(Google Search Engine)

Test Plan

**Table of Contents**

1 Introduction 4

1.1 Purpose 4

1.2 Overview of System 4

Overview 4

Project Objectives 4

References and Requirements 4

Industry Specific 4

Project specific 4

Requirement/Scope: 4

1.3 References & Source Documentation 5

2 Verification & Validation Strategy 6

2.1 Unit Testing 6

2.2 Integration Testing 6

2.3 System Testing 7

2.4 Regression Testing 7

2.5 Production Environment Testing (If required) 8

2.6 User Acceptance Testing (If required) 8

3 Team Development 9

3.1 Testing Team Formation 9

3.2 Roles & Responsibilities 9

4 Test Environment 10

4.1 Hardware components required 10

Environment 10

4.1.1 Workstations 10

4.2 Software Requirements 10

4.2.1 Software/Server Requirements 10

4.2.2 Production Environment 10

4.3 Tools to be used 10

4.4 Test Databases 10

5 Effort & Schedules 11

6 Standards and Procedures 12

6.1 Scenario/Check List 12

6.2 Test Cases 12

6.3 Bug Report 12

6.4 Automation Script 12

7 Test Items 13

7.1 Inclusions- Features to be tested 13

7.2 Exclusions- Features not to be tested 13

7.3 Prioritization 13

8 Defect Management 14

8.1 Defect Tracking Tool 14

8.2 Defect Resolution Procedure 15

8.3 Status of Bugs 15

9 Tools 16

10 Deliverables 17

11 Status Reporting 18

##### Revision Chart

A new record should be added every time a user updates this document.

|  |  |  |  |
| --- | --- | --- | --- |
| Ver. # | Date | Author | Change |
| v.10052016 | 18/10/2023 | Aparna Asthana | Initial Draft |
| v.07182017 |  |  |  |
| v.09052017 |  |  |  |
| v.12042017 |  |  |  |
| v.02072019 |  |  |  |

# Introduction

## Purpose

Develop Google Search Engine to satisfy requirements / wish list of needs.

## Overview

## Google is a fully-automated search engine that uses software known as "web crawlers" that explore the web on a regular basis to find sites to add to our index.

## References and Requirements

### Industry Specific

### Google began as an online search firm, but it now offers more than 50 Internet services and products, from e-mail and online document creation to software for mobile phones and tablet computers. In addition, its 2012 acquisition of Motorola Mobility put it in the position to sell hardware in the form of mobile phones.

### Project specific

NA

### Requirement/Scope:

NA

## References & Source Documentation

* JIRA Link
* Customer Documents
* Implementation Documents
* MOM
* Archive
* FYI\_Emails
* QA
* Connectivity

## Intended Audience

The audiences for this document include

* Project Manager :
* Dev. Team Leader :
* Dev. Team Members :
* Test Engineer :

# Verification & Validation Strategy

The following Validation Strategies would be implemented for the testing of Google Search Engine.

1. Unit Testing
2. Integration Testing
3. System Testing
4. Regression Testing
5. Production Environment Testing
6. User Acceptance Testing

## 2.1 Unit Testing

Goal of Unit testing is to uncover defects using formal techniques like Boundary Value Analysis (BVA), Equivalence Partitioning, and Error Guessing. Defects and deviations in Date formats, Special requirements in input conditions (for example Text box where only numeric or alphabets should be entered), selection based on Combo Box’s, List Box’s, Option buttons, Check Box’s would be identified during the Unit Testing phase.

Test Cases would be written based on the design document for all the requirements mentioned in the test conditions.

Test Conditions

* User Interface
* Boundary Value Analysis
* Equivalence Partitioning
* Error Guessing

Entry Criteria

* Base lined design document
* Base lined code
* Base lined test plan document
* Unit test case document

Work Items

* Test execution

Exit Criteria

* Test execution report
* Unit testing review checklist

## 2.2 Integration Testing

During Integration Testing, behavior of the system on the whole is observed. Individual modules are integrated and tested for the whole functionality.

