CS691 – Computer Science, Spring 2023

Pace University

Logo

Description automatically generated

SYSTEM TEST PLAN

MaidEase

Authors: Tahera, Nidhi, Jahid, Bhakti, Sai Manish, Sai Teja

Project Manager: Sai Teja

Date: 05/01/2023

Table of Contents

[1. INTRODUCTION 3](#_Toc133854202)

[2. TESTING SCOPE 3](#_Toc133854203)

[2.1 Core Features to be Tested 3](#_Toc133854204)

[2.2 Non-Functional Features to be Tested 4](#_Toc133854205)

[2.3 Features to be Tested 4](#_Toc133854206)

[3. TEST PROCESS DEFINITION 4](#_Toc133854207)

[3.1 Test Process Phrases and Tasks 4](#_Toc133854208)

[3.2 Deliverables 5](#_Toc133854209)

[4. APPROACH TO SYSTEM TESTING 5](#_Toc133854210)

[4.1 Approach to Functional Testing 5](#_Toc133854211)

[4.2 Approach to Non-Functional Testing 6](#_Toc133854212)

[5. ENTRY/ EXIT CRITERIA 6](#_Toc133854213)

[6. SYSTEM TEST ENVIRONMENT 6](#_Toc133854214)

[7. ROLES AND RESPONSIBILITIES 7](#_Toc133854215)

[8. TEST CYCLES & SCHEDULE 7](#_Toc133854216)

[9. RISK AND CONTINGENCIES 8](#_Toc133854217)

# **INTRODUCTION**

The primary purpose of the System Test Plan document is to establish a common understanding among the "MaidEase" project stakeholders about the scope, objectives, and approach to performing the system testing. In addition, the document covers such topics as environmental needs, testing entry/exit criteria, test schedule, roles and responsibilities, and risks and contingencies.

# **TESTING SCOPE**

The testing scope includes two perspectives - the functional scope and technical scope.

The functional scope of system testing includes the modules of the “MaidEase” system: Create Account, Update Account, Maintain Account Summary, Close Account.

The technical scope includes the following architectural components:

* Web server
* Application server
* Database server
* Middleware messaging

## **Core Features to be Tested**

User Experience

* I Have Input
  + To test whether the user is successfully able to input data.
* I have Item Delete
  + To test whether the user is successfully able to delete the account that they had previously created.
* I have Item update
  + To test whether the user is successfully able to delete the account that they had previously created.
* Maintain Account Summary
  + Test whether the user is successfully generate the account summary

User Register/Login

* User Registration
  + Test whether a user can register/create account.
* User Login
  + Test whether a user can login once they have registered.

In addition to the above core features, testing will cover crosscutting concerns applicable to the context of individual core features, see the Requirements Composition Table for reference.

## **Non-Functional Features to be Tested**

The System Test includes the following objectives to test non-functional requirements:

* **Volume testing**

The system must be able to handle large number of user registrations when the product goes online. The system must be able to handle concurrent transactions such as booking a same service at the same time by two different users. The system should be able to adhere to large number of service booking transactions.

* **Portability testing**

The system should have all the features displayed correctly on all major browsers such as Google Chrome, Safari, MS Edge and Mobile.

* **Extreme layout testing**

The system should not pixelate or lose its layout when the display of the browser is changed, or the resolution is changed. The buttons, font and text should appear of the same size.

## **Features to be Tested**

###### Social Media Integration: Website has links to social media, this feature may not require extensive testing. Social media platforms logins are often tested by the social media platform itself.

###### Terms and Conditions/Privacy Policy: These typically static pages with minimal functionality may not require extensive testing.

###### About Us/Contact Us Pages: Similar to the terms and conditions and privacy policy pages, these pages are typically static and require little to no functionality testing.

###### Footer Links: The footer links may not require as much testing as other links on your site. Footer links typically lead to pages that are less critical to the overall user experience.

# **TEST PROCESS DEFINITION**

## **Test Process Phrases and Tasks**

**Test Planning**

* + User should be able to register, login and search through the websites. Register, Login, Search are the three main modules to be tested in this scope.
  + The tester on the team is responsible for testing all inputs and the developers are responsible for their respective modules.
  + The testing will be done after the module is completely built and will go back to the developer for bug fixing. There will be another round of testing after the bugs are fixed.

**Test Design**

* + There will be 5-7 test cases for each of the which might be repetitive in few cases. For e.g. Font style or color.
  + The cases will be finalized and approved before test execution.
  + The test data will contain all boundary cases and values for each of the input.

