# Assignment 02 – JavaRev (version 2.0)

Individual Assignment

Assignment announcement: Sunday, 24 February, 2018

**Version 2.0 is due on Friday, 1st March 13:00 hrs thru a zip file of java code files.**

**Marks: 10 Marking would be done thru demo of code. A list of demo timing will be announced**

**Note: Late assignment will not be marked at all.**

You have to make a Java Code Static Analyzer. The application you are going to develop is called Java Code Reveler (JavaRev). The working of the application is explained by user stories and working examples. This would be developed in iterations and new features will be added to each version.

**Version 2.0.0**

A user opens the JavaRev application.

* **The application presents a simple GUI Frame menu to the user. The frame should have two J Menus. File and About. File Menu should have Open folder ,close folder ,exit as menu items.**  (GUI feature 01)
* About should open a JDialog with developer’s name, email address and student id number. These details should come from Java properties file (use java.util.Properties) (GUI feature 02)

The main area of the frame should have three text areas with labels on top and text area should show the information required by the user on the files in the folder.

1. (Feature 1) List of all files with complete path and extension
2. (Feature 2) List all java files in the folder with complete path and extension
3. (Feature 3) Present following information for each java file: **complete name of the public class, complete name** of the public methods in the file. (Name and args only for now)
4. (Feature 4) Application should record all its interaction in a log file (log.txt) at back end. The log file should note when application was started and when was and which folder was accessed.

Application Working:

Application starts and the user wants to choose a folder, he/she clicks on File-Open Folder menu and a JFileChooser dialog box opens up. He/she should use JFileChooser to make his choice of folder and after that the application should immediately present the information about the folder in the given text areas. (See figure for frame’s reference). The user should be able to quit application

Design Considerations:

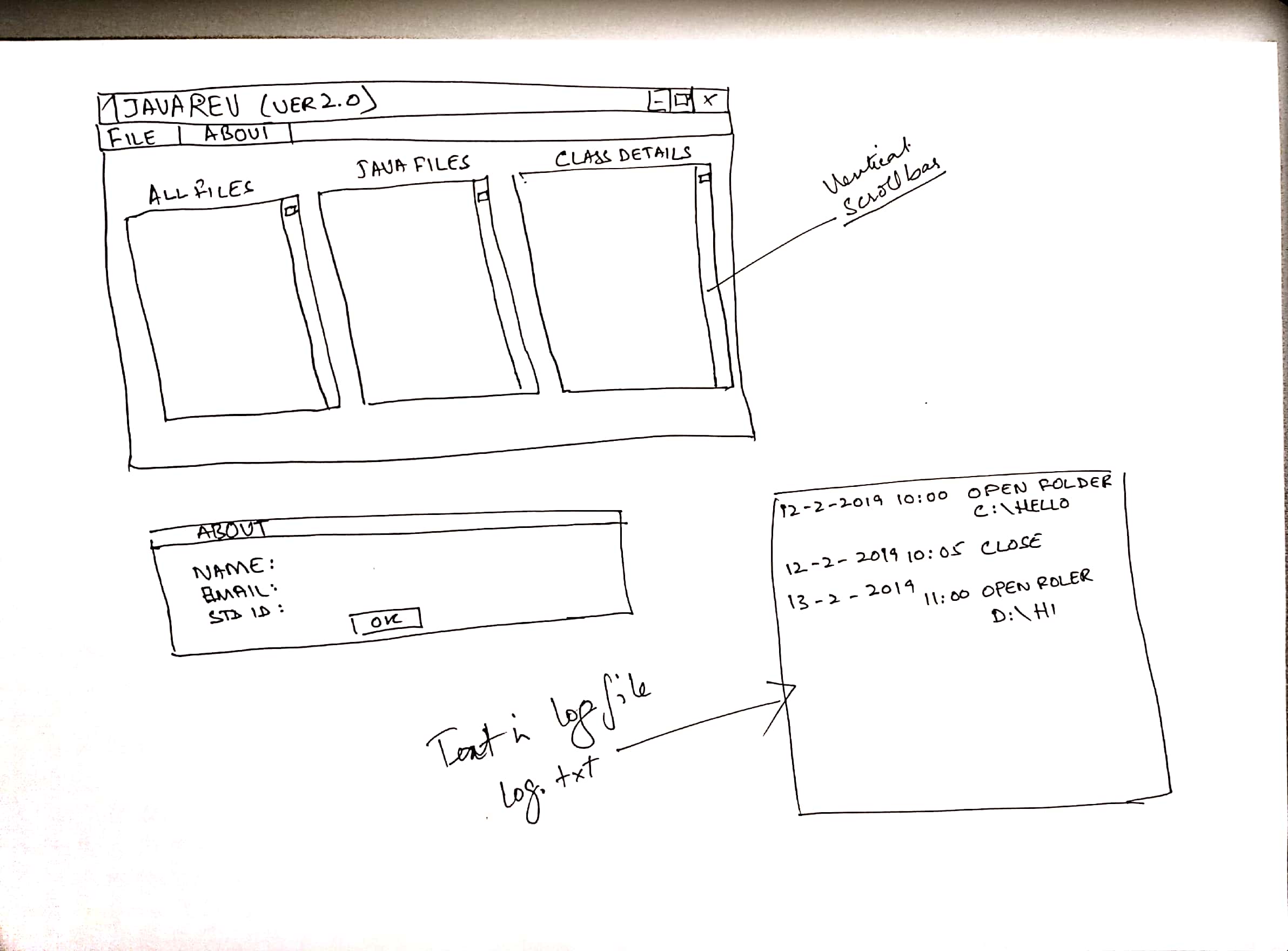
* Java codes should be segregated in packages against their concerns (business logic, user interface, persistence etc.
* The GUIFrame should have layout manager (gridlayout manager could be an option) that should ensure that the look and feel is not destroyed once application frame is resized. You should build on your previous version and add new functionality.
* Your code should be written in such way that minimal code is in JFrame and a special class called GUIController interacts with the GUI and the application layer. (Note there should be no business logic processing in the Frame code)
* Feature four could be applied in plain simple text based filing. If you want to us can use java.util.logger but you will not get any additional marks for this.
* All work should be written in classes in such way the future enhancement can be done in class of the application layers.
* **Appropriateness of work be: Application should have all four features working**

Code Submission:

Note this is individual assignment and a same level question might come in exam thus do it on your own using your own skills. If you use code from internet or tutorial sites, do give detail in code comments.

Marks would be deducted if there is no commenting in the code. **BTW, You have to write your own JFrame code not and use drag a drop feature of IDE or GUI builders to make components. looks and feel is required.**

Application Mockup



Note:

By definition complete name of the class or method is it signature, but for now you can only take the name of the class and method.

public class MyWorld{

public void hello(){

}

public void foo (int i){

}

public String myShow(String a, String b){

return a+b;

}

}//enc of class

should show

Public Class: MyWorld

Method

hello args none

foo args Int i

myShow args String a, String b

For more detail on method signature:

https://docs.oracle.com/javase/tutorial/java/javaOO/methods.html