MCIT 591 Investment Planning Statement (IPS) Planner

PART I PROGRAM DETAILS

Completion Date:
5/10/2019

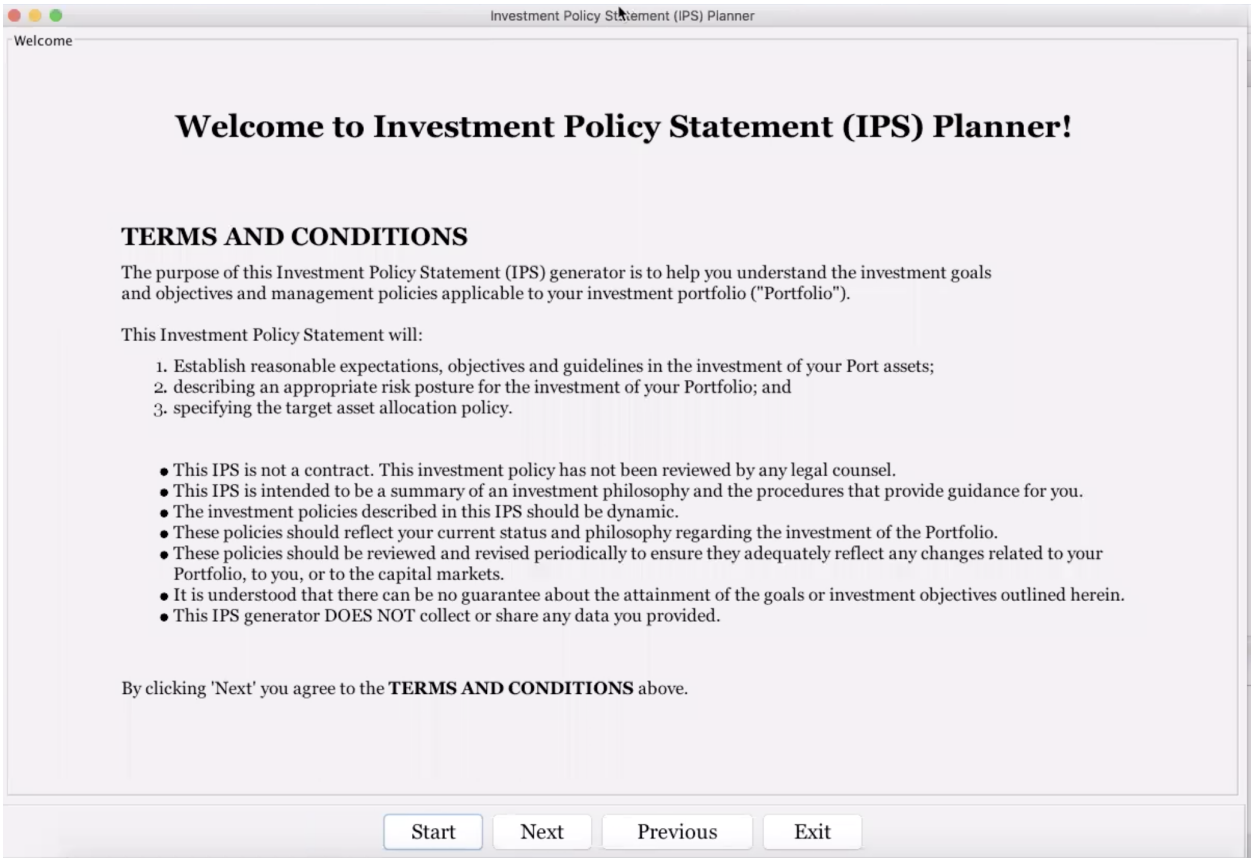
Authors:
Mengxue (Kristy) Li mengxli@seas.upenn.edu
Pradeep Ramachandran pradeepr@seas.upenn.edu
Yubei (Eva) Liu yubeiliu@seas.upenn.edu

