COVER PAGE

Tilted Programmers

Kevin Le

Michael Ha

David Luong

David Tran

Jonathan Peng

Stephen Chan

3/2/16

Introduction to Software Engineering

Professor Ehteshami

**Revision History**

**Jan 30th 2016:**

First team meeting.

David T: Not present

David L: Present

Kevin: Present

Jonathan: Present

Stephen: Present

Michael: Present

Group decided to pick Chapter 18 for project. Worked on team charter along with Assignment 1. Left a few questions open in order to ask professor.

**Feb 5th 2016**

David Luong: Present, Chapter 3

Kevin Le: Present, Chapter 4

David Tran: Present, Chapter 8

Stephen Chan: Present, Chapter 5

Jonathan Peng: Present, Chapter 9

Michael Ha: Present, Chapter 10

Group worked on Assignment #2 and also decided on which mockups will be used for our program. We each were assigned chapters to study for and prepare questions. Decided on team name to be Tilting programmers

**Feb 12th 2016**

David T: Not present

David L: Present

Kevin: Present

Jonathan: Present

Stephen: Present

Michael: Present

Group worked on Assignment #3. Group also began coding for the Login Screen, Chapter menu. Jonathan was working on the screens and posted the code for the forms on the google drive. Group must have register screen,login screen, chapter menu working by class meeting on wednesday.

**Feb 15th 2016**

David T: Present

David L: Not Present

Kevin: Present

Jonathan: Present

Stephen: Present

Michael: Present

Group discussed plan for wednesday class that includes running the code in front of our professor. Each member was also assigned to read their chapters and have questions prepared for the meeting on friday in order to be ready to implement the questions forms for the program.

**Feb 20th 2016**

David T: Present

David L: Present

Kevin: Present

Jonathan: Present

Stephen: Present

Michael: Present

Group worked on test case templates. Each individual was in charge of creating class diagrams in order to be ready for the communication diagram.

Table of Contents

Chapters pg #

Chapter 3: Software Process Structure………………………………………………………….30-40

Chapter 4: Process Models……………………………………………………………………….40-66

Chapter 5: Agile Development…………………………………………………………………….66-87

Chapter 8: Understanding Requirements……………………………………………………..131-166

Chapter 9: Requirements Modeling: Scenario-Based Methods…………..………………....166-184

Chapter 10: Requirements Modeling: Class-Based Methods………………………..….…..184-202

**Pg 4.Explain how your team worked on Pre-game Planning ( Write 2-3 paragraphs explaining what you did as well as attaching items 5,6 and 7)**

**5. Project Plan ?( From assignment 2)**

We are addressing the problem of efficient studying for university students in their Introduction to Software Engineering class. Our product intends to address this by developing a tool that will act as a supplement with the *Software Engineering: A Practitioner’s Approach* book*.* With this in mind, we have decided to develop a desktop application since the majority of our customer base will have access to a personal computer. Our primary goals while developing our application will be the UI of the application, which includes the various functions, and properly meeting the goals of the stakeholders.

Throughout the process, we may encounter risks such as not meeting the project deadline, loss of data due to unforeseen circumstances and it is possible our end product may not be accepted. The stakeholder for our application is Professor Ehteshami. The time frame for developing the product will be approximately three months and we will be using the SCRUM methodology to run the group.

**6. Use Case Diagram**



**7. Use-case (textual) - descriptions for each requirement (5 functional Requirements and 5 nonfunctional requirements).**

## **Use Case:** Making a new account on the Register Screen

**ID :** UC\_1

**Description:** The user needs to make an account in order to login to the program.

**Level:** High

**Primary Actor:** Students, professors, and admin.

Supporting Actors:

## **Main Success Scenario**

1. User enters their first name
2. User enters their email address
3. User enters creates unique username
4. User creates password for account.
5. User hits register button and is brought to login screen

## **Extensions**

3a. User enters a username that already exists.

The system shall prompt the user the username is already taken and to input a different username

3b. User enters less than 5 characters for their username.

The system shall prompt the user to enter more than 5 characters for their username.

3c. User leaves the username box empty.

The system shall prompt the user to input a username.

4a. User enters less than 8 characters for their password.

The system shall prompt the user to enter more than 8 characters for their password.

4b. User leaves the password box empty.

The system shall prompt the user to input a password.

**Use Case:** Accessing the Login page

**ID :** UC\_2

**Description:** The user needs to login to the program.

**Level:** High

**Primary Actor:** Students, professors, and admin.

Supporting Actors:

## **Main Success Scenario**

1. User enters username into textbox.
2. User enters password into password text box.
3. User clicks on login button.
4. The system brings user to chapter menu for selection.

## **Extensions**

1a. User enters the wrong username or password.

System should prompt message to the user in order to fix username or password

1b. User enters more than 20 characters for the password.

The system shall prompt the user to use less than 20 characters.

1c. User enters less than 6 characters for the password.

The system shall prompt the user to enter at least 6 characters.

## **Use Case:** Viewing/Selecting a chapter in the Chapter Menu

**ID :** UC\_3

**Description:** The user is brought to the chapter menu to select a chapter to study about.

**Level:** High

**Primary Actor:** Students, professors, and admin.

Supporting Actors:

## **Main Success Scenario**

1. User clicks on a chapter button.
2. System should show chapter question UI.

## **Extensions**

## **Use Case:** Select/Viewing the chapter’s True and False Questions

**ID :** UC\_4

**Description:** The user is able to view and is able to select the chapter’s True and False questions.

**Level:** Medium

**Primary Actor:** Students, professors, and admin.

Supporting Actors:

## **Main Success Scenario**

1. User is able to view the chapter’s true and false questions.
2. User is able to see the chapter’s buttons allowing them to select true or false.
3. User is able to see and select the buttons allowing them to select true or false.
4. User is able to select one of the choices.
5. User is able to click on the next button to go to the results screen

## **Extensions**

4a. User does not choose an answer for one of the questions.

The question is counted wrong.

**Use Case:** Selecting/Viewing all chapters Multiple Choice Question Page

**ID :** UC\_5

**Description:** The user is able to view and select all chapter’s Multiple Choice questions.

**Level:** Medium

**Primary Actor:** Students, professors, and admin.

Supporting Actors:

## **Main Success Scenario**

1. User is able to view the Multiple Choice questions.
2. User is able to see the choices for the Multiple Choice questions
3. User is able to make a selection for the question.
4. User is able to click on the next button to go to the results screen

## **Extensions**

3a. User does not enter a choice for some/all questions.

The question is counted wrong.

## **Use Case:** Selecting/Viewing all chapters Fill In The Blank Question Page

**ID :** UC\_6

**Description:** The user is able to view and is able to enter all chapter Fill In The Blank questions.

**Level:** Medium

**Primary Actor:** Students, professors, and admin.

Supporting Actors:

## **Main Success Scenario**

1. User is able to view the Fill In The Blank questions.
2. User is able to see the box allowing them to input the answer for the Fill In The Blank question.
3. User is able to enter text into the box in order to submit the answer.
4. User is able to click on the next button to go to the result screen.

## **Extensions**

3a. User leaves one or all of the boxes empty

The question is counted wrong.

## **Use Case:** Selecting/Viewing all chapter matching questions page.

**ID :** UC\_7

**Description:** The user is able to view and match correct keywords with definitions.

**Level:** Medium

**Primary Actor:** Students, professors, and admin.

Supporting Actors:

## **Main Success Scenario**

1. User is able to view boxes of keywords and definitions.
2. User is able move keywords with their respective definitions.
3. User is able to click on the next button to go to the results screen

## **Extensions**

3a. User does not drag any/some of the keywords with a definition.

The questions are counted wrong.

## **Use Case:** Viewing the Results Screen

**ID :** UC\_8

**Description:** The user is able to view the answers that they got correct and also see the questions they got wrong along with the ability to return back to a question and logout.

**Level:** Medium

**Primary Actor:** Students, professors, and admin.

Supporting Actors:

## **Main Success Scenario**

1. User is able to see a list of questions that reveals which questions they got correct and also the ones that they got wrong.
2. User is able to see the clickable buttons that allow them to logout and return back to the chapter menu.
3. User also sees a calculated percentage of the questions they got correct.

## **Use Case:** Choosing to return back to a question in the results screen.

**ID :** UC\_9

**Description:** The user is able to view the answer they got wrong and click on a button to return back to the question.

**Level:** Medium

**Primary Actor:** Students, professors, and admin.

Supporting Actors:

## **Main Success Scenario**

1. User is able to see a list of questions that reveals which questions they got correct and also the ones that they got wrong.
2. User is able to click on the button to return back to a question.

## **Extensions**

1a. User clicks on the button that they already received the correct answer for.

User will still be brought to the question and will see their original answer.

**Staging - Grooming** Our team first looked over our essential requirements and assured that they were all working correct. If there were problems with our essential requirements, they were noted and brought up in our meeting for further discussion. This took some time as we had to go through our entire product multiple times to assure quality. After identifying problems with our essential requirement, we prioritized the urgency by how critical they were to the core program.

For example, if the problem dealt with our login, it would be highly prioritized as it will keep the user from using the program at all. However, problems such as minor typos were prioritized very lowly as this can be taken care of in a short period of time and did not affect the main functionality. Once we gathered all the high and medium priority tasks, we met as a group and decided who would work on each of the problems. We used cards to decide the amount of time each task would take and agreed as a group if it was fair allotment or not.

