CF\_PL\_DotNet\_SocialNetworking\_Test Plan

Version No.:1.0

Date: 05-Apr-2011

Project Name: Practice Labs Social Networking Application

Project Code:

Copyright Notice

This document contains proprietary information of HCL Technologies Ltd. No part of this document may be reproduced, stored, copied, or transmitted in any form or by means of electronic, mechanical, photocopying or otherwise, without the express consent of HCL Technologies. This document is intended for internal circulation only and not meant for external distribution.

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Version No | Date | Prepared by / Modified by | Significant Changes |
| 1.0 | 05-Apr-2011 | Santhosh Kumar G | New |
| 1.0 | 07-Apr-2011 | Santhosh Kumar G | Edited Test Items Requirements |

Table of Contents

[1 Introduction 4](#_Toc271212072)

[1.1 Purpose 4](#_Toc271212073)

[1.2 Scope 4](#_Toc271212074)

[1.3 Intended Audience 4](#_Toc271212075)

[1.4 Definitions, Acronyms and Abbreviations 4](#_Toc271212076)

[1.5 References 4](#_Toc271212077)

[2 Testing Types 5](#_Toc271212078)

[3 Test Items 8](#_Toc271212079)

[4 Item Requirement / Design Specification 9](#_Toc271212080)

[5 Test Design & Documentation Strategy 10](#_Toc271212081)

[6 Test Coverage Criteria 11](#_Toc271212082)

[7 Entry & Exit Criteria 12](#_Toc271212083)

[8 Test Execution Strategy 17](#_Toc271212084)

[9 Testing Tools (If Applicable) 17](#_Toc271212085)

[10 Test Environment 18](#_Toc271212086)

[11 Item Pass / Fail Criteria 18](#_Toc271212087)

[12 Defects Analysis and closure 19](#_Toc271212088)

[13 Suspension and Resumption Requirements 19](#_Toc271212089)

[14 Test Deliverables 20](#_Toc271212090)

[15 Testing Tasks](#_Toc271212091) 21

[16 Responsibilities 21](#_Toc271212092)

[17 Staffing & Training Needs 22](#_Toc271212093)

[18 Schedule 22](#_Toc271212094)

[19 Traceability to Requirements 22](#_Toc271212095)

# Introduction

## Purpose

The purpose of this document is to provide a Test Plan for the Practice Labs Social Networking Application. It describes the testing strategy and approach to testing QA will use to validate the quality of this product prior to release. It also contains various resources required for the successful completion of this project

## Scope

This document covers Unit testing, N-Unit testing, integration testing and system testing

## Intended Audience

This document is intended to developers, testers and everyone involved in the project.

## Definitions, Acronyms and Abbreviations

NA

## References

# Testing Types

### Fault Injection Testing

|  |  |
| --- | --- |
| Test Objective: | Verify system can handle corrupted source code or index files. |
| Technique: | Corrupt files, delete source code files… |
| Completion Criteria: | All planned tests have been executed. |
| Special Considerations: | Highly sophisticated corruption may not be possible. |

### 2.1.2 Function Testing

|  |  |
| --- | --- |
| Test Objective: | Verify use cases main scenarios and extensions can be achieved. |
| Technique: | System tests should test the boundary conditions as well as any special values.  All query keywords should be tested. All language keywords should be tested.  Exploratory testing should involve:  Finding the max number of query terms.  Finding the max query length.  Using terms incorrect for the language  All use cases and use case extensions should have test cases. |
| Completion Criteria: | All planned tests have been executed. |
| Special Considerations: | None |

### User Interface Testing

|  |  |
| --- | --- |
| Test Objective: | Verify the following:  Page navigation  text, images, buttons, menus, dialog boxes, icons, toolbars  Tab order |
| Technique: | None |
| Completion Criteria: | None |
| Special Considerations: | None |

### Performance Profiling

|  |  |
| --- | --- |
| Test Objective: | Verify the system under a simple usage scenario |
| Technique: | Use a simple search use case scenario |
| Completion Criteria: | Can search with multiple users |
| Special Considerations: | None |

### Load Testing

|  |  |
| --- | --- |
| Test Objective: | Verify performance behavior time for designated transactions or business cases under varying workload conditions. |
| Technique: | Verify these in combination:  Multiple users doing search  Background indexing  Small, medium and large code base  Small, medium and large query size (in number of terms) |
| Completion Criteria: | Verify against criteria specified in HLD document. |
| Special Considerations: | None |

### 

### Security and Access Control Testing

|  |  |
| --- | --- |
| Test Objective: | Application-level Security: Verify that an actor can access only those functions or data for which their user type is provided permissions. |
| Technique: | Attempt log in as various roles.  Verify permission limitations. |
| Completion Criteria: | Verify against criteria specified in Architecture document. |
| Special Considerations: | None |