PART II USER INSTRUCTIONS

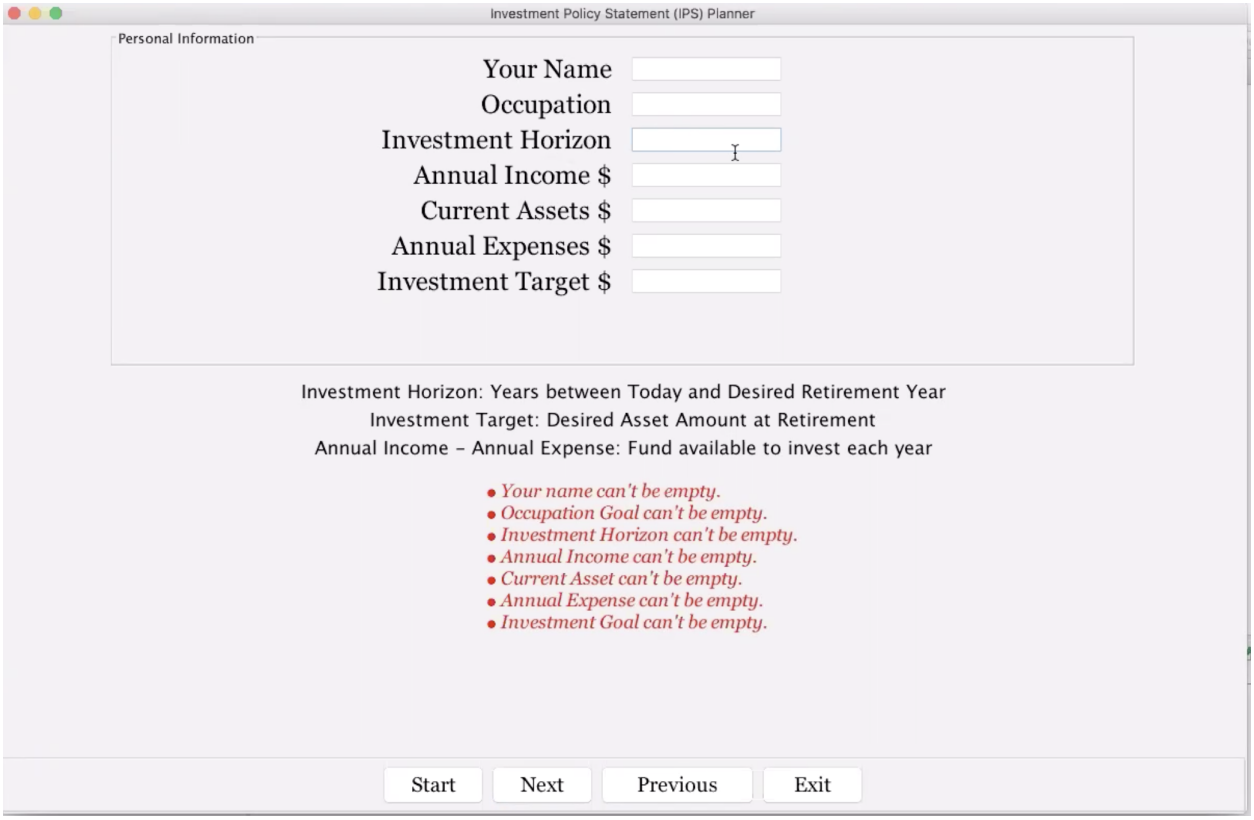
MCIT 591 Investment Planning Statement (IPS) Planner (" the Program") is to help the user understand the investment return and risk objectives, as well as providing recommended asset allocation and corresponding portfolio return and risks based on the user's risk tolerance. During the use of the Program, the user is able to perform the following functions with the buttons below.

"Start" to go back to the first panel of the Program
"Next" to go to the next panel
"Previous" to go to the previous panel
"Exit" to end the program during any step