Within our meetings, we spent some time on improving the overall functionality of the program as well. For instance, we standardized much of the menus to make it easy on both the user to navigate. We also made some standard for everyone to follow when creating each new page to assure consistency across our program. All these were implemented as a whole team to assure a quality product once finished.

**Pg 12 8. User Story (5 user stories).**

**US-1: Login**

As a user, I would like to login to the system so that I can begin studying.

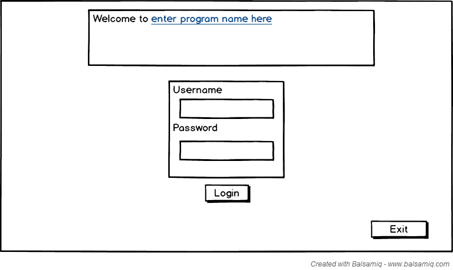
Relation to Use Case: 2 as it allows the user to login to the program

Assigned Developers: David T, Kevin

Due Date: 2-17-16

Priority: High

Mockup:



**US-2: Register Screen**

As a user, I would like to be able to make an account for the program so that I can login to begin studying.

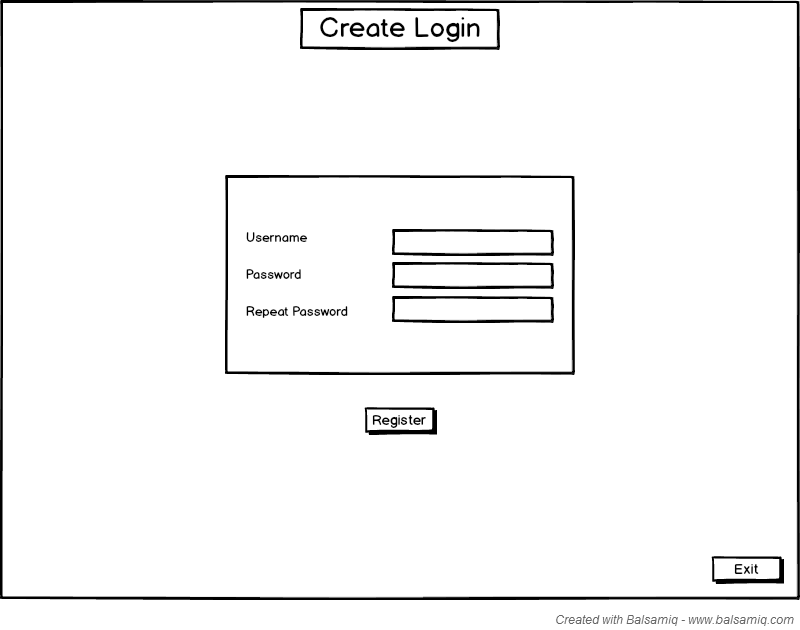
Relation to Use Case: 1 as it allows the user to create an account login.

Assigned Developers: David L, Jonathan

Due Date: 2-17-16

Priority: High

Mockup:



**US-3: Chapter Screen**

As a user, I would like to click on a chapter button so that I can begin being quizzed.

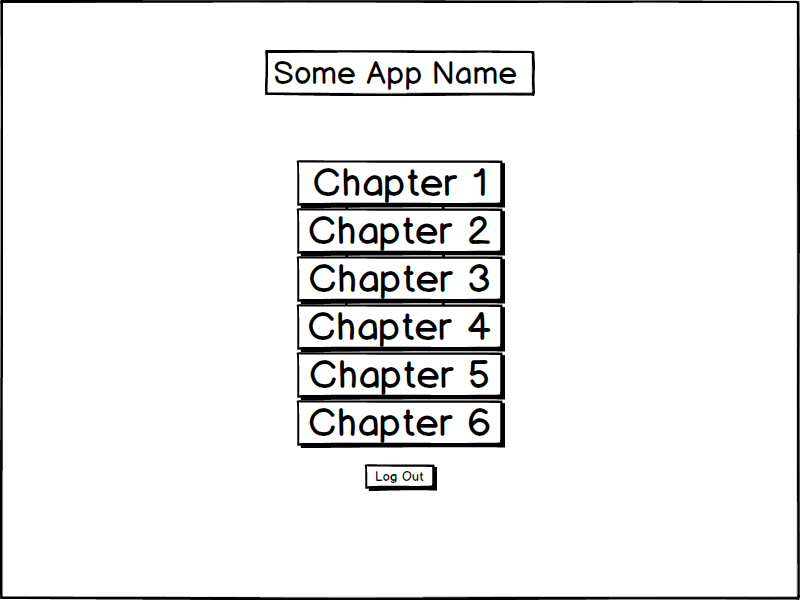
Relation to Use Case: 3 as it allows the user to view and select chapters.

Assigned Developers: Michael, Stephen

Due Date: 2-17-16

Priority: High

Mockup:



**US-4: Matching**

As a user, I would like to select my answer for matching questions in order to see if I got the correct answer.

Relation to Use Case: 7 as it allows the user to view and answer matching questions.

Assigned Developers: Michael, David T

Due Date: 2-17-16

Priority: High

Mockup:

**US-5: Fill in the Blank**

As a user, i would like to select my answer for all chapters Fill in Blank questions in order to see if I got the correct answer.

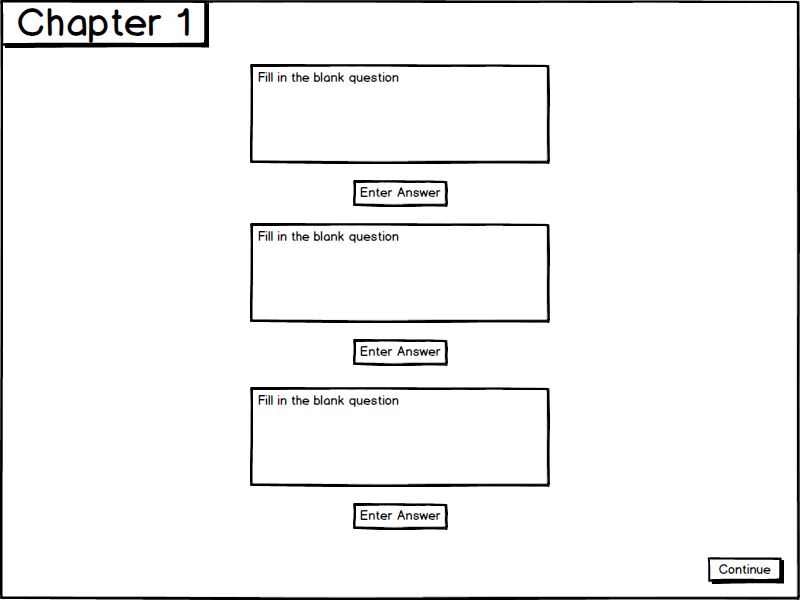
Relation to Use Case: 6 as it allows the user to view and answer fill in the blank questions.

Assigned Developers: David L, Kevin

Due Date: 2-17-16

Priority: High

Mockup:



**US-6: True/False**

As a user, i would like to select my answer for all chapter true/false questions in order to see if I got the correct answer.

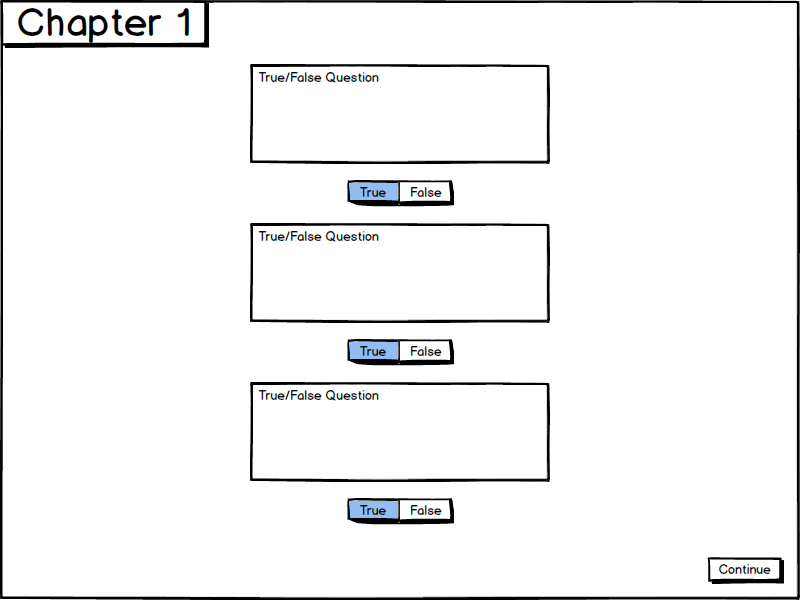
Relation to Use Case: 4 as it allows the user to view and answer true/false questions.

