High Level Test Cases

Graduate School Recommendation System (GradU)

Alok Asok
Kapil Sharma
Sagar Dhamija
Varun Khandelwal
Ramanpreet Singh Khinda

Contents

1. Introduction

1.1 Overview

The document outlines the high-level test strategy for the project Graduate School Recommendation System (GradU). This document will be completed and utilized by the QA team to manage testing for this project. The test effort will be prioritized and executed based on the priorities as defined in the project's Software Requirements specification document. This is a living document that may be refined as the project progresses.

1.2 References

- Software Requirements Specification Document for GradU
- Test strategy
- Test Case Template Development – How to?
- Test Suite Definition
- Test Case : Wikipedia
- Requirement : Wikipedia
- Functional Testing
- System Testing
- Load Testing
- Performance Testing

1.3 Glossary

Term	Definition
Graduate School Recommendation System(Gradu)	It's the name of the project which is intended to produce a university recommendation system for students intending to apply for admission to the Master of Science course in graduate schools in the United States of America.
Test Suite	In software development, a test suite, less commonly known as a 'validation suite', is a collection of test cases that are intended to be used to test a software program to show that it has some specified set of behaviours.
Test Suite ID	It is a unique id to represent each test suite
Test Case	A test case, in software engineering, is a set of conditions under which a tester will determine whether an application, software system or one of its features is working as it was originally established for it to do.

Test Case ID	It's a unique id assigned to each test case
Requirement ID	It's a unique id assigned to each requirement
Requirement Description	In product development and process optimization, a requirement is a singular documented physical and functional need that a particular design, product or process must be able to perform.
Test Objective	It refers to the chief purpose of the test case i.e. what is tested as a part of that test case
Test Predecessor	It refers to the test case which must be performed before performing the current test case
Pre-condition	It refers to the set of conditions that must be true for performing the current test case
Test Steps	It refers to the steps that must be followed for testing the current test case
Expected Behaviour	It refers to the expected behaviour of the application after performing the mentioned test steps
Actual Behaviour	It refers to the actual behaviour of the application after the test steps were performed
Post- condition	It refers to the conditions that hold true after the test case has been performed
Status(Pass/Fail)	If the expected behaviour is same as the actual behaviour then it is said that the application has passed the test case else it means the application failed the particular test case
Functional Testing	Functional testing is a quality assurance (QA) process and a type of black box testing that bases its test cases on the specifications of the software component under test.
System Testing	System testing of software or hardware is testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements. System testing falls within the scope of black box testing, and as such, should require no knowledge of the inner design of the code or logic.
Load testing	Load testing is the process of putting demand on a software system or computing device and measuring its response. Load testing is performed to determine a system's behaviour under both normal and anticipated peak load conditions.

Performance Testing	Performance testing is the process of determining the speed or effectiveness of a computer, network, software program or device.

2. Test Strategy

Based on the requirements provided in the software requirements specification document, the testing is split into system, functional, load, performance and security testing. We have tried to take into the consideration all the possible scenarios and have provided the test cases for the same. Since, the design document is still not out we have taken into consideration the basic UI and have provided test cases for the same.

3. Test Suites

We have maintained four test suites uniquely identified by their respective test suite ids. The following are the test suits that we have maintained:

- TS_001: includes test cases for the actions performed by the user
- TS_002: includes test cases for the actions performed by administrator
- TS_003: includes test cases for security testing
- TS_004: includes test cases for load/performance testing

4. Test Cases

Based on the requirements provided to us in the software requirements specification document, we have tried to capture as many scenarios as possible and have documented the test cases for the same in the above mentioned test suites. The test cases were documented with utmost caution keeping in mind all the possible scenarios that may come to play when the application goes live. Moreover, while writing the test cases we also made sure to include the pre and post conditions along with detailed description of how to perform that test case. We have also provided the expected behaviour of the test case. Since, the design and implementation is still in progress we couldn't capture the actual behaviour of the application for a particular test case.

A detailed description of the test suits and the test cases is provided in the excel sheet.

GRADU_TestCases_v1.0.xlsx

5. Test Plan

Most of the testing will be manual. Once the design document is finalized, many aspects such as the database design, data types, table structure, load, number of users would be available and hence, automated testing if required would be done accordingly.

6. Installation and Configuration

Installation / Configuration testing verifies that the system will install and function on all required operating platforms, under all specified configurations.

1) The software will be able to run on Android 2.3 (Gingerbread) or higher. All devices using this operating system should be compatible with this software.
2) The application will get the required data, such as information related to universities, from the application database stored on a server.

Minimum hardware requirements

- An android 2.3 or higher compatible smartphone with 1 Giga hertz processor
- 512 Mb ram
- 10 Mb of available disk space
- Access to internet connection

Minimum Software requirements

- Android 2.3 or higher

7. Uncovered Use Cases

Following use cases aren't mentioned in the required document and have been agreed by Development team to be included:

- Administrator can delete a university
- User can complete his profile at any point later
- User can reset/change his password