

# Quality Assurance Plan

## Assignment #2

Website : <http://www.code.google.com/p/busapp>

iHouse group number 12 members:

- David Kim
- Song Bae Choi
- Mario Elena
- Rory Rock

CMPT 275

Due: October 13<sup>th</sup>, 2010

Instructor: Herbert Tsang





## Software Tools

### 1.0 Software Tools for Testing

The tools that will be used to test our application will be the xCode Software Development Kit (SDK) and the iPhone device. The SDK contains a source editor, debugger, testing environment, iPhone simulator, and a performance measurement tools kit.

#### 1.1 xCode Software Development Kit

The xCode SDK offers two types of unit tests:

Logic Tests - This test checks the functionality of the program and exercises the code at a very low level or part of many classes working together.

Application Tests – This type of test checks the functionality of the application along with the feasibility of the user interface. It also checks for controls and controllers within the application commutation.

#### 1.2 iPhone Simulator

The iPhone simulator will be a major testing tool for our group. It will be used to test the functionality of the application before it is run on a real device.

#### 1.3 iPhone device

The application will be run on a real iPhone device to test the real time functionality of our application.

## Internal Deadlines

	Begin Coding	Ending Coding	# of Days	Begin Testing	End Testing	# of Day
Assignment3						
• Version1	10/14/2010	10/31/2010	18	11/01/2010	11/08/2010	8
Assignment4						
• Version2	11/09/2010	11/18/2010	10	11/19/2010	11/22/2010	4
Assignment5						
• Version3	11/23/2010	11/24/2010	2	11/25/2010	11/25/2010	1

# Testing

## **Integration Testing and System Testing**

Most features will be implemented as an independent unit and tested individually to ensure the correct functionality and output generated. Then we will slowly integrate our classes that are dependent of each other and make sure that those classes interact correctly. All other classes will be tested in this manner to ensure that our system will be built correctly. This is known as a Bottom Up approach. An example of how our integration testing could be like is as followed:

1. Build Main Menu, Prospective Student Menu, Current Student Menu, and News & Events Menu.
  - a. Main Menu will hold 3 links for each category.
  - b. Each category will be in its simplest form and contain only a “Back” option.
  - c. Verify successful navigation from Main Menu to each category then back.
2. Build each subcategory within Prospective Student Menu.
  - a. Subcategory Menus.
    - i. About BUS Menu will contain only “Back” option.
    - ii. About BUS Menu will contain only “Back” option.
    - iii. Cost Menu will contain only “Back” option.
    - iv. Advisor Menu will contain only “Back” option.
    - v. Why BUS Menu will contain only “Back” option.
    - vi. Possible Career Paths Menu will contain only “Back” option.
  - b. Verify successful integration of sub menus from Prospective Student Menu to each subcategory then back.
3. Build each subcategory for Current Student Menu and News & Events Menu.
  - a. Perform similar tests to subcategory menus as defined in 2.

## **Unit Testing**

Completed versions will be demonstrated by a real user to test for overall application functionality.

User testing will be performed 12:00 PM on November 1, 2010 in the ASB Common Area. The user testing will have a minimum of 3 participants since the application is designed for three different audiences: visitors, prospective students, and current students. Each participant will be aware of their specific roles and test/use the application as needed.

### Testing Guidelines:

1. Provide each participant with a role: visitor, prospective student, or current student.
2. Present participants with the iPhone and ask them to launch the iHouse busapp application by selecting it from the main screen.
3. Ask participants to interact with the application to find necessary information they would need as specified by their role.
4. Ask participants to fill out a survey when they believe they are finished with the application. Their questions will be based on the users experience and on how they felt their needs were met.
5. Compare the results to our “Intended Audience” goals in the Requirements Document.

All testing of our versions will be carried out on the following dates:

Version 1 : Novemeber 1, 2010.

Version 2: November 19, 2010.

Version 3: November 25, 2010.

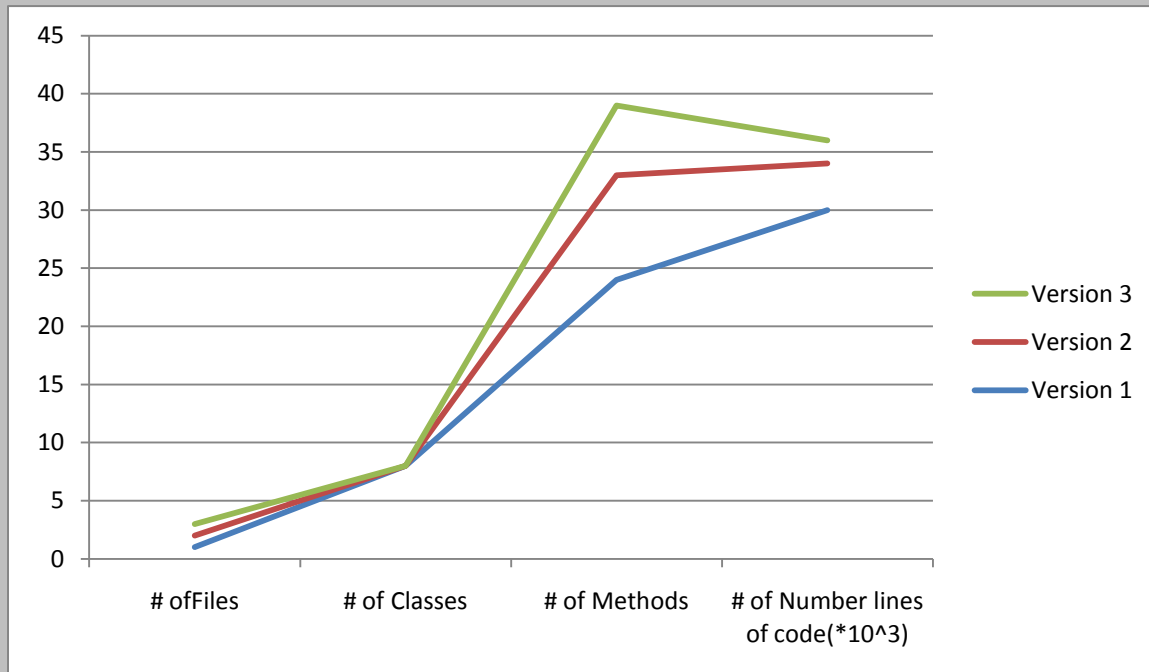
### Integration Testing vs. Unit Testing

The difference between integration testing and unit testing in our group will be that in integration testing, we will pass values and inputs that we would like to check for correctness. In unit testing, the user(s) will test random values based on their own ideas. Any incorrect output presented by user will mean that we did not do enough testing in the integration phase.

## Software Tools

Measuring size & complexity:

xCode will be used to measure the project size and complexity. In the xcode SDK, there is a feature called xCode Assistant, which is used to keep track of the number of files, classes, methods, and the number of lines of code.



The xCode SDK has other tools that can be used to gather performance data. Such as:

Instruments is a set of analysis tools that includes:

- Measurement metrics on file system reads and writes
- Stats on graphic operation and performance
- Information regarding memory leaks, thread activity and events

## Additional Quality Measurements

Any sick individual will not attend any meetings till they are healthy as it could jeopardize the teams' health and possibly impair their overall performance towards the project.

Any work done on the project will be documented and appendix to indicate work has been done.

Therefore group members will be able to compare previous versions to current versions and ensure quality remains high.

Testing must be done on legitimate devices that correspond with Apple's warranty agreement to ensure accurate results.