Assigned Developers: Stephen, Michael

Due Date: 2-17-16

Priority: High

Mockup:



**US-7: Multiple Choice**

As a user, i would like to select my answer for all chapter's multiple choice questions in order to see if I got the correct answer

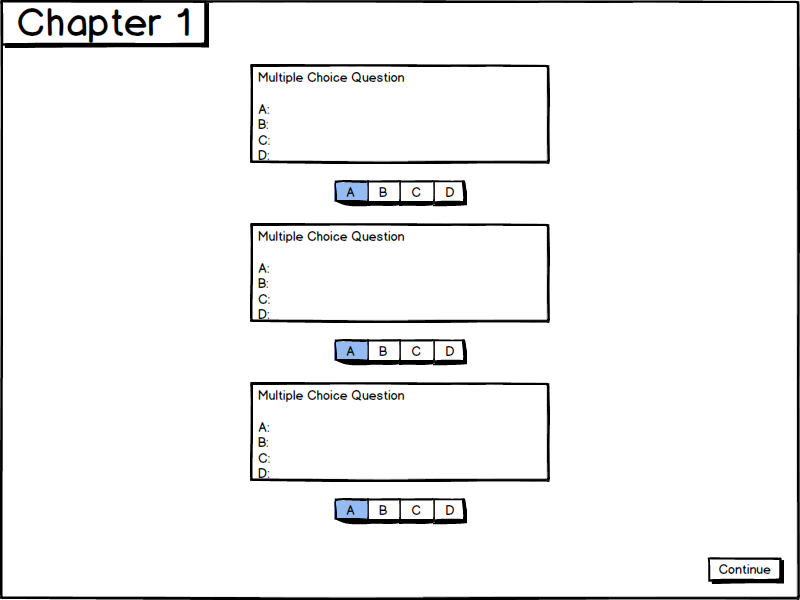
Relation to Use Case: 5 as it allows the user to login to view and answer multiple choice questions.

Assigned Developers: Jonathan, Kevin

Due Date: 2-17-16

Priority: High

Mockup:



**US-8: Results Screen**

As a user, I would like to access my results screen so that I can review my answers and go back to the questions I missed.

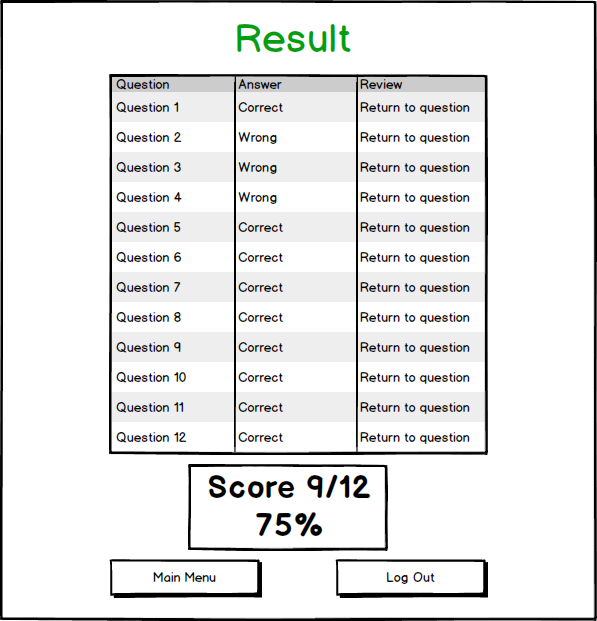
Relation to Use Case: 8 and 9, as it allows the user to view their results and return back to questions.

