CS691 – Computer Science, Spring 2020

Pace University



SYSTEM TEST PLAN

BundleBid

Authors: Swetha Bitla, Weili Zhong

Project Manager: Siddharth Jhaveri

Date:13-4-2020

**Table of Contents**

[INTRODUCTION 3](#_Toc37343243)

[1. TESTING SCOPE 3](#_Toc37343244)

[2. TESTING OBJECTIVES 3](#_Toc37343245)

[2.1 Core Features to be Tested 4](#_Toc37343246)

[2.2 Non-Functional Features to be Tested 4](#_Toc37343247)

[2.3 Features not to be Tested 5](#_Toc37343248)

[3. TEST PROCESS DEFINITION 5](#_Toc37343249)

[3.1 Test Process Phases 5](#_Toc37343250)

[3.2 Testing Tasks and Deliverables 6](#_Toc37343251)

[4. APPROACH TO SYSTEM TESTING 7](#_Toc37343252)

[4.1 Approach to Functional Testing 7](#_Toc37343253)

[4.2 Approach to Non-Functional Testing 7](#_Toc37343254)

[5. ENTRY/EXIT CRITERIA 7](#_Toc37343255)

[5.1 Entry Criteria 7](#_Toc37343256)

[5.2 Exit Criteria 7](#_Toc37343257)

[6. SYSTEM TEST ENVIRONMENT 8](#_Toc37343258)

[7. ROLES AND RESPONSIBILITIES 8](#_Toc37343259)

[8. TEST CYCLES AND SCHEDULE 8](#_Toc37343260)

[9. RISKS AND CONTINGENCIES 8](#_Toc37343261)

# INTRODUCTION

This document is for the establishment of the System Test Plan that provides a common understanding among the “BundleBid” project stakeholders on the following aspects, including the scope, objectives, and approach to performing the system testing. In addition, the document also involves a few more topics, which are features to be tested, entry/exit criteria, resource and responsibilities, and testing schedule.

# 1. TESTING SCOPE

The testing scope consists of mainly two aspects, the functional scope and the technical scope.

The functional scope of the system includes the following functional areas:

* Users ability to register and login
* View the products and bid online
* Users ability to update existing bid
* Provide purchasing instructions and details after successful bids
* Seller’s ability to list a product with description and a starting bid price.

The technical scope of the system includes the following technical areas:

* Web server
* Application server
* Firebase NoSQL
* Content server

# 2. TESTING OBJECTIVES

The main objective to system testing is to validate the implemented system features so that they comply with the functional and non-functional requirements. The implemented test cases should validate if the feature being testing is free of defects as well as verify that the requirements are fulfilled by the application feature. This section lists the features to be testing and the features that will be excluding out of the testing scope.

The list of project documents that will be used as a basis for designing test cases include:

* Project Initiation Document (PID)
* Business Requirements Document (BRD)
* User Roles
* Requirements Composition Table (RCT)
* Requirements Traceability Matrix (RTM)
* Feature Short Descriptions
* Entitlements Specification
* Responsibility Assignment Matrix (RACI table)

## 2.1 Core Features to be Tested

This section lists all core features that will be tested.

* List a product
  + Creating a Product
  + Modifying a Product
  + Deleting a Product
  + Listing History
* User Account
  + Register Account
  + Update Account
  + Close Account
  + Purchase History
  + Sign in to Account
* Browse for Products
  + Search Product
  + Product Category
  + Product Subcategory
* Bidding a Product
  + Submit a Bid
  + Buy at Max Bid
* Payment Gateway
  + Send Payment Detail
  + Payment Verification

## 2.2 Non-Functional Features to be Tested

The system test cases will be implemented in compliance with the following objectives according to the non-functional requirements:

