



Group 13 Stock Market Simulator Test and Inspection Report

Version <2.0>

<12/01/2015>

Team members:

Akhil Kateja
Adil Qaisar
Andriy Serafyn
Himanshu Anand

TABLE OF CONTENTS

TABLE OF CONTENTS	2
1.VERSION HISTORY	3
2.Introduction	4
2.1 PURPOSE	4
3.Project Overview	4
4.Test Summary	4
4.1 In Scope	4
4.2 Out of Scope	4
5.Test Execution Status	5
6.Defect Status	6
7.Defect Status by Type	7

1.VERSION HISTORY

Version #	Implemented By	Revision Date	Approved By	Approval Date	Reason
1.0	Himanshu Anand	11/20/2015	Akhil Kateja	11/21/2015	
2.0	Adil Qaisar	11/27/2015	Akhil Kateja	11/28/2015	Defect Status was added

2.Introduction

2.1 PURPOSE

This *Stock Market Simulator* Test Report provides a summary of the results of test performed as outlined within this document.

3.Project Overview

The Stock Market Simulation Program is going to be a web application that will be designed *to simulate* buying and selling stocks without the user having to invest real currency. It will deal with registering a user, logging in, viewing the dashboard, updating user's profile, searching companies and adding them to watchlist. It will also enable the user to seamlessly move from one page/feature of the application to the other. The *Dashboard* page is going to enable the user to view the list of stocks owned by him/her with the details such as live share price, quantity, day high/low, percentage change, percentage gain/loss and total gain/loss related to that share/transaction. The add to *Watchlist* feature will enable the user to search for a company and add the company to watchlist with the page displaying the list of the user's watchlist shares along with their details such as live price of the share, day high/low and the percentage change in the price of the share. The *Update Profile* feature will enable the user to update his/her details such as his/her username, address, occupation etc. The application will connect to the yahoo finance API to get the real time share prices and statistics. The dashboard and the user watchlist pages will provide the share names as clickable links, which will enable the user to view the graphical data associated with that share. The application will also display the *Contact Us* and the *About Us* information. It will add the additional functionalities of buying/selling share, logging into the application using social networking platforms and refreshing the application on stock price updation.

4.Test Summary

4.1 In Scope

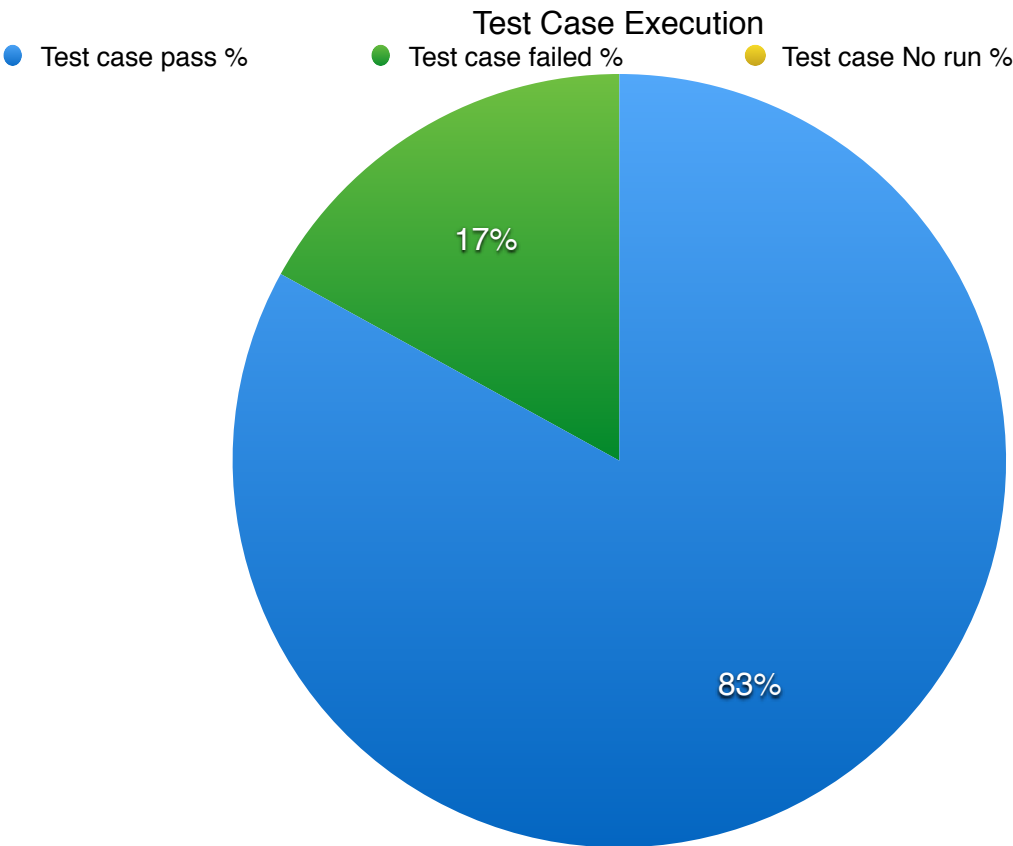
Functional Testing for the following modules are in Scope of Testing

- Login/Registration
- Portfolio
- Watchlist
- Buy Share
- Sell Share
- Profile

4.2 Out of Scope

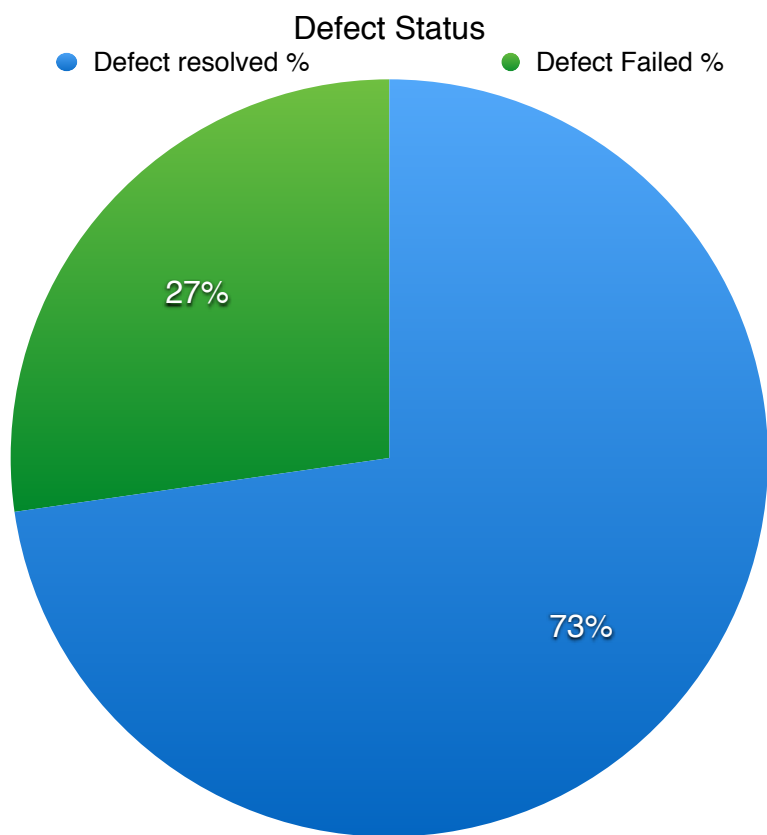
Performance Testing was not done for this application.