Assigned Developer: Stephen, Jonathan

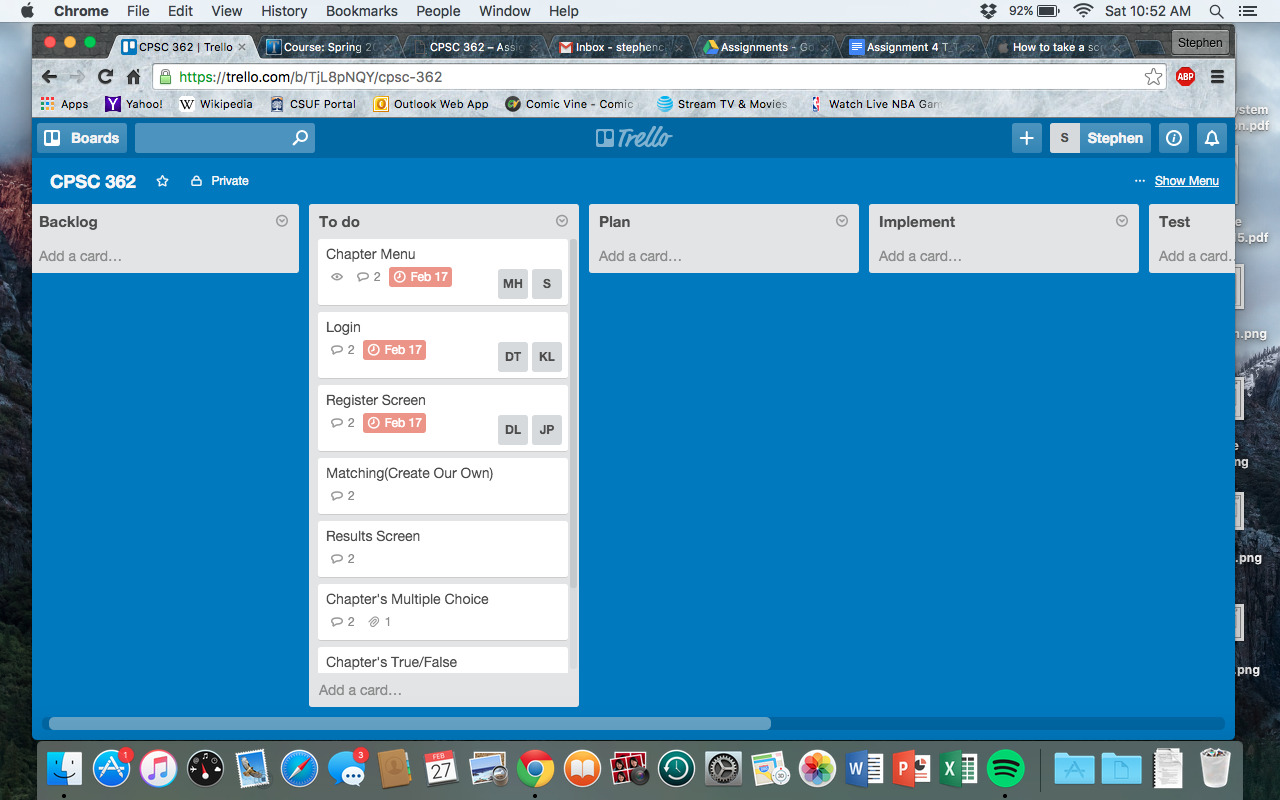
Due Date: 4/3/16

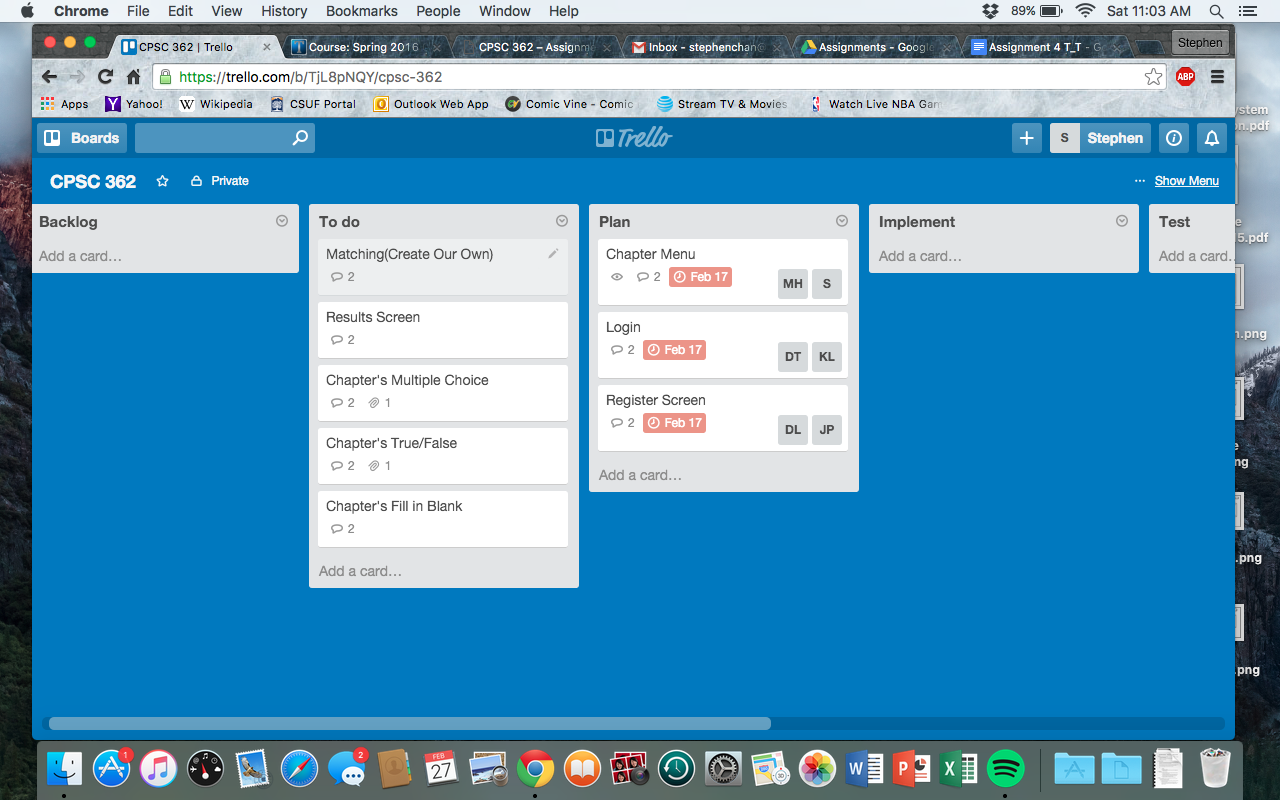
Priority: High

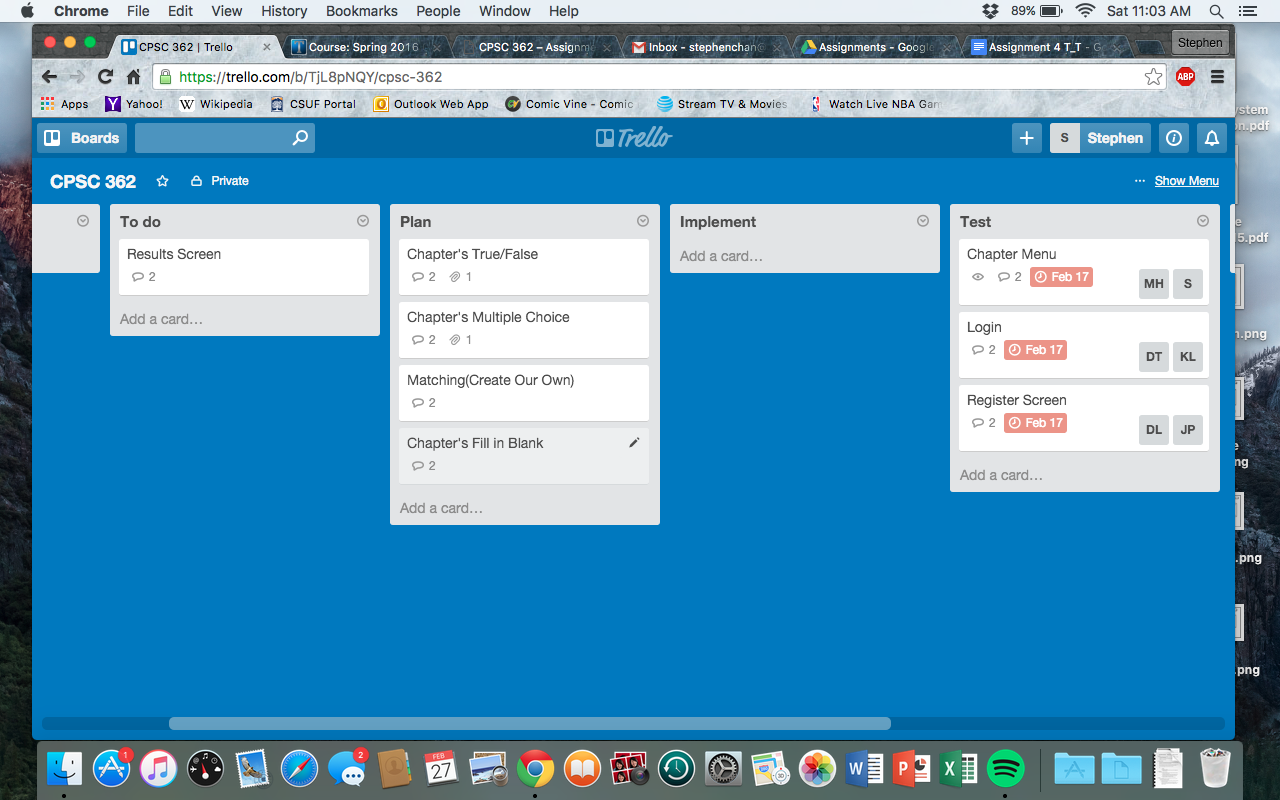
Mockup:

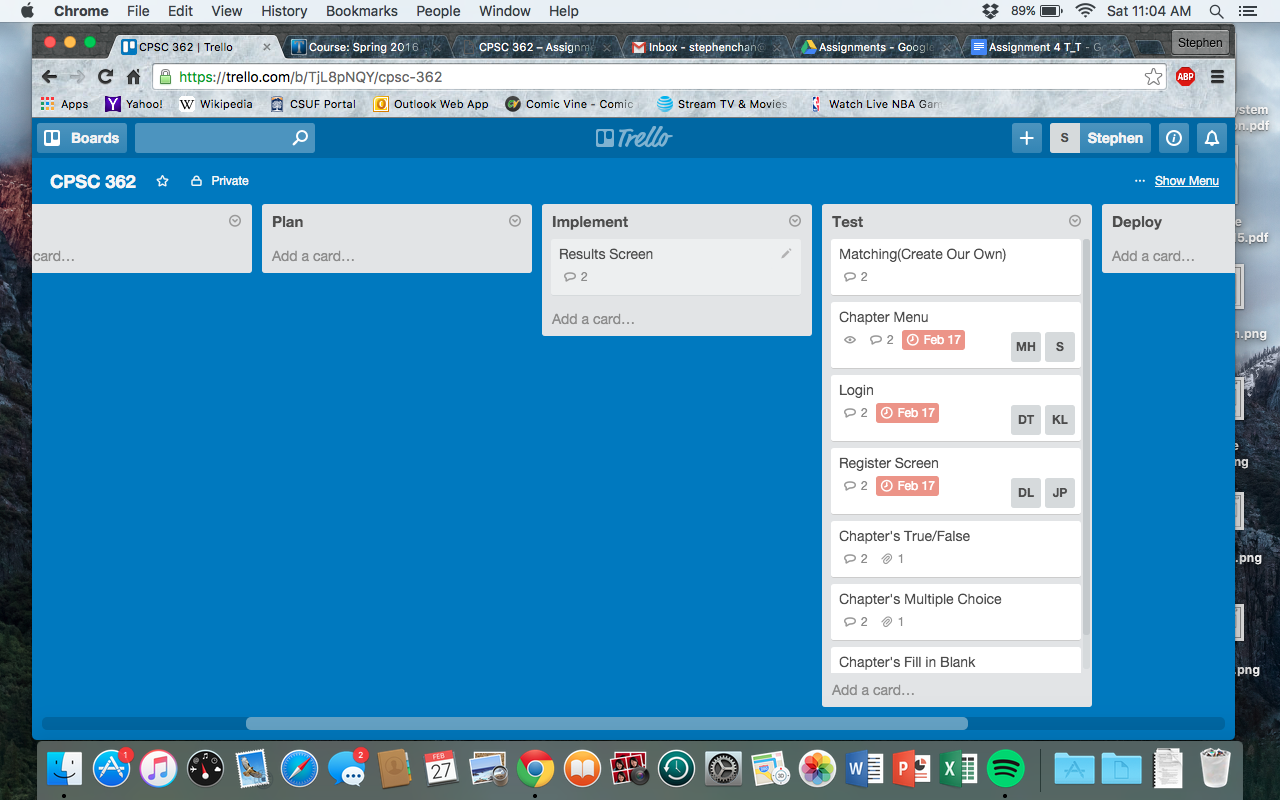


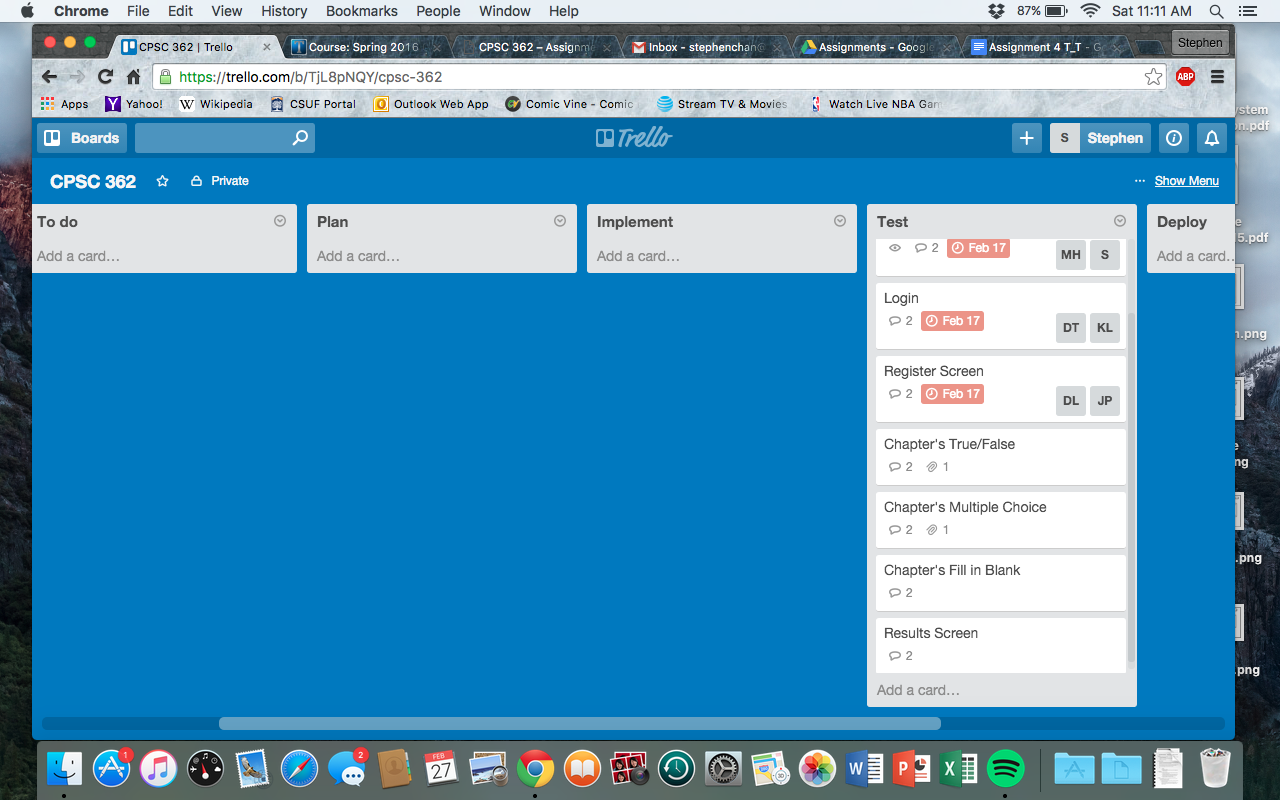
**9. Sprint Backlog Screen shots of your Trello Kanban-Board.**











