**

CMPE 287 - Software Quality Assurance and Testing

**Fall - 2014**

**Project – Testing Instagram mobile application**

****

**Instructor:**

Dr. Jerry Gao

**TEST PLAN**

**Submission Date: 10/10/2014**

**Submitted by:**

Team - Session1- Group 6

Divanshu Arneja - SJSU ID: 009437064

Piyush Seth - SJSU ID: 009425247

Saumil Jain - SJSU ID: 009428419

Shilpa - SJSU ID: 009392513

Surbhi Rautji - SJSU ID: 009317555

Table of Contents

[**1.** **Introduction** 3](#_Toc400662950)

[1.1. Project Overview 3](#_Toc400662951)

[1.2. Testing Objective 6](#_Toc400662952)

[1.3. Document purpose and structure 9](#_Toc400662953)

[1.4. Scope 9](#_Toc400662954)

[1.5. Team interaction 9](#_Toc400662955)

[**2.** **Coverage and Testing Target** 10](#_Toc400662956)

[2.1. Non Functional Testing 11](#_Toc400662957)

[2.2. Functional Testing 11](#_Toc400662958)

[2.2.1. System level function testing 11](#_Toc400662959)

[2.2.2. Automation System Testing 12](#_Toc400662960)

[**3.** **Testing Requirements** 14](#_Toc400662961)

[3.1. Test Environment 14](#_Toc400662962)

[3.2. Testing Platform 14](#_Toc400662963)

[3.3. Hardware Requirements 14](#_Toc400662964)

[3.4. Software Requirements 15](#_Toc400662965)

[**4.** **Testing Tasks and Schedule** 16](#_Toc400662966)

[4.1. Roles 16](#_Toc400662967)

[4.2. Responsibilities 16](#_Toc400662968)

[4.3. Project Deliverables 17](#_Toc400662969)

[4.4. Project Schedule 18](#_Toc400662970)

[4.5. Individual Testing Schedule 18](#_Toc400662971)

[**5.** **Testing Tools** 20](#_Toc400662972)

[5.1. Appium 20](#_Toc400662973)

[5.2. Elementool 22](#_Toc400662974)

[5.3. NeoLoad 24](#_Toc400662975)

[**6.** **Cost Analysis** 26](#_Toc400662976)

[**7.** **Testing Standards** 27](#_Toc400662977)

[7.1. Template for test case 27](#_Toc400662978)

[7.2. Template for test result 28](#_Toc400662979)

[7.3. Template for Bug/Issue Reporting 28](#_Toc400662980)

[**8.** **References** 30](#_Toc400662981)

## **Introduction**

The test plan document covers the test planning and strategies for testing the mobile application “Instagram”. Instagram is a mobile application which allows the users to take and share photos and videos. Digital filters can be applied to the images. The application also provides an ability to share photos and videos on a variety of social networking platforms, such as Facebook, Twitter etc. The application also provides geo tagging feature. Instagram is designed using various advanced technologies and can be used on various mobile operating system. In order to ensure that the application works as per the designed functionality, we have to use various black box and white box testing strategies.

The test plan includes the following –

* Requirement for testing
* Tools to be used
* Bug reporting
* Correcting and Tracking

## Project Overview

Instagram is a photo/video sharing application which allows user to click/record and share the photo/video on social networks. We can also apply various filters while clicking the pictures. The photos can be geo tagged too. User can view the pictures based on the geo location which they have visited. The various features and functions of the application are – clicking of photos, sharing on social networks, geo tagging, user profiles etc. Thus the main focus of the project is to test the various functionalities of the application using various testing methodologies and techniques to test the functioning, stability and performance. To ensure the quality of the application, the project includes black box testing, white box testing and performance testing of the application.



Figure 1 – Instagram logo

The application is available for various Mobile operating systems like iOS and Android. The application will be tested for functionality, performance and GUI testing on the various operating systems.

The below are the features of the application –

**Home Screen** – The user is able to view the feed of the users he is following

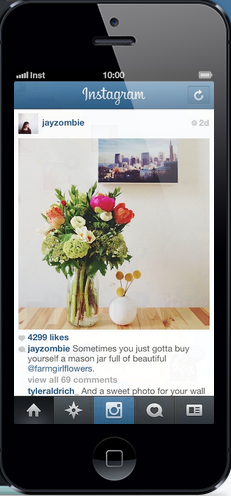


Figure 2 – Home Screen

**Camera** – Photo/Video capture feature



Figure 3 - Camera

**Geo tagging** – This feature lists out the photos based on locations on the map.



Figure 4 – Geo Tagging

**Following/News** – This feature provides the feed and updates.

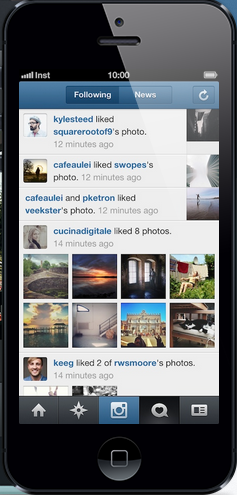


Figure 5 – Following/News

**User profile** – The user profile page.



Figure 6 – User Profile

## Testing Objective

The objective of testing this application is to validate and verify the functionality of the application. All of the components, features and functionality should perform as per the requirement and specification.

The features we test for the project are as follows –

1. Home Screen –

* The home screen of the Instagram application is a feed page which provides the feeds of the users that we are following.
* The homepage shows us the shared content that the users have shared.