5.Test Execution Status



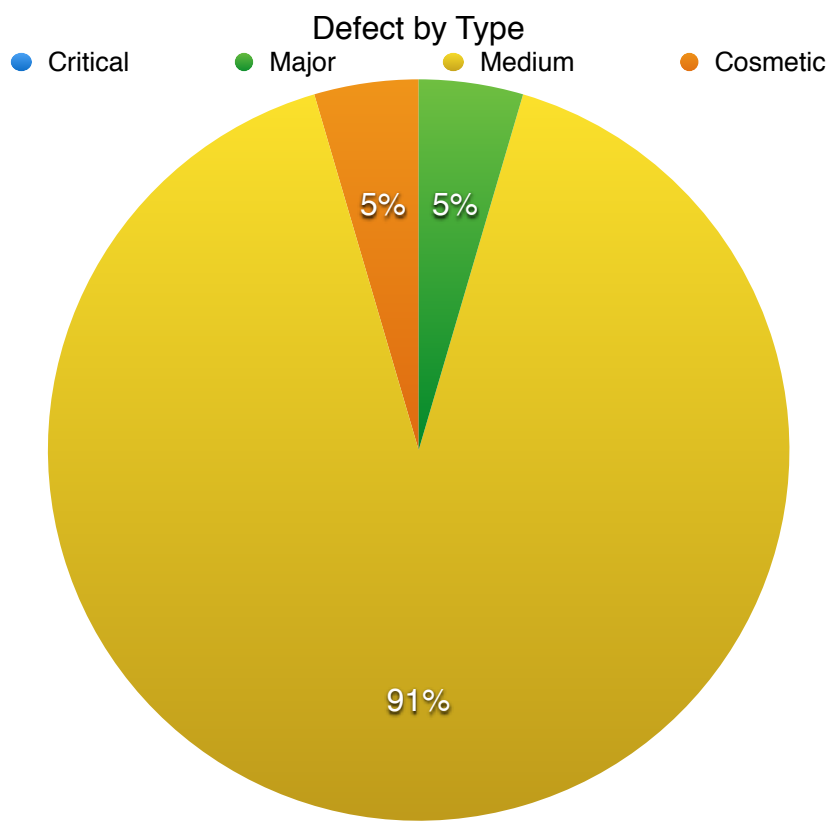
Test Execution Status	
Total Test cases planned	53
Total Test cases executed	53
Test Case No Run	0
Out of scope	0
Test Cases passed	44
Test Cases Failed	9
Test case pass %	83.0188679245283
Test case failed %	16.9811320754717
Test Case No Run %	0

6.Defect Status



Defect Status	
Total Defects	22
Resolved Defects	16
Failed Defects	6
Defect resolved %	72.7272727272727
Defect failed %	27.2727272727273

7.Defect Status by Type



	Login/Registration	Profile	Portfolio/Sell Share	Watchlist	Buy Share	Contact	About	Total
Critical								0
Major					1			1
Medium	5	2	6	3	4			20
Cosmetic						1		1
Total	5	2	6	3	5	1		22

Detailed Test Scripts

Test Case #	Test Objective	Pre-requisites	Step Name	Test Steps	Expected Result	Actual Result	Status (Pass/Fail)
TC1_Registrati on_Panel_disp lay	To verify that the Registration Panel gets displayed with the following options: 1)Username 2>Password 3)Confirm Password 4>Email	User should have entered the application URL and navigated to the Home page	Step 1	Hover the mouse on the Register Tile	The Registration Panel should be displayed with the following options: 1)Username 2>Password 3)Confirm Password 4>Email and a Register button	The Registration Panel is displayed with the options	Pass
TC2_Registrati on_Tile_slide right	To verify that the Registration Panel Tile slides to the right on hovering the mouse over the panel	User should have entered the application URL and navigated to the Home page	Step 1	Hover the mouse on the Register Tile	The Registration Panel tile should slide to the right	The Registration Panel tile slides to the right	Pass
TC3_Registrati on_Register_u sername_blan k	To verify that the following error message displays on clicking the Register button keeping the username field blank: "Please fill out this field."	User should have entered the application URL and navigated to the Home page	Step 1	Hover the mouse on the Register Tile	The Registration Panel tile should slide to the right and Registration Panel should be displayed with the following options: 1)Username 2>Password 3)Confirm Password 4>Email and a Register button	The Registration Panel is displayed with the options	Pass
			Step 2	Fill out the required fields keeping the username field as blank and click on Register button	An error message "Please fill out this field." should be displayed below the username text field.	Error message is displayed correctly.	Pass
TC4_Registrati on_Register_P assword_blank	To verify that the following error message displays on clicking the Register button keeping the Password field blank: "Please fill out this field."	User should have entered the application URL and navigated to the Home page	Step 1	Hover the mouse on the Register Tile	The Registration Panel tile should slide to the right and Registration Panel should be displayed with the following options: 1)Username 2>Password 3)Confirm Password 4>Email and a Register button	The Registration Panel is displayed with the options	Pass
			Step 2	Fill out the required fields keeping the Password field as blank and click on Register button	An error message "Please fill out this field." should be displayed below the username text field.	Error message is displayed correctly.	Pass

Test Case #	Test Objective	Pre-requisites	Step Name	Test Steps	Expected Result	Actual Result	Status (Pass/Fail)
TC5_Registrati on_Register_Confirm Password_blank	To verify that the following error message displays on clicking the Register button keeping the Confirm Password field blank: "Please fill out this field."	User should have entered the application URL and navigated to the Home page	Step 1	Hover the mouse on the Register Tile	The Registration Panel tile should slide to the right and Registration Panel should be displayed with the following options: 1)Username 2>Password 3)Confirm Password 4>Email and a Register button	The Registration Panel is displayed with the options	Pass
			Step 2	Fill out the required fields keeping the Confirm Password field as blank and click on Register button	An error message "Please fill out this field." should be displayed below the Confirm Password text field.	Error message is displayed correctly.	Pass
TC6_Registrati on_Register_Email_blank	To verify that the following error message displays on clicking the Register button keeping the Email field blank: "Please fill out this field."	User should have entered the application URL and navigated to the Home page	Step 1	Hover the mouse on the Register Tile	The Registration Panel tile should slide to the right and Registration Panel should be displayed with the following options: 1)Username 2>Password 3)Confirm Password 4>Email and a Register button	The Registration Panel is displayed with the options	Pass
			Step 2	Fill out the required fields keeping the Email field as blank and click on Register button	An error message "Please fill out this field." should be displayed below the Email text field.	Error message is displayed correctly.	Pass
TC7_Registrati on_Password_less_than_six_characters	To verify that the following error message displays when the user enters a password of less than six characters "Error: password must contain atleast six characters."	User should have entered the application URL and navigated to the Home page	Step 1	Hover the mouse on the Register Tile	The Registration Panel tile should slide to the right and Registration Panel should be displayed with the following options: 1)Username 2>Password 3)Confirm Password 4>Email and a Register button	The Registration Panel is displayed with the options	Pass
			Step 2	Fill out the required fields and entering fewer than six characters in the Password field	The following error message should be displayed: "Error: password must contain atleast six characters."	Error message is displayed correctly.	Pass

Test Case #	Test Objective	Pre-requisites	Step Name	Test Steps	Expected Result	Actual Result	Status (Pass/Fail)
TC8_Registrati on_Password_ no_lower_case _letter	To verify that the following error message displays when the user enters a password without any lower case characters "Error: password must contain at least one lowercase letter (a-z)!"	User should have entered the application URL and navigated to the Home page	Step 1	Hover the mouse on the Register Tile	The Registration Panel tile should slide to the right and Registration Panel should be displayed with the following options: 1)Username 2>Password 3)Confirm Password 4>Email and a Register button	The Registration Panel is displayed with the options	Pass
			Step 2	Fill out the required fields and entering six or more upper case characters and numbers in the Password field	The following error message should be displayed: "Error: password must contain at least one lowercase letter (a-z)!"	Error message is displayed correctly.	Pass
TC9_Registrati on_Password_ no_upper_cas e_letter	To verify that the following error message displays when the user enters a password without any upper case characters "Error: password must contain at least one uppercase letter (A-Z)!"	User should have entered the application URL and navigated to the Home page	Step 1	Hover the mouse on the Register Tile	The Registration Panel tile should slide to the right and Registration Panel should be displayed with the following options: 1)Username 2>Password 3)Confirm Password 4>Email and a Register button	The Registration Panel is displayed with the options	Pass
			Step 2	Fill out the required fields and entering six or more lower case characters and numbers in the Password field	The following error message should be displayed: "Error: password must contain at least one uppercase letter (A-Z)!"	Error message is displayed correctly.	Pass
TC10_Registra tion_Password _no_numbers	To verify that the following error message displays when the user enters a password without any number "Error: password must contain at least one number (0-9)!"	User should have entered the application URL and navigated to the Home page	Step 1	Hover the mouse on the Register Tile	The Registration Panel tile should slide to the right and Registration Panel should be displayed with the following options: 1)Username 2>Password 3)Confirm Password 4>Email and a Register button	The Registration Panel is displayed with the options	Pass
			Step 2	Fill out the required fields and entering six or more lower case characters and upper case characters in the Password field	The following error message should be displayed: "Error: password must contain at least one uppercase letter (0-9)!"	Error message is displayed correctly.	Pass