* **Usability** 
  + The application’s GUI interface should feature a simple and intuitive design, with menu options clearly labeled and field function obvious to any user.
  + The purpose of any section of the application should be immediately evident to any user.
  + The application should be easy to navigate and users should be able to immediately access any significant functionality that they may need within a few clicks, ideally in one click.
* **Performance** 
  + The response time of the application should not be impacted by latency that will significantly impact the evident availability of application features or information
  + The application should be available for users 24 hours a day, every day of the year.
  + The application should be not make excessive demands on customer’s computer or device memory capabilities
  + The application should support several users using simultaneously.
* **Security**
  + The application will be protected from hackers and intrusion by various security measures
  + Encryption for customer’s personal data that is maintained on the web application’s server
* **Database** 
  + Application will use the Firebase NoSQL database
* **External** **System**
  + Application will be able to interface successfully with social media websites, travel service websites, credit card payment services, and other online resources

## 2.3 Features not to be Tested

There are a few features that will not be covered in testing. These features include:

* Reliability of the system
* Maximum serve time and run time of the website
* Encryption bypass/failure of some parameters

# 3. TEST PROCESS DEFINITION

## 3.1 Test Process Phases

The testing process for the system will be split into five phases which include:

* Test Planning
  + Define the scope and objectives of testing
  + Testing tool selection and effort estimation
  + Roles and responsibilities
* Test Design
  + Determine test logic
  + Design test cases specifications
  + Determine requirements for test
* Test Preparation
  + Setting up test environment
  + Provision test data
  + Install testing software in a proper environment
* Test Execution
  + Execute all system tests
  + Find and report bugs and defects
  + Validate that system complies with requirements
  + Validate all target features
* Test Reporting
  + Create bug reports
  + Create test reports for stakeholders
  + Report defect metrics and test execution reports
  + Test completion report

## 3.2 Testing Tasks and Deliverables

|  |  |  |
| --- | --- | --- |
| **Process Phase** | **Tasks** | **Deliverables** |
| Test Planning | * Define the scope and objectives of testing * Testing tool selection and effort estimation * Roles and responsibilities | System Test Plan Document |
| Test Design | * Determine test logic * Design test cases specifications * Determine requirements for test | * Test Design Specification * Test-Case Specifications * Test Management System |
| Test Preparation | * Setting up test environment * Provision test data * Install testing software in a proper environment | * Testing system establishment * Test availability of the data in an application environment * Implement Defect Tracking System |
| Test Execution | * Execute all system tests * Find and report bugs and defects * Validate that system complies with requirements * Validate all target features | * Report all defects to the Defect Tracking System |
| Test Reporting | * Create bug reports * Create test reports for stakeholders * Report defect metrics and test execution reports * Test completion report | * Test Summary Report * Defect metrics * Test execution status reports * Final test report |

# 4. APPROACH TO SYSTEM TESTING

## 4.1 Approach to Functional Testing

The approach to the system’s functional testing will be performed based on black-box techniques. This means that a black-box testing technique will be used to check whether or not a specific functionality fulfills a design goal. These goals will be based off of the functional requirements section listed in the Business Requirements Document. A brief list of these requirements can be sound in Section 2.1.

Using a black-box approach can be helpful in the fact that it can be used to examine the functionalities of the system without knowledge of the systems internal code structure, implementation details and internal paths of the system. A few useful black-box testing techniques include boundary-value analysis, cause-effect graphing, decision table testing, etc.

## 4.2 Approach to Non-Functional Testing

Both black-box and white-box testing will be necessary and used for non-functional testing. Black-box testing will be utilized in performing tests to ensure usability, performance and external systems. For example, usability tests can be performed based off us ease of access and navigation to core features which do not require knowledge of internal structures of the system. On the other hand, white-box testing is necessary for testing non-functional aspects such as security and efficiency of the Firebase NoSQL database. Some white-box testing techniques include control flow testing, data flow testing, branch testing, decision coverage, etc.

# 5. ENTRY/EXIT CRITERIA

This section addresses both entry and exit criteria for the system.