**Test Preparation**

* The test environment will be the localhost.

**Test Execution**

* Execute all test cases.
* Find and report software defects.
* Evaluate the system stability.
* Validate all target features.

**Test Reporting**

* Summarize and report the test execution results.
* Report defect metrics
* Text exit criteria will depend on the module to be tested.
* Create a test completion report, submit for stakeholder approval.
* Obtain stakeholder signoff on system testing.

## **Deliverables**

On this project, the test process deliverables include:  
• System Test Plan document.  
• Test Design specifications.  
• Test Case specifications.  
• Software Defects.  
• Test Execution Logs.

Test Completion Report.

# **APPROACH TO SYSTEM TESTING**

## **Approach to Functional Testing**

The overall approach to functional testing will be based on the Black-box method:

* Test cases will be designed using some formal black-box techniques such as boundary-value analysis, equivalent-class partitioning, cause-effect graphing, decision tables, and state-transition testing, where applicable.
* Test execution will be conducted manually, from the user perspective and based on formal test case specifications.

The test execution results will be captured and reported in test execution logs.

## **Approach to Non-Functional Testing**

All non-functional test objectives specified above can be tested using the black-box approach, that is from the user perspective. The volume test should be performed for a complete production scenario. The portability and extreme layout tests should cover all functions of the system and validate that each function works under the specified test conditions.

# **ENTRY/ EXIT CRITERIA**

The **Test Entry criteria** is used to formally evaluate the conditions necessary to begin test execution, it includes the following conditions:

* Development tasks and integration testing have been completed.
* System Test Plan document has been approved.
* QA environment is ready.
* QA team members have access to the QA environment.
* Test case specifications have been completed and reviewed.
* Release Notes document has been sent to the QA team.

The **Test Exit criteria** is used to evaluate the conditions necessary to conclude that testers can stop test execution and the system is ready for the final user acceptance testing, it includes the following conditions:

* All requirements, in scope of testing, are covered by test cases.
* All test cases have been executed.
* Zero defects of Critical and Hi-severity remain open.
* Open defects of Medium and Low severity have known work-around.
* A Test Completion Report has been produced and communicated to stakeholders.
* QA testing sign-off has been provided.

# **SYSTEM TEST ENVIRONMENT**

The system test will be performed in New York on the following test servers:

* Allocation System – abc024
* Conditional Offering System – abc025

The project team uses SourceSafe as a Software Configuration Management  
tool and HP Quality Center as a defect-tracking tool and a repository of test  
cases.

Diagram

Description automatically generated

# **ROLES AND RESPONSIBILITIES**

The project team has eight members that are assigned various project roles including Project Manager, Product Owner, Lead Business Analyst, Lead Developer, DBA, Lead QA Analyst. Their responsibilities are defined in the table below.

|  |  |
| --- | --- |
| **Project Role** | **Role Responsibilities** |
| Project Manager | Reviewing and approving the test design parameters and the system test plan.  Organizing the testing environment's setup.  keeping track of the testing schedule and outcomes. |
| Lead QA Analyst | Creating a test strategy, setting up a test repository, creating test case specifications, carrying out testing, and submitting bug reports. |
| Product Owner | Contributing to test cases and test plan specs. looking through results from tests. |
| Lead Business Analyst | Generating the specifications of the test plan and test cases. examining test outcomes. |
| Lead Developer | Creating and maintaining the test environment and working with a Lead QA Analyst to complete the tests. |
| DBA | Helping to create and manage the test environment with the lead developer. |

# **TEST CYCLES & SCHEDULE**

The system test execution will be conducted as four test cycles that are aligned with four application modules as follows:

Cycle 1. Create Account

* This cycle concentrates on testing the part (input and filtering) of the User Account Module

Cycle 2. Authenticate Account

* This cycle concentrates on testing the part (Verification)user account Module.

Cycle 3. Update Account

* This cycle concentrates on testing the User Account Update Module.

Cycle 4. Maintain Account Summary

* This cycle concentrates on testing the User Account summary module.

See the schedule of the test execution cycles in the project plan.

# **RISK AND CONTINGENCIES**

The risks and contingencies that might have arisen during the system testing are highlighted in this section.

* Lack of testing resources could cause a delay.
* Any modifications to the scope objectives may result in a delay or additional effort.
* Lack of team member collaboration can hinder testing progress and make it take longer to repair problems and finish testing when there are many of them.
* The execution of the test can be delayed if there are too many flaws.
* The test execution schedule may be impacted by the test environment's instability.