Test Case #	Test Objective	Pre-requisites	Step Name	Test Steps	Expected Result	Actual Result	Status (Pass/Fail)
TC11_Registration_Password_only_numbers	To verify that the following error message displays when the user enters a password without any lower case and uppercase characters "Error: password must contain at least one number (0-9),one lowercase letter(a-z) and one uppercase letter(A-Z)!"	User should have entered the application URL and navigated to the Home page	Step 1	Hover the mouse on the Register Tile	The Registration Panel tile should slide to the right and Registration Panel should be displayed with the following options: 1)Username 2>Password 3)Confirm Password 4>Email and a Register button	The Registration Panel is displayed with the options	Pass
			Step 2	Fill out the required fields and entering six or more numbers in the Password field	The following error message should be displayed: "Error: password must contain at least one number (0-9),one lowercase letter(a-z) and one uppercase letter(A-Z)!"	Incorrect error message "Error: password must contain at least one lowercase letter (a-z)!" is displayed.	Fail
TC12_Registration_email_without_@	To verify that the following error message displays on clicking the Register button without entering an "@" in the email field: "Please include '@' in the email address.'(text entered)' is missing an '@'.	User should have entered the application URL and navigated to the Home page	Step 1	Hover the mouse on the Register Tile	The Registration Panel tile should slide to the right and Registration Panel should be displayed with the following options: 1)Username 2>Password 3)Confirm Password 4>Email and a Register button	The Registration Panel is displayed with the options	Pass
			Step 2	Fill out the required fields and enter text without an '@' in the email field.	The following error message should be displayed: "Please include '@' in the email address.'(text entered)' is missing an '@'.	Error message is displayed correctly	Fail
TC13_Registration_password_mismatch	To verify that the following error message displays on clicking the Register button entering different passwords in password and confirm password fields: "passwords do not match."	User should have entered the application URL and navigated to the Home page	Step 1	Hover the mouse on the Register Tile	The Registration Panel tile should slide to the right and Registration Panel should be displayed with the following options: 1)Username 2>Password 3)Confirm Password 4>Email and a Register button	The Registration Panel is displayed with the options	Pass

Test Case #	Test Objective	Pre-requisites	Step Name	Test Steps	Expected Result	Actual Result	Status (Pass/Fail)
			Step 2	Fill out the required fields and enter different characters in the password and confirm password fields	The following error message should be displayed: "passwords do not match".	Error message is displayed correctly	Pass
TC14_Registration_Success	To verify that the following message displays on entering the valid details in the Registration fields and clicking on the Register button: "Registration Successful.Please proceed to Login"	User should have entered the application URL and navigated to the Home page	Step 1	Hover the mouse on the Register Tile	The Registration Panel tile should slide to the right and Registration Panel should be displayed with the following options: 1)Username 2>Password 3)Confirm Password 4>Email and a Register button	The Registration Panel is displayed with the options	Pass
			Step 2	Fill out the required fields and click on Register button	The following message should be displayed: "Registration Successful.Please proceed to Login".	Success message is displayed correctly	Pass
TC15_Login_Failure	To verify that the following message displays on Unsuccessful Login	User should have entered the application URL and navigated to the Home page	Step 1	Hover the mouse on the Login Tile	The Login Panel tile should slide to the right and Login Panel should be displayed with the following options: 1)Username 2>Password and a Login button	The Login Panel is displayed with the options	Pass
			Step 2	Fill out the required fields with incorrect credentials and click on Login button	The following message should be displayed: "Login Failed,Invalid Username Password Entered".	Error message is displayed correctly	Pass
TC16_Login_Success	To verify that the user is navigated to the Portfolio page on successful login	User should have entered the application URL and navigated to the Home page	Step 1	Hover the mouse on the Login Tile	The Login Panel tile should slide to the right and Login Panel should be displayed with the following options: 1)Username 2>Password and a Login button	The Login Panel is displayed with the options	Pass
			Step 2	Fill out the required fields with correct credentials and click on Login button	The user should be navigated to the Portfolio page	The user is navigated to the portfolio page.	Pass
TC17_Login_Facebook_Success	To verify that the user is able to login to the application through facebook	User should have entered the application URL and navigated to the Home page	Step 1	Click on the "Login with Facebook" panel	Facebook Login Page should be displayed	Facebook Login Page is displayed	Pass

Test Case #	Test Objective	Pre-requisites	Step Name	Test Steps	Expected Result	Actual Result	Status (Pass/Fail)
			Step 2	Fill out the required fields with correct credentials and click on Login	The user should be navigated to the Profile page	The user is navigated to the Profile page	Pass
TC18_Login_Twitter_Success	To verify that the user is able to login to the application through Twitter	User should have entered the application URL and navigated to the Home page	Step 1	Click on the "Login with Twitter" panel	Twitter Login Page should be displayed	Nothing happens	Fail
			Step 2	Fill out the required fields with correct credentials and click on Login	The user should be navigated to the Profile page	Nothing happens	Fail
TC19_Login_Gmail_Success	To verify that the user is able to login to the application through Gmail	User should have entered the application URL and navigated to the Home page	Step 1	Click on the "Login with Gmail" panel	Gmail Login Page should be displayed	Nothing happens	Fail
			Step 2	Fill out the required fields with correct credentials and click on Login	The user should be navigated to the Profile page	Nothing happens	Fail
TC20_Login_LinkedIn_Success	To verify that the user is able to login to the application through LinkedIn	User should have entered the application URL and navigated to the Home page	Step 1	Click on the "Login with LinkedIn" panel	LinkedIn Login Page should be displayed	Nothing happens	Fail
			Step 2	Fill out the required fields with correct credentials and click on Login	The user should be navigated to the Profile page	Nothing happens	Fail
TC21_FindUs_Facebook_Success	To verify that the user is able to navigate to the application's Facebook page.	User should have entered the application URL and navigated to the Home page	Step 1	Click on the "Find Us on Facebook" panel	Application's Facebook page should be displayed.	Application's Facebook page is displayed	Pass
TC22_ContactUs_Facebook_Success	To verify that the user is able to navigate to the developer's Facebook page.	User should have entered the application URL and navigated to the Home page	Step 1	Click on the "Contact Us" panel	Application developer's Facebook page should be displayed.	Application developer's Facebook page is displayed	Pass
TC23_Login_Tiles	To verify that all the tiles on the Login page are working correctly	User should have entered the application URL and navigated to the Home page	Step 1	Hover mouse over various tiles	The Login and Registration tiles should slide to the right whereas the remaining slides should slide to the the bottom	Tiles are working as expected.	Pass