1. Clicking photos and recording videos –

* Instagram is camera based application which allows used to take photos and records videos.
* User is able to apply various filters on the photos and can share them on their profile.

1. User profile and sharing on social networking –

* A user can create a profile through Sign Up.
* An existing user can login to the user profile.
* Share location.
* Share pictures.
* View total number of friends.
* View recent activity.

1. Following/News –

* User Profile and sharing on social networking.
* Following/News.
* It gives the recent notifications.
* Most trending news.
* Updates on people you are following.

1. Geo tagging –

* In Instagram, geotagging is a way to store data in terms of longitude and latitude of a particular location with the photo.
* The data is gathered through the GPS device in your phone or tablet and it will be accessed through Instagram if we provide permission to access current location.
* The photos/video can be geo tagged.
* Once the photos are geo tagged, they are available on the map by group the photos based on same location.

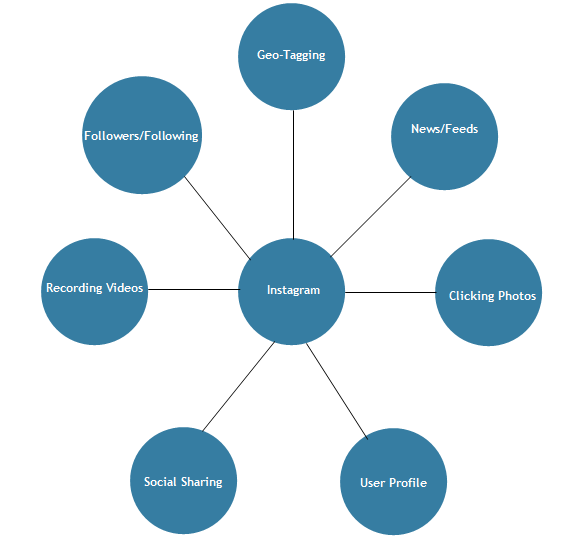


Figure 7 - Instagram Components

The gestures that we test for the features of the application are as follows –

* Tap: Briefly touch surface with fingertip.
* Double Tap: Rapidly touch surface twice with fingertip.
* Drag: Move fingertip over surface without losing contact.
* Flick: Quickly brush surface with fingertip.
* Pinch: Touch surface with two fingers and bring them closer together.
* Spread: Touch surface with two fingers and move them apart.
* Press and Drag: Press surface with one finger and move second finger over surface without losing contact.
* Rotate: Touch surface with two fingers and move them in a clockwise or counter-clockwise direction.
* Scroll: Move two fingers up or down across surface
* Scroll fast: Quickly brush surface with fingertip in the direction you want to scroll
* Press and Hold: Press surface with one finger and briefly touch surface with second finger.

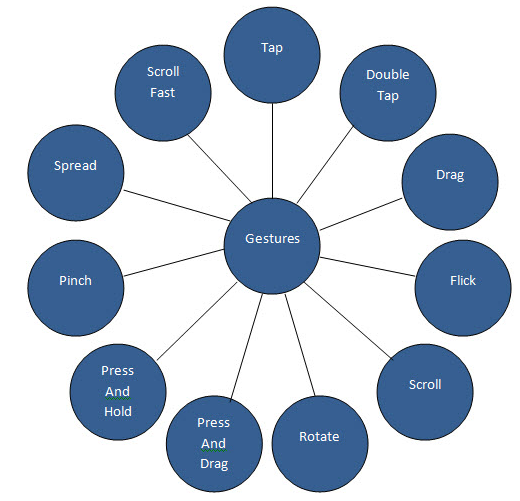


Figure 8 - Instagram Components

## Document purpose and structure

This document describes the testing plan for Instagram mobile application. The document provides us the high level details on the methods and tools that will be used for the testing of the application.

The entire document structure will be as follows:

* Coverage – The overall testing of the application.
* Testing Target – The target functionalities that will be tested to ensure the application with working as per the requirements and meets the desired quality.
* Testing requirement: The team should ensure that the requirements are met and documentation for the test plans is created.
* Task and Schedule: This area tells about the roles that are assigned to the tester. The timelines are also mentioned in order to complete the testing within in timelines.
* Tool Selection: The relevant tools must be selected for the testing and the cost of the testing process should be kept to the minimum.
* Test Standards: The specified standards that are required to be followed for the test cases and bug reports.

## Scope

The test strategy, schedule for the testing and the clear objective of the testing is specified in this document. For our testing, we will use various testing strategies like white box testing, black box testing which will ensure the proper working of the functionalities of the application. Validation and Verification methods will ensure that the requirements and specifications are met. The testing tools must be selected such that the cost of the testing is kept low. The deadlines for the testing are specified and the plan is specified that is based on the scheduled timelines.

## Team interaction

Team interaction is an essential component to have a successful product. The qualities a team should have are -

* Team work is essential part while performing the testing of the project. Effective communication is the key for a successful testing of the product.
* The team must ensure to visualize the test cases for the project, ensure the quality and documentation for the testing process.
* The standards should be followed and requirements specified should be met.