# Test Items

| Test Item | Version | Description |
| --- | --- | --- |
| Login Form | 1.0 | User Authentication |
| Registration Form | 1.0 | User Registers with the network |
| Search | 1.0 | User Searches for a friend by entering a name |
| Add Friend | 1.0 | Accepts Friend Request and Add user to friends list |
| Send Friend Request | 1.0 | User send friend request asking the other user to add him to his friends list. |
| Post Scrap | 1.0 | Post message on to the profile |
| Delete Scrap | 1.0 | Delete posted scrap |
| Delete Friend | 1.0 | Removes user from Friend Request. |
| Delete Friend Request | 1.0 | Deletes friend request from profile. |
| Admin windows Form | 1.0 | Administration can be done here. |

## 

# Item Requirement / Design Specification

| Test Item | Item Requirements/Design Specification |
| --- | --- |
| Login Form | Software Requirements Specification (SRS) Version 1.0, Section 4.5.1 |
| Registration Form | SRS Version 1.0, Section 4.5.2 |
| Search | SRS Version 1.0, Section 4.5.4 |
| Add Friend | SRS Version 1.0, Section 4.5.6 |
| Send Friend Request | SRS Version 1.0, Section 4.5.5 |
| Post Scrap | SRS Version 1.0, Section 4.5.2 |
| Delete Scrap | SRS Version 1.0, Section 4.5.2 |
| Delete Friend | SRS Version 1.0, Section 4.5.6 |
| Delete Friend Request | SRS Version 1.0, Section 4.5.6 |
| Admin windows Form | SRS Version 1.0 |
|  |  |

Non-testable items:

| Test Item | Item Requirements / Design Specification | Reason |
| --- | --- | --- |
| NA |  |  |
|  |  |  |

# Test Design & Documentation Strategy

* For User Acceptance testing, the Developer team has completed unit, system and integration testing and met all the Requirement’s (including quality requirements) based on Requirement Traceability Matrix.
* User Acceptance testing will be conducted by End-users
* Test results will be reported on daily basis using BugZilla. Failed scripts and defect list from BugZilla with evidence will be sent to Developer directly.
* Use cases have been developed by Developers for User Acceptance testing. Use cases are approved by test lead.
* Test scripts are developed and approved.
* Test Team will support and provide appropriate guidance to Developers to conduct testing
* Major dependencies should be reported immediately after the testing kickoff meeting.
* Developers should clearly understand on test procedures and recording a defect or enhancement. Testing Process Management Team will schedule a teleconference with Developers and Developers to train and address any testing related issues.
* Developer will receive consolidated list of request for test environment set up, user accounts set up, data set (actual and mock data), defect list, etc. through BugZilla after the initial Adopter testing kick off meeting.
* Developer will support ongoing testing activities based on priorities
* Test scripts must be approved by Test Lead prior test execution
* Test scripts, test environment and dependencies should be addressed during testing kickoff meeting in presence of a SME and request list should be submitted within 3 days of the kickoff meeting
* The Developer cannot execute the User Acceptance and End to End test scripts. After debugging, the developer can conduct their internal test, but no results from that test can be recorded / reported.
* Developers are responsible to identify dependencies between test scripts and submit clear request to set up test environment

# Test Coverage Criteria

Test coverage criteria for various type of testing (Unit Testing, Integration Testing, Validation Testing, System Testing) Test cases are identified as adequate when they address all/most of the following points in general.

However any specialized test that is required to be performed before system delivery has to be taken into account and should cover the following points.

* Code Coverage
* Statement Coverage
* Decision Coverage
* Condition Coverage
* Multiple Condition Coverage
* Function Coverage
* Data Flow Coverage

# Entry & Exit Criteria

This section describes the general criteria by which testing commences, temporarily stopped, resumed and completed within each testing phase. Different features/components may have slight variation of their criteria, in which case, those should be mentioned in the feature test plan. The testing phase also maps to the impact level definition when a defect is entered in the bug-tracking phase.

## Unit Testing

Unit Testing is done at the source or code level for language-specific programming errors such as bad syntax, logic errors, or to test particular functions or code modules. The unit test cases shall be designed to test the validity of the programs correctness.

### Black Box Phase

Black box testing typically involves running through every possible input to verify that it results in the right outputs using the software as an end-user would. We will use Equivalence Partitioning and Boundary Value Analysis complexity metrics in order to quantifiably determine how many test cases needed to achieve maximum code coverage.

#### Black Box Entry Criteria

The Black Box Entry Criteria will rely on the component specification, and user interface requirements. Things that must be done on entry to the Black Box stage:

* Enter invalid inputs, such as username and password trying login.
* Enter invalid data in to search bar.
* Try sending friend requests to users who are already friends.
* Enter invalid data while registration and try to complete it.

