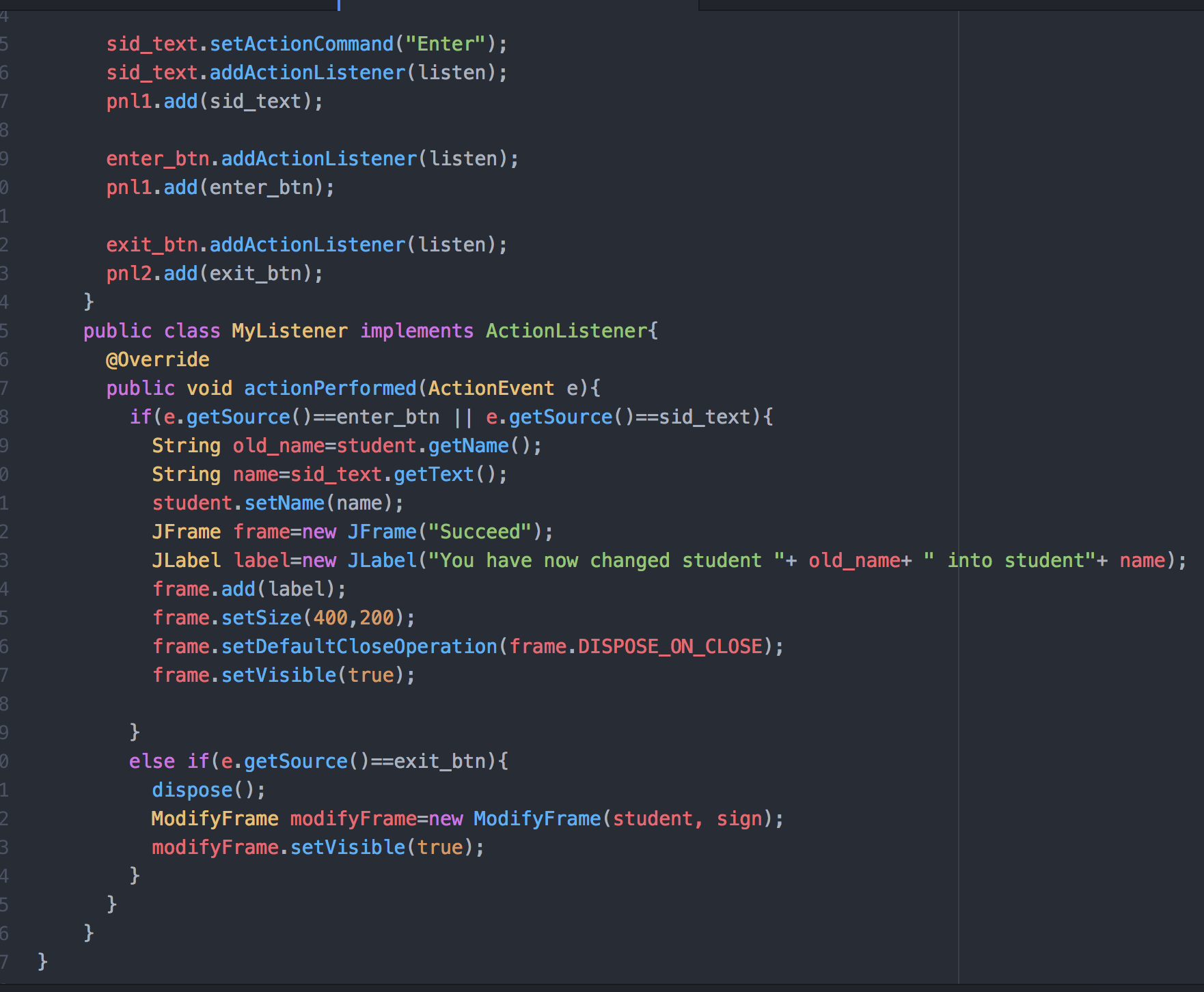
If the user clicks “Change name,” the following window will be displayed by calling NameFrame class