## 5.1 Entry Criteria

The purpose of the test entry criteria is to form an evaluation standard that will be used to begin test executions. An amount of conditions of the entry criteria include:

* Complete the development of the all tasks
* Accomplish the integration testing
* Approve the system test plan
* Establish the testing (QA) environment
* Make the testing environment accessible
* Finish and review test case specifications
* Distribute notes documents to team members

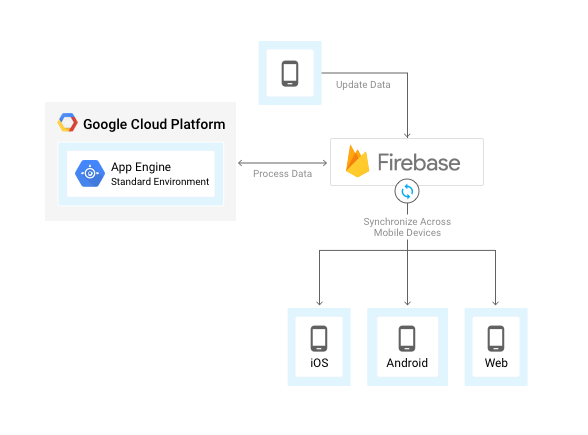
## 5.2 Exit Criteria

The purpose of the test exit criteria is to determine when and how the testing is complete. It is a criteria showing the system is ready for the application to users. Main conditions of the exit criteria include:

* Have executed all testing cases
* Zero defects of Critical and Hi-severity remain open
* Open defects of Medium and Low severity have known work-around
* Have completed a summary testing report
* Have approved a testing sign-off

# 6. SYSTEM TEST ENVIRONMENT

The system test will mainly appear on the local shot 80 port. Further tests will be operated on Google Cloud Platform (GCP) and Firebase test lab after completing all function testing locally. Lead QA Analyst will work with Lead Developer and DBA to determine cloud service offerings on GCP. The target GCP products include Google Compute Engine – IaaS (virtual machine services) and Google Cloud SQL (database).



# 7. ROLES AND RESPONSIBILITIES

The project team has seven members, including Project Manager, Product Owner, Lead Business Analyst, Lead Developer, DBA, Lead QA Analyst, and Professor. Aligning with the project’s RACI Table, the following table provides descriptions of roles and responsibilities for each team member during the testing period.

|  |  |
| --- | --- |
| **Project Role** | **Role Responsibilities** |
| Project Manager | Assist the testing operation throughout the process of system testing; assist to govern the overall project timelines; review and approval of the System Test Plan, escalation of issues. |
| Lead QA Analyst | Responsible for designing a test plan, establishing a test repository, developing test specifications, executing testing and report defects, conducting defect review calls, and producing/delivering defect metrics. Also consulting the establishment and maintenance of the test environment. |
| Product Owner | Consulting test plans, test repository, and test specifications. Also keeping up-to-date on other work’s progress. |
| Lead Business Analyst | Work with Lead QA Analyst and be responsible for conducting defect review skills and producing/delivering defect metrics. Participate in other work if necessary. |
| Lead Developer | Responsible for establishing and maintaining the test environment and assist Lead QA Analyst throughout the testing process. |
| DBA | Responsible for assisting Lead Developer and Lead QA Analyst to establish and maintain the test environment. Keep informed throughout the testing period. |
| Professor | Guide/advise the project team. |

# 8. TEST CYCLES AND SCHEDULE

The system testing consists of five test cycles in line with three modules. Details of the module cycles and the corresponding schedules are presented in the followings.

Cycle 1. Product Module

This cycle concentrates on creating, editing and modifying products.

Cycle 2. User Account Module

This cycle concentrates on testing the User Account Module.

Cycle 3. Browse for Product(s) Module

This cycle concentrates on searching product based on its category and

subcategory.

Cycle 4. Bidding on products Module

This cycle concentrates on submitting a bid and buying the max bid.

Cycle 5. Payment Gateway Module

This cycle concentrates on testing the Payment Gateway Module.

See the timelines of the testing cycles in the project plan.

# 9. RISKS AND CONTINGENCIES

This section highlights a few potential risks and contingencies that maybe happened during the system testing.

* Limited testing resource may result in a delay.
* Collaboration of the team has an impact on the testing progress.
* Any changes on the scope objectives can cause a delay or extra work.
* Many defects require a longer time to fix the system.