#### Black Box Exit Criteria

To exit the Black Box phase all code bugs that are exposed are corrected and 100% success rate must be achieved.

### White Box Phase

The White Box criteria apply for purposes of focusing on internal program structure, and discover all internal program errors. Defects will be categorized and the quality of the product will be assessed.

#### White Box Entry Criteria

The White Box Entry Criteria will rely on the QA engineers verifying that the major features work alone but not necessarily in combination; exception handling will not be implemented. The design and human interface are stable. The type of White Box testing Methods will be determined upon entry. We will use Function Validation testing. Black Box Testing should be in its late stages. After the White Box criteria have been met, the product enters the White Box stage. During White Box stage Development Engineering’s emphasis is on refining the product and fixing defects. Information Design’s emphasis is on developing product user documentation.

#### White Box Exit Criteria

The Binary Tree in the White Box stage should have a generally stable feel to it. White Box testing continues until the Black Box or next milestone criteria are met. To exit the White Box phase 100% success rate must be achieved. The following describes the state of the product upon exit from the White Box Stage:

* All Product functions Login, Register, Search, post scraps, delete scraps, Add Friends, and Delete Friends are implemented, operational and tested.
* The test cases will be generated from the Control Flow diagrams of all functions.
* The User interface has been reviewed and found to satisfactory by development Engineers, and QA Engineers, and is stable, that is, no further changes to dialog boxes or other interface elements are planned. Minor changes (word-smiting, etc.) are acceptable, but must be arranged with the Development and Test Engineers.
* All code bugs that are exposed are corrected.

## Integration Test

There are two modules that will be integrated for Integration Testing. The two modules are The Graphic User Interface module and the Admin windows Forms Module. The following describes the entry and exit criteria for Integration testing.

### Integration Test Entry Criteria

The Integration Test Entry Criteria will rely on both modules to be operational. The Web site (human interface) and Admin Interface must be stable. Things that must be done on entry to the Integration Test stage:

* All Product functions Login, Register, Search, post scraps, delete scraps, Add Friends, and Delete Friends are created or coded
* The Graphical User Interface must be developed. It implemented to facilitate test case input and output values.
* Interfaces and interactions between the Website and the Admin Interface must be operational.
* A bottom-up Integration Test Strategy will be conducted.
* Black Box Testing should either be in its late stages or completed.
* White Box Testing should have begun.

### Integration Test Exit Criteria

The Integration Test Exit Criteria will rely on both modules to be operational. To exit the Integration Testing phase 100% success rate must be achieved. Things that must be done on exit from the Integration Test stage:

* All code bugs that are exposed are corrected.
* The Website module and Admin interface Module will interact together with complete accuracy, according to the System Specification Design. All discrepancies are corrected.
* Both Modules are ready for System Testing. Stubs are replaced with fully functional code.
* Black Box Testing is completed.
* White Box Testing should either be in its late stages or completed.

## System Test

The System Test criteria apply for purposes of categorizing defects and the assessing the quality level of the product. All elements of the Social Networking website and Admin Windows’s forms are meshed together and tested as a whole. System test focuses on functions and performance, reliability, instillation, behavior during special conditions, and stress testing.

### System Test Entry Criteria

The Entrance Criteria specified by the Development Engineers, should be fulfilled before System Test can commence. In the event, that any criterion has not been achieved, the System Test may commence if both Development and Test Engineers are in full agreement that the risk is manageable.

* The Web User Interface and the back-end Module must be fully functional.
* All developed code must be unit tested. Unit and Link Testing must be completed and signed off by the development team.
* All test hardware and environments must be in place, and free for System test use.
* All Black Box testing must be complete and exposed bugs must be corrected.
* All White Box testing must be complete and exposed bugs must be corrected.
* Integration Testing must be complete and exposed bugs must be corrected
* Function Validation Testing is the accepted method of testing for all functions: Login, Register, Search, post scraps, delete scraps, Add Friends, and Delete Friends. The Web User Interface will be the method of interacting with the system, so the GUI will be tested thoroughly.
* Development and Test Engineers agree that Function Validation Testing will cover function performance, reliability, stress and load testing.

### System Testing Exit Criteria

The Exit Criteria must satisfy all the criteria listed below. This verifies that all elements of the project mesh properly. This is to make sure that all the system functions and performs according to the System Specification Document.

* All Function Validation Testing is 100 percent successful. All Product functions Login, Register, Search, post scraps, delete scraps, Add Friends, and Delete Friends interact with complete accuracy.
* No degradation of System performance across different platforms of Windows operating system will be affected. (Windows 95 or above is acceptable)
* The Graphical User Interface performs to System Specification Requirements.
* All the properties are expressed correctly through the Graphical User Interface.
* All input fields on the Web User Interface are working correctly.
* All high priority errors from System Testing must be fixed and tested.
* If any medium or low-priority errors are outstanding – the Development Engineers and Test manager must sign off the implementation risk as acceptable.