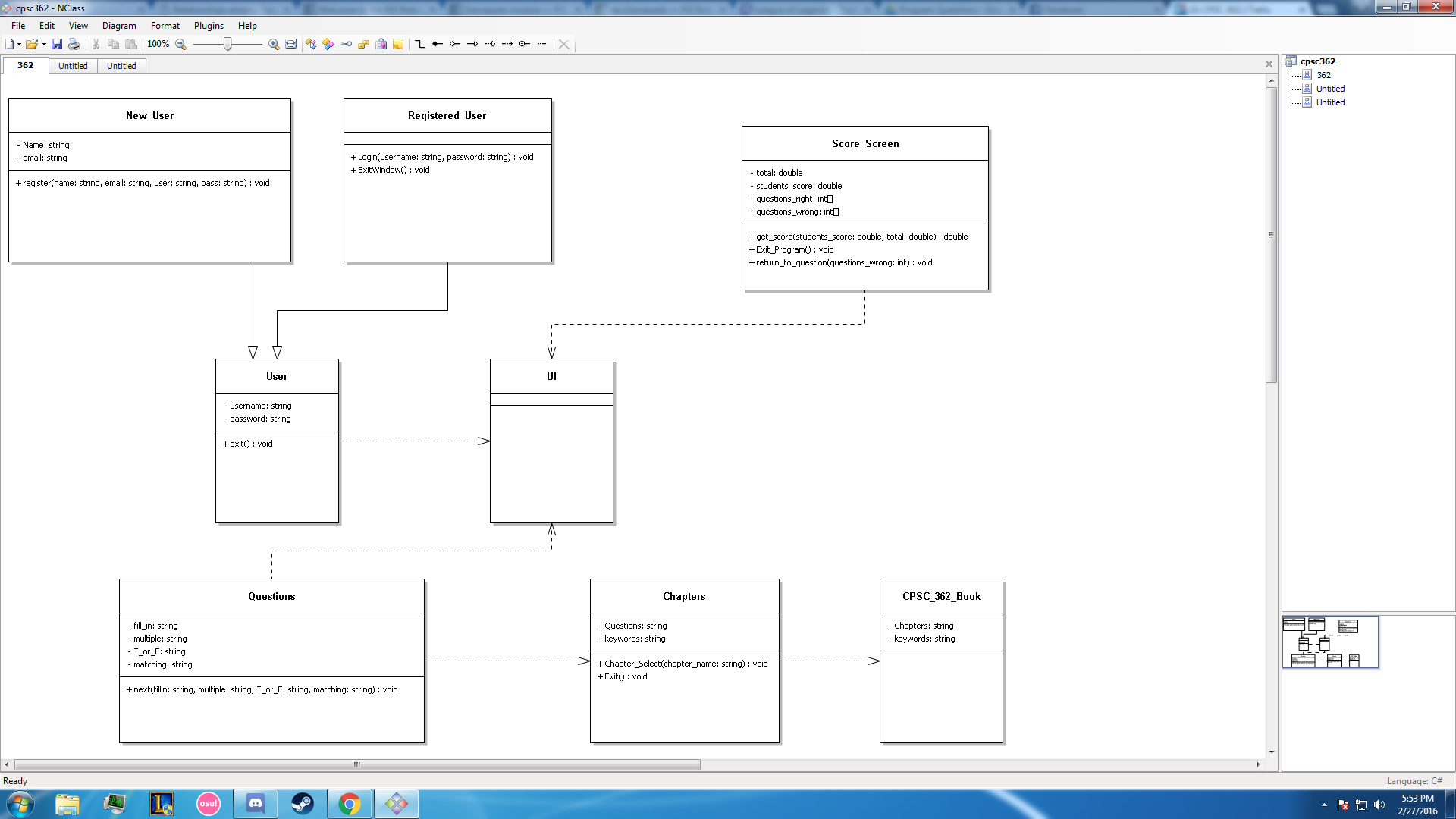
**Explain how your team developed your system and documented activities by producing work products related to this phase.(Write 2-3 paragraphs explaining what you did as well as attaching items 10,11,12 and 13)**

In order to develop our system correctly, we first made sure components of our program ran efficiently without any errors. We focused on 3 aspects of our user stories which were, Register screen, Login Screen, Chapter Select. We had our programmers develop these screens immediately without any errors while other members of our group were each in charge of developing questions for their own respective chapters. These screens were tested periodically with our programmers in order to fix bugs.

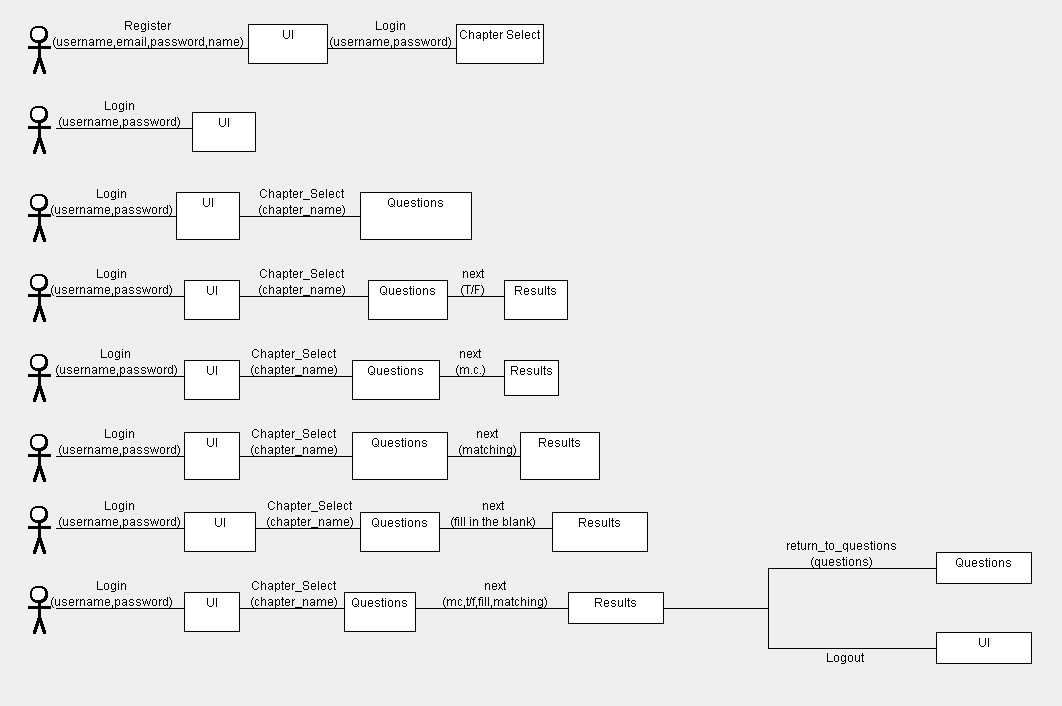
We then worked on implementing the questions page with all of our created questions for each chapter. We developed the questions page in accordance to each chapter and made sure each form had a similar template with each other. Most chapters have the same question template with their correct functions. These pages were tested with multiple choices being selected.

The last page was the results screen page that was programmed in sync with our questions page. We each tested the functionality of the results screen “return to question” button in order to see if answers were saved. The program was developed with few hiccups and room for improvement.