## **Coverage and Testing Target**

The testing targets of this test plan as well as the criteria and coverage are listed below-

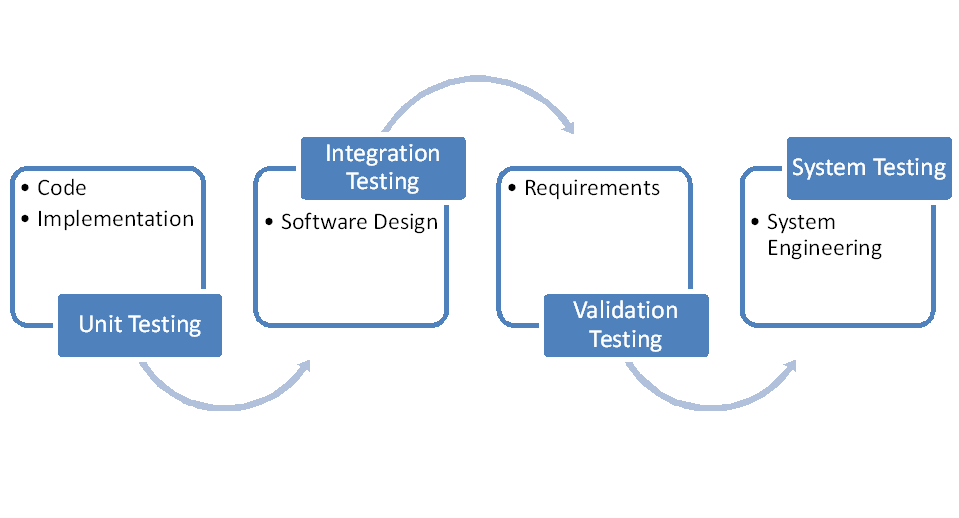


Figure 9 – Software Testing process

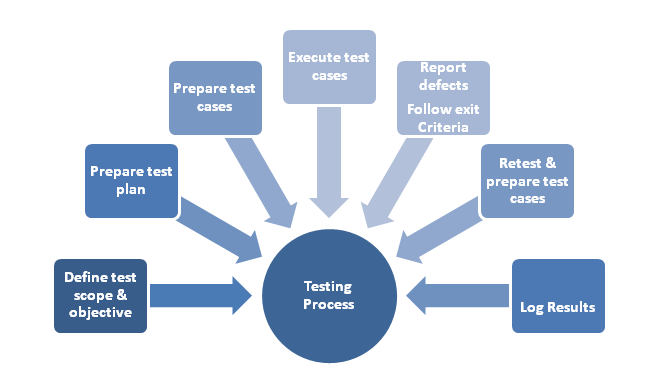


Figure 10 – Steps in Software testing

## Non Functional Testing

Non-functional testing helps us in testing of the non-functional requirements of the system. By the help of this testing we ensure the quality of the software product. The attributes are security, behavior, performance and scalability. The various types of non-functional testing are -

* Load Testing
* Performance Testing
* Installation Testing

## Functional Testing

## System level function testing

Testing target

In this application we will be performing testing on components for appropriate results. Our focus is to test input and output to the system from the point of view of system level functional testing. We will be using black box testing to test functional components. We will be using testing strategies like decision table testing and category partition testing etc., which we would choose on basis of functionalities of our mobile application Instagram. Our aim is to test each and every component of mobile application to find bugs and flaws by using appropriate testing types.

Knowing the application

It is important to know what we are testing in order to perform testing efficiently. Aim of our team is to understand the functionality of the application completely so that we should be clear about what is desirable and what’s not. There should be clear line of separation between functionality and bug.

Decision Table testing

Decision table testing is method to deal with different combination of inputs with their associated outputs. It is also called cause effect table. Decision table is derived on basis of different combinations. It is a black box testing technique in which each component is identified and listed in tabular form with their outputs.

Equivalence Partitioning testing

Equivalence Partitioning testing is method to deal with different combination of inputs into different classes and then test the system with input from each class. It is a black box testing technique.

## Automation System Testing

Targeted testing

Components of the system are tested by test scripts generated by process of automation system testing.

There are many aspects which defines scope of automation testing.

Performance of the system is measured and analysis is done on system after the collection of performance data and defining performance metrics and report of the system is generated in the end.

Test Coverage of automation system testing

While performing automation system testing, scripts are created to test the performance using the data required for the software to function.

These scripts generate the data based on defined functionalities of the mobile application, also it covers testing of the non-functional requirement specifications.

Automation system testing involves coverage of cases like high traffic and its effect on performance of the mobile application.

Test Criteria

Test criteria is specifying the Entry and Exit criteria for testing the software, further test results and standards are compared.

System Level Testing Criteria

Testing is performed at different levels, and there are different methods and criteria for each level. The system level testing involves in using the entry and exit criteria, black box testing methods.

* Entry Criteria: Determining the start of the test activity for specific test activity
* Exit Criteria: Exit criteria is mainly when the black box methods is used thoroughly

Decision-Table Testing Method

In Decision Table testing different combinations of input are provided to the test cases and which results in different actions being processed for every input, which is not possible to show with Equivalence Table Testing or Boundary Value Testing.

Equivalence Partitioning testing

Equivalence Partitioning testing is method to deal with different combination of inputs into different classes and then test the system with input from each class. It is a black box testing technique.

## **Testing Requirements**

This section covers the components that are targeted to be tested as well as what strategy will be useful for completing this requirement:

## Test Environment

Once when the testing stage starts, the most important part is the working of the testing environment. The testing environment for the testing phase should be decided prior to the testing phase starts. It should provide us with all the necessary tools and also with the software and hardware support which is required for the testing process. Only a well setup testing environment with proper hardware and software tools can provide the exact and precise testing results. Proper integration between all the software and hardware components should be done for the proper functioning and efficiency of the testing components.

* The system will require high speed internet connection.
* The system will be testing for the mobile application (mobile app).