Test Cases would be written based on the design document, for all the requirements mentioned in the test conditions.

Test Conditions

* Behavior of Individual Modules.
* Integration between different modules

Entry Criteria

* Base lined design document
* Base lined Functional specification document
* Base lined code
* Base lined test plan document
* Unit test execution report
* Integration test case document

Work Items

* Test execution

Exit Criteria

* Test execution report
* Integration review checklist

## 2.3 System Testing

The goal of system testing is to uncover defects while integrating hardware and software system to verify that the system meets its specified requirements.

Test Conditions

* Functionality of the application with respect to its behavior on the whole, with compliance to its hardware and software.

Entry Criteria

* Unit test review checklist
* Integration test review checklist
* Base lined test plan document
* Unit / Integration / System test case documents

Work Items

* Test execution

Exit Criteria

* Test execution report
* System review checklist

## 2.4 Regression Testing

The goal of Regression Testing is to observe the behavior of the system after certain rounds of test phases.

Test cases / scenarios for the regression round would be derived from the unit and integration test case documents.

Test Conditions

* Functionality of the application on the whole. (Refer Unit / Integration / System Checklists).

Entry Criteria

* Unit test review checklist
* Integration test review checklist
* Base lined test plan document
* Unit / Integration test case documents

Work Items

* Test execution

Exit Criteria

* Test execution report
* Regression review checklist

## 2.5 Production Environment Testing (If required)

The behavior of the application after the standard setup is made in the production environment and at the live environment is tested during this phase.

Test cases and scenarios would be derived from Unit / Integration / System / Regression / Performance test case documents.

Test Conditions

* The behavior of the completely built application at the production and the client site. (Refer the checklists for Unit / Integration / System and User Acceptance Testing).

Entry Criteria

* Base lined design document
* Base lined Test plan document
* Compiled EXE file of the application
* Unit / Integration / Regression / Performance Test execution reports

Work Items

* Test execution

Exit Criteria

* Test execution report
* Production Environment review checklist

## 2.6 User Acceptance Testing (If required)

User Acceptance Testing (UAT) is performed to test the application on the whole and its behavior in the live environment based on the clients requirements.

Test cases and scenarios would be derived from the Unit / Integration / Regression / Performance test case documents and also the Software Requirements Document.

Test Conditions

* The behavior of the application as per the client’s requirements.

Entry Criteria

* Base lined software requirement document
* Base lined design document
* Base lined test plan document
* Unit / Integration / Regression / Performance / Production environment test execution reports

Work Items

* Test execution

Exit Criteria

* Test execution report
* UAT review checklist

**Note:** We deployed the build on staging server fort the client to test as UAT release.

# Team Development

## Testing Team Formation

QA

QA Lead

PM.

## Roles & Responsibilities

**QA:**

(I). Assigned QA is responsible to prepare the test cases based on provided scenarios.

(ii) Prepare the positive and negative test cases and lock in JIRA.

(iii) Review and Update the Test Cases

(iv) Execute Test Cases.

(v) Defect Reporting and defect tracking through JIRA.

(vi) Attend Scrum Calls.

**QA Lead: -** Responsible to prepare the positive business flow scenarios as per the requirements decided in Grooming Sessions.

Task and Sub task assignment to QA team,Prepare Test Plan, Defect Report, Test Estimation and Closure Report.

**PM**: - Will review the scenarios. Confirm the requirements.

Development Team: - Will be responsible for fixing the reported defects.

# Test Environment

## Hardware components required

## Environment

### Workstations

A test environment that uses Firefox, Windows XP, and WAS is a configuration of these environment types: browser, operating system, and application server.

## Software Requirements

### Software/Server Requirements

## Staging Environment

## Production Environment

## Tools to be used

JIRA

## Test Databases

For test separate Test database should be provided by development team.

# Effort & Schedules

Assigned QA first run the positive business flow test cases. UI test cases will cover in positive testing, secondly will execute the network scenarios and lastly the negative scenarios will be executed.