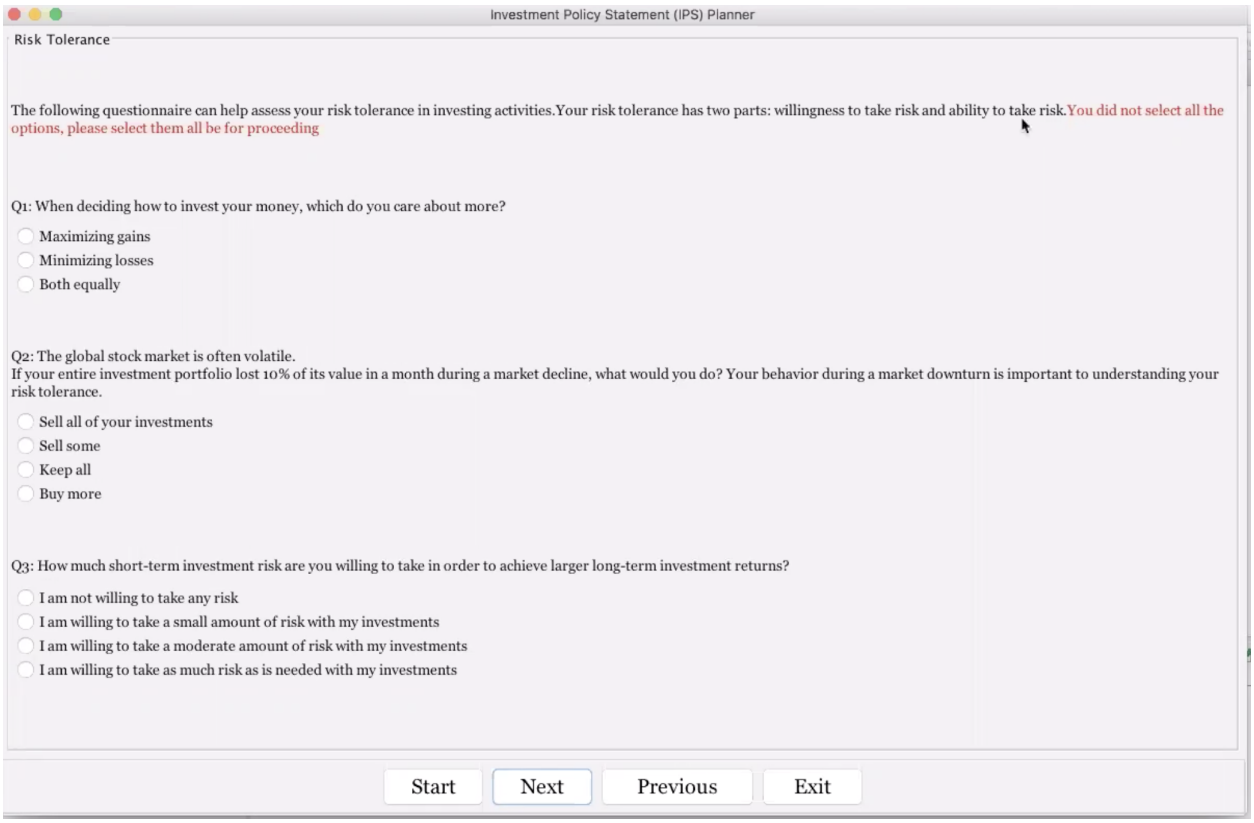
Step 1: the user can read and accept the terms and agreements in order to use the Program.