## Testing Platform

The app (or application) is basically a mobile application and is needed to be tested on most of the devices and OS platforms. For instance application is available for iOS, Android. So the application needs to be tested on iOS 7 (mostly used on iPhone) onwards on devices like iPhone 5, iPhone 5S and even if possible on iPhone 6/plus. Android is widely distributed on various devices and comes in different OS versions like 2.3 onwards. But the target operating systems or Android OS 4 onwards on devices like Nexus, Samsung Galaxy and Moto series.

## Hardware Requirements

The minimum hardware requirements that is necessary for the testing of the application- Instagram:

|  |  |
| --- | --- |
| **Device** | **Specification** |
| Mobile Device (Android) | Android OS 4.0/4.1/4.4 and 1GB RAM |
| Mobile Device (iOS) | iOS 7.0, iOS 8.0 and IGB RAM |
| Laptop/Desktop | Intel Core Processor, 4 GB RAM |

## Software Requirements

The software requirements are:

|  |  |
| --- | --- |
| Operating System | Android OS 4.0/4.1/4.4, iOS 7.0, iOS 8.0, Windows 8 |
| Documenting Tool | Microsoft Office |
| Automation Tool | PerfectoMobile |
| Project Management and Bug Tracking Tool | Elementool and Bugzilla |
| Performance Tool | Neoload |

## **Testing Tasks and Schedule**

## Roles

This table shows the roles of the team members.

|  |  |
| --- | --- |
| **Role** | **Team Member** |
| Test Manager | Divanshu Arneja |
| Test Engineers | Shilpa, Saumil Jain |
| Performance test automation Engineers | Piyush Seth, Surbhi Rautji |

## Responsibilities

The below table lists the responsibilities matching each role for the testing process.

|  |  |
| --- | --- |
| **Role** | **Responsibilities** |
| Test Manager | 1. Planning the project and defining the scope/requirement of the project. 2. Setting up the team structure and managing the test engineers. 3. Product evaluation. 4. Tracking project status by planning and organizing the test schedules and timelines. 5. Scheduling meetings. 6. Define test methods, test strategy, standards, criteria and roles of the test engineers. 7. Review all the test reports prepared by the team and ensure the quality of the project. 8. Provide motivation and leadership to team members. 9. Administer the overall process for the project. |
| Test Engineers | 1. Design test cases 2. Document the testing performed 3. Adhere to the standards and techniques as planned 4. Test cases Implementation. 5. Test data and test procedures are developed after getting the knowledge of specifications & requirements of the project. 6. Document the test execution and problem reports. 7. Appropriate utilization of the testing tools. 8. Keep the log of the bugs and make sure that the bugs are fixed on priority basis. |
| Performance and Test Automation Engineer | 1. Develop and set the network for test automation process and execute tests for the security and performance of the software. 2. Check appropriate tools that are required for test automation process. 3. Design, develop and apply the different test strategies and also to build the automated testing frameworks. 4. Run the test script on the system and record the results in the form of bug reports which can be used for future reference. 5. Record test results and report and verify software bug fixes. |

## Project Deliverables

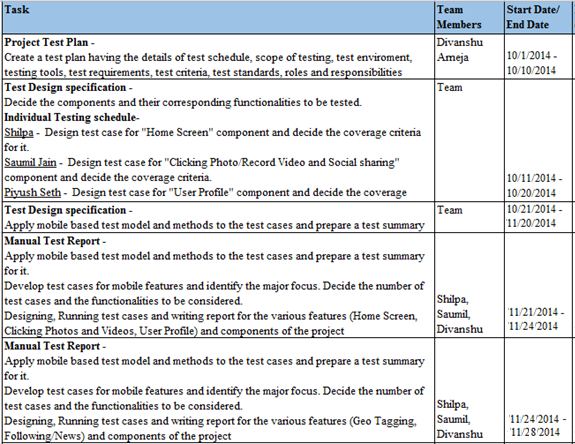
|  |  |
| --- | --- |
| **Deliverables** | **Date** |
| Test plan document | 10/10/2014 |
| Test Design specification | 11/20/2014 |
| Manual Test Report | 11/30/2014 |
| Automatic Performance Test Report | 12/08/2014 |
| Entire Report | 12/08/2014 |
| Project Presentation and Test Automation Demo | 12/04/2014 |

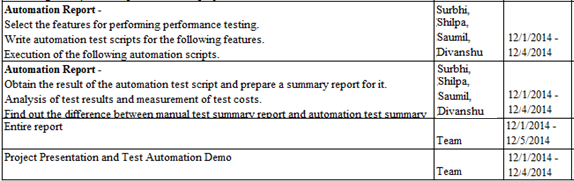
## Project Schedule

The below table defines the project schedule that will be followed -

|  |  |  |
| --- | --- | --- |
| **Tasks** | **Start date** | **End Date** |
| Test Plan - Initial document | 09/28/2014 | 10/10/2014 |
| Test Design specification | 10/11/2014 | 11/20/2014 |
| Manual Test Report | 11/21/2014 | 11/30/2014 |
| Automation GUI Test Report | 12/01/2014 | 12/04/2014 |
| Final reports | 12/01//2014 | 12/08/2014 |
| Project Presentation and Test Automation Demo | 12/01/2014 | 12/04/2014 |

## Individual Testing Schedule





## **Testing Tools**

Following section describes the tools planned to be used for our testing project.