Types of testing to be performed:-

* Functional Testing
* Business Flow test
* UI testing
* Regression testing
* Boundary testing

**Note**: The Schedule for the Testing Activity is synchronized with the Development Test Plan. Any changes made to the Development Plan would reflect in the schedules of the Testing activity.

# Standards and Procedures

## Scenario/Check List

## Test Cases

We can use the user interface document for creating the high level test cases initially.

## Bug Report

## Automation Script

On Client’s requirements.

# Test Items

## Inclusions- Features to be tested

Based on UI and functional requirement of the site.

## Exclusions- Features not to be tested

Which are not in scope like third party sites.

## Prioritization

Execute functional test cases first then UI and then the negative test cases.

# Defect Management

During Testing, Defects will be recorded using JIRA. The Defects will be reviewed and prioritized and assigned for fix, if required. Defects, which are agreed as valid, will be categorized as follows:

|  |  |
| --- | --- |
| Defect Severity | Description |
| **Critical** | Defects that would result in complete omission of a requirement, from the scope of the document / code or applicable standards and guidelines. crashes, loss of data, severe memory leak |
| **Major** | Defects with observable departures from the standard or guidelines resulting in incomplete and ambiguous documentation and program code. Major loss of function |
| **Minor** | Cosmetic problems like spelling, formatting etc. minor loss of function, or other problem where easy workaround is present |
| **Trivial** | Cosmetic problem like misspelled words or misaligned text. Suggestions: These are primarily suggestions for improving / enhancing the functionality. |
| **Blocker** | Blocks development and/or testing work |
|  |  |
| Defect Status | Description |
| **New** | Defects when reported and not reviewed by development team |
| Fixed | The fixed status indicates that a change was made to the code and will be available in the next build |
| **Invalid** | The problem described is not a bug. |
| **Wont Fix** | The problem described is a bug which will never be fixed. There may be extenuating circumstances where Defects will not be fixed because of technology, time constraints, and a risk of de stabilizing the code or other users. |
| **Closed** | The bug is considered dead, the resolution is correct |
| **Duplicate** | The problem is a duplicate of an existing bug. Marking a bug duplicate requires the bug# of the duplicating bug and will at least put that bug number in the description field. |
|  |  |
|  |  |
|  |  |

## 8.1 Defect Tracking Tool

We would be using JIRA. All the Development Team Members and Test Engineers would be given user privileges to log in any type of error identified by them.

## 8.2 Defect Resolution Procedure

Defects logged on to the defect tracking tool should be resolved at the earliest. The person, who identifies the bug, logs it on to the JIRA and intimates the person who is assigned the job to fix the bug via JIRA.

## 8.3 Status of Bugs

The status of bugs will have four options. They are briefed below:

|  |  |
| --- | --- |
| **Status** | **Description** |
| Open | The error reported is open. |
| Resolved | The error is resolved. |
| Close | The error is re-tested and closed. |
| Re-Open | The error is re-opened. |

When the test engineer opens a bug, the developer resolves the bug and changes the status of the error to “Resolved”. The test engineer re-tests the error and Closes. If the error is repeated while in any phase of testing, the error would be re-opened.

**8.4 Test Bed**

The testing of Google Search Engine application is carried out on a separate test bed, different from the development environment. The original database is replicated here for the purpose.

# Tools

The following tools would be used by the team members during the course of the project.

|  |  |
| --- | --- |
| **Purpose** | **Tool** |
| **Defect Tracking Tool** | JIRA |
|  |  |
|  |  |

# Deliverables

The following documents would be maintained by the Test team.

1. Test Plan.
2. Test Estimation.
3. Test Case Documents.
4. Test Execution Reports.
5. Closure Report

# Status Reporting

Test preparation and progress will be formally reported during a *Scrum Meeting*. The attendees at this meeting are:

1. Project Manager
2. Development Lead
3. Development Team
4. QA Team.