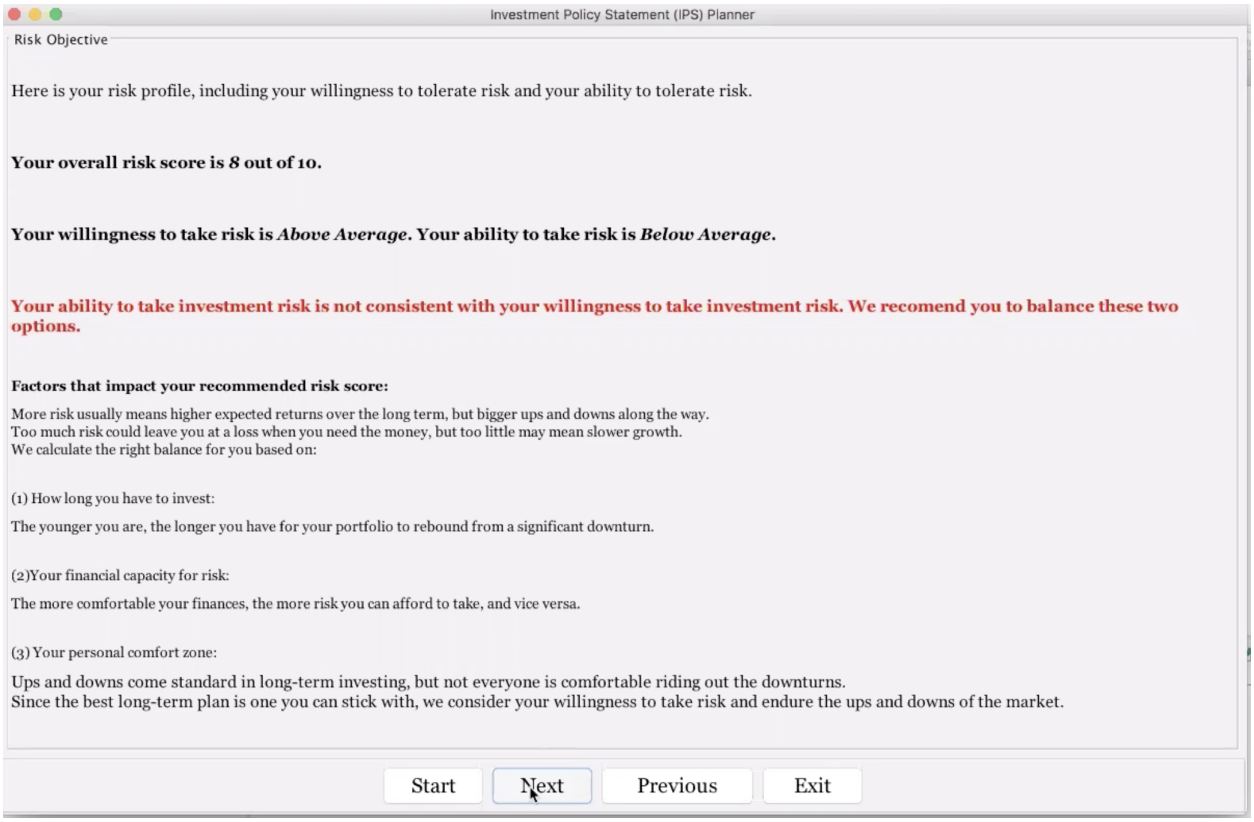
Step 2: the user can enter the personal investment information including the user name, user occupation, desired investment horizon, user annual income, user current asset, user annual income and expense, and the desired investment target amount at the end of the investment horizon.