**10. Class diagram (includes all the classes that are included to your code).**



**11. Communication Diagrams based on your Class diagram.**



**12. Test Plan, Test suit and Test Cases.**

|  |  |
| --- | --- |
| **Test Designed by: David Luong** | **Module Name: Register screen** |
| **Test Designed date: 2-17-16** | **Test Title: Successfully registering new account** |
| **Test Case ID: UC-1,** | **Description: Registering a new user without any errors.** |
| **Test Priority (Low/Medium/High): High** |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** |
| **1** | **Navigate to registration screen** |  | **User should be able to create a new account** | **User is allowed to register their account** | **Pass** |
| **2** | **Provide valid name** | **Name = SomeName** |  |  | **Pass** |
| **3** | **Provide valid email** | **email = somerandom@kappa.com** |  |  | **Pass** |
| **4** | **Provide valid username** | **User= Kappa123** |  |  | **Pass** |
| **5** | **Provide valid password** | **Password: Kappa** |  |  | **Pass** |
| **6** | **Click on register button** |  | **User receives a message saying information is registered.** | **Success message appears on screen** | **Pass** |

**Post-conditions:**

**User is registered into the database and the user is now able to login with the new user at the login screen.**

|  |  |
| --- | --- |
| **Test Designed by: David Luong** | **Module Name: Register screen** |
| **Test Designed date: 2-17-16** | **Test Title: User enters a username that already exists** |
| **Test Case ID:2 UC-1, extension 3-a** | **Description: User should not be able to make a new account with a username that already exists in the system.** |
| **Test Priority (Low/Medium/High): High** |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** | **Notes** |
| **1** | **Navigate to registration screen** |  | **User should be able to create a new account** | **User is allowed to register their account** | **Pass** |  |
| **2** | **Provide valid name** | **Name = SomeName** |  |  | **Pass** |  |
| **3** | **Provide valid email** | **email = somerandom@kappa.com** |  |  | **Pass** |  |
| **4** | **Provide invalid username** | **User= alreadyExists** |  |  | **Pass** |  |
| **5** | **Provide valid password** | **Password: Kappa** |  |  | **Pass** |  |
| **6** | **Click on register button** |  | **User receives a message saying “username already exists”** | **Message is displayed on screen requiring user to change the username** | **Pass** |  |

**Post-conditions:**

**User is registered into the database and the user is now able to login with the new user at the login screen.**

|  |  |
| --- | --- |
| **Test Designed by: David Luong** | **Module Name: Register screen** |
| **Test Designed date: 2-17-16** | **Test Title: User enters less than 5 characters for their username** |
| **Test Case ID:3 UC-1, extension 3b** | **Description: A message should prompt the user to enter more than the enough characters for the username** |
| **Test Priority (Low/Medium/High): High** |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** | **Notes** |
| **1** | **Navigate to registration screen** |  | **User should be able to create a new account** | **User is allowed to register their account** | **Pass** |  |
| **2** | **Provide valid name** | **Name = SomeName** |  |  | **Pass** |  |
| **3** | **Provide valid email** | **email = somerandom@kappa.com** |  |  | **Pass** |  |
| **4** | **Provide valid username** | **User=**  **Kapp** |  |  | **Pass** |  |
| **5** | **Provide valid password** | **Password: Kappa** |  |  | **Pass** |  |
| **6** | **Click on register button** |  | **User receives a message saying that there is not enough characters** | **Success message appears on screen** | **Pass** |  |

**Post-conditions:**

**User is registered into the database and the user is now able to login with the new user at the login screen.**

|  |  |
| --- | --- |
| **Test Designed by: David Luong** | **Module Name: Register screen** |
| **Test Designed date: 2-17-16** | **Test Title:User has no inputs for username** |
| **Test Case ID:4 UC-1, extension 3c** | **Description: A message should prompt the user to enter a username.** |
| **Test Priority (Low/Medium/High): High** |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** | **Notes** |
| **1** | **Navigate to registration screen** |  | **User should be able to create a new account** | **User is allowed to register their account** | **Pass** |  |
| **2** | **Provide valid name** | **Name = SomeName** |  |  | **Pass** |  |
| **3** | **Provide valid email** | **email = somerandom@kappa.com** |  |  | **Pass** |  |
| **4** | **Provide valid username** | **User= “”** |  |  | **Pass** |  |
| **5** | **Provide valid password** | **Password: Kappa** |  |  | **Pass** |  |
| **6** | **Click on register button** |  | **User receives a message saying that the user must input a username** | **Success message appears on screen** | **Pass** |  |

**Post-conditions:**

**User is registered into the database and the user is now able to login with the new user at the login screen.**

|  |  |
| --- | --- |
| **Test Designed by: David Luong** | **Module Name: Register screen** |
| **Test Designed date: 2-17-16** | **Test Title: User enters less than 8 characters for their password** |
| **Test Case ID:5 UC-1, extension 4a** | **Description: A message should prompt the user to enter more than the enough characters for the password.** |
| **Test Priority (Low/Medium/High): High** |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** | **Notes** |
| **1** | **Navigate to registration screen** |  | **User should be able to create a new account** | **User is allowed to register their account** | **Pass** |  |
| **2** | **Provide valid name** | **Name = SomeName** |  |  | **Pass** |  |
| **3** | **Provide valid email** | **email = somerandom@kappa.com** |  |  | **Pass** |  |
| **4** | **Provide valid username** | **User= Kappa123** |  |  | **Pass** |  |
| **5** | **Provide valid password** | **Password: Kappa** |  |  | **Pass** |  |
| **6** | **Click on register button** |  | **User receives a message saying that there is not enough characters** | **Success message appears on screen** | **Pass** |  |

**Post-conditions:**

**User is registered into the database and the user is now able to login with the new user at the login screen.**

|  |  |
| --- | --- |
| **Test Designed by: David Luong** | **Module Name: Register screen** |
| **Test Designed date: 2-17-16** | **Test Title: Use has no inputs for password** |
| **Test Case ID:6 UC-1, extension 4b** | **Description: A message should prompt the user to enter a password.** |
| **Test Priority (Low/Medium/High): High** |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** | **Notes** |
| **1** | **Navigate to registration screen** |  | **User should be able to create a new account** | **User is allowed to register their account** | **Pass** |  |
| **2** | **Provide valid name** | **Name = SomeName** |  |  | **Pass** |  |
| **3** | **Provide valid email** | **email = somerandom@kappa.com** |  |  | **Pass** |  |
| **4** | **Provide valid username** | **User= Kappa123** |  |  | **Pass** |  |
| **5** | **Provide valid password** | **Password:**  **“”** |  |  | **Pass** |  |
| **6** | **Click on register button** |  | **User receives a message saying that the user must input a password** | **Success message appears on screen** | **Pass** |  |

**Post-conditions:**

**User is registered into the database and the user is now able to login with the new user at the login screen.**

|  |  |
| --- | --- |
| **Test Designed by: David Luong** | **Module Name: Register screen** |
| **Test Designed date: 2-17-16** | **Test Title: Use has no inputs for password** |
| **Test Case ID:7 UC-1, extension 4b** | **Description: A message should prompt the user to enter a password.** |
| **Test Priority (Low/Medium/High): High** |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** | **Notes** |
| **1** | **Navigate to registration screen** |  | **User should be able to create a new account** | **User is allowed to register their account** | **Pass** |  |
| **2** | **Provide valid name** | **Name = SomeName** |  |  | **Pass** |  |
| **3** | **Provide valid email** | **email = somerandom@kappa.com** |  |  | **Pass** |  |
| **4** | **Provide valid username** | **User= Kappa123** |  |  | **Pass** |  |
| **5** | **Provide valid password** | **Password:**  **“”** |  |  | **Pass** |  |
| **6** | **Click on register button** |  | **User receives a message saying that the user must input a password** | **Success message appears on screen** | **Pass** |  |

**Post-conditions:**

**User is registered into the database and the user is now able to login with the new user at the login screen.**

|  |  |
| --- | --- |
| **Test Designed by: David Luong** | **Module Name: Accessing the login page** |
| **Test Designed date: 2-17-16** | **Test Title: User logins into program** |
| **Test Case ID:8 UC-2** | **Description: The user needs to login to the program** |
| **Test Priority (Low/Medium/High): High** | **Pre-conditions: Valid Username and password** |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** | **Notes** |
| **1** | **User enters username into textbox.** | **Username =SomeName** | **Username is entered into text box.** | **Username is entered into text box and awaits password** | **Pass** |  |
| **2** | **User enters password into password text box.** | **Password = kappa123** | **Password is entered into text box.** | **Password is entered into textbox and awaits user to click login button.** | **Pass** |  |
| **3** | **User clicks on login button** |  | **The program checks username and password** | **Username and password checks are successful.Moves user to the next screen if successful.** | **Pass** |  |
| **4** | **The System brings user to chapter menu selection window.** |  | **Chapter menu is displayed** | **Chapter menu is displayed** | **Pass** |  |

|  |  |
| --- | --- |
| **Test Designed by: David Luong** | **Module Name: Accessing the login page** |
| **Test Designed date: 2-17-16** | **Test Title: User logins into program** |
| **Test Case ID:9 UC-2, Extension 1a** | **Description: The user needs to login to the program** |
| **Test Priority (Low/Medium/High): High** | **Pre-conditions: Incorrect Username and password** |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** | **Notes** |
| **1** | **User enters wrong username or password into textbox.** | **Username =wrongUser** | **A pop up screen will state that the username or password is invalid** | **The pop up screen appears** | **Pass** |  |
| **2** | **User enters wrong password** | **Password = wrongPass** | **A pop up screen will state that the username or password is invalid** | **The pop up screen appears** | **Pass** |  |