## Perfectomobile

Perfecto Mobile is a cloud based automation testing tool used for testing mobile and web application. It provides testing of various different devices running on different operating system.

We can create scripts and run them on the real devices on the real network.

It is secure, easy to learn and provides simple UI for automation.

. 

Figure 10: Perfectomobile Tool for Mobile Automation Testing

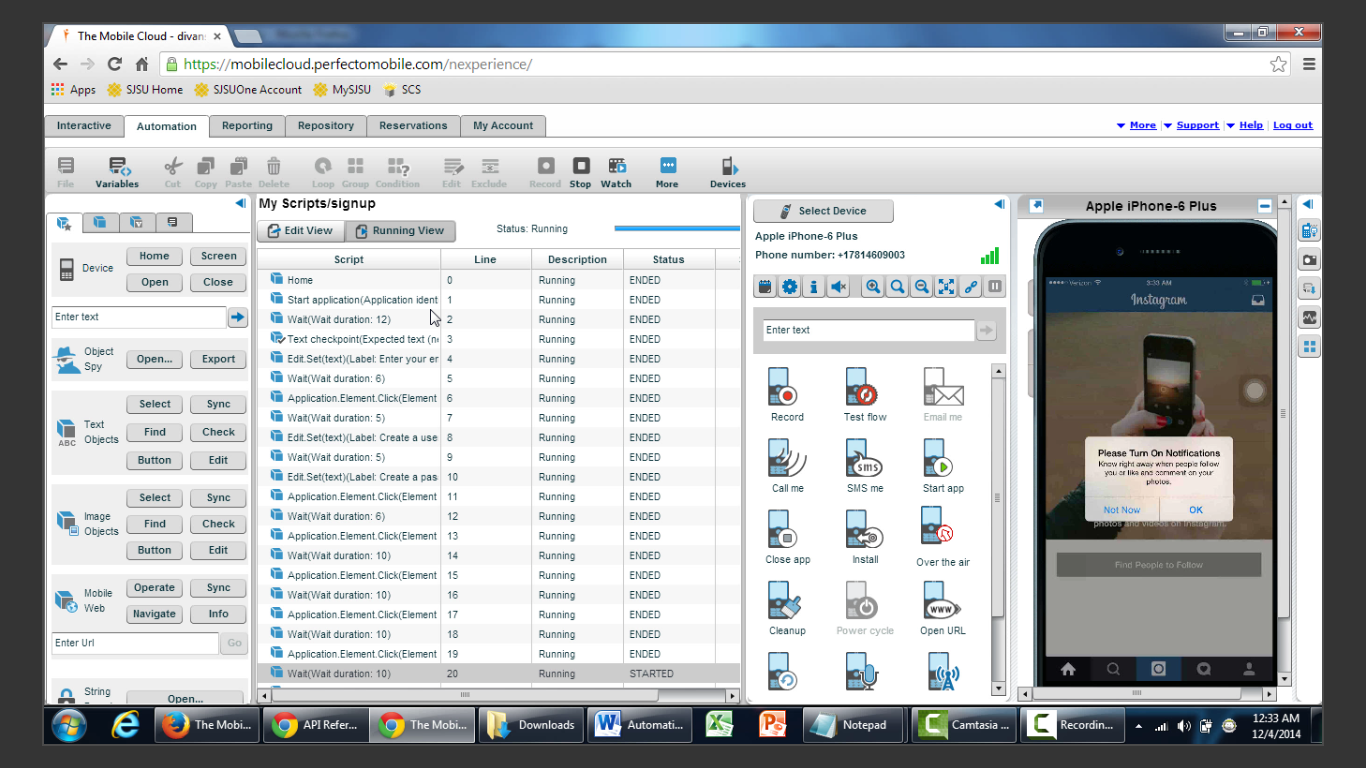


Figure 11: Perfectomobile for Mobile Automation Testing

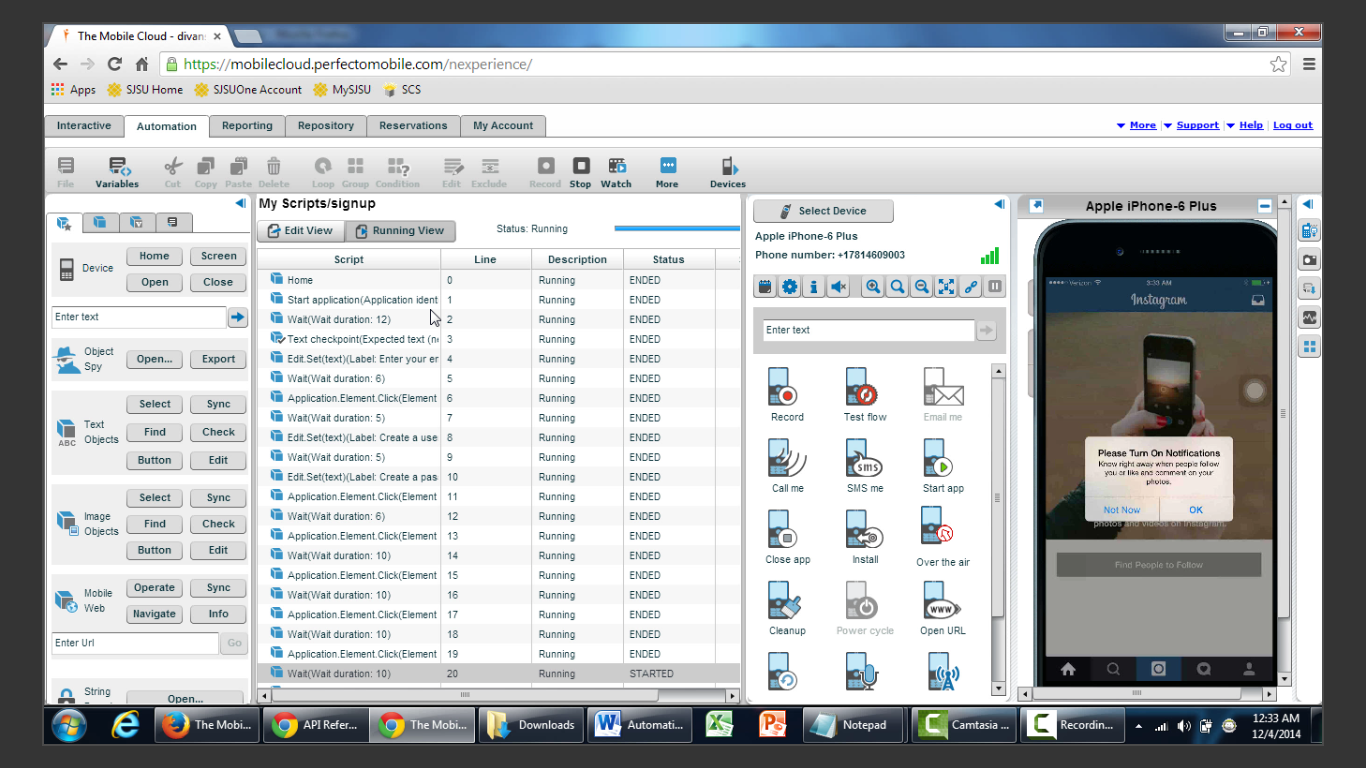


Figure 12: Perfectomobile for Mobile Automation Testing

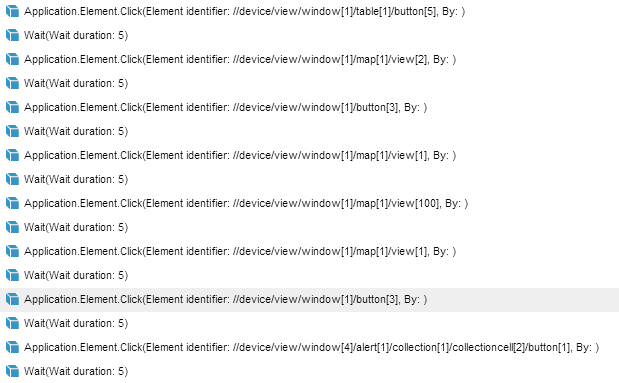


Figure 13: Perfectomobile sample script

## Elementool

The elementool is a bug tracking and management tool which is used in testing process all over the industry. The tool helps in reporting the bugs that are found while testing the application. The features of the testing tool are –