Test Case #	Test Objective	Pre-requisites	Step Name	Test Steps	Expected Result	Actual Result	Status (Pass/Fail)
TC24_PortFolio_table_fields	To verify the fields of the table on portfolio page and their order	User should have valid credentials and should be logged in	Step 1	Verify the names and order of the following table fields on portfolio page: 1)Company/Sector 2)Live Price 3)Change 4)Quantity 5)Inv. Price 6)Day's High 7)Day's Low 8)Overall Gain 9)Overall Gain% 10)Latest Value 11)Actions	The field names should be correct and should appear in the proper order	The field names are correct and appear in proper order	Pass
TC25_Portfolio_New_User_display	To verify that the table fields are blank and "No data available in table" displayed inside the table	User should have valid credentials and should be logged in	Step 1	Verify that the table fields are blank and "No data available in table" displayed inside the table	The table fields should be blank and "No data available in table" should be displayed	The table fields are blank and "No data available in table" is displayed.	Pass
TC26_Portfolio_Sell_Stocks_fields_disabled	To verify that the Sell Stocks text fields and button are disabled until the user selects a stock to sell	User should have valid credentials and should be logged in	Step 1	Verify the Sell Stocks text fields	The Sell stocks fields and button should be disabled	The fields and button are disabled.	Pass
TC27_BuyStocks_table_fields	To verify the fields of the table on Buy stocks page and their order	User should have valid credentials and should be logged in	Step 1	Click on the Buy menu in the navigation bar	The Buy stock page should be displayed	The Buy stock page is displayed	Pass
			Step 2	Verify the names and order of the following table fields on portfolio page: 1)Company/Sector 2)Live Price 3)Change% 4)Day's High 5)Day's Low 6)Actions 7)Graph	The field names should be correct and should appear in the proper order	The field names are correct and appear in proper order	Pass
TC28_BuyStocks_fields_disabled	To verify that the Buy Stocks text fields and button are disabled until the user selects a stock to buy	User should have valid credentials and should be logged in	Step 1	Click on the Buy menu in the navigation bar	The Buy stock page should be displayed	The Buy stock page is displayed	Pass
			Step 2	Verify the buy Stocks text fields	The Buy stocks fields and button should be disabled	The fields and button are disabled.	Pass
TC29_BuyStocks_Search_Company	To verify that the user is able to search for a company by typing its name in the Search Company text field	User should have valid credentials and should be logged in	Step 1	Click on the Buy menu in the navigation bar	The Buy stock page should be displayed	The Buy stock page is displayed	Pass
			Step 2	Type the name of the company in the Search Company text field	The Company name should be displayed	The Company name is displayed	Pass

Test Case #	Test Objective	Pre-requisites	Step Name	Test Steps	Expected Result	Actual Result	Status (Pass/Fail)
TC30_BuyStocks_View_History	To verify that the company's stocks history data gets populated in the table fields when the user clicks on the View History button	User should have valid credentials and should be logged in	Step 1	Click on the Buy menu in the navigation bar	The Buy stock page should be displayed	The Buy stock page is displayed	Pass
			Step 2	Type the name of the company in the Search Company text field	The Company name should be displayed	The Company name is displayed	Pass
			Step 3	Click on View History button	The Company's stock history data should be populated in the table fields	The Company's stock history data is populated in the table fields	Pass
TC31_BuyStocks_Search_Company_overlap	To verify that the names of the companies that appear in the Search Company drop down do not overlap with the other components on the page	User should have valid credentials and should be logged in	Step 1	Click on the Buy menu in the navigation bar	The Buy stock page should be displayed	The Buy stock page is displayed	Pass
			Step 2	Type the name of the company in the Search Company text field	The Company name should not overlap with other components on the page	The Company name overlaps with other components	Fail
TC32_BuyStocks_Company_Graph	To verify that the user is able to view the graphical representation of the company's stock fluctuations over the following time periods: 1)3 Days 2)5 Days 3)15 Days 4)1 Month 5)3 Months 6)6 Months 7)1 Year 8)5 Years	User should have valid credentials and should be logged in	Step 1	Click on the Buy menu in the navigation bar	The Buy stock page should be displayed	The Buy stock page is displayed	Pass
			Step 2	Type the name of the company in the Search Company text field	The Company name should be displayed	The Company name is displayed	Pass
			Step 3	Click on View History button	The Company's stock history data should be populated in the table fields	The Company's stock history data is populated in the table fields	Pass
			Step 4	Click on the time period under Graph column for which you want to view the stock's history	A graph should be displayed on a new tab	Graph is displayed on a new tab	Pass

Test Case #	Test Objective	Pre-requisites	Step Name	Test Steps	Expected Result	Actual Result	Status (Pass/Fail)
TC33_BuyStocks_Fields_enabled	To verify that the Buy Stocks fields get enabled when the user clicks on the Buy button under Actions column	User should have valid credentials and should be logged in	Step 1	Click on the Buy menu in the navigation bar	The Buy stock page should be displayed	The Buy stock page is displayed	Pass
			Step 2	Type the name of the company in the Search Company text field	The Company name should be displayed	The Company name is displayed	Pass
			Step 3	Click on View History button	The Company's stock history data should be populated in the table fields	The Company's stock history data is populated in the table fields	Pass
			Step 4	Click on the Buy button in the Actions column	The Buy Stock fields should be enabled	The Buy Stocks fields are enabled	Pass
TC34_BuyStocks_Buy_table_field_blank	To verify that the table fields become blank when the user clicks on the buy button after the entering the allowable quantity of stocks	User should have valid credentials and should be logged in	Step 1	Click on the Buy menu in the navigation bar	The Buy stock page should be displayed	The Buy stock page is displayed	Pass
			Step 2	Type the name of the company in the Search Company text field	The Company name should be displayed	The Company name is displayed	Pass
			Step 3	Click on View History button	The Company's stock history data should be populated in the table fields	The Company's stock history data is populated in the table fields	Pass
			Step 4	Click on the Buy button in the Actions column	The Buy Stock fields should be enabled	The Buy Stocks fields are enabled	Pass
			Step 5	Enter a valid entry in the Enter Quantity field and click on Buy	The table data should disappear	The table data gets disappeared	Pass
TC35_BuyStocks_Buy_Success	To verify that the user is buy the stocks successfully and the stock information gets updated on the Portfolio page and the updated Virtual currency should be displayed on the Profile page	User should have valid credentials and should be logged in	Step 1	Click on the Buy menu in the navigation bar	The Buy stock page should be displayed	The Buy stock page is displayed	Pass
			Step 2	Type the name of the company in the Search Company text field	The Company name should be displayed	The Company name is displayed	Pass

Test Case #	Test Objective	Pre-requisites	Step Name	Test Steps	Expected Result	Actual Result	Status (Pass/Fail)
			Step 3	Click on View History button	The Company's stock history data should be populated in the table fields	The Company's stock history data is populated in the table fields	Pass
			Step 4	Click on the Buy button in the Actions column	The Buy Stock fields should be enabled	The Buy Stocks fields are enabled	Pass
			Step 5	Enter a valid quantity to buy and click on Buy button	The table data should disappear	The table data gets disappeared	Pass
			Step 6	Click on the Portfolio menu	Portfolio page should be displayed	Portfolio page is displayed	Pass
			Step 7	To verify that the stock bought is visible on the portfolio page with the actual quantity bought	the stock bought should be visible on the portfolio page with the actual quantity bought	the stock bought is visible on the portfolio page with the actual quantity bought	Pass
			Step 8	Click on the Profile menu in the navigation bar	The profile page should open	The profile page opens	Pass
			Step 9	To verify that the virtual currency value	Virtual currency value should get updated correctly on the profile page.	Virtual Currency value gets updated correctly	Pass
TC36_BuyStocks_negative quantity	To verify that an error should be displayed if the user enters a negative value in the Enter Quantity field	User should have valid credentials and should be logged in	Step 1	Click on the Buy menu in the navigation bar	The Buy stock page should be displayed	The Buy stock page is displayed	Pass
			Step 2	Type the name of the company in the Search Company text field	The Company name should be displayed	The Company name is displayed	Pass
			Step 3	Click on View History button	The Company's stock history data should be populated in the table fields	The Company's stock history data is populated in the table fields	Pass
			Step 4	Click on the Buy button in the Actions column	The Buy Stock fields should be enabled	The Buy Stocks fields are enabled	Pass
			Step 5	Enter a negative value in the quantity to buy and click on Buy	An error should be displayed	The stock is displayed on the user's portfolio page with the negative quantity entered.	Fail
TC37_BuyStocks_Enter_quantity_invalid_characters	To verify that an error is displayed if the user enters an alphabet or a special character in the Enter Quantity and clicks on the Buy button	User should have valid credentials and should be logged in	Step 1	Click on the Buy menu in the navigation bar	The Buy stock page should be displayed	The Buy stock page is displayed	Pass