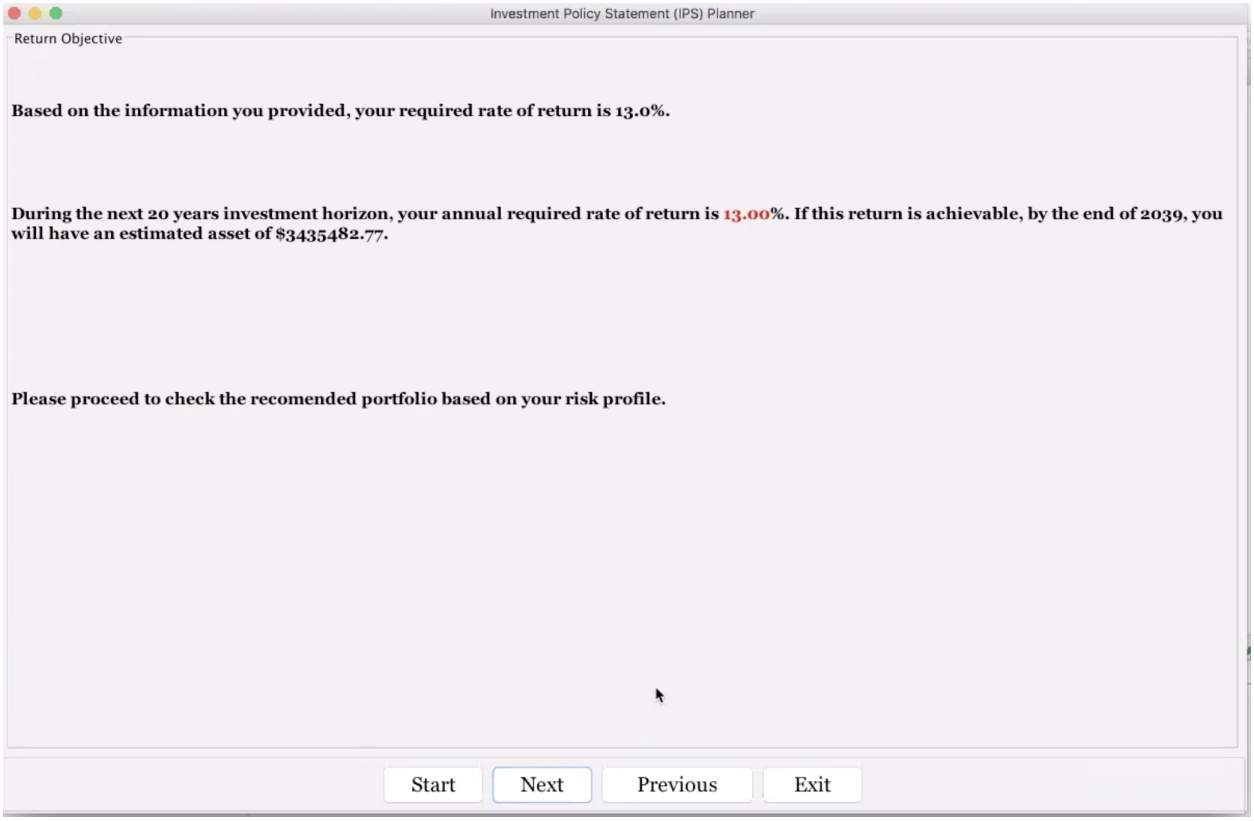
Step 3: the user can answer six questions. These questions determine the user's overall investment risk tolerance, the willingness and ability to tolerate investment risks. When there are inconsistency between the willingness and ability, the Program will generate an alert. Such as alert will not stop the user to proceed to the next steps, because in reality, many investors do not have consistent willingness and ability to tolerate risk.



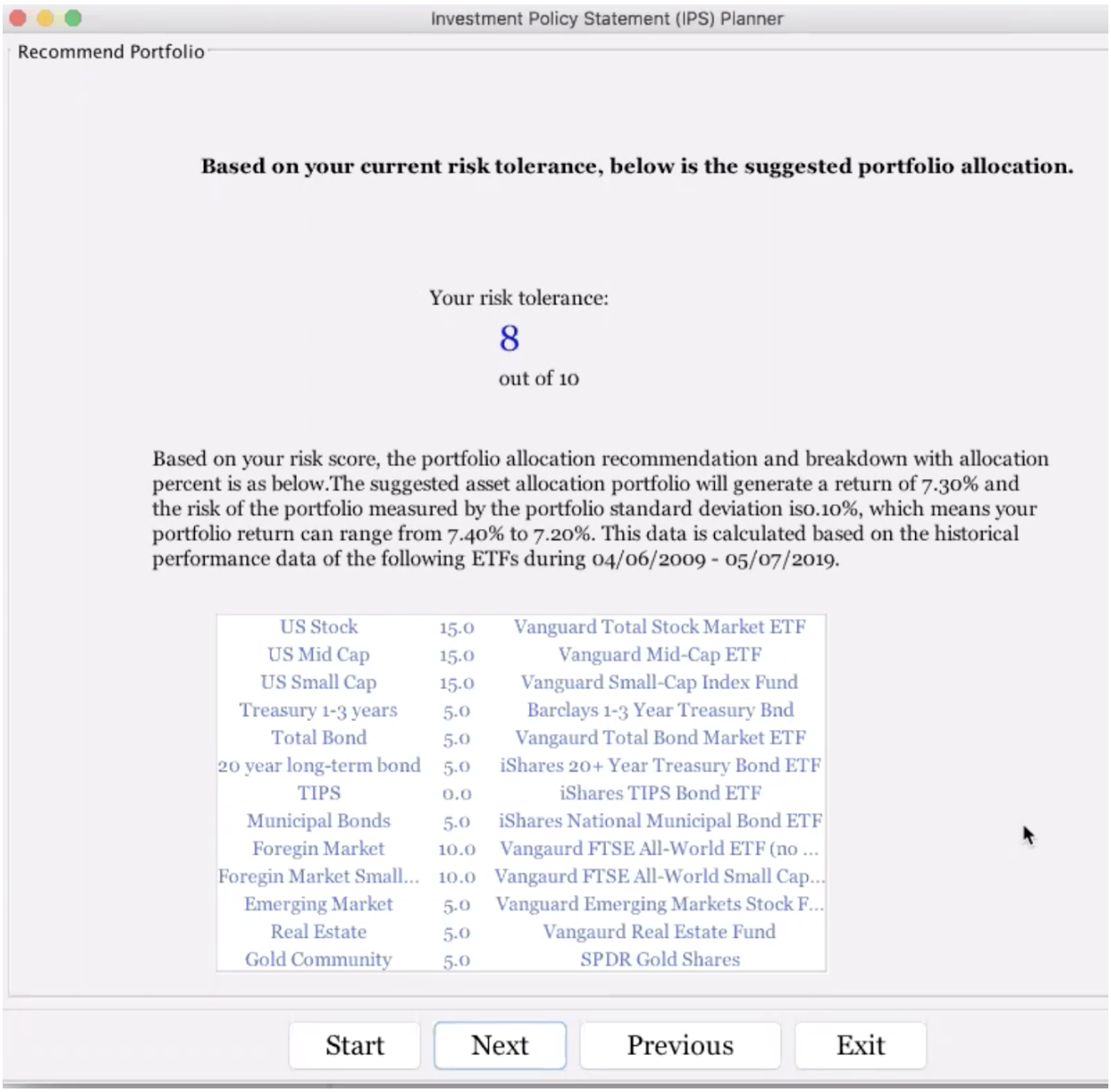
Step 4: the Program displays a screen with the analysis of the user's risk tolerance profile, including the overall risk score, willingness and ability to tolerate investment risks.