* Bug tracking
* Issue tracking.
* Creating the test cases.

Elementool is a web based tool, which is very convenient and helps the testers with documenting the bugs. It rules out the need to installing a bug tracking tool on the personal system.

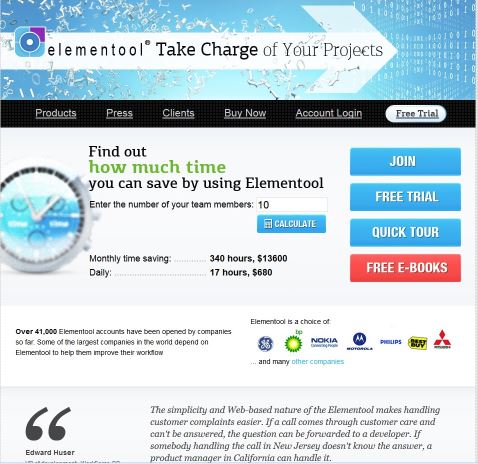


Figure 14: Elementool webpage

The features of Element Tool are:

• Adding Test Cases -   
The application permits testers to include or adjust custom test cases so as to test the application. It additionally gives the client an alternative to assign a specific test case to a dedicated tester.  
• Issue Tracking -   
Elementool has a feature of issue tracking. This feature provides the ability to the group of test engineers to track the issues that were experienced while applying the test cases during the running application.   
• Issue Reporting -   
Issue Reporting is used by the tester to report the bug to the developer so that it can be fixed.