# Test Execution Strategy

1. Define testing strategies for each area and sub-area to include all the functional and quality (non-functional) requirements.
2. Divide Design Spec into testable areas and sub-areas (do not confuse with more detailed test spec). Be sure to also identify and include areas that are to be omitted (not tested) also.
3. Define bug-tracking procedures.
4. Identify testing risks.
5. Identify required resources and related information.
6. Provide testing Schedule.

# Testing Tools (If Applicable)

Not Applicable

# Test Environment

| Items | Configuration |
| --- | --- |
| Hardware | PC at least with Pentium IV processor, 160GB hardisk, 1GB Ram. |
| Software | Web browser, Visual Express C# developer 2008 , Visual Express Web Developer 2008, Microsoft SQL Express 2008 |
| BugZilla, Tortoise SVN |
| Test Data | Database “PracticeLabsDB” |

# Item Pass / Fail Criteria

Defects will be classified as follows according to severity of the impact on the system:

| Severity | Impact |
| --- | --- |
| Major | System inoperable, An implementation that does not meet the requirements (or any other input document), Defects that may cause the system to hang, crash, produce incorrect/ unexpected results or behavior, or corrupt user data with no known work around. |
| Minor | Major function disabled/ incorrect, Non-conformance to standards. Defects that cause incorrect results or behavior with known work around. Large and/or critical portion of the system is affected that would not cause operational failure. |
| Trivial | Defects that affect limited areas of functionality that either can be work around or ignored. |

Pass / Fail Criteria

| Item | To pass the total number of defects on running the complete suite of test cases not to exceed | | |
| --- | --- | --- | --- |
|  | Major | Minor | Trivial |
| COMSOFREP | 0 | 2 | 5 |
| PROGFNLN | 0 | 0 | 2 |
| FNDBLINK | 0 | 1 | 2 |

# Defects Analysis and closure

The following are done while identifying defects

* Logging of defects
* Analysis of defects
* Fixing of defects
* Re-testing of fixes such as regression or full testing
* Defect tracking till closure

The tool used for bug tracking and defect analysis is BugZilla.

# Suspension and Resumption Requirements

Testing will be suspended if there is any Major Bugs or any crash.

Resumption criteria: When each of the suspended criteria is addressed and closed

# Test Deliverables

Below is the list of artifacts that are process driven and should be produced during the testing lifecycle. Certain deliverables should be delivered as part of test validation, you may add to the below list of deliverables that support the overall objectives and to maintain the quality.

This matrix should be updated routinely throughout the project development cycle in you project specific Test Plan.

|  |
| --- |
| **Deliverable** |
| **Documents** |
| Test Approach |
| 🡪 Test Plan |
| 🡪 Test Schedule |
| 🡪 Test Specifications |
| **Test Case / Bug Write-Ups** |
| Test Cases / Results |
| Test Coverage Reports |
| BugZilla Bug tracker for bug reporting |
| **Reports** |
| Test results report |
| Test Final Report - Sign-Off |

# Testing Tasks

The following are the testing tasks

* Write a Test Plan
* Build Test Cases
* Develop Escalation Procedure
* Conduct Tests and Evaluate Tests
* Document Test Results

# Responsibilities

### Developer

Responsible to:

(a) Develop the system/application

(b) Develop Use cases and requirements in collaboration with the Developers

(c) Conduct Unit, system, regression and integration testing

(d) Contribute to Use case, requirement development through review

(e) Contribute to develop and execution of the development test scripts through review

(f) Conduct Full User Acceptance, N-unit, and end-to-end testing; this includes identifying testing scenarios, building the test scripts, executing scripts and reporting test results

(g) Support user acceptance testing

### Testing Process Management Team

Responsible to:

(a) Monitor and manage testing integrity and Support testing activities

(b) Coordinate members in the team.

# Staffing & Training Needs

Not Applicable.

# Schedule

The Test Schedule is the responsibility of the Test Lead (or Department Scheduler, if one exists) and will be based on information from the Project Scheduler (done by Product Manager). The project specific Test Schedule may be done in MS Project.

# Traceability to Requirements

A Requirements Traceability Matrix (RTM) which is used to link the test scenarios to the requirements and use cases is a required part of the Test Plan documentation for all projects. Requirements traceability is defined as the ability to describe and follow the life of a requirement, in both a forward and backward direction (i.e. from its origins, through its development and specification, to its subsequent deployment and use, and through periods of ongoing refinement and iteration in any of these phases)