**Login User Story**

**Test Plan**

**Version 1.0**

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 05/December/2017 | 1.0 | Test Plan for Login User Story | Maka Sampath Kumar |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

[1. Objectives 4](#_Toc500325970)

[2. Scope 4](#_Toc500325971)

[3. References 4](#_Toc500325972)

[4. Test Requirements 4](#_Toc500325973)

[4.1 Data and Database Integrity Testing 4](#_Toc500325974)

[4.2 System Testing (i.e. functional testing) 5](#_Toc500325975)

[4.3 Cross browser/Responsive Testing 5](#_Toc500325976)

[4.4 User Interface Testing 5](#_Toc500325977)

[4.5 Performance Testing 5](#_Toc500325978)

[4.6 Load Testing 5](#_Toc500325979)

[4.7 Stress Testing 5](#_Toc500325980)

[4.8 Security and Access Control Testing 5](#_Toc500325981)

[5. Test Strategy 6](#_Toc500325982)

[5.1 Testing Types 6](#_Toc500325983)

[5.1.1 Data and Database Integrity Testing 6](#_Toc500325984)

[5.1.2 System Testing 6](#_Toc500325985)

[5.1.3 Cross browser/Responsive Testing 7](#_Toc500325986)

[5.1.4 User Interface Testing 7](#_Toc500325987)

[5.1.5 Performance Testing 8](#_Toc500325988)

[5.1.6 Load Testing 9](#_Toc500325989)

[5.1.7 Stress Testing 9](#_Toc500325990)

[5.1.8 Security and Access Control Testing 10](#_Toc500325991)

[5.2 Tools 11](#_Toc500325992)

[6. Resources 11](#_Toc500325993)

[7. Project Milestones 12](#_Toc500325994)

[8. Deliverables 13](#_Toc500325995)

[8.1 Test Suite 14](#_Toc500325996)

[8.2 Defect Reports 14](#_Toc500325997)

[9. Project Tasks 14](#_Toc500325998)

**Test Plan**

# 1. Objectives

This document describes the plan for testing the Login User Story of Gotouche. This Test Plan document supports the following objectives:

* + Identify existing project information and the software components that should be tested.
  + List the recommended test requirements (high level).
  + Recommend and describe the testing strategies to be employed.
  + Identify the required resources and provide an estimate of the test efforts.
  + List the deliverable elements of the test activities.

# 2. Scope

This Test Plan applies to the integration and system tests that will be conducted on the Login User Story of Gotouche System. It is assumed that unit testing already provided thorough black box testing through extensive coverage of source code and testing of all module interfaces.

This Test Plan applies to testing all requirements of the Login User Story of Gotouche System as defined in the Use Case Specifications.

# 3. References

Applicable references are:

* 1. Login User Story of Gotouche System
  2. Information required to test
  3. Design Assets

# 4. Test Requirements

The listing below identifies those items (use cases, functional requirements, non-functional requirements) that have been identified as targets for testing. This list represents *what* will be tested. Details on each test will be determined later as Test Cases are identified and Test Procedures developed.

## 4.1 Data and Database Integrity Testing

Verify the data is properly getting added in the corresponding Database.

Verify simultaneous record read accesses.

Verify correct retrieval of update of database data.

## 4.2 System Testing (i.e. functional testing)

Verify Login Use Case with email and password

Verify Login Use Case with phone number and password

## 4.3 Cross browser/Responsive Testing

Verify Login use case in different browsers (Firefox, Chrome, IE)

Verify the Login use case in different mobile devices (Android and iOS)

## 4.4 User Interface Testing

Verify ease of navigation through a sample set of screens.

Verify sample screens conform to GUI standards.

## 4.5 Performance Testing

Verify response time to access given URL from internal subsystem (Internal to Touche).

Verify response time to access given URL from external subsystem.

Verify response time for login. The system shall provide access to the logged in page with no more than a 10 second latency

## 4.6 Load Testing

Verify system response when loaded with 200 logged on users.

Verify system response when 50 simultaneous user accesses to the Touche.

## 4.7 Stress Testing

Verify system response during prime time use of the Server.

Verify system response during maximum user logins.

## 4.8 Security and Access Control Testing

Verify Logon from a local PC.

Verify Logon from a remote PC.

Verify Logon security through user name and password mechanisms.