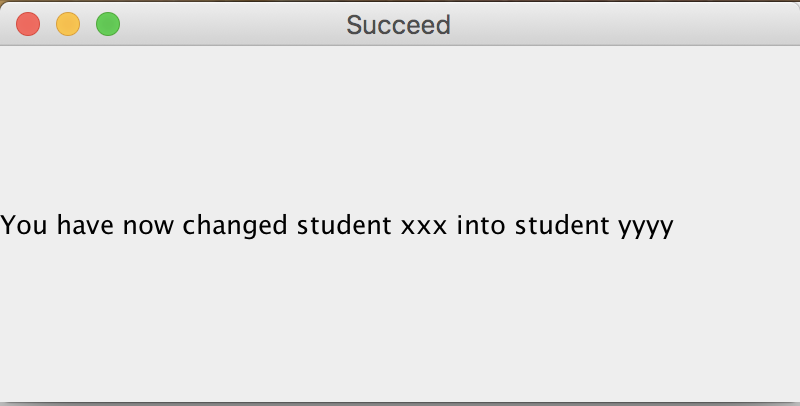
NameFrame class:

* When the user enters the new name, it will store the change to the student’s Student object and display another window to show the change made

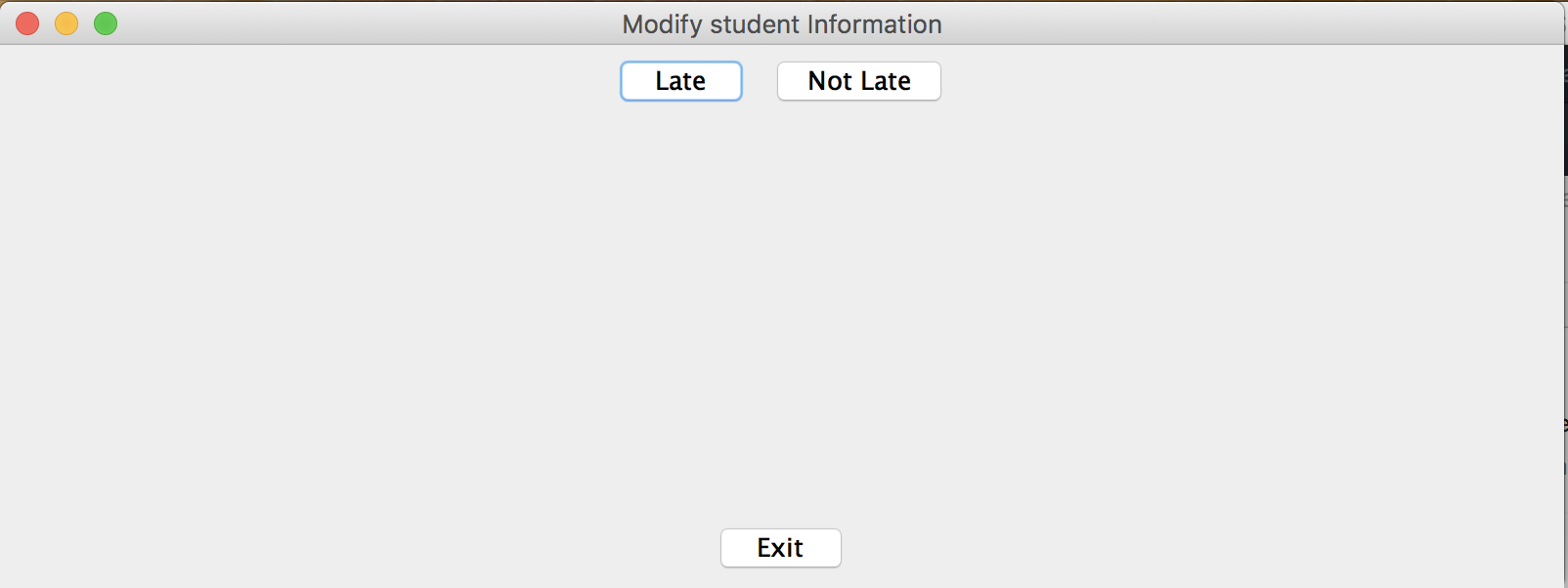


****

* If I type “yyyy” in the text field, I change “xxx”(default name in my program) and this following window will be shown meaning the change is successfully done.