|  |  |
| --- | --- |
| **Test Designed by: David Luong** | **Module Name: Accessing the login page** |
| **Test Designed date: 2-17-16** | **Test Title: User logins into program** |
| **Test Case ID:10 UC-2, Extension 1b** | **Description: The user needs to login to the program** |
| **Test Priority (Low/Medium/High): High** | **Pre-conditions: Invalid password greater than ten characters** |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** | **Notes** |
| **1** | **User enters username into textbox.** | **Username =correctuser** | **Username is entered into text box.** | **Username is entered into text box and awaits password** | **Pass** |  |
| **2** | **User enters password that is more than 20 characters** | **Password = password>20char** | **Password is entered into text box.** | **Prompts user to enter in shorter password.** | **Pass** |  |
| **3** | **User clicks on login button** |  | **The error message should pop up** | **A pop up screen will appear stating that the password is invalid** | **Pass** |  |

|  |  |
| --- | --- |
| **Test Designed by: David Luong** | **Module Name: Accessing the login page** |
| **Test Designed date: 2-17-16** | **Test Title: User logins into program** |
| **Test Case ID:11 UC-2, Extension 1c** | **Description: The user needs to login to the program** |
| **Test Priority (Low/Medium/High): High** | **Pre-conditions: Invalid password shorter than six characters** |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** | **Notes** |
| **1** | **User enters username into textbox.** | **Username =wrongUser** | **Username is entered into text box.** | **Username is entered into text box and awaits password** | **Pass** |  |
| **2** | **User enters password that is less than 6 characters** | **Password = password<6char** | **Password is entered into text box.** | **Prompts user to enter in shorter password.** | **Pass** |  |
| **3** | **User clicks on login button** |  | **The error message should pop up** | **A pop up screen will appear stating that the password is invalid** | **Pass** |  |

|  |  |
| --- | --- |
| **Test Designed by: David Tran** | **Module Name: Select a chapter in the Chapter menu** |
| **Test Designed date: 2-21-16** | **Test Title: Verify to see that a successful login would allow the user to be able to access and select a chapter in the chapter menu page.** |
| **Test Case ID:12 UC-3** | **Description: Test to see if the user, who is logged in, can access the chapter menu.** |
| **Test Priority (Low/Medium/High): High** | **Pre-conditions: User is already logged in.** |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** | **Notes** |
| **1** | **User selects a chapter from the chapter menu** |  | **User should be navigated to that chapter’s page with questions.** | **User is taken to the chapter page that they selected.** | **Pass** |  |

**Post-conditions:**

**User is validated with database and successfully login to account. The account sessio details are logged in database.**

|  |  |
| --- | --- |
| **Test Designed by: Michael Ha** | **Module Name: Chapter Questions** |
| **Test Designed date: 2-20-16** | **Test Title: Select/Viewing the chapter’s True and False Q’s** |
| **Test Case ID:13 UC-4** | **Description: Test the question pages** |
| **Test Priority (Low/Medium/High): Med** | **Pre-conditions: User has logged on and has chosen a chapter** |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** | **Notes** |
| **1** | **User is able to view the true and false questions** |  | **User is able to view the true and false questions for this chapter** | **The true and false questions appear** | **Pass** |  |
| **2** | **User is able to answer the true and false questions** |  | **User is able to select the true or false radio buttons** | **True and false radio buttons are selectable for the question** | **Pass** |  |

**Post-conditions: Users score is shown and can be seen**

|  |  |
| --- | --- |
| **Test Designed by: Michael Ha** | **Module Name: Chapter Questions** |
| **Test Designed date: 2-20-16** | **Test Title: Select/Viewing the chapter’s True and False Q’s** |
| **Test Case ID:14 UC-4, extension a** | **Description: User does not answer the questions** |
| **Test Priority (Low/Medium/High): Med** | **Pre-conditions: User has logged on and has chosen a chapter** |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** | **Notes** |
| **1** | **User does not answer the question** |  | **The question is counted wrong if there is no input.** | **The user chose not to select either radio buttons.** | **Pass** |  |

**Post-conditions:**

**Users score is shown and can be seen**

|  |  |
| --- | --- |
| **Test Designed by: Kevin Le** | **Module Name: Multiple choice Questions** |
| **Test Designed date: 2-20-16** | **Test Title: Answer the multiple choice questions** |
| **Test Case ID:15 UC-5** | **Description: Test the multiple choice** |
| **Test Priority (Low/Medium/High): Med** | **Pre-conditions: User has already selected a chapter.** |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** | **Notes** |
| **1** | **User looks at the multiple choice questions** |  | **User should be able to answer the multiple choice** | **Questions are viewable on screen** | **Pass** |  |
| **2** | **User answers the three questions by choosing the options available** |  | **User can select the options displayed** | **User can select and also change their answers.** | **Pass** |  |
| **3** | **User submits the answers.** |  | **User can submit questions by clicking on the button.** | **User is brought to the next screen** | **Pass** |  |

**Post-conditions:**

**System allows the user to go to the results screen.**

|  |  |
| --- | --- |
| **Test Designed by: Kevin Le** | **Module Name: Multiple choice Questions** |
| **Test Designed date: 2-20-16** | **Test Title: Answer the multiple choice questions** |
| **Test Case ID:16 UC-5, extension a** | **Description: Test the multiple choice** |
| **Test Priority (Low/Medium/High): Med** | **Pre-conditions: User has already selected a chapter.** |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** | **Notes** |
| **1** | **User is able to view the multiple choice questions** |  | **User should be asked to select an answer to the question** | **User is able to view the multiple choice questions for this chapter** | **Pass** |  |
| **2** | **User does not select an answer for any of the multiple choice** |  | **User is able to select the multiple choice radio buttons** | **Multiple choice radio buttons are selectable for the question** | **Pass** |  |

**Post-conditions:**

**System allows the user to go to the results screen.**

|  |  |
| --- | --- |
| **Test Designed by: Jonathan Peng** | **Module Name: Fill in the Blank Questions** |
| **Test Designed date: 2-20-16** | **Test Title: Test if the user can view/answer fill in the blank questions** |
| **Test Case ID:17 UC-6** | **Description: Test to see if the user can view and answer fill in the blank questions.** |
| **Test Priority (Low/Medium/High): High** | **Pre-conditions: User is logged in.** |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** | **Notes** |
| **1** | **View the fill in the blank questions** |  | **Able to see the questions listed for the chapter** | **User should be able to see the questions** | **Pass** |  |
| **2** | **Enter text into the box to submit the answer** | **Answer = “deployment”** | **Users should be able to input characters in the textbox** | **Users should be able to input their answers** | **Pass** |  |

|  |  |
| --- | --- |
| **Test Designed by: Jonathan Peng** | **Module Name: Fill in the Blank Questions** |
| **Test Designed date: 2-20-16** | **Test Title: Verify to see if users answers the fill in the blank questions** |
| **Test Case ID:18 UC- 6, extension 3a** | **Description: Test to see if the user, who logged in, answer all the questions for fill in the blank.** |
| **Test Priority (Low/Medium/High): High** | **Pre-conditions: User did not answer every fill in the blank question.** |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** | **Notes** |
| **1** | **View the fill in the blank questions** |  | **Able to see the questions** | **User should be able to see the questions** | **Pass** |  |
| **2** | **Enter text into the box to submit the answer** | **Answer = “ “** | **The question should be counted wrong.** | **The question is counted wrong** | **Pass** |  |

|  |  |
| --- | --- |
| **Test Designed by: Stephen Chan** | **Module Name: Matching Questions page.** |
| **Test Designed date: 2-22-16** | **Test Title: User is able to view and match correct keywords with definitions** |
| **Test Case ID:19 UC-7** | **Description: User is able to match buttons with respective definitions.** |
| **Test Priority (Low/Medium/High): High** | **Pre-conditions: User is logged in.** |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** | **Notes** |
| **1** | **User views keywords and definitions** |  | **Buttons are and boxes are aligned with definitions on the other side** | **Question with keywords are displayed.** | **Pass** |  |
| **2** | **User can enter respective letter next to a definition.** |  | **Box allows for user to make an entry.** |  | **Pass** |  |

|  |  |
| --- | --- |
| **Test Designed by: Stephen Chan** | **Module Name: Matching Questions page.** |
| **Test Designed date: 2-22-16** | **Test Title: User does not match any of the keywords with a definition before hitting submit.** |
| **Test Case ID:20 UC-7 , extension 3a** | **Description: User attempts to proceed without matching any of the keywords.** |
| **Test Priority (Low/Medium/High): High** | **Pre-conditions: User is logged in.** |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** | **Notes** |
| **1** | **User views keywords and definitions** |  | **Buttons are and boxes are aligned with definitions on the other side** | **Questions are displayed** | **Pass** |  |
| **2** | **User leaves boxes empty.** |  | **The questions are counted wrong.** | **Questions are counted wrong.** | **Pass** |  |

**Post-conditions:**

**User finishes matching questions**

|  |  |
| --- | --- |
| **Test Designed by:David Luong** | **Module Name: View Results** |
| **Test Designed date: 2-17-16** | **Test Title: View Results Screen** |
| **Test Case ID:21 UC-8** | **Description: Shows the results of the review questions. Shows wrong and right answers.** |
| **Test Priority (Low/Medium/High): High** | **Pre-conditions: The user has finished last question and have answered all review questions for the chapter.** |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** | **Notes** |
| **1** | **User is able to see the questions they got correct and the ones they got wrong.** |  | **User sees their questions along with right and wrong answers.** | **User sees questions indicating whether they are correct or not** | **Pass** |  |
| **2** | **User is able to see the clickable buttons that allow them to logout and return back to the chapter menu.** |  | **User is able to see the buttons and use them correctly.** | **User is able to visually see the buttons and click them as needed.** | **Pass** |  |
| **3** | **User also sees a calculated percentage of the questions they got correct.** |  | **Percentage is shown and is visible to user** | **The percentage is shown and is visible to user.** | **Pass** |  |