Step 5: the Program displays a screen with the analysis of the user's desired return objective. If the objective is too challenging to achieve, the Programs will generate an alert. Such as alert will not stop the user to proceed to the next steps, because the Program will recommend a realistic return based on the user's profile.



Step 6: the Program displays a screen with the recommended portfolio allocation to each class based on the user's risk score. The Program also shows the portfolio return and standard deviation based on the user's risk profile. Please note that the recommended portfolio return and standard deviation may be different from the user's desired return and standard deviation. However, the recommended portfolio return can be more realistic.



Step 7: the user can end the program by clicking on “Exit” or “Next” in the last panel.

PART III CLASSES (to code reviewers)

1. Classes for User Interface Design:

(1) IpsFrame
This class designs the panels structure and the buttons of the entire program.

(2) WelcomePanel
This class is the 1st screen the user encounters. This page displays the terms and conditions as a disclaimer of the product function from a legal perspective.

(3) FormPanel
This class sets the display of the 2nd screen the user encounters. This class focuses on the format and display of the text boxes.

(4) PersonalInfoPenel (Error Check for User Input)
This class incorporates the error check to validate the user's input format for "Personal Info"

(5) FormEvent
This class sets the events and collet the user's input information needed for further analysis.

(6) FormListener
This class sets interface for form event listener

(7) RiskTolerancePanel
This class designs the first part (Question 1, 2, 3) of questionnaire that the user needs to answer for the risk assessment.

(8) RiskTolerancePanel2
This class designs the second part (Question 4, 5, 6) of questionnaire that the user needs to answer for the risk assessment.

(9) RiskObjectivePanel
This class designs the display of the user's risk tolerance analysis.

(10) ErrorControl (Error Check for User Input)
This class incorporates the error check to validate the user's input format for "Risk Tolerance Multiple Choices". It also passes errors from the Perosnal Infopanel to the Panel display control to display any errors in user input.

(11) ReturnObjectivePanel
This class designs part of the display on the return objective panel. The algorithms below processed the user personal information input and generated return results. The results are passed to ReturnObjectivePanel and display for the user.