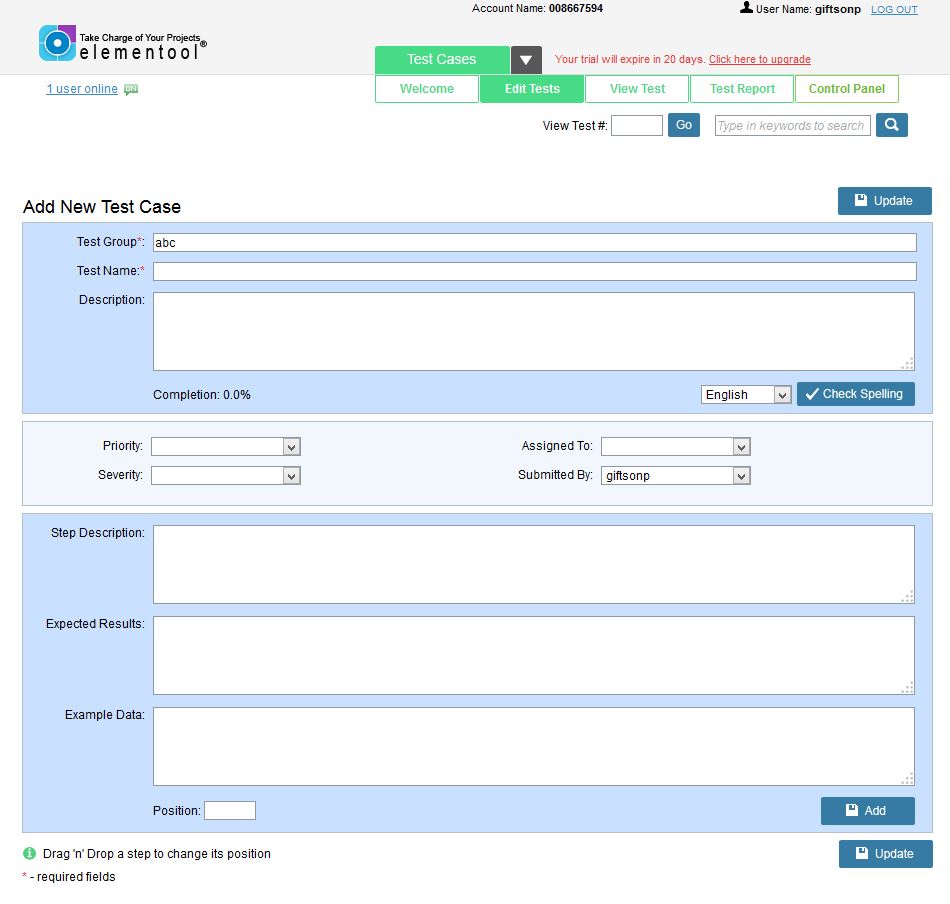


Figure 15: Elementool adding a test case

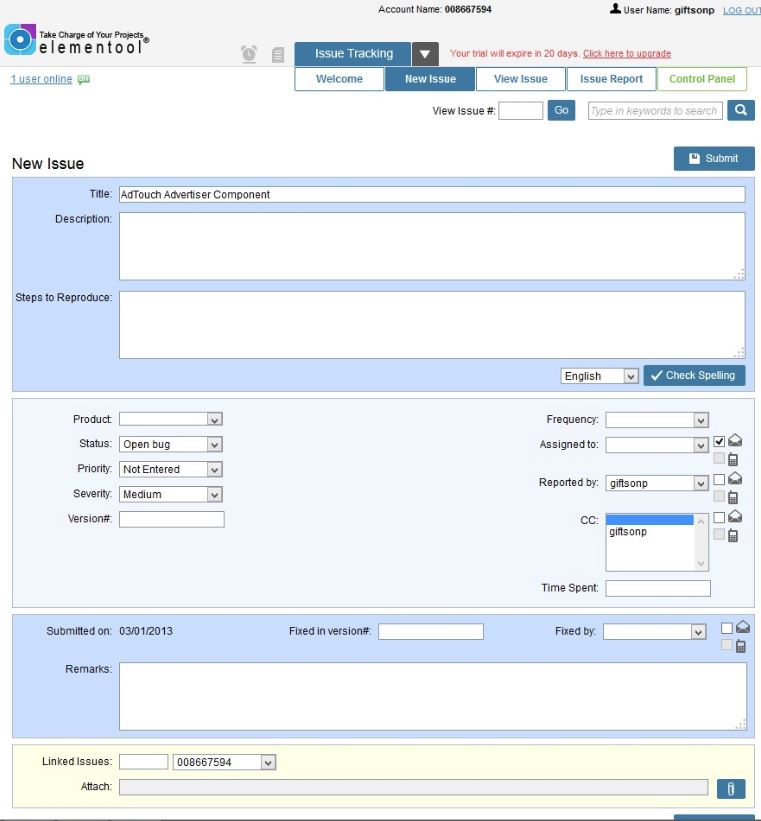


Figure 16: Elementool issue tracking

## NeoLoad

NeoLoad is a tool for performance testing of mobile applications. The tool is helpful to developers and testers to increase the performance of the application. By using NeoLoad, we can simulate the traffic upto millions of users such that we are able to test the application performance under heavy load.

We can also write the scripts using the GUI and can use various advance feature of the NeoLoad which will help us in testing the application.

Steps -

* Create a login for the application.
* Select the search criteria and data to be recorded.
* Modify and submit

Creating Test Cases in NeoLoad

* Click start button to record steps.
* A virtual user will start recording the feature.
* Browser setting can also be provided and assign HTTP traffic or Adobe RMTP traffic.

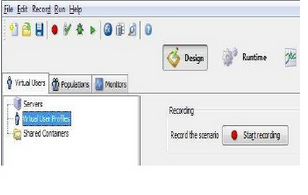


Figure 17: Test Case of NeoLoad

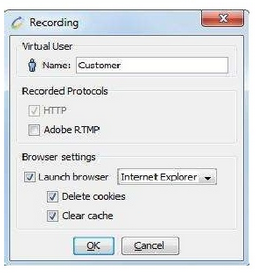


Figure 18: Recording template of NeoLoad

* Specify the functionality that needs to be testing.
* This will start the recorder and it can be stopped by using the stop button.
* Create the virtual user and mention the validity of the user.
* Run the test by clicking run from the toolbar.