Test Case #	Test Objective	Pre-requisites	Step Name	Test Steps	Expected Result	Actual Result	Status (Pass/Fail)
			Step 2	Type the name of the company in the Search Company text field	The Company name should be displayed	The Company name is displayed	Pass
			Step 3	Click on View History button	The Company's stock history data should be populated in the table fields	The Company's stock history data is populated in the table fields	Pass
			Step 4	Click on the Buy button in the Actions column	The Buy Stock fields should be enabled	The Buy Stocks fields are enabled	Pass
			Step 5	Enter an alphabet or a special character in the quantity to buy and click on Buy	An error should be displayed	An error is displayed	Pass
TC38_BuyStocks_amount_exceeding_virtual_currency	To verify that the following error gets displayed if the user tries to purchase stocks of amount greater than the amount available in their virtual currency: "You have exceeded your available currency value!"	User should have valid credentials and should be logged in	Step 1	Click on the Buy menu in the navigation bar	The Buy stock page should be displayed	The Buy stock page is displayed	Pass
			Step 2	Type the name of the company in the Search Company text field	The Company name should be displayed	The Company name is displayed	Pass
			Step 3	Click on View History button	The Company's stock history data should be populated in the table fields	The Company's stock history data is populated in the table fields	Pass
			Step 4	Click on the Buy button in the Actions column	The Buy Stock fields should be enabled	The Buy Stocks fields are enabled	Pass
			Step 5	Enter a value in the Enter quantity field such that the total value of shares to be bought exceeds the amount available in the virtual currency and click on Buy	"You have exceeded your available currency value!" error message should be displayed	Error message displayed correctly	Pass
TC39_BuyStocks_autopopulate_company_name	To verify that the name of the company of which the user wishes to buy shares gets auto populated in the Buy Stock field when the user clicks on the Buy button in the Actions column	User should have valid credentials and should be logged in	Step 1	Click on the Buy menu in the navigation bar	The Buy stock page should be displayed	The Buy stock page is displayed	Pass

Test Case #	Test Objective	Pre-requisites	Step Name	Test Steps	Expected Result	Actual Result	Status (Pass/Fail)
			Step 2	Type the name of the company in the Search Company text field	The Company name should be displayed	The Company name is displayed	Pass
			Step 3	Click on View History button	The Company's stock history data should be populated in the table fields	The Company's stock history data is populated in the table fields	Pass
			Step 4	Click on the Buy button in the Actions column	The Buy Stock fields should be enabled	The Buy Stocks fields are enabled	Pass
			Step 5	Verify that the Company name gets auto populated in the Buy Stocks field and it should be disabled	The company name should get auto populated and it should be disabled	The company name gets auto populated and is disabled	Pass
TC40_SellStocks_Success	To verify that the user is able to sell the stocks from the Portfolio page	User should have valid credentials and should be logged in and the user should have some stocks in their account	Step 1	Click on the Sell menu in the navigation bar	The Sell stock page should be displayed	The Sell stock page is displayed	Pass
			Step 2	Click on the Sell button under Actions column	The Sell Stocks fields and Sell button should be enabled	The fields and button gets enabled	Pass
			Step 3	Enter an allowable quantity in the Enter Quantity field and click on Sell	The updated quantity should be selected on the Portfolio page	The updated quantity is reflected on the Portfolio page	Pass
			Step 4	Click on the Profile menu in the navigation bar	The Profile page should open	The Profile page gets opened	Pass
			Step 5	Verify the Virtual currency amount on the Profile page	The virtual currency amount should get updated	The virtual currency is updated successfully	Pass
TC41_SellStocks_quantity exceeding the quantity available	To verify the following error message if the user enters a quantity more than the quantity available with him to sell: "Entered quantity greater than owned quantity!"	User should have valid credentials and should be logged in and the user should have some stocks in their account	Step 1	Click on the Sell menu in the navigation bar	The Sell stock page should be displayed	The Sell stock page is displayed	Pass
			Step 2	Click on the Sell button under Actions column	The Sell Stocks fields and Sell button should be enabled	The fields and button gets enabled	Pass
			Step 3	Enter a quantity greater than the quantity available and click sell	"Entered quantity greater than owned quantity!" error message should be displayed	Correct error message gets displayed.	Pass

Test Case #	Test Objective	Pre-requisites	Step Name	Test Steps	Expected Result	Actual Result	Status (Pass/Fail)
TC42_SellStocks_negative_quantity	To verify that an error is displayed if the user attempts to sell a negative quantity	User should have valid credentials and should be logged in and the user should have some stocks in their account	Step 1	Click on the Sell menu in the navigation bar	The Sell stock page should be displayed	The Sell stock page is displayed	Pass
			Step 2	Click on the Sell button under Actions column	The Sell Stocks fields and Sell button should be enabled	The fields and button gets enabled	Pass
			Step 3	Enter a negative Quantity to sell and click on Sell	An error message should be displayed	The transaction is successful and the absolute value of the quantity gets added	Fail
TC43_SellStocks_sell_all_stocks	To verify that the stock entry gets removed from the Portfolio page when the user sells all the stocks available with him	User should have valid credentials and should be logged in and the user should have some stocks in their account	Step 1	Click on the Sell menu in the navigation bar	The Sell stock page should be displayed	The Sell stock page is displayed	Pass
			Step 2	Click on the Sell button under Actions column	The Sell Stocks fields and Sell button should be enabled	The fields and button gets enabled	Pass
			Step 3	Enter the quantity to sell equivalent to the quantity currently available with the user and click on Sell	The stock entry should get deleted from the Portfolio page	The stock entry is not deleted and quantity is displayed with zero value	Fail
TC44_SellStocks_SearchStock	To verify that the user is able to search the stock they currently hold by entering the name of the stock in the Search text field	User should have valid credentials and should be logged in and the user should have some stocks in their account	Step 1	Click on the Sell menu in the navigation bar	The Sell stock page should be displayed	The Sell stock page is displayed	Pass
			Step 2	Search the company by entering its name in the Search text field	The stock entry of the company should get displayed on top	The stock name of the company gets displayed on top	Pass
			Step 3	Search the company by entering its name in the Search text field	The stock entry of the company should get displayed on top	The stock name of the company gets displayed on top	Pass
TC45_Login_With_Facebook_Success	To verify that the user is able to Login successfully with Facebook	User should have entered the application URL and navigated to the Home page and they should have valid Facebook credentials	Step 1	Click on Login with Facebook Panel	Facebook Login page should open	Facebook Login page is opened	Pass

Test Case #	Test Objective	Pre-requisites	Step Name	Test Steps	Expected Result	Actual Result	Status (Pass/Fail)
			Step 2	Enter the credentials and click on Login	Login should be successful and the user's portfolio page should be displayed	Login is successful and user's portfolio page gets displayed.	Pass
			Step 3	Click on the Profile menu in the navigation bar	The user's email address registered with Facebook should get auto populated	The user's email address is auto populated	Pass
TC46_Profile_Field	To verify the following elements on the Profile page: 1)Name 2)Email 3)Gender radio button 4)Date of Birth selector 5)Occupation 6)Industry 7)Annual Income 8)Virtual Currency	User should have valid credentials and should be logged in	Step 1	Click on the Profile menu in the navigation bar	The Profile page should open	The Profile page gets opened	Pass
			Step 2	Verify the following fields on the Profile page: 1)Name 2)Email 3)Gender radio button 4)Date of Birth selector 5)Occupation 6)Industry 7)Annual Income 8)Virtual Currency	The fields should be displayed and should	The fields are displayed	Pass
TC47_Profile_VirtualCurrency_Autopopulate	To verify that the virtual currency gets auto populated with amount 20000 dollars for a new user	User should have valid credentials and should be logged in	Step 1	Click on the Profile menu in the navigation bar	The Profile page should open	The Profile page gets opened	Pass
			Step 2	Verify the Virtual Currency field	Virtual currency field should get auto populated with 20000 dollars	Virtual currency field is auto populated with 20000 dollars	Pass
TC48_Profile_Submit	To verify that the user is able to update their profile information	User should have valid credentials and should be logged in	Step 1	Click on the Profile menu in the navigation bar	The Profile page should open	The Profile page gets opened	Pass
			Step 2	Fill up the fields on the Profile page and click on submit	The Profile should get updated	The Profile is getting updated.	Pass
TC49_About	To verify that the user is able to navigate to the About page by clicking on the About menu in the menu bar	User should have valid credentials and should be logged in	Step 1	Click on the About menu in the menu bar	The user should be navigated to About page	The user is navigated to the About page	Pass