(12) RiskObjectivePanel

This class designs part of the display on the risk objective panel.The algorithms below processed the user multiple choices restyles and generated risk score. The analysis are passed to RiskObjectivePanel and display for the user.

(13) RecommendPortfolioPanel
This class designs the display the recommended allocation and the corresponding return and standard deviation. The algorithms below processed the user's overall information and produce a more realistic recommended investment portfolio. Additionally, the algorithms below also calculates the expected return and standard deviation

2. Classes for Finance Algorithms:

(1) TimeValueOfMoney
This class contains the methods to generate the user's return objective, which is the minimum return required to realize the user's desired retirement goal, based on the information user entered related to the goals.

Present Value = Current Asset Holding entered by the user;
Future Value = Desired Retirement Amount entered by the user;
Annuity = Annual Income - annual expense entered by the user;
Number of Years = Desired Retirement Age - User's Current Age entered by the user;
Internal Rate of Return (IRR) = Minimum Portfolio Return can achieve the user's desired retirement goals.

Method "calculateIRR" gives the final IRR output and Method "analyzeReturnObjective" tests whether the user's investment objective is realistic.

(2) RiskScore
This class returns the user's risk objective based on the survey taken by the user.
Question 1 to Question 4 returns the risk score for the user's willingness to tolerate risks. Question 5 and Question 6, together with the user's investment time horizon, which was denoted as Question 7, determine the user's ability to tolerate investment risks. All seven questions as a whole determine the user's overall risk tolerance.

Method "analyzeRisk" interprets the risk preference behind the user's risk score.

(3) SuggestedAllocation (Note 1 and Note 4)
This class returns the suggested asset allocation based on user's risk objective. Please note that this allocation is not based on the user's return objective.

The algorithm is based on publications of several financial institutions, such as Charles Schwab and Wealthfront.

(4) FinanceEquations (Note 2, Note 3, and Note 4)
This class contains the algorithm to return the Portfolio Return and Risk with the suggested asset allocation from the class Suggested Allocation. The portfolio return is weighted average

return of each individual asset in the portfolio. The portfolio risk is measure by the portfolio standard risk.

Please note the suggested portfolio return and risk may not be consistent with the user's desired return and risk. However, the suggested portfolio return and risk are more realistic based on the analysis from the financial institutions.

(5) FutureValueCalc
This class contains the method to calculate the future portfolio value based on the current asset holding, annuity, investment horizon, etc.

3. Classes for Integration between design and algorithm

(1) DataManager
This class handles the data transfer between logic and UI. It uses static variables and methods to set, retrieve and change declared variables. This class collects value and passes value between user inputs and algorithms classes. It also contains a few validation test for some results generated by the algorithms.

(2) ErrorControl
This class handles the error checking between the user input and UIview. It ensures that the textfields/radiobuttons in panels that require user input is properly filled-in/selected. It ensures that proper strings are used for the textfields.

4. JUNIT Tests Classes
15 JUNIT tests across 7 classes have been implemented. Due to the lack in the number of methods we decided to test method "AnalyzeRisk" with different input parameters to ensure it proper functions with different parameters.

1. TimeValueMoneyTest.java
    Methods Tested (2)
    Methods Names:
        A) testCalculatePresentValue
        B) testCalculateIRR

2. SuggestedAllocationTest.java
    Methods Tested (1)
    Methods Names:
        A) testGetBenchmarkCoefficient

3. RiskScoreTest.java
    Methods Tested(7)
    Methods Names:
        A) testGetQ7Score
        B) testGetOverallRiskScore
        C) testGetWillingnessRiskScore
        D) testGetAbilityRiskScore

E) testAnalyzeRisk
F) testAnalyzeRisk2 (different input parameters)
G) testAnalyzeRisk3 (different input parameters)