If the user clicks “Change late status” button, StatusFrame class will be declared to create a new interface window as following:

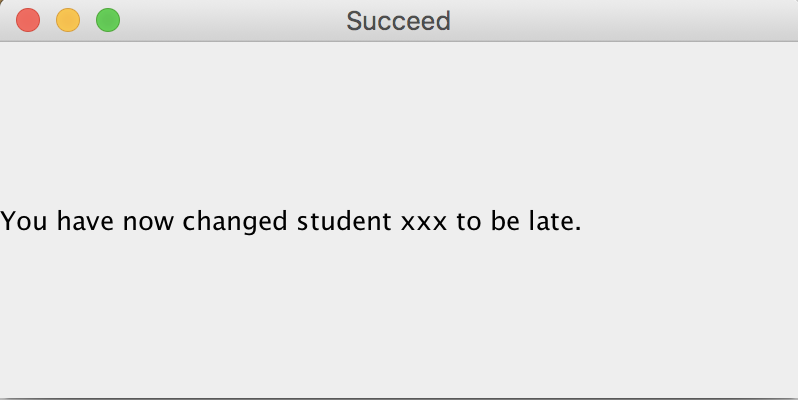


StatusFrame class:

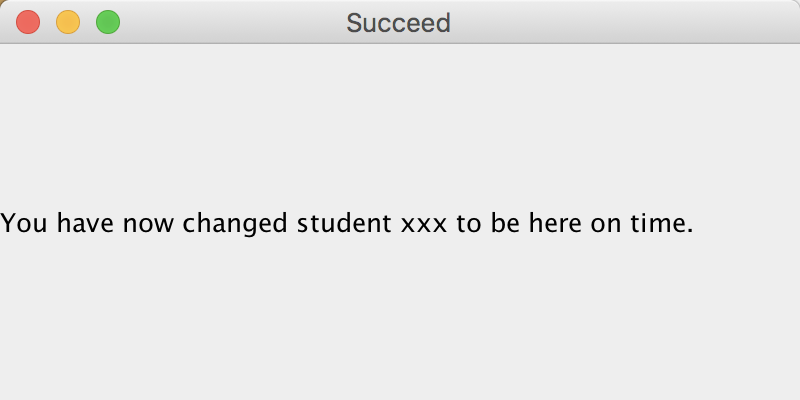
* Set up a new interface window to let the user to control what to do



* Will give different window that display the confirmation of the change the user made
  + If click “Late”



* + If click “Not Late”

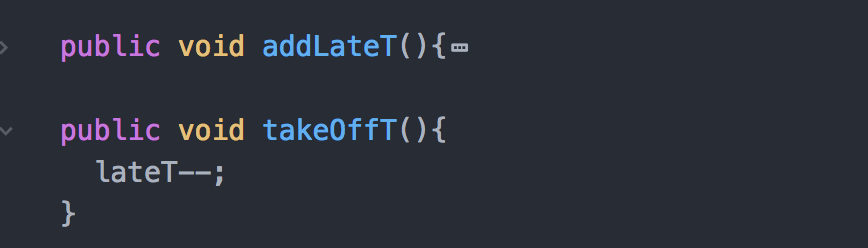


MyListener class:

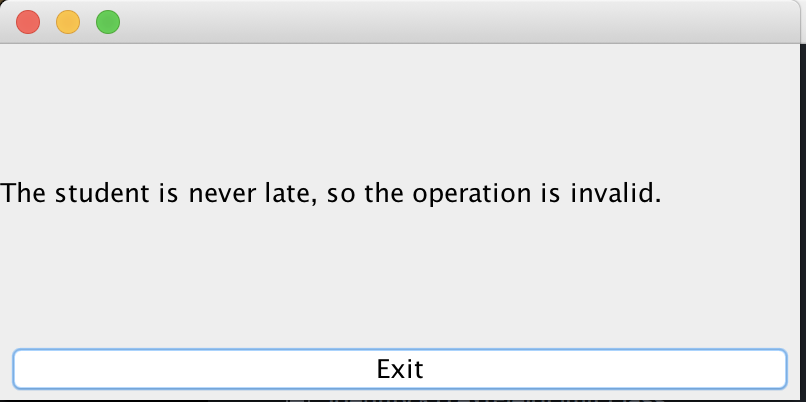
* Allows the buttons to have meaning so that depending on the button that the user has clicked different method will be called.



If the user clicks “Take out one late time” button, the student’s late time will be subtract by one by using the takeoff method in Student class



However, if the student total late time is zero, the following window will appear



If the users want to end the program, they have to click on the exit button on the tool bar. The “Exit” button I creates will only allow user to go back to the previous window they are at.