Test Case #	Test Objective	Pre-requisites	Step Name	Test Steps	Expected Result	Actual Result	Status (Pass/Fail)
TC50_Contact	To verify that the user is able to navigate to the Contact page by clicking on the About menu in the menu bar	User should have valid credentials and should be logged in	Step 1	Click on the Contact menu in the menu bar	The user should be navigated to Contact page	The user is navigated to the Contact page	Pass
TC51_Contact	To verify that the table of Contacts is aligned in the centre of the screen	User should have valid credentials and should be logged in	Step 1	Click on the Contact menu in the menu bar	The user should be navigated to Contact page	The user is navigated to the Contact page	Pass
			Step 2	To verify that the Table of Contacts is aligned in the centre	Table of Contacts should be aligned in the centre	Table is not aligned in the centre	Fail
TC52_Logout	To verify that the user is able to successfully logout of the application	User should have valid credentials and should be logged in	Step 1	Click on the Logout menu in the top right corner in the menu bar	The user should be navigated to the homepage and their session should close	The user is navigated to the Homepage	Pass
TC53_Refresh	To verify that the stock data refreshes successfully after one minute	User should have valid credentials and should be logged in and the user should have some stocks in their account	Step 1	Wait for one minute and verify whether the stock data refreshes to display the current data	Stock data should refresh after one minute	Stock data gets refreshed after one minute	Pass

CODE INSPECTION REPORT

I. Introduction

1. Purpose of Code Inspection

Code Inspection is systematic examination (often known as peer review) of computer source code. It is intended to find and fix mistakes overlooked in the initial development phase, improving both the overall quality of software and the developers' skills. The basic purpose of code inspection includes:

- Ensures that code has sufficient quality to be released/ committed.
- Ensures that the code is properly formatted/ commented.
- Ensures that the code is correct, efficient and performant.
- Helps to maintain a level of consistency in design and implementation.
- Helps to identify common defects across the team thus reducing rework.
- Team cohesiveness. Working together helps draw team members closer. It also brings a brief respite from the isolation that coding often brings.

2. Coding Conventions

The following coding conventions were followed for the code inspection of Stock Market Simulator application:

- Each Java source file contains a single public class or interface. When private classes and interfaces are associated with a public class, you can put them in the same source file as the public class. The public class should be the first class or interface in the file.
- All source files should begin with a c-style comment that lists the class name, version information, date, and copyright notice.
- The first non-comment line of most Java source files is a package statement. After that, import statements can follow.
- The declaration of class as well instance variables should follow the following order:
First the public class variables, then the protected, then package level (no access modifier), and then the private.
- Methods should be grouped by functionality rather than by scope or accessibility. For example, a private class method can be in between two public instance methods. The goal is to make reading and understanding the code easier.
- Avoid lines longer than 80 characters, since they're not handled well by many terminals and tools. When an expression will not fit on a single line, break it according to these general principles:
 - Break after a comma.
 - Break before an operator.
 - Prefer higher-level breaks to lower-level breaks.
 - Align the new line with the beginning of the expression at the same level on the previous line.

- If the above rules lead to confusing code or to code that's squished up against the right margin, just indent 8 spaces instead.
- Block comments are used to provide descriptions of files, methods, data structures and algorithms. Block comments may be used at the beginning of each file and before each method. They can also be used in other places, such as within methods. Block comments inside a function or method should be indented to the same level as the code they describe.
- Short comments can appear on a single line indented to the level of the code that follows. If a comment can't be written in a single line, it should follow the block comment format. A single-line comment should be preceded by a blank line.
- Very short comments can appear on the same line as the code they describe, but should be shifted far enough to separate them from the statements. If more than one short comment appears in a chunk of code, they should all be indented to the same tab setting.
- The `//` comment delimiter can comment out a complete line or only a partial line. It shouldn't be used on consecutive multiple lines for text comments; however, it can be used in consecutive multiple lines for commenting out sections of code
- Doc comments describe Java classes, interfaces, constructors, methods, and fields. Each doc comment is set inside the comment delimiters `/**...*/`, with one comment per class, interface, or member
- One declaration per line is recommended since it encourages commenting
- Try to initialize local variables where they're declared. The only reason not to initialize a variable where it's declared is if the initial value depends on some computation occurring first.
- Put declarations only at the beginning of blocks. (A block is any code surrounded by curly braces `"{"` and `"}"`.) Don't wait to declare variables until their first use; it can confuse the unwary programmer and hamper code portability within the scope.
- When coding Java classes and interfaces, the following formatting rules should be followed:
 - No space between a method name and the parenthesis `"("` starting its parameter list.
 - Open brace `"{"` appears at the end of the same line as the declaration statement.
 - Closing brace `"}"` starts a line by itself indented to match its corresponding opening statement, except when it is a null statement the `"}"` should appear immediately after the `"{"`.
- Each line should contain at most one statement.
- Compound statements are statements that contain lists of statements enclosed in braces `"{ statements }"`.
- The enclosed statements should be indented one more level than the compound statement.
- The opening brace should be at the end of the line that begins the compound statement; the closing brace should begin a line and be indented to the beginning of the compound statement.

- Braces are used around all statements, even single statements, when they are part of a control structure, such as an if-else or for statement. This makes it easier to add statements without accidentally introducing bugs due to forgetting to add braces.
- A return statement with a value should not use parentheses unless they make the return value more obvious in some way
- The if-else class of statements should have the following form:

```
if (condition) {
    statements;
}
```

```
if (condition) {
    statements;
} else {
    statements;
}
```

```
if (condition) {
    statements;
} else if (condition) {
    statements;
} else {
    statements;
}
```

- A for statement should have the following form:

```
for (initialization; condition; update) {
    statements;
}
```

- An empty for statement (one in which all the work is done in the initialization, condition, and update clauses) should have the following form:

```
for (initialization; condition; update);
```
- A while statement should have the following form:

```
while (condition) {
    statements;
}
```

- An empty while statement should have the following form:

```
while (condition);
```

- A do-while statement should have the following form:

```
do {
    statements;
} while (condition);
```

- A switch statement should have the following form:

```
switch (condition) {
case ABC:
    statements;
```

```

        /* falls through */
case DEF:
    statements;
    break;
case XYZ:
    statements;
    break;
default:
    statements;
    break;
}

```

- Every switch statement should include a default case. The break in the default case is redundant, but it prevents a fall-through error if later another case is added.
- A try-catch statement should have the following format:

```

try {
    statements;
} catch (ExceptionClass e) {
    statements;
}

```

- A try-catch statement may also be followed by finally, which executes regardless of whether or not the try block has completed successfully.

```

try {
    statements;
} catch (ExceptionClass e) {
    statements;
} finally {
    statements;
}

```

- Blank lines improve readability by setting off sections of code that are logically related
- Two blank lines should always be used in the following circumstances:
 - Between class and interface definitions.
- One blank line should always be used in the following circumstances:
 - Between methods.
 - Between the local variables in a method and its first statement.
 - Before a block.
 - Between logical sections inside a method to improve readability.
- Blank spaces should be used in the following circumstances:
 - A keyword followed by a parenthesis should be separated by a space
 - A blank should appear after commas in argument lists.
 - All binary operators except . should be separated from their operands by spaces. Blank spaces should never separate unary operators such as unary minus, increment ("++"), and decrement ("--") from their operands.
 - The expressions in a for statement should be separated by blank spaces.