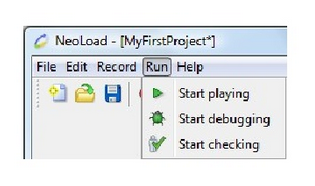


Figure19: NeoLoad showing how to run

## **Cost Analysis**

The cost of this project is based on various factors like –

* People
* Tools/Resources

Estimated Costs will include:

|  |  |
| --- | --- |
| 5 x $25000 (for 3 months) = $125000 | Resources ($100000/year) |
| $0 | Software –  Licenses - Testing Tool |
| Computers:  (5 x $500) = $2500  Mobile Devices:  iPhone (3 at $800) = $2400  Android (2 at $500) = $1000 | Hardware –  Devices  Accessories (Cables etc) |
| $130900 | Total |

The above cost is based on the market salaries of the test engineers and the cost of the equipment.

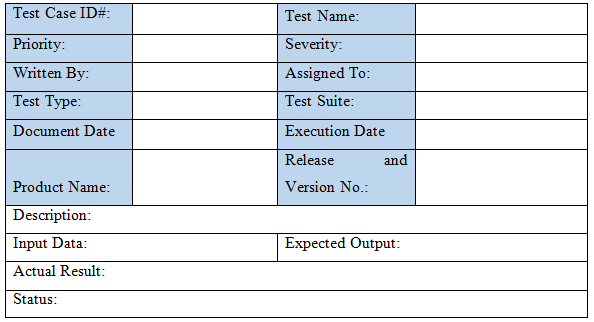
However this is a school project and the resources does not have any salaries and the devices are personally owned. So the overall cost of the project is $0.

## **Testing Standards**

## Template for test case

The Test cases that will be performed during the testing process should be well structured and documented with details to attain maximum efficiency at bug tracking and bug fixes. The following is the template for documenting test cases.

Description of the fields in above test case template:

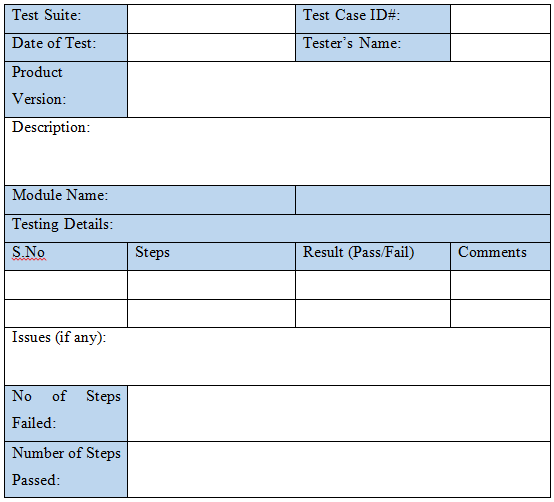


* Test Case ID - Number assigned to the test case.
* Test Name – Name of the Test Case.
* Priority – The priority of the Test case.
* Severity – The importance of the test case.
* Written By – Author of the test case.
* Assigned To – The person who will be executing the test case.
* Document Date - The date when this test case was designed/written.
* Execution Date - The date when this test case is being executed.
* Product Name - The product being tested.
* Release and Version No - The date product will be released.
* Description – The brief description of the test case which is used for testing.
* Input Data –The sample values that are given as input for testing.
* Expected Result – The desired value for a test case.
* Actual Result - The result of a test case.
* Status - The Pass/Fail result of the test case.

## Template for test result

Test outcomes should to be well organized to distinguish and get the issues that are existing in the testing application. A general sorted out test result template should contain the test result tagging whether the application has passed or failed the test cases, the issues regarding the failure of test case and also provide a list of all the test cases passed/failed.

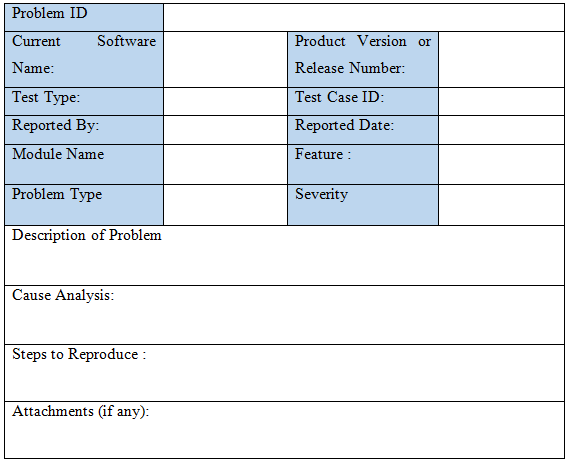
The following is the template for documenting the test results.



## Template for Bug/Issue Reporting

Bugs are issues that are accomplished amid the testing process that cause obstruction to the smooth running of the software. They must be recorded in a composed way with the objective that it gives the developer a legitimate understanding of the issue so it can be settled without any inconvenience besides to be reused later on if in case the same issues rise later on. A general composed issue reporting format should contain an issue or issue\_id, test case\_id and all other details.

The table below displays the template for documenting bugs.



## **References**

1. Appium - <http://appium.io/>
2. Elementool - <http://www.elementool.com/>
3. NeoLoad - <http://www.neotys.com/product/overview-neoload.html>
4. Instagram - <http://instagram.com/>
5. Mobile Testing-as-a-Service by Jerry Gao, Ph.D. Professor, Computer Engineering Department, San Jose State University
6. Mobile Application Testing: A Tutorial by Jerry Gao- San Jose State University, Xiaoying Bai- Tsinghua University, Wei-Tek Tsai- Arizona State University Tadahiro Uehara- Fujitsu Laboratories
7. CMPE 287 slides and class notes