**Post-conditions:**

|  |  |
| --- | --- |
| **Test Designed by: Stephen Chan** | **Module Name: Choosing to return back to a question in the results screen.** |
| **Test Designed date: 2-22-16** | **Test Title: User sees results from the questions.** |
| **Test Case ID:22 UC-9** | **Description: User is able to return back to questions they got wrong or restart the program.** |
| **Test Priority (Low/Medium/High): High** | **Pre-conditions: User is logged in. User has attempted all questions.** |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** | **Notes** |
| **1** | **User is able to see a list of questions that reveals which questions they got correct.** |  | **List of the questions with an indicator of whether they got it right or wrong.** | **Questions apprear with an indicator of if they’re right or wrong.** | **Pass** |  |
| **2** | **User is able to click on a return back to questions button** |  | **User is returned back to all questions.** | **User is returned back to all questions** | **Pass** |  |

**Post-conditions:**

**User is finished with the program.**

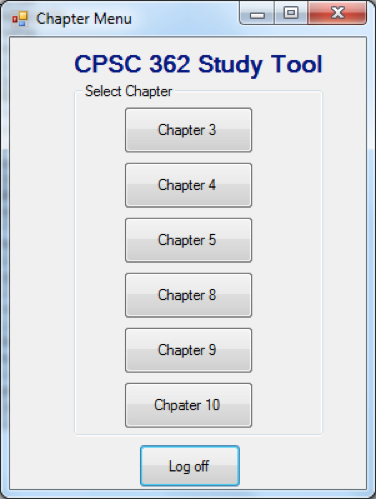
|  |
| --- |
|  |
|  |
|  |
|  |

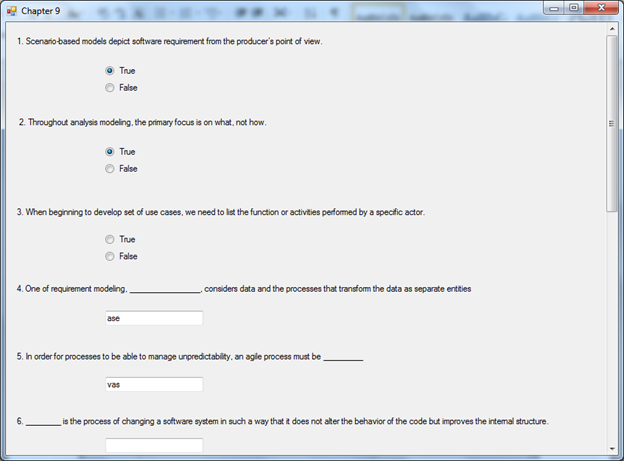
**13. Your code for the 5 functional user stories.**

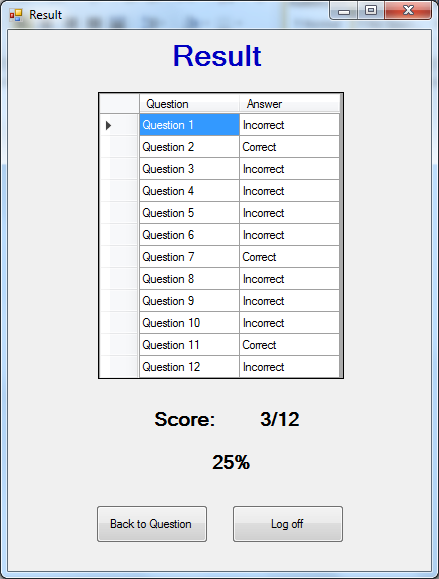
**14. User manual – Screen shots of your working product with explanation on How to use your system.**











**15. References (list references here, and cite them in appropriate places in the report)**

Software Engineering: A Practitioner’s Approach by Roger Pressman

**16. Team Charter (in the given format)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Course Title** | CPSC 362 |  | All team members participated in the creation of this charter and agree with its content. **Date** 01/28/2014 |
| **Instructor** | Yasamin Ehteshami |  |
| **Course Dates** | MW 5:00-6:50 |  |

**Team Members** (Contact Information)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Address (city, state, country) | Phone | Cell | Email |
| Stephen Chan | La Mirada, CA | 562-650-2700 |  | stephenchan@csu.fullerton.edu |
| David Tran | Anaheim, CA | 714-487-2653 |  | DavidTran794@csu.fullerton.edu |
| Michael Ha | Westminster, CA | 714-725-9569 |  | Mha94@csu.fullerton.edu |
| Kevin Le | Westminster, CA | 714-925-3210 |  | lekevin42@csu.fullerton.edu |
| Jonathan Peng | Chino Hills, CA | 909-348-4201 |  | jspeng@csu.fullerton.edu |
| David Luong | Santa Ana, CA | 714-360-3083 |  | davidluong@csu.fullerton.edu |

**Team Member Skill Inventory** (Areas individual members can contribute)

|  |  |
| --- | --- |
| Stephen Chan | § C++,SQL,Python |
| David Tran | § C++, Python, HTML5, SQL |
| Michael Ha | § C++, Python, Assembly |
| Kevin Le | § C++, python, C, Assembly |
| Jonathan Peng | § C++,Assembly, Java, C# |
| David Luong | § C++, Java, Python, CSS, HTML server management and hosting experience. |

**Team Goals** (Project goals, team process goals, quality goals, etc.)

|  |
| --- |
| § Efficient Application, Finish a few days before deadline. Program will compile without errors. |

**Team Roles** (Define roles of each member to achieve goals)

|  |  |
| --- | --- |
| Stephen Chan | § Scrum Master, Recorder, Developer |
| David Tran | § Developer, Tester |
| Michael Ha | § Developer, tester |
| Kevin Le | § Developer, tester |
| Jonathan Peng | § Developer, tester |
| David Luong | § Developer, Tester |

**Ground Rules** (Meeting schedule/locations, attendance expectations, agenda, assignment completion, communication methods, etc.)

|  |
| --- |
| § Group will meet and discuss project in Google Hangout sessions.  § All team members must be punctual and prepared for each team meeting.  § Participation and input is expected from all team members. All opinions will be considered and equally valued.  § The team will meet at least once each week via chat or conference call to discuss current and upcoming projects or assignments (tentatively scheduled for every Saturday @ 8:30 pm).  § Team members will notify the lead in advance if they are not going to be able to attend a scheduled meeting.  § Team members should check email at least once a day to stay on top of things.  § Team members should reply to email within 24 hours.  § Team members will turn in team assignments no later than two weeks prior to the due date.  § All team members will be held accountable for their portions of the projects and are expected to complete them in a timely manner and doing the best job they can.  § Notify team of emergencies that may result in not being able to meet deadlines or meetings. The rest of the team will do their best to pitch in on the team assignment.  § The team must maintain open, clear, and effective communication at all times.  § Assist fellow team members when they are in need.  Team will collectively decide when to meet.  Team will not form alliances or teams against one another.  § Maintain a positive, honest, and open atmosphere by respecting other members’ suggestions, using constructive criticism, and encouragement. |

**Time Commitments/Availability** (Pacific Time)

|  |  |
| --- | --- |
| Stephen Chan | § Friday ALL DAY, Weekend(free) |
| David Tran | § Friday(afternoon-night), Sunday(Morning,Night) |
| Michael Ha | § All day fri-sun, mon-thurs anytime except 5-9pm |
| Kevin Le | § Friday - Sunday |
| Jonathan Peng | § Weekend(free) |
| David Luong | § Everyday(morning) |

**Conflict Management** (What are potential conflicts that might arise among or between team members during this course? How will team members deal with these and other conflicts?)

|  |
| --- |
| Scrum Master will decide course of action.  § In order to avoid conflict clear roles and responsibilities must be assigned, so that there is no confusion.  § If a team member is not performing, the team lead will speak to the member and try and resolve the issue.  § If conflicts arise, please bring them up to the whole team so that everybody can help to resolve the issue in a peaceful and harmonious manner.  § All team members must settle conflicts within the group as quickly as possible. |

**Risk Management** (What are potential barriers to the achievement of these goals?)

|  |
| --- |
| Project Must be in a programming language all developers are comfortable with.  § Scrum Master and Recorder will be in charge of managing  § Any issues between the Scrum Master and the Recorder will be resolved by involving the entire group  § List risks that are chances or possibilities of suffering loss or danger in the project.   * Computer breaks.   + Solution: Make sure to backup files on USB or email * Files are lost for any reason.   + Solution: Upload to Google Drive. * The possibility that we will not finish project on time.   + Solution: Make sure to keep an eye on progress made throughout the semester, set deadlines * Arguments that threaten the group project   + Solution: Make sure to address everyone’s concerns before moving forward |

**Team Evaluation Criteria** (List evaluation criteria that will be used to evaluate team members objectively.)

|  |
| --- |
| * Response time for emails, texts. * Updates on current progress of assignments. * Whether they’re able to attend meetings. * Actual completion of project components. * Evaluate performance based upon efficient code and time of submission. |

**17. Team Evaluation (in the given format).**