- Casts should be followed by a blank space
- Don't make any instance or class variable public without good reason. Often, instance variables don't need to be explicitly set or gotten-often that happens as a side effect of method calls.
- Avoid using an object to access a class (static) variable or method. Use a class name instead.
- Numerical constants (literals) should not be coded directly, except for -1, 0, and 1, which can appear in a for loop as counter values.
- Avoid assigning several variables to the same value in a single statement. It is hard to read.
- It is generally a good idea to use parentheses liberally in expressions involving mixed operators to avoid operator precedence problems. Even if the operator precedence seems clear to you, it might not be to others-you shouldn't assume that other programmers know precedence as well as you do
- If an expression containing a binary operator appears before the ? in the ternary ?: operator, it should be parenthesized
- Use XXX in a comment to flag something that is bogus but works. Use FIXME to flag something that is bogus and broken.
- *Naming Conventions*

Identifier Type	Rules for Naming	Examples
Packages	<p>The prefix of a unique package name is always written in all-lowercase ASCII letters and should be one of the top-level domain names, currently com, edu, gov, mil, net, org, or one of the English two-letter codes identifying countries as specified in ISO Standard 3166, 1981.</p> <p>Subsequent components of the package name vary according to an organization's own internal naming conventions. Such conventions might specify that certain directory name components be division, department, project, machine, or login names.</p>	com.sun.eng com.apple.quicktime.v2 edu.cmu.cs.bovik.cheese
Classes	<p>Class names should be nouns, in mixed case with the first letter of each internal word capitalized. Try to keep your class names simple and descriptive. Use whole words-avoid acronyms and abbreviations (unless the abbreviation is much more widely used than the long form, such as URL or HTML).</p>	class Raster; class ImageSprite;

Interfaces	Interface names should be capitalized like class names.	interface RasterDelegate; interface Storing;
Methods	Methods should be verbs, in mixed case with the first letter lowercase, with the first letter of each internal word capitalized.	run(); runFast(); getBackground();
Variables	Except for variables, all instance, class, and class constants are in mixed case with a lowercase first letter. Internal words start with capital letters. Variable names should not start with underscore _ or dollar sign \$ characters, even though both are allowed. Variable names should be short yet meaningful. The choice of a variable name should be mnemonic- that is, designed to indicate to the casual observer the intent of its use. One-character variable names should be avoided except for temporary "throwaway" variables. Common names for temporary variables are i, j, k, m, and n for integers; c, d, and e for characters.	int i; char c; float myWidth;
Constants	The names of variables declared class constants and of ANSI constants should be all uppercase with words separated by underscores ("_"). (ANSI constants should be avoided, for ease of debugging.)	static final int MIN_WIDTH = 4; static final int MAX_WIDTH = 999; static final int GET_THE_CPU = 1;

3. Review Process

- The code review for the *Stock Market Simulator Application* was completed on November 12, 2015 by a review team consisting of *Akhil Kateja, Himanshu Anand, Adil Quaisar and Adriy Serafyn*.
- A meeting between the review team was held at the beginning of the first sprint of the project, at which time information about the code structure and coding conventions was presented to the team.
- The code was periodically added to the *Code Collaborator* server in which the programmer of the piece of code acted as the *Author* while the rest of the team acted as the *Reviewer*.
- The *Reviewers* could test the code on the basis of the set coding conventions for the project and also in terms of performance, efficiency and/or logic.
- The *Reviewers* could give comment(s) corresponding to specific line number(s) or raise a defect(s) depending on the severity of the violation.
- Based on the comments from the reviewers, the *Author* would incorporate the code changes into his/ her program/ code and resubmit his/ her code to the code collaborator server for a fresh cycle of code review.
- This will continue until, the code review is approved/ passed by all the *Reviewers* after which the *Author* could check in the code into the repository.
- Before checking any code into the repository, it was mandatory for the *Author* of the code to get his code reviewed and approved from rest of the *Developers/ Reviewers*

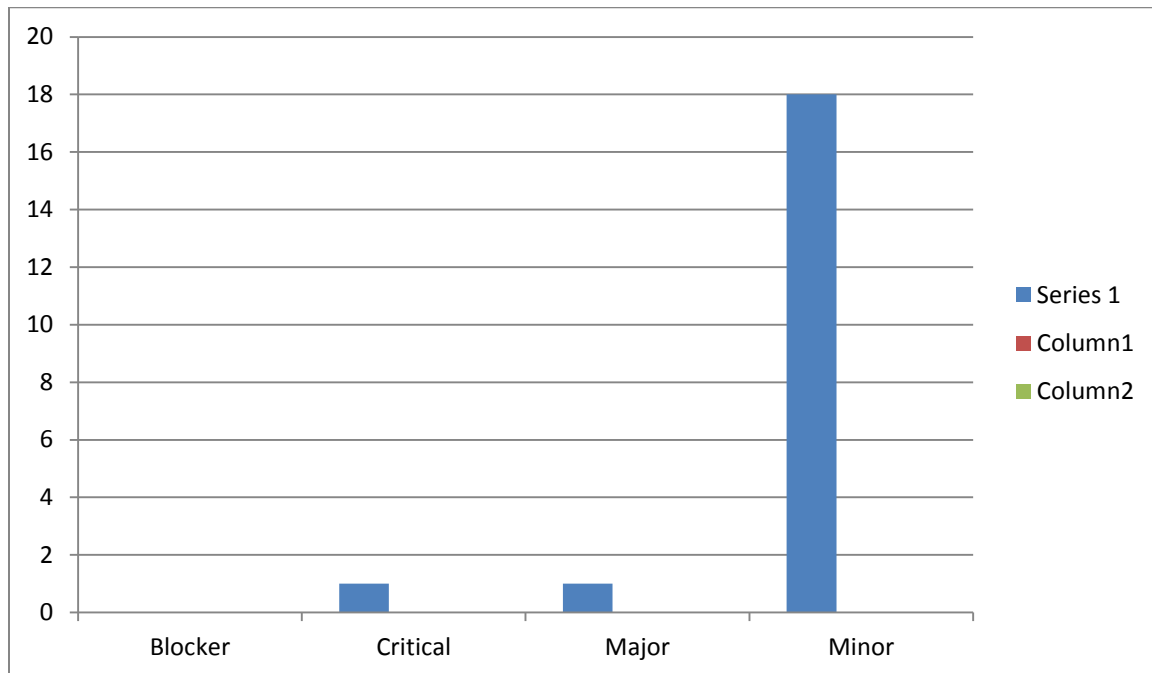
II. Defects checklist

Source File	Line Number	Description	Risk	Module	Status
<i>src/main/java/com/uic/se/stockapp/dbcon/DatabaseCon</i>	26	Empty Catch Block	Major	Database Connection	Resolved
<i>src/main/java/com/uic/se/stockapp/dbcon/DatabaseCon</i>	16	Cyclomatic Complexity of 4	Critical	Database Connection	Resolved
<i>src/main/java/com/uic/se/stockapp/dbcon/DatabaseCon</i>	26	Catching generic exceptions such as NullPointerException, RuntimeException, Exception in try-catch block	Major	Database Connection	Resolved
<i>src/main/java/com/uic/se/stockapp/dbcon/DatabaseCon.java</i>	112	Local variable 'listOfStocks' could be declared final	Minor	Database Connection	Resolved
<i>src/main/java/com/uic/se/stockapp/dbcon/DatabaseCon.java</i>	114	Ensure that resources like this ResultSet object are closed after use.	Minor	Database Connection	Resolved
<i>src/main/java/com/uic/se/stockapp/server/registration/ SearchStockServlet.java</i>	13	Method argument could be final	Minor	Buy/ Watchlist	Resolved
<i>src/main/java/com/uic/se/stockapp/server/registration/ SearchStockServlet.java</i>	17	Local variable could be final	Minor	Buy/Watchlist	Resolved

<i>src/main/java/com/uic/se/stockapp/server/registration/ SearchStockServlet.java</i>	11	Each class should declare at least one constructor.	Minor	Buy/Watchlist	Resolved
<i>src/main/java/com/uic/se/stockapp/server/registration/ SearchStockServlet.java</i>	11	Denotes whether comments are required (or unwanted) for specific language elements.	Minor	Buy/Watchlist	
<i>src/main/java/com/uic/se/stockapp/server/registration/ LoginServlet.java</i>	16	Unused Imports	Minor	Login	Resolved
<i>src/main/java/com/uic/se/stockapp/server/registration/ LoginServlet.java</i>	20	Method argument could be final	Minor	Login	Resolved
<i>src/main/java/com/uic/se/stockapp/server/registration/ LoginServlet.java</i>	18	Denotes whether comments are required (or unwanted) for specific language elements.	Minor	Login	Resolved
<i>src/main/java/com/uic/se/stockapp/server/registration/ LoginServlet.java</i>	18	Each class should declare at least one constructor.	Minor	Login	Resolved
<i>src/main/java/com/uic/se/stockapp/server/registration/ RegistrationServlet.java</i>	16	Method argument could be final	Minor	Registration	Resolved
<i>src/main/java/com/uic/se/stockapp/server/registration/ UpdateProfileServlet.java</i>	16	Method argument could be final	Minor	Update Profile	Resolved
<i>src/main/java/com/uic/se/stockapp/server/registration/ WatchListServlet.java</i>	19	Local Variable could be final	Minor	WatchList	Resolved
<i>src/main/java/com/uic/se/stockapp/entity/USER.java</i>	72	Use explicit scoping instead of the default package level.	Minor	Entity	Resolved
<i>src/main/java/com/uic/se/stockapp/entity/USER.java</i>	69	Fields should be declared at the top of the class, before any method declarations, constructors, initializers or inner classes.	Minor	Entity	Resolved
<i>src/main/java/com/uic/se/stockapp/entity/USER.java</i>	74	Method argument could be final	Minor	Entity	Resolved

III. Findings Summary

This section provides a summary of the findings resulting from this review. The figures below graphically outline the review team's findings by risk level.



IV. Screenshots

This section contains the screenshots of the *Code collaborator* that was used during the code inspection process.

The screenshot displays the Code Collaborator web application interface. The browser address bar shows the URL `127.0.0.1:8080/ui#review:id=1`. The application header includes the logo, version `v9.3.9300`, user `Akhil Kateja`, and navigation links: `Home`, `Settings`, `Admin`, `Reports`, `Logout`, and `Help`. A search bar is also present. A banner indicates `EVALUATION COPY - Expires in 28 days`.

The main content area is titled `Review #1: Scenario-I changes`. It shows the current state as `Waiting for any activity`. A progress bar indicates the review stages: `PLANNING`, `ANNOTATING`, `INSPECTION` (active), and `COMPLETED`. Below the progress bar, statistics are shown: `2 Participants`, `13 Files`, `0 Chats`, and `0 Defects`. Buttons for `DETAILS`, `CANCEL`, `REJECT`, and `EDIT` are available.

A table lists the review details:

Field	Value
Review Title	Scenario-I changes
Role	Author
Created	Mon Nov 30 18:17:06 GMT-600 2015 by Akhil Kateja
Template	Default
Completed On	N/A
Restrict Access	Anyone
Restrict Uploads/Deletions	No
Overview	Scenario-I changes.

Below the table, the `Participants` section is shown. It includes a `Group by None` dropdown and a table of participants:

Name	Role	State	Action
Akhil Kateja	Author	Waiting	<input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
Himanshu Anand	Reviewer	Active	<input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>

At the bottom, there are links for `Jira ticket Everyone`, `Email Everyone`, and `Poke Everyone`, along with an `EDIT` button. The footer shows the status `javascript:void(0);` and a download link `Show all downloads...`.

Review #1: Scenario-I cha

127.0.0.1:8080/ui#reviewid=1

AppsInstalling Python, Ec...Inbox (20,043) - akhi...https://www.facebo...Herman Survivor M...How I Met Your Mot...Ruby - Code SchoolRuby on Rails Scree...Other bookmarks

Review materials

DownloadUpload

View as: OverlaySeparateExternal DiffDelete

Name:
Author: Akhil Kateja
Date: 11/30/2015, 6:17:09 PM

Local changes

Himanshu Anand
Akhil Kateja

Files

	Location				Status	Notes
StockMarketApp/src/main						
java/com/uic/se/stockapp						
dbcon						
DatabaseCon.java		144	3	0	0	+
DatabaseConstants.java		2	0	0	0	+
entity						
User.java		8	79	0	0	+
server/registration						
LoginServlet.java		7	13	0	0	+
RegistrationServlet.java		1	0	0	0	+
UpdateProfileServlet.java		1	22	0	0	+
WatchListServlet.java		0	0	0	0	+
webapp						
Aboutus.html		0	1	1	0	+
Contactus.html		0	1	1	0	+
dashboard.jsp		103	28	1	0	+
index.jsp		7	15	4	0	+
UserProfile.jsp		36	30	2	0	+
...		11	11	0	0	+

Git-2.6.3-64-bit.exe

Regulatory Complian....jpg

Show all downloads...

Review #1: Scenario-I char x

127.0.0.1:8080/ui#file:review=1/version=15

Apps | Installing Python, Ec... | Inbox (20,043) - akhi... | https://www.facebo... | Herman Survivor M... | How I Met Your Mot... | Ruby - Code School | Ruby on Rails Scree... | Other bookmarks

Display | File | Prev | Next | File | External | Download Diff

MARK ALL READ

Overall

Himanshu Anand on 11/30/2015 at 6:22:43 PM

Remove printn statements.

MARK READ

Start conversation at: Line 346

Empty catch clause, consider putting some logger/print stack trace there

Empty catch clause, consider putting some logger/print stack trace there

Severity: Major

Type: Maintainability

ADD DEFECT | CANCEL

MARK ALL READ

Review: #1 File: StockMarketApp/src/main/java/com/uic/se/stockapp/dbcon/DatabaseCon.java

After: Version 9344c8ff95e2f5c2d... checked in on 2015-11-30 18:17

Before: Version 938f420c0456049a4... checked in on 2015-10-31 09:31

```
340         if (rs.getString("email").equalsIgnoreCase(email)) {
341             context.setAttribute("USER_ID", rs.getInt("id"));
342             context.setAttribute("CURRENCY", rs.getInt("currency"));
343             return true;
344         }
345     }
346 } catch (Exception e) {
347
348 }
349     return false;
350 }
351
352 public static boolean updateCurrency(int userId, int currency){
353
354     con = getConnection();
355     String query = EMPTY_STRING;
356     PreparedStatement pstmt = null;
357     query = "update user set currency=? where id="
358           + userId;
359
360     try {
361         pstmt = con.prepareStatement(query);
362         pstmt.setString(1, new Integer(currency).toString());
363         int count=pstmt.executeUpdate();
364         if(count>0){
365             context.setAttribute("CURRENCY", currency);
366             return true;
367         }
368         con.close();
369         pstmt.close();
```

Git-2.6.3-64-bit.exe | Regulatory Complian...jpg | Show all downloads...

Review #1: Scenario-I cha x

127.0.0.1:8080/ui#file:review=1/version=15

Apps Installing Python, Ec... Inbox (20,043) - akhi... https://www.facebo... Herman Survivor M... How I Met Your Mot... Ruby - Code School Ruby on Rails Scree... Other bookmarks

Display File Prev Next File External Download Diff

MARK ALL READ

Overall

Himanshu Anand on 11/30/2015 at 6:22:43 PM

Remove printn statements.

MARK READ

Line 348

Himanshu Anand on 11/30/2015 at 6:24:01 PM

Created defect D1: Empty catch clause, consider putting some logger/print stack trace there : Major / Maintainability

MARK READ

ADD ADD AS DEFECT

Empty catch clause, consider putting some logger/print stack trace there

EDIT MARK FIXED

To start a new conversation, click on a line at right.

MARK ALL READ

Review: #1 File: StockMarketApp/src/main/java/com/uic/se/stockapp/dbcon/DatabaseCon.java

After: Version 9344c8ff95e2f5c2d... checked in on 2015-11-30 18:17

Before: Version 938f420c0456049a4... checked in on 2015-10-31 09:31

```
340         if (rs.getString("email")!=null) {
341             context.setAttribute("USER_ID", rs.getInt("id"));
342             context.setAttribute("CURRENCY", rs.getInt("currency"));
343             return true;
344         }
345     }
346 } catch (Exception e) {
347
348 }
349 return false;
350 }
351
352 public static boolean updateCurrency(int userId, int currency){
353
354     con = getConnection();
355     String query = EMPTY_STRING;
356     PreparedStatement pstmt = null;
357     query = "update user set currency=? where id="
358           + userId;
359
360     try {
361         pstmt = con.prepareStatement(query);
362         pstmt.setString(1, new Integer(currency).toString());
363         int count=pstmt.executeUpdate();
364         if(count>0){
365             contxt.setAttribute("CURRENCY", currency);
366             return true;
367         }
368         con.close();
369         pstmt.close();
370     }
```

Git-2.6.3-64-bit.exe

Regulatory Complian...jpg

Show all downloads...