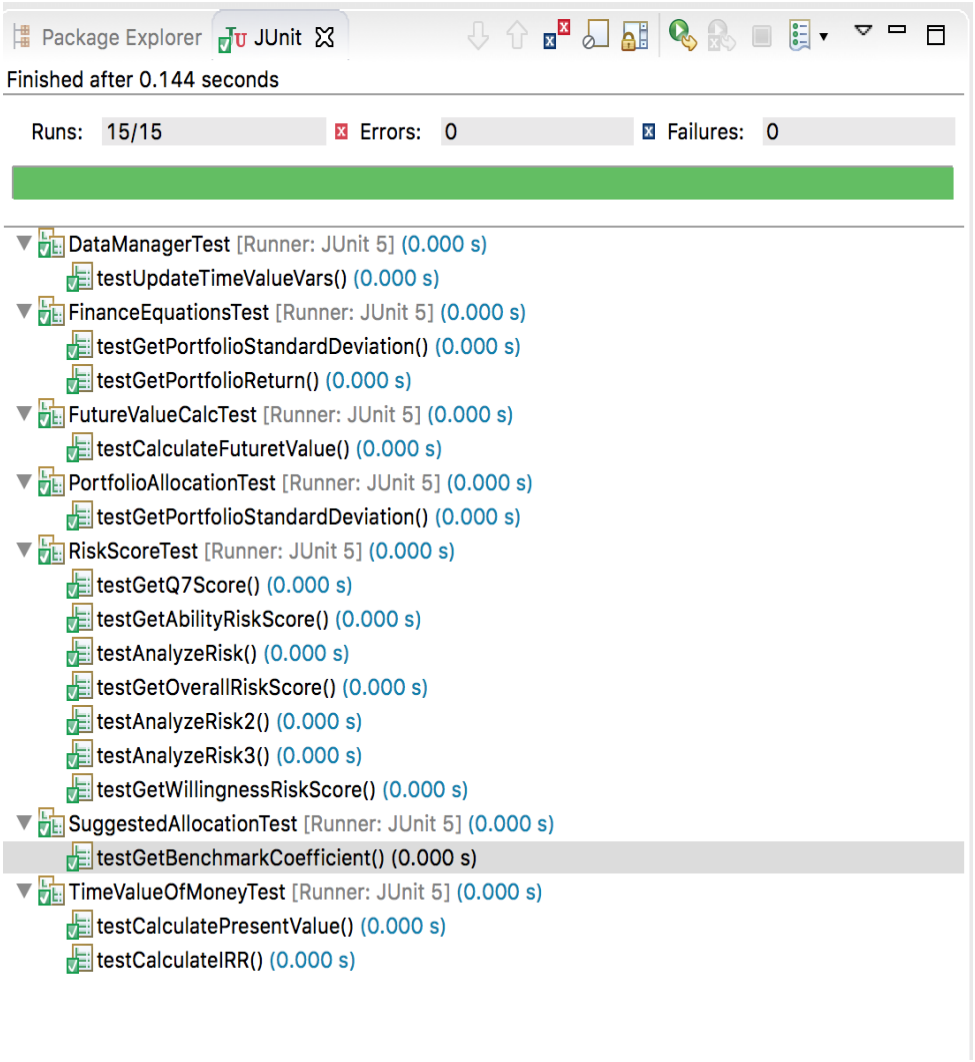
4. PortfolioAllcoation.java
Methods Tested(1)
Method Names:
A) testGetPortfolioStandardDeviation

5. FutureValueCalcTest.java
Methods Tested(1)
Method Names:
A) testCalculateFuturetValue

6. FinanceEquationsTest.java
Methods Tested(2)
Method Names:
A) testGetPortfolioReturn
B) testGetPortfolioStandardDeviation

7. DataManagerTest
Methods Tested(1)
Method Names:
A) testUpdateTimeValueVars

Result of Junit Tests:



PART IV DATA SOURCES AND REFERENCES (to code reviewers)

Note 1: Schwab Center for Financial Research with data provided by Morningstar, Inc.
https://www.schwab.com/public/file/P-778947/InvestorProfileQuestionnaire.pdf

Note 2: Asset class correlations for time period 04/06/2009 - 05/07/2019 based on daily returns.
https://www.portfoliovisualizer.com/asset-class-correlations

Note 3: Finance Algorithm demonstration example for the reviewer to understand
https://www.wallstreetmojo.com/portfolio-standard-deviation/

Note 4: Wealthfron Investment Methodology White Paper
https://research.wealthfront.com/whitepapers/investment-methodology/#2-finding_asset_classes