

# Diary - First Iteration - COMP 354 - Software Engineering

Vincent Boivin

February 11 2018

## **1 Jan 18,2018 (10:40 PM - 12:40 AM)**

Created the driver class called App354.java and created the class Panel.java which extends JFrame. I added the layout, the user text field(JTextField object called userArea), the password text field(JTextField object called passwordArea) and a "log in" button(JButton object called btn). I pushed the first version of App354.java and Panel.java to the github repository master branch.

## **2 Jan 18,2018 (9:50 AM - 10:56 AM)**

I modified the panel class by adding two new JLabel objects and modifying the password text field called passwordArea. The first JLabel object added is called "user" and the second JLabel object added is called "pass"(short for password). The password area has been changed from a JTextField object to a JPasswordField object, this will allow the user to enter his/her password without displaying it in the text area. I pushed the modifications to the github repository master branch.

## **3 Jan 19,2018 (12:40 AM - 2:30 AM)**

I modified the panel class again. I added a JButton object called cancelBtn. I also added a BufferedImage object called appIcon. The appIcon will use the image called "money.png". Finally, I added two methods called "closeOnCancelClick" and "setIconImage". The method closeOnCancelClick will close the application when the user click on the cancel button. The method setIconImage can be used to modify the icon image on the top left corner of the JFrame by providing a BufferedImage object and a path in the form of a String object. I pushed the modifications of Panel.java and the "money.png" image for the icon to the github repository master branch. I modified the panel class by adding a JMenuBar object called "menuBar", a JMenu object called "menu" and a JMenuItem object called "menuItem". I added the method "placeMenu"

to the panel class. The placeMenu method will accept a JMenuBar object, a JMenu object and a JMenuItem object as its parameters and will add them to the JFrame. I pushed the last modifications to the github repository master branch.

#### **4 Jan 19,2018 (9:30 AM - 11:10 AM)**

I changed the name of the class Panel.java to LogInPanel.java. In the LogInPanel.java class I added the "placeButtons", "setupForPanel" and "logInOnClick" methods. The placeButtons method is using the overlay to place all the JButton objects on the JFrame. The setupForPanel method is used to set various attributes of the JFrame such as the size,the icon image, etc. The logInOnClick is a placeholder method that will be replace to another method. When you click on the log in button, the logInOnClick method move you to the FinancePanel JFrame. I added a FinancePanel.java class which extends the LogInPanel. The finance panel will be used for use case 3. I pushed the LogInPanel.java and its modifications and the FinancePanel.java to the github repository master branch.

#### **5 Jan 19,2018 (11:40 AM - 12:20 PM)**

I changed the name of the class LogInPanel.java to MainPanel.java and I created the BankPanel.java class which will be used for use case 2. The BankPanel.java extends MainPanel.java class. I pushed the modifications to the github repository master branch.

#### **6 Jan 30, 2018 (4:15 PM - 5:30 PM)**

I looked for a java library to easily make graphs or charts. I found the xchart library on github. I read up on the library and saved the github address for later. The github address is : <https://github.com/timmolter/XChart> .

#### **7 Jan 30, 2018 (9:30 PM - 11:00 PM)**

As a team, we presented the project to the TA at the lab.

#### **8 Feb 8, 2018 (12:30 AM - 6:15 AM)**

I modified the FinancePanel class by adding JLabel objects,JPanel objects,JComboBox objects, JButton objects,JTextField objects and PieChart objects imported from the outside library xchart. In the setupForPanel method, I placed all the previously mentioned objects on the FinancePanel JFrame by using the GridBagLayout. I created the FinanceController.java class which will make the

finance panel interactive. I added the "resetBudget" method, the "resetSpending" method, the "createBudgetChart" method, the "createSpendingChart" method, the "getBudgetDropInfo" method, the "getSpendingDropInfo" method and the "setup" method. The resetBudget method will assign the value 0 to all the fields corresponding to a category(home,food,hobbies,savings,other) on the budget side when the user click on the JButton object called resetBudget. The resetSpending method will do the same thing as the resetBudget method, but it will do it for the spending side of the panel by clicking on the JButton object called resetSpending. The createBudgetChart method will create a pie chart with the corresponding values from the budget side. The createSpendingChart will create a pie chart with the corresponding values from the spending side. The getBudgetDropInfo method will get the amount from the JTextField object on the budget side and the category from the JComboBox object on the budget side and add it to the corresponding category and to the total budget when the user click on the JButton object addToBudget. The getSpendingDropInfo method does the same action as the getBudgetDropInfo, but it does it for the spending side and when the user click on the addToSpending JButton object. The setup method is used to connect the FinancePanel class to the FinanceController class which will make the finance panel interactive. Then, I added FinanceController.setup() to the driver class App354.java to make the finance panel appear. Finally, I pushed the modifications to the App354.java class, the FinancePanel.java class and the new FinanceController.java to the github repository Iteration-1-Update branch and to the github repository master branch.

## **9 Feb 9, 2018 (7:00 PM - 9:30 PM)**

I created the FinanceTest.java class which will be used for testing on the FinanceController.java class. To unit testing on the FinanceController.java class I used the JUnit library and the Hamcrest library. I added comments and javaDoc comments to the FinancePanel.java class and to the FinanceController.java class. I added my tests to the App354.java driver class by using the JUnitCore class, a Result object called result and a Failure object called failure. Finally, I pushed the new FinanceTest.java class and the modifications of the FinanceController.java class and the FinancePanel.java class to the github repository master branch.