# 5. Test Strategy

The Test Strategy presents the recommended approach to the testing of the software applications. The previous section on Test Requirements described *what* will be tested; this describes *how* it will be tested.

The main considerations for the test strategy are the techniques to be used and the criterion for knowing when the testing is completed.

In addition to the considerations provided for each test below, testing should only be executed using known, controlled databases, in secured environments.

The following test strategy is generic in nature and is meant to apply to the requirements listed in this document.

## 5.1 Testing Types

### 5.1.1 Data and Database Integrity Testing

The databases and the database processes should be tested as separate systems. These systems should be tested without the applications (as the interface to the data). Additional research into the DBMS needs to be performed to identify the tools / techniques that may exist to support the testing identified below.

|  |  |
| --- | --- |
| Test Objective: | Ensure Database access methods and processes function properly and without data corruption. |
| Technique: | * Invoke each database access method and process, seeding each with valid and invalid data (or requests for data). * Inspect the database to ensure the data has been populated as intended, all database events occurred properly, or review the returned data to ensure that the correct data was retrieved (for the correct reasons) |
| Completion Criteria: | All database access methods and processes function as designed and without any data corruption. |
|  |  |

### 5.1.2 System Testing

Testing of the application should focus on any target requirements that can be traced directly to use cases (or business functions), and business rules. The goals of these tests are to verify proper data acceptance, processing, and retrieval, and the appropriate implementation of the business rules. This type of testing is based upon black box techniques, that is, verifying the application (and its internal processes) by interacting with the application via the GUI and analyzing the output (results). Identified below is an outline of the testing recommended for each application:

|  |  |
| --- | --- |
| Test Objective: | Ensure proper application navigation, data entry, processing, and retrieval. |
| Technique: | * Execute each use case, use case flow, or function, using valid and invalid data, to verify the following: * The expected results occur when valid data is used. * The appropriate error / warning messages are displayed when invalid data is used. * Each business rule is properly applied. |
| Completion Criteria: | * All planned tests have been executed. * All identified defects have been addressed. |

5.1.3 Cross browser/Responsive Testing

Cross browser/Responsive Testing should emulate the activities performed on the different systems and different browsers to verify the functionality is as per the specifications.

|  |  |
| --- | --- |
| Test Objective | Ensure proper application and background processes function according to required business models in different browsers and different systems. |
| Technique: | * Testing will simulate several business cycles by performing the following: * The tests used for application function testing will be performed multiple times on different machines, systems and different devices to record the results of the tests in each cycle |
| Completion Criteria: | * All planned tests have been executed. * All identified defects have been addressed. |

5.1.4 User Interface Testing

User Interface testing verifies a user’s interaction with the software. The goal of UI Testing is to ensure that the User Interface provides the user with the appropriate access and navigation through the functions of the applications. In addition, UI Testing ensures that the objects within the UI function as expected and conform to corporate or industry standards.

|  |  |
| --- | --- |
| Test Objective: | Verify the following:   * Navigation through the application properly reflects business functions and requirements, including window to window, field to field, and use of access methods (tab keys, mouse movements, accelerator keys) * Window objects and characteristics, such as menus, size, position, state, and focus conform to standards. |
| Technique: | * Create / modify tests for each window to verify proper navigation and object states for each application window and objects. |
| Completion Criteria: | * Each window successfully verified to remain consistent with benchmark version or within acceptable standard |

5.1.5 Performance Testing

Performance testing measures response times, transaction rates, and other time sensitive requirements. The goal of Performance testing is to verify and validate the performance requirements have been achieved. Performance testing is usually executed several times, each using a different "background load" on the system. The initial test should be performed with a "nominal" load, similar to the normal load experienced (or anticipated) on the target system. A second performance test is run using a peak load.

Additionally, Performance tests can be used to profile and tune a system’s performance as a function of conditions such as workload or hardware configurations.

NOTE: Transactions below refer to "logical business transactions." These transactions are defined as specific functions that an end user of the system is expected to perform using the application, such as add or modify a given contract.

|  |  |
| --- | --- |
| Test Objective: | Validate System Response time for designated transactions or business functions under a the following two conditions:  - normal anticipated volume  - anticipated worse case volume |
| Technique: | * Use Test Scripts developed for Business Model Testing (System Testing). * Modify data files (to increase the number of transactions) or modify scripts to increase the number of iterations each transaction occurs. * Scripts should be run on one machine (best case to benchmark single user, single transaction) and be repeated with multiple clients (virtual or actual, *see special considerations below).* |
| Completion Criteria: | * Single Transaction / single user: Successful completion of the test scripts without any failures and within the expected / required time allocation (per transaction) * Multiple transactions / multiple users: Successful completion of the test scripts without any failures and within acceptable time allocation. |
|  |  |

5.1.6 Load Testing

Load testing measures subjects the system-under-test to varying workloads to evaluate the system’s ability to continue to function properly under these different workloads. The goal of load testing is to determine and ensure that the system functions properly beyond the expected maximum workload. Additionally, load testing evaluates the performance characteristics (response times, transaction rates, and other time sensitive issues).

NOTE: Transactions below refer to "logical business transactions." These transactions are defined as specific functions that an end user of the system is expected to perform using the application, such as add or modify a given contract.

|  |  |
| --- | --- |
| Test Objective: | Verify System Response time for designated transactions or business cases under varying workload conditions. |
| Technique: | * Use tests developed for Business Cycle Testing. * Modify data files (to increase the number of transactions) or the tests to increase the number of times each transaction occurs. |
| Completion Criteria: | * Multiple transactions / multiple users: Successful completion of the tests without any failures and within acceptable time allocation. |

5.1.7 Stress Testing

Stress testing is intended to find errors due to low resources or competition for resources. Low memory or disk space may reveal defects in the software that aren't apparent under normal conditions. Other defects might result from competition for shared resource like database locks or network bandwidth. Stress testing identifies the peak load the system can handle.

NOTE: References to transactions below refer to logical business transactions.

|  |  |
| --- | --- |
| Test Objective: | Verify that the system and software function properly and without error under the following stress conditions:   * little or no memory available on the server (RAM and DASD) * maximum (actual or physically capable) number of clients connected (or simulated) * multiple users performing the same transactions against the same data / accounts * worst case transaction volume / mix (see performance testing above).   NOTES: Stress testing’s goal might also be stated as identify and document the conditions under which the system FAILS to continue functioning properly. |
| Technique: | * Use tests developed for Performance Testing. * To test limited resources, tests should be run on single machine, RAM and DASD on server should be reduced (or limited). * For remaining stress tests, multiple clients should be used, either running the same tests or complementary tests to produce the worst case transaction volume / mix. |
| Completion Criteria: | All planned tests are executed and specified system limits are reached / exceeded without the software or software failing (or conditions under which system failure occurs is outside of the specified conditions). |
|  |  |

5.1.8 Security and Access Control Testing

Security and Access Control Testing focus on two key areas of security:

- Application security, including access to the Data or Business Functions, and  
- System Security, including logging into / remote access to the system.

Application security ensures that, based upon the desired security, users are restricted to specific functions or are limited in the data that is available to them. For example, everyone may be permitted to enter data and create new accounts, but only managers can delete them. If there is security at the data level, testing ensures that user "type" one can see all customer information, including financial data, however, user two only sees the demographic data for the same client.

System security ensures that only those users granted access to the system are capable of accessing the applications and only through the appropriate gateways.

|  |  |
| --- | --- |
| Test Objective: | Function / Data Security: Verify that user can access only those functions / data for which their user type is provided permissions.  System Security: Verify that only those users with access to the system and application(s) are permitted to access them. |
| Technique: | * Function / Data Security: Identify and list each user type and the functions / data each type has permissions for. * Create tests for each user type and verify permission by creating transactions specific to each user type. * Modify user type and re-run tests for same users. In each case verify those additional functions / data are correctly available or denied. * System Access (see special considerations below) |
| Completion Criteria: | For each known user type the appropriate function / data are available and all transactions function as expected and run in prior Application Function tests |
| Special Considerations: | * Access to the system must be reviewed / discussed with the appropriate network or systems administrator. This testing may not be required as it maybe a function of network or systems administration. |

## 5.2 Tools

The following tools will be employed for testing of the system:

|  |  |  |
| --- | --- | --- |
|  | **Tool** | **Version** |
| Test Management | HP ALM/JIRA | TBD |
| Test Design | HP ALM | TBD |
| Defect Tracking | HP ALM/ JIRA | TBD |
| Functional Testing | HP ALM | TBD |
| Performance Testing | Load Runner/JMeter | TBD |
| Test Coverage Monitor or Profiler | HP ALM | TBD |
| Other Test Tools | Selenium Web Driver | TBD |
| Project Management | Microsoft Project  Microsoft Word  Microsoft Excel | TBD |
| DBMS tools | TBD | TBD |

# 6. Resources

This section presents the recommended resources for testing the Login User Story of Gotouche System, their main responsibilities, and their knowledge or skill set.

This table shows the staffing assumptions for the test activities.

|  |  |  |
| --- | --- | --- |
| **Human Resources** | | |
| **Role** | **Minimum Resources Recommended**  (number of workers allocated full-time) | **Specific Responsibilities/Comments** |
| Test Manager | 1 – Sampath Kumar | Provides management oversight  Responsibilities:   * Provide technical direction * Acquire appropriate resources * Management reporting |
| Test Designer | Sampath Kumar | Identifies, prioritizes, and implements test cases  Responsibilities:   * Generate test plan * Generate Test Suite * Evaluate effectiveness of test effort |
| System Tester | Sampath Kumar | Executes the tests  Responsibilities:   * Execute tests * Log results * Recover from errors * Document defects |
| Test System Administrator | Sampath Kumar | Ensures test environment and assets are managed and maintained.  Responsibilities:   * Administer test management system * Install / manage worker access to test systems |

# 7. Project Milestones

The test activities and milestones are very much dependant upon the development iterations. The Construction Phase will be split into 2 iterations. Each iteration contains a full test cycle of test planning, designing, development, execution, and evaluation.

Refer to the Iteration Plans for the master schedule and Construction Phase plan that shows the development iterations.

The following table shows the Test Milestones. Effort, start date, and end date can be completed as the iteration content is planned.

|  |  |  |  |
| --- | --- | --- | --- |
| **Milestone Task** | **Effort (pd)** | **Start Date** | **End Date** |
| Iteration C1: Beta Release  Test Planning  Test Design  Test Development  Test Execution  Test Evaluation | TBD | TBD | TBD |
| Iteration C2: R1.0 Release  Test Planning  Test Design  Test Development  Test Execution  Test Evaluation | TBD | TBD | TBD |

# 8. Deliverables

The deliverables of the test activities as defined in this Test Plan are outlined in the table below.

Note that some of these deliverables are produced multiple times; once for each test cycle or iteration. Other deliverables, such as the Test Plan, are updated each development iteration.

|  |  |  |
| --- | --- | --- |
| **Deliverable** | **Review / Distribution** | **Due Date** |
| Test Plan | Senior Project Mgmt Team | TBD |
| Test Suite | Internal Peer Review | TBD |
| Test Data Sets | Internal Peer Review | TBD |
| Test Scripts | Internal Peer Review | TBD |
| Test Scripts | - | TBD |
| Test Defect Reports  (for each iteration) | Senior Project Mgmt Team | TBD |
| Test Results  (for each iteration) | Test Manager | TBD |
| Test Evaluation Report  (for each iteration) | Senior Project Mgmt Team | TBD |

## 8.1 Test Suite

The Test Suite will define all the test cases and the test scripts which are associated with each test case.

## 8.2 Defect Reports

HP ALM/JIRA (TBD) will be used for logging and tracking individual defects.

# 9. Project Tasks

Below are the test related tasks for testing the Login User Story of Gotouche System:

|  |
| --- |
| **Plan Test** |
| Identify Requirements for Test |
| Assess Risk |
| Develop Test Strategy |
| Identify Test Resources |
| Create Schedule |
| Generate Test Plan |
| **Design Test** |
| Workload Analysis |
| Develop Test Suite |
| Identify and Describe Test Cases |
| Identify and Structure Test Scripts |
| Review and Access Test Coverage |
| **Implement Test** |
| Setup Test Environment |
| Record or Program Test Scripts |
| Develop Test Stubs and Drivers |
| Identify Test-Specific functionality in the design and implementation model |
| Establish External Data sets |
| **Execute Test** |
| Execute Test Scripts |
| Evaluate Execution of Test |
|  |
| Verify the results |
| Investigate Unexpected Results |
| Log Defects |
| **Evaluate Test** |
| Evaluate Test-Case Coverage |
| Evaluate Code Coverage |
| Analyze Defects |
| Determine if Test Completion Criteria and Success Criteria have been achieved |
| Create Test